Definición de un Framework para el Análisis Predictivo de Datos no Estructurados

José Luis Jiménez Márquez

en cumplimiento parcial de los requisitos para el grado de Doctor en Ciencia y Tecnología Informática

Universidad Carlos III de Madrid

Directores:

Dr. José Luis López Cuadrado

Dr. Israel González Carrasco

Tutor: Dr. Israel González Carrasco

Leganés, Marzo de 2019

A Rosario y Pilar

AGRADECIMIENTOS

Es complejo condensar en unas cuantas líneas el aprecio y el afecto que tengo por tantas personas de ambos lados del charco que me han acompañado física o virtualmente y que me han ayudado a llegar a mi meta.

En primer lugar, a mis directores José Luis e Israel, muchas gracias por su tiempo y apoyo.

A Rosario, gracias por tu amor, confianza y apoyo a este proyecto. Viajar a tu lado en el tren, un sueño difícil de creer.

A Pilar, gracias por darme la vida. Por tu infinito amor, por estar ahí cada día, por darme la fortaleza para ser quien soy. Pero sobre todo, por tanto apoyo.

A Luis Ángel y Erick Alfredo, a ustedes más que nada les ofrezco disculpas por tanto tiempo no compartido a su lado, ahora es el tiempo de retribuirles todo.

A Rubén, muchas gracias amigo por haberme brindado la mano en la oportunidad que me cambió la vida. Tu nobleza es un gran ejemplo para mí

A Édgar, por haber puesto el listón tan alto que me dieran ganas de alcanzarlo. Espero poder alcanzar los listones que has seguido poniendo. Gracias por tus consejos también.

A Alondra, como te dije, la admiración es mutua. Gracias por tu aprecio. Y espero sigas alcanzando más metas.

A Germán y Manuel, mis amigos. Los recuerdo con mucho aprecio y también agradezco el apoyo que me han brindado.

A Sofía, Alejandra, Laura y Betty. Muchas gracias por el afecto y la buena vibra que me han brindado.

A Elvia, Julio César, Juan, Michelle, Sergio, Héctor, Luzma, Paola, Ernesto y demás amigos y familiares que saben que están en mi corazón. Les agradezco el apoyo y el afecto que me han brindado por diversos medios.

Pero sobre todo a Luis, este agradecimiento va hasta el cielo, a ti te debo todo lo que soy. Espero puedas verlo y sentirte orgulloso de mí, allá donde tú estés con mi hermanita.

CONTENIDOS PUBLICADOS Y PRESENTADOS

1) Título: Challenges and Opportunities in Analytic-Predictive Environments of Big Data and Natural Language Processing for Social Network Rating Systems.
Revista y Año: IEEE Latin America Transactions, 2018.
Autores: Jose Luis Jimenez-Marquez, Israel Gonzalez-Carrasco, José Luis Lopez-Cuadrado.
Rol: Autor principal.
URL: https://ieeexplore.ieee.org/document/8327417
DOI: https://doi.org/10.1109/TLA.2018.8327417

- El artículo está incluido parcialmente en la tesis.
- Capítulos en los que se incluye el material de dicha contribución: 2 y 3.
- Todo material de esta fuente incluido en la tesis está señalado por medios tipográficos y una referencia explícita.

2) Título: Towards a Big Data Framework for Analyzing Social Media Content
Revista y Año: International Journal of Information Management, 2019
Autores: Jose Luis Jimenez-Marquez, Israel Gonzalez-Carrasco, José Luis Lopez-Cuadrado, Belén Ruiz-Mezcua.
Rol: Autor principal.
URL: https://www.sciencedirect.com/science/article/pii/S0268401218305073
DOI: https://doi.org/10.1016/j.ijinfomgt.2018.09.003

- El artículo está incluido parcialmente en la tesis.
- Capítulos en los que se incluye el material de dicha contribución: 2, 4 y 5.
- Todo material de esta fuente incluido en la tesis está señalado por medios tipográficos y una referencia explícita.

Resumen

La cantidad de información que se genera segundo a segundo en Internet aumenta en volumen y variedad cada día. La web 2.0, el Internet de las cosas y los dispositivos móviles son tan sólo algunos de los elementos que han generado tal incremento en el volumen de los datos. En el futuro cercano, la introducción de la tecnología 5G propiciará un incremento exponencial en la generación de datos al permitir una mayor transferencia de Gb/s. Por lo anterior, la investigación en esta área debe establecer las pautas que guíen el camino mediante el cual se puedan establecer metodologías para el análisis de los datos, así como medios para tratarlos.

No obstante, el tamaño y la diversidad de estos datos hacen que tengan que conjuntarse diversas disciplinas científicas para poder analizar los datos y obtener hallazgos relevantes dentro de la información. Es decir, que no sólo se aplicarán las técnicas tradicionales para realizar el análisis, sino que se tendrán que conjuntar otras áreas de la ciencia para poder extraer la denominada *'información oculta'* que se encuentra tras estos datos. Por otra parte, dentro de esta disponibilidad de datos que se está generando, la web 2.0 contribuye con el paradigma de las redes sociales y los tipos de datos (no estructurados) que estos generan, comúnmente texto libre. Este texto libre puede venir asociado a otros elementos dependiendo de la fuente de donde procedan, por ejemplo, pueden estar asociados a una escala de valoración de algún producto o servicio.

Por todo lo anterior, esta tesis plantea la definición de un framework que permita el análisis de datos no estructurados de redes sociales mediante técnicas de aprendizaje automático, procesamiento de lenguaje natural y big data. Dentro de las características principales de este framework se tienen:

- El framework está dividido en dos fases, cada una de las cuáles consta de un conjunto de etapas definidas con el propósito de analizar un volumen de datos ya sea pequeño (inferior a lo considerado big data) o grande (big data).
- El elemento central de la fase uno del framework es el modelo de aprendizaje automático el cual consiste de dos elementos: *(i)* una serie de técnicas de procesamiento de lenguaje natural orientadas al preprocesamiento de datos y *(ii)* una serie de algoritmos de aprendizaje automático para la clasificación de la información.
- El modelo de aprendizaje automático construido en la primera fase tiene como intención el poder ser empleado en la segunda (big data) para analizar el mismo origen de datos, pero a un volumen mucho mayor.

- El modelo de aprendizaje automático no está relacionado directamente con la aplicación de determinados algoritmos para su uso, lo que lo convierte en un modelo versátil para emplear.

De tal manera que como se observa, el marco en que se desenvuelve esta investigación es multidisciplinar al conjuntar diversas disciplinas científicas con un mismo propósito. Por lo cual, el resolver el problema de análisis de datos no estructurados provenientes de redes sociales requiere de la unión de técnicas heterogéneas procedentes de diversas áreas de la ciencia y la ingeniería. La metodología de investigación seguida para la elaboración de esta tesis doctoral ha consistido en:

1. **Estado del Arte:** Se presenta una selección de estudios que otros autores en las áreas de Big Data, Machine Learning y Procesamiento de Lenguaje Natural han realizado al respecto, así como la unión de estos temas con el área de análisis de sentimientos y los sistemas de calificación de redes sociales. También se presenta una comparativa que integra los temas abordados con el propósito de conocer el estado del arte en cuanto a lo que otros autores han propuesto en sus estudios al combinar las tres áreas cubiertas por el framework.
2. **Estado de la Técnica:** En esta fase se analizaron los diversos elementos que componen el framework y a partir de esto se presenta una retrospectiva teórica al respecto. Se abordan temas más técnicos, para lo cual se presenta un panorama de las tecnologías que se están empleando en la investigación actual.
3. **Solución Propuesta:** En esta fase se presenta el framework propuesto analizándolo desde dos perspectivas: los aspectos teóricos que comprende cada fase y los aspectos de implementación, en los cuáles se abordan temas como la complejidad de llevar a la práctica cada fase en una situación real.
4. **Evaluación y Validación:** Se definen una serie de pruebas destinadas a comprobar las hipótesis establecidas al principio de la investigación, para demostrar la validez del modelo propuesto.
5. **Documentación y Conclusiones.:** Esta actividad consistió en documentar todos los aspectos relacionados con esta tesis y presentar las conclusiones que surgen al término de la investigación.

Por consiguiente, se construyó un framework que contempla dos fases a través de las cuáles se realiza el análisis de un conjunto de datos no estructurados, siendo una distinción de este framework la construcción de un modelo de aprendizaje automático durante la primera fase, que pretende servir como base en la segunda, la cual se caracteriza por el procesamiento de datos de gran volumen. Para poder validar este trabajo de tesis, se emplearon datos de Yelp, concretamente del sector de la hotelería. De igual manera, se evaluó el framework mediante la ejecución de diversas pruebas empleando clasificadores

de aprendizaje automático, obteniendo porcentajes altos de predicción en la búsqueda binaria llevada a cabo tanto en el entorno no big data como en big data.

Las conclusiones obtenidas tras haber diseñado el framework, así como haber analizado y validado los resultados conseguidos demuestran que el modelo presentado es capaz de analizar datos no estructurados de redes sociales tanto a una escala menor (no big data) como mayor (big data) de análisis. Por otra parte, interesantes retos y futuras líneas de investigación surgen tras haber concluido el modelo tanto para extenderlo hacia el análisis de otro tipo de información, como en el aspecto de la integración y adaptación del modelo de aprendizaje automático de la primera hacia la segunda fase.

Abstract

The amount of information generated continuously on the Internet increases in volume and variety each day. Web 2.0, the Internet of things and mobile devices are just some of the elements that have generated such an increase in the volume of data. In the near future, the introduction of 5G technology will lead to an exponential increase in data generation by allowing a greater Gb/s transfer. Therefore, research in this area should establish the guidelines that guide the way by which methodologies can be established for the analysis of data, as well as means to deal with them.

However, the size and diversity of these data mean that different scientific disciplines have to be combined in order to analyze the data and obtain relevant findings within the information. That is, not only traditional techniques will be applied to carry out the analysis, but other areas of science will have to be combined in order to extract the so-called 'hidden information' found behind these data. On the other hand, in this availability of data being generated, web 2.0 contributes with the paradigm of social networks and the types of (unstructured) data that these generate, commonly free text. This free text may be associated with other elements depending on the source they come from, for example, they may be associated with a rating scale of a product or service.

For all the above, this thesis proposes the definition of a framework that allows the analysis of unstructured data of social networks using machine learning, natural language processing and big data techniques. The main features of this framework are:

- The framework is divided into two phases, each of which consists of a set of stages defined for the purpose of analyzing a volume of data either small (less than big data) or large (big data).
- The central element of phase one of the framework is the machine learning model which consists of two elements: (i) a series of natural language processing techniques for data preprocessing and (ii) a series of machine learning algorithms for the classification of information.
- The machine learning model built in the first phase is intended to be used in the second phase (big data phase) to analyze the same data source, but at a much larger volume.
- The machine learning model is not directly related to the application of certain algorithms for its use, which makes it a versatile model to adopt.

Therefore, the framework where this research is developed is multidisciplinary by combining diverse scientific disciplines with a same purpose. Therefore, to solve the problem of unstructured data analysis of social networks requires the union of heterogeneous techniques from various areas of science and engineering. The research methodology for the preparation of this doctoral thesis consisted of the following:

1. **State of the Art:** It presents a selection of studies where other authors in the Big Data, Machine Learning and Natural Language Processing areas have done research about them, as well as the union of these topics with sentiment analysis and social network rating systems. It also presents a comparison that integrates the mentioned topics with the purpose of knowing the state of the art in terms of what other authors have proposed in their studies by combining the three areas covered by the framework.
2. **State of the Technique:** In this phase, the various elements that make up the framework were analyzed, presenting a theoretical retrospective about. More technical issues are addressed, presenting an overview of the technologies that are being used in current research.
3. **Proposed Solution:** In this phase, the proposed framework is presented analyzing it from two perspectives: the theoretical aspects that each phase comprises and the aspects of implementation, where topics as complexity of carrying out each phase in a real situation are addressed.
4. **Evaluation and Validation**: A series of tests are defined to verify the hypotheses established at the beginning of the research, to demonstrate the validity of the proposed model.
5. **Documentation and Conclusions:** This activity consisted of documenting all the aspects related to this thesis and presenting the conclusions that emerge at the end of the research.

Therefore, a framework was built including two phases that perform the analysis of a set of unstructured data, a distinction of this framework is the construction of a machine learning model during the first phase, which aims to serve as a basis in the second, characterized by the processing of large volume of data. In order to validate this thesis, Yelp data was used, specifically in the hotel sector. Likewise, the framework was evaluated by executing several tests using machine learning classifiers, obtaining high prediction percentages in the binary search carried out both in the non-big data and the big data environment.

The conclusions obtained after having designed the framework, as well as having analyzed and validated the results obtained show that the presented model is capable of analyzing unstructured data of social networks both on a smaller scale (not big data) and a higher scale (big data) of analysis. On the other hand, interesting challenges and future lines of research arise after having completed the model for both extending it to the

analysis of another type of information, as in the aspect of integration and adaptation of the machine learning model from the first to the second phase.

Índice general

Índice de figuras

Índice de tablas

Acrónimos

Adam. Adaptive moment estimation.

API. Application Programming Interfaces.

BD. Big Data.

BDA. Big Data Analytics.

BOW. Bag of words.

BSD. Big or Small Data.

CNN. Convolutional Neural Networks o Redes neuronales convolucionales

CRM. Cluster Resource Manager.

DFS. Distributed File System.

GVDNE. Grandes Volúmenes de Datos No Estructurados.

IA. Inteligencia Artificial.

IoT. Internet of Things.

LBFGS. Limited-memory Broyden–Fletcher–Goldfarb–Shanno.

LDA. Latent Dirichlet Analysis.

LR. Logistic Regression.

LSVC. Linear Support Vector Classification.

ML. Machine Learning o Aprendizaje Automático.

MLM. Machine Learning Model o Modelo de Aprendizaje Automático.

MLP. Multi-layer Perceptron.

MNB. Multinomial Naïve Bayes.

NB. Naïve Bayes.

PLN. Procesamiento de Lenguaje Natural.

NLTK. Natural Language Toolkit.

RNA. Redes Neuronales Artificiales.

SGD. Stochastic Gradient Descent.

SVC. Support Vector Classifier.

SVM. Support Vector Machine.

TFIDF. Term Frequency – Inverse Document Frequency.

UGC. User Generated Content.

Capítulo 1. Introducción

El objetivo principal de la presente investigación es la definición de un framework que mediante técnicas de aprendizaje automático, procesamiento de lenguaje natural y big data sea capaz de analizar datos de redes sociales mediante algoritmos supervisados de machine learning para extraer la información que normalmente no se obtiene mediante técnicas convencionales. Este capítulo realiza una descripción del problema del análisis de datos mediante las técnicas referidas.

Una vez que se ha presentado lo anterior, se explica la motivación de llevar a cabo este trabajo y se definen los objetivos a conseguir a lo largo de la investigación, describiéndose también las hipótesis de partida establecidas para esta tesis doctoral. Posteriormente, se propone el framework como alternativa de solución al problema descrito mencionando de manera resumida las ventajas de este. Finalmente, se describe la metodología seguida en el desarrollo de la tesis para conseguir los objetivos y se presenta un resumen de la estructura del resto del documento.

1.1. Contexto

Con el desarrollo de la Web 2.0, el Internet de las cosas (IoT) y el uso masivo de los dispositivos móviles, la cantidad de información que existe en el universo digital es algo tan grande que merece ser considerado, ya que casi contiene la misma cantidad de bits de información como estrellas en nuestro universo físico (Gantz & Reinsel, 2011). A partir de la integración de tales sistemas o entes tan heterogéneos se seguirá creando y aumentando cada vez más el volumen de la información. En 2014 la International Data Corporation presentó una estadística que asegura que el universo digital se está duplicando cada dos años, y que alcanzará un tamaño de 40 zettabytes en 2020, habiendo partido de 4,4 zettabytes en 2013 (Turner, Gantz, Reinsel, & Minton, 2014).

Diversas técnicas han sido creadas para poder gestionar, almacenar, manipular e interpretar los inmensos volúmenes de información, la más popular es BD (en lo sucesivo referido como GVDNE por Grandes Volúmenes de Datos No Estructurados), una disciplina que involucra toda una serie de conceptos y técnicas novedosas para poder hacer frente a tales demandas del mercado. Se dice que los datos están no estructurados ya que no están organizados ni almacenados siguiendo los métodos establecidos por los modelos de bases de datos tradicionales, siendo el más conocido de estos el Modelo

Relacional. Respecto al acrónimo GVDNE es importante destacar lo que este denota dentro del marco de esta tesis: se le refiere como el conjunto de datos empleado en esta investigación el cual no tiene un volumen cercano a lo considerado típicamente como BD, pero que sin embargo son tratados (almacenados y analizados) mediante este tipo de técnicas

Son varias las acepciones que se han formulado respecto a la definición formal de BD, pero este es algo tan grande y complejo de tratar a través de técnicas tan diferentes en el sector de las tecnologías de la información, que en ocasiones pareciera que los autores se contradicen y en otras que dichas definiciones están incompletas. Mauro, Greco, & Grimaldi (2015) se avocan al estudio de las definiciones que se le han dado y lo definen así: “Big data representa los activos de información caracterizados por volúmenes, velocidades y variedades tan altos que requieren métodos analíticos y tecnologías específicas para su transformación en valor”.

GVDNE no tiene que ver únicamente con el almacenaje y recuperación de los grandes volúmenes de datos, justamente son los datos los que se consideran en el siglo XXI como el “nuevo oro”, es decir, quien posee tales datos y principalmente sabe cómo interpretarlos y darles un uso es el que tiene en sus manos una riqueza. No obstante, la mayoría de los datos almacenados bajo el esquema GVDNE no son analizados o interpretados, resultando en que ningún valor real se obtenga de ellos, por lo que una de las motivaciones de realizar este trabajo es proporcionar a científicos e investigadores un modelo que facilite no sólo en obtener el valor oculto en los datos, sino que ayude a generar predicciones sobre los mismos.

Otra de las tendencias que marca el mercado es que los GVDNE deben contribuir a los enfoques tanto analítico como predictivo. El análisis de los datos es el conjunto de técnicas que permiten obtener una interpretación real y fidedigna de los datos, de tal forma que el valor que estos faciliten a la toma de decisiones o a la conclusión de estudios sea preponderante. La predicción es el área de GVDNE que reúne los datos históricos, los datos en tiempo real y los datos de terceras fuentes para construir pronósticos de lo que sucederá en determinado ámbito de interés científico en los próximos meses, semanas, días e incluso horas.

Para el desarrollo de esta tesis se estableció desde el inicio, poder conseguir y tener acceso a conjuntos de datos que fueran del tipo GVDNE. Tras un periodo de búsqueda en Internet, se encontró que la empresa de Internet Yelp publica semestralmente un subconjunto de sus conjuntos de datos (datasets) los cuáles se encuentran todos en el idioma inglés. La razón de este idioma es que el dataset se expone para ser utilizado en diversos concursos académicos y científicos, por lo anterior, el inglés se emplea como el

idioma base de la competición y en consecuencia es el idioma base de la información textual que se recuperó para esta tesis.

En esta tesis se emplea el PLN para realizar minería de datos en los datasets de Yelp que, como se ha explicado, se encuentran en idioma inglés. PLN es un dominio interdisciplinario que combina las tecnologías de la ciencia computacional (como la lingüística, el ML y la probabilidad y estadística), con el objetivo de hacer posible la comprensión y el procesamiento asistidos por ordenador (Gudivada, Rao, & Raghavan, 2015). Esta es un área de gran interés para la ciencia informática ya que permite, por decirlo de alguna manera, habilitar a los ordenadores a entender el texto. Se integra también en este estudio la aplicación de las técnicas de ML, dichas técnicas se aplican a lo largo de la tesis ya que los algoritmos ML que son empleados para el análisis de datos de texto, son empleados tanto en la fase previa a la aplicación de las herramientas GVDNE como durante esta misma fase.

El análisis predictivo de grandes volúmenes de información es una de las áreas con mayor auge dentro del sector de los GVDNE y que cada vez tiene mayor importancia para la comunidad científica. Existen diversos trabajos que abarcan este tipo de problemas desde distintos enfoques, pero esta tesis doctoral aporta un modelo que permite conjuntar las técnicas más adecuadas para el tratamiento de este tipo de información en un conjunto de datos multidominio. Estas técnicas comprenden, entre otras, el uso de ML (algoritmos ML aplicables al tratamiento de información textual supervisada), BD (técnicas que amplían o mejoran los métodos anteriores para el tratamiento de cantidades ingentes de datos) y PLN (*tokenización*, palabras vacías, extracción de entidades). Por lo que, en resumen, son estas tres disciplinas científicas: ML, BD y PLN los ejes sobre los cuáles se centra y dirige esta investigación.

Dado que el dataset permite identificar los dominios a los cuales pertenecen los datos, es factible que los métodos empleados para el análisis de datos de un dominio en particular puedan ser replicados hacia otros dominios, no solamente variando los datos de entrada, sino modificando las variables de prueba sin que esto altere el núcleo central del framework propuesto. De esta manera los métodos y procedimientos realizados a lo largo de esta tesis podrán ser replicados por otros investigadores y estudiantes hacia estos mismos dominios, o incluso dominios distintos si se cuenta con la cantidad de datos necesaria.

1.2. Planteamiento del problema

La cantidad de información que se genera a través de diversos medios tales como Internet, dispositivos móviles y sensores entre otros, plantea retos en el envío, almacenamiento, procesamiento y estructuración de la información. La interacción que

se da entre los usuarios de Internet a raíz de la Web 2.0, el 5G y tecnologías venideras, hace pensar que los nuevos comportamientos humanos podrán comprenderse mejor a través del análisis de la información de lo que ellos mismos van generando. La información generada actualmente a través de los medios mencionados crece aceleradamente día a día, y como tal, es necesario ir creando modelos que permitan la gestión y el análisis de tales volúmenes de información.

GVDNE es algo tan complejo que es imposible de procesar y trabajar con los sistemas tradicionales y las herramientas comunes de datawarehousing (Ishwarappa & Anuradha, 2015). Los sistemas de bases de datos tradicionales no pueden hacer frente a las cargas de gestión de información como el almacenamiento y, sobre todo, la recuperación de información. Es decir, que estamos ante un escenario en el que es cada vez más requerido brindarle al usuario un conjunto de información muy corto que necesita ser extraído de enormes bancos de datos. Esto conlleva la adecuada gestión de la memoria disponible y el empleo de técnicas que accedan de forma rápida y eficiente a los datos.

Los modelos de predicciones revisten gran importancia en la comunidad científica, a través de variadas técnicas estadísticas se han logrado pronosticar las tendencias de estudios tanto cualitativos como cuantitativos. La informática ha hecho uso de estos modelos de predicciones para distintos quehaceres del saber humano, tanto en áreas científicas y de alta investigación, como en áreas comerciales. GVDNE es un área que por su naturaleza de estudio tiene que utilizar la información y el análisis para generar modelos también predictivos.

1.3. Motivación

Al conjuntar las disciplinas presentadas anteriormente, estamos ante un escenario en el que se podrán interpretar y analizar grandes volúmenes de datos. El análisis predictivo extrae la información de los conjuntos de datos con que se trabaja en esta tesis para determinar los posibles patrones y realizar las predicciones sobre los datos. Los modelos predictivos se emplean para analizar los datos actuales y los hechos históricos para entender mejor a los usuarios, también se identifican los riesgos potenciales y las oportunidades (Gordon, 2016).

El análisis predictivo ayuda a responder preguntas importantes tales como ¿por qué evalúan los usuarios de tal manera a cierta entidad de estudio? ¿cómo ha sido el comportamiento de dicha entidad de estudio en los últimos años y cómo se prevé la evolución de ésta en el futuro cercano? ¿Cuáles son los gustos actuales en determinado mercado y cuáles serán los gustos futuros de estos mismos usuarios en dicho mercado? De tal manera que el análisis predictivo no puede garantizar un resultado, pero sí puede establecer una tendencia certera.

El análisis de la información de GVDNE proveniente de redes sociales es un área que ha cobrado gran relevancia en tiempos recientes. No obstante, dicho análisis involucra la integración de diversas técnicas las cuáles son, en ocasiones, muy complejas de formular y aún más de llevar a cabo, ya sea por capacidad del equipo investigador o la limitante de no contar con una infraestructura adecuada para realizar la investigación a gran escala. Por lo que el framework propuesto en esta tesis está encaminado a simplificar las etapas que conlleva el análisis referido mediante un enfoque menos complejo, pero a la vez más simple de poner a prueba; todo esto con el enfoque de obtener información de gran valor.

A lo largo del presente trabajo se hace referencia a la palabra framework para referirse de una manera muy concreta al trabajo de tesis. Dado que la palabra framework puede referirse a múltiples situaciones y contextos científicos, se procede a dar una explicación, si bien abstracta, a lo que en el contexto de la presente tesis puede referirse. El framework al que esta tesis hace referencia es un contexto de trabajo que enmarca las técnicas esquematizadas anteriormente, el flujo de trabajo dentro de las fases de cada una y la misma comunicación entre estas técnicas, produciendo a la postre un determinado conjunto de salidas que satisfarán las condiciones requeridas.

1.4. Objetivos

1.4.1. Objetivo general

El objetivo final de esta tesis doctoral es definir un modelo de trabajo que combine el uso de técnicas empleadas en el aprendizaje automático, el análisis predictivo de datos no estructurados y el procesamiento del lenguaje natural, para poder establecer predicciones basadas en el análisis y la comparación de datos cualitativos contra cuantitativos.

Para poder alcanzar este objetivo, es necesario emplear las técnicas de ML para encontrar y definir la correlación que existe entre opiniones textuales y la calificación numérica relacionada con dicha opinión. Posterior a esto se aplica el análisis predictivo en GVDNE mediante la aplicación del MLM obtenido en la fase previa. Para lo cual se han establecido los siguientes objetivos:

1.4.2. Objetivos específicos

1) Establecer el Corpus para el cual se definirán las reglas de valoración. En primera instancia se tendrá que investigar en el estado de la técnica el área más factible para poder llevar cabo la investigación.

2) Identificar y recopilar las fuentes de datos de tipo texto en las que se realizarán los estudios. Habiendo identificado el dominio idóneo para obtener las fuentes de datos se proceden a llevar a cabo las siguientes actividades:

- Analizar cuáles son los proveedores líderes en el mercado que albergan el tipo de información con volúmenes cercanos a los GVDNE.
- Identificar si el formato y la estructura de la información que se obtenga por parte de esos proveedores puede ser empleada para fines de la investigación.
- Establecer los métodos necesarios para obtener la información de estas fuentes.
- Determinar si los métodos del punto anterior son los más factibles para el desarrollo de la tesis.

3) Aplicar técnicas de PLN para el preprocesamiento de la información. Las tareas que involucran dar cumplimiento a este objetivo son:

- Investigar sobre los frameworks de PLN que ya son empleados por la comunidad científica y que además permiten su integración dentro de herramientas para el análisis de GVDNE.
- Explorar en el estado del arte sobre las técnicas comúnmente empleadas para el procesamiento de datos textuales.

4) Analizar la información obtenida en el punto anterior mediante las técnicas de ML que se adapten mejor a los conjuntos de datos. Para ello es necesario aplicar las técnicas de ML a un conjunto de datos de prueba tanto para encontrar los algoritmos más idóneos para los datos disponibles, como para entrenar estos algoritmos y que generen las predicciones más acertadas posibles.

5) Definir la estructura lógica y física para la integración de los datos que permita su tratamiento en una arquitectura GVDNE. Para ello habrán de resolverse las siguientes cuestiones:

- Indagar en el estado del arte los modelos propuestos en esta área para conocer las mejores prácticas existentes.
- A partir de lo anterior, construir el modelo de la arquitectura que mejor satisfaga las condiciones requeridas por el estudio.
- Identificar la infraestructura de hardware y software disponible en la que pueda ser implementada la arquitectura GVDNE propuesta.

6) Emplear las técnicas de análisis de GVDNE para obtener predicciones sobre la información definida anteriormente. Para lograr este objetivo se debe integrar el MLM obtenido en la fase previa, para poder replicarlo hacia los métodos propios de GVDNE.

7) Realizar una comparativa entre el MLM obtenido en la primera fase y el aplicado en la segunda fase para contrastar los resultados obtenidos en ambas fases. Para ello se ejecutan diversas pruebas empleando los mismos clasificadores en ambas etapas y entonces comparar el rendimiento de cada uno, con lo cual se espera validar las hipótesis establecidas.

1.5. Hipótesis

Las hipótesis de partida que se han formulado de acuerdo al planteamiento inicial del problema y los objetivos propuestos son las siguientes:

1. Es posible definir un framework en el que se puedan aplicar diversos indicadores de estudio independientemente del volumen de datos a analizar.
2. Las técnicas de análisis de aprendizaje automático propuestas por el framework detectan patrones que permiten establecer predicciones sobre los datos de cualquier volumen.
3. Los modelos de aprendizaje automático definidos para trabajar en un entorno con un volumen reducido de información se pueden extrapolar a un entorno GVDNE para realizar el mismo tipo de análisis.

1.6. Justificación

El análisis predictivo de grandes volúmenes de información es una de las áreas con mayor auge dentro del sector del GVDNE y que cada vez será de mayor importancia para la comunidad científica en informática. Existen diversos trabajos desarrollados que abarcan este tipo de problemas desde distintos enfoques, pero en esta tesis doctoral se aporta un modelo que permita conjuntar las técnicas ya mencionadas para el tratamiento de un conjunto de datos que, si bien es multidominio, las técnicas se aplican al dominio de la hotelería, pero puede replicarse hacia otros dominios.

Por lo anterior, se busca proveer a la comunidad académico-científica de un framework que permita realizar el análisis de la información de GVDNE para, a su vez, definir un MLM que pueda ser empleado para extraer información oculta en datos de redes sociales, misma que sea capaz de aportar valor tanto a ámbitos científico-académicos como empresariales-comerciales. De esta manera los métodos y

procedimientos desarrollados a lo largo de esta tesis podrán ser replicados por investigadores, científicos y profesionales hacia otros dominios.

¿Por qué es importante la definición del framework?

Porque es importante aportar un modelo al estado del arte que permita establecer una metodología para el análisis de GVDNE mediante la integración de diversas técnicas que permitan descubrir la información oculta en los datos. De tal manera que la principal aportación del framework es un MLM que podrá ser empleado tanto en entornos de cómputo no BD como BD. La intención de crear el MLM en dos fases distintas es que en la primera se obtenga un mayor conocimiento acerca de los datos, además de que esta fase es independiente de la segunda, es decir, que por sí sola puede realizar predicciones sobre los datos. Mientras que la segunda fase, aunque depende de la primera, contempla un despliegue más rápido de resultados al aprovechar los métodos creados en la fase anterior.

¿Por qué utilizar técnicas de ML, PLN y GVDNE?

Por una parte, los algoritmos de ML aprenden sobre los datos y mientras mayor cantidad hay de estos, mayor es su capacidad de aprendizaje. Por otra parte, el integrar PLN al estudio, permite la preparación del corpus previo a la fase de análisis mediante ML, de tal manera que es mediante PLN que se transforma el corpus original para que las técnicas de ML puedan procesarlo. Mientras que las técnicas de GVDNE permiten el procesamiento llevado a cabo tanto por aquellas de ML y PLN pero a una mayor escala, siendo esto importante en la era en que está generando los mayores cúmulos de información.

¿Por qué el análisis de datos mediante ML, PLN y GVDNE se empleará para establecer predicciones sobre la información?

Porque al integrar estas técnicas se obtendrán mayores aciertos al llevar a cabo las predicciones sobre los datos. No obstante que existen otras técnicas (matemáticas, estadísticas, etc.) para analizar este tipo de información, las cuales están probadas y documentadas en el estado del arte, lo cierto es que la tendencia reciente es emplear las técnicas propuestas en la tesis para generar las predicciones que aporten un mayor conocimiento sobre el valor oculto en los datos así como poder escalar dicho análisis hacia cantidades ingentes de información, sin que esto conlleve una degradación en la calidad del análisis de dicha información

¿Qué aportaciones tendrá esta tesis doctoral?

La información que se genera a través de la interacción de los usuarios de redes sociales principalmente por sus opiniones representa un valor. Pero este valor por lo general no es explotado para encontrar qué hay más allá de lo ya expresado, y cuando se le explota, no se realiza desde diversos enfoques. Por lo que se hace necesario aportar al estado de la técnica modelos que permitan el análisis y la interpretación de la información. El presente trabajo aporta a la ciencia informática un framework de análisis predictivo que a través de un MLM podrá analizar información supervisada textual y categórica extraída de redes sociales. Dicho modelo podrá ser empleado tanto en ambientes BD como no BD.

1.7. Metodología

Este apartado detalla las actividades en las que se divide la investigación para alcanzar los objetivos expuestos en esta tesis doctoral, como se detalla a continuación:

1. Estado del arte. El principal objetivo de esta actividad consiste en detallar los trabajos realizados tanto con técnicas de ML, BD y PLN, en el ámbito del establecimiento de predicciones a través del análisis de información supervisada no estructurada. Los principales temas a desarrollar son:

- Antecedentes de la aplicación de soluciones basadas en ML, BD y PLN dentro del dominio estudiado.

- Análisis crítico de las principales alternativas clásicas para la resolución de problemas de datos no estructurados.

2. Recolección de datos. El principal reto de la investigación asociado al área de BD es la recolección de datos a partir de diferentes fuentes. Por ello, en esta etapa se procede a emplear las técnicas de recolección necesarias para allegarse de la mayor información posible.

3. Definición del framework. El trabajo a realizar por el autor en este apartado consiste en definir los elementos que conforman el framework, para su posterior aplicación mediante el empleo de las técnicas referidas anteriormente hacia la información recolectada en el punto anterior.

4. Desarrollo de la primera fase del framework. La primera fase del framework implica diversas etapas que van desde la obtención de la información hasta generar el MLM

teniendo un mayor énfasis las actividades enfocadas al procesamiento y análisis de la información mediante técnicas de ML y PLN.

5. Desarrollo de la segunda fase del framework. La segunda fase del framework contempla la definición de una arquitectura básica de BD para poder realizar el análisis de GDVNE, no obstante, el mayor trabajo a desarrollar en esta fase es la integración del MLM obtenido en la primera fase para comprobar la eficacia de los métodos desarrollados en el análisis de la información. Posterior a esto, la etapa de visualización permite mostrar los resultados de esta etapa de forma gráfica para una mejor interpretación de la información.

6. Evaluación y Validación. Las actividades de evaluación y validación incluidas pretenden estimar la efectividad de las diferentes soluciones planteadas durante la investigación. En esta etapa se evalúan las técnicas que se hayan empleado en las dos fases propuestas del framework y se comparan los resultados obtenidos en ambas para comprobar la efectividad del modelo y la aceptación o rechazo de las hipótesis.

7. Documentación y Conclusiones. En esta actividad se documentan cada uno de los aspectos desarrollados durante la tesis. Se describen las alternativas seguidas en cada etapa de la investigación, así como sus contribuciones a los resultados obtenidos. Por último, se presentan las conclusiones del estudio a través de la validación del framework.

1.8. Estructura de la Tesis Doctoral

Este trabajo de tesis doctoral está planteado para su desarrollo de acuerdo a la Figura 1.1. El presente documento ha sido estructurado en una serie de capítulos como se detalla a continuación:

- **Estado del Arte.** En este capítulo se revisan los trabajos realizados en las áreas de ML, PLN y BD. Se expone para cada área los antecedentes de cada una de estas mencionando los aspectos más relevantes que además forman la base de este trabajo doctoral, así como los trabajos resultantes de la intersección entre las áreas.
- **Estado de la Técnica.** En este capítulo se presentan los aspectos técnico-científicos que forman parte de las etapas descritas en las dos fases del framework.
- **Solución Propuesta.** En este capítulo se describe cuál es el problema que se pretende resolver, por lo que se presenta entonces el framework como una alternativa de solución a dicho problema. Para cada etapa del framework se detallan dos situaciones: por un lado, se presentan los aspectos teóricos que fundamentan la presencia de dicha etapa y cómo contribuye a la siguiente etapa que le sucede. Por el otro, se presenta la aplicación real de dicha etapa y cómo

contribuye de forma directa a la solución del problema demostrando la integración de las distintas partes para alcanzar un objetivo común.

- **Evaluación y Validación.** En este capítulo se evalúa el modelo propuesto a través de la validación de los métodos empleados en las dos fases del framework y comparando las alternativas de las técnicas de ML y PLN que se emplearon en cada caso.
- **Conclusiones y Futuras Líneas de Investigación.** En este capítulo se presentan las conclusiones de la investigación realizada y se proponen las líneas de trabajo en las que puede continuarse el trabajo desarrollado.

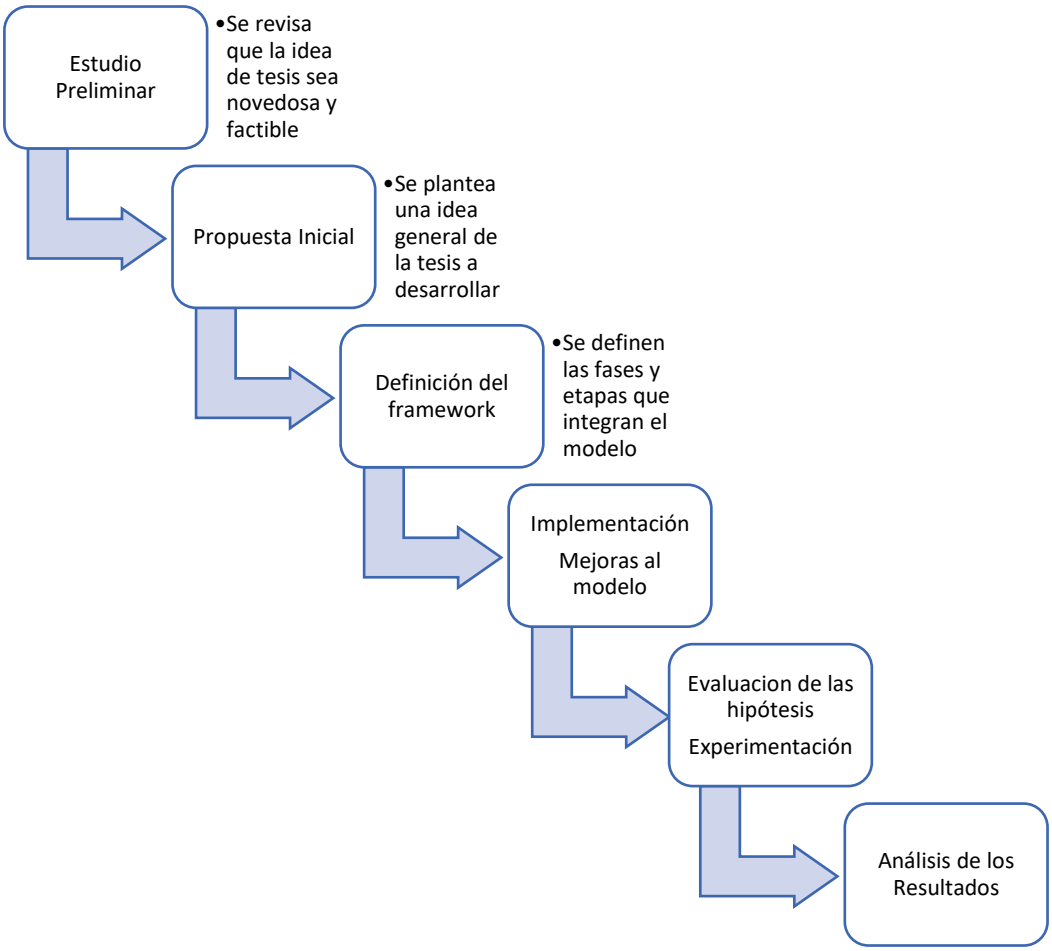

Figura 1.1: Etapas de la investigación

Capítulo 2. Estado del Arte

En este capítulo se presentan las tres áreas de investigación que se abarcan de forma transversal en esta tesis: el machine learning, el big data y el procesamiento de lenguaje natural. Para este propósito se presentan de forma introductoria cada una de estas áreas para conocer su descripción y características. Posterior a esto, se presentan cada una de estas áreas relacionadas con respecto a otra, para lo cual en cada caso se realizó una investigación en el estado del arte para conocer los trabajos más importantes que involucraran las áreas respectivas.

A continuación, se enlazan las tres disciplinas científicas descritas para dar a conocer algunos de los trabajos existentes en el estado del arte que se han llevado a cabo integrando tales disciplinas; se menciona en este caso, la pertinencia de llevar a cabo esta tesis doctoral ubicándola dentro del estado del arte. Al final del capítulo se desarrolla la discusión sobre estos temas para dar una perspectiva del presente y futuro de la investigación en estas áreas.

2.1. Introducción

Desde que se ha producido la globalización de Internet 2.0 y el IoT, la cantidad de datos que se genera día a día crece exponencialmente. El BD permite a la comunidad científica llevar a cabo gran diversidad de estudios en casi cualquier disciplina, ya que existen ahora volúmenes de información con los que era impensable contar antes. Además, el sector de BD está proporcionando herramientas analíticas y predictivas a especialistas para el tratamiento de estos datos. En esta sección se presenta el estado del arte de diversos elementos que conforman la investigación, el objetivo común que comparten y algunos ejemplos de los campos científicos en los que pueden aplicarse.

Existen diversas definiciones de BD y no hay un acuerdo sobre cómo definir algo que crece tan aceleradamente cada segundo. Se prefiere definir el término BD como el trabajo de reunir, organizar, limpiar y asegurar la privacidad de enormes conjuntos de datos procedentes de diversas fuentes para obtener información valiosa de tales datos. Mauro et al. (2015) han hecho un análisis más profundo de lo que el término significa basado en la definición de muchos autores. ¿Es BD una moda en informática? el futuro sigue siendo

incierto. Según Özköse, Sertac, & Gencer (2015) a partir de 2012 el interés y el número de estudios que se han realizado a través, o relacionados con BD ha aumentado cada año.

El BD tiene un alto potencial para emplearlo en investigación, pero primero se necesitan dos cosas: (1) pensar en cuál es el valor exacto que desea obtener de los datos, y (2) utilizar inteligentemente las herramientas apropiadas para establecer una arquitectura que conduzca a responder las preguntas previamente formuladas. Existen conjuntos enormes de datos de medios sociales que contienen información de comentarios en línea sobre varios tipos de actividades de negocios. El valor que se espera obtener de estos datos es encontrar patrones y comportamientos de usuario que puedan ayudar a predecir indicadores tales como: preferencias, disgustos, tendencias actuales, tendencias futuras, etc.

Por otra parte, el comercio en línea constituye actualmente uno de los mayores activos para las grandes corporaciones, de hecho, muchas de las más exitosas startups están basados en los comentarios en línea vertidos por los usuarios pertenecientes a determinada red social. Esas expresiones y la posterior interacción entre usuarios se están convirtiendo en un factor clave para encontrar un consenso común sobre un tema determinado a través de millones de registros en datos no estructurados[1].

2.2. Áreas de la Investigación

La presente investigación se puede entender desde la unión de tres grandes disciplinas científicas informáticas: el ML, el PLN y BD (este último compuesto por BDP y BDA). En el diagrama de Venn de la Figura 2.1 se representan las áreas de investigación que se relacionan en la tesis.

[1] Este apartado es la adaptación de un extracto del artículo en (Jimenez-Marquez, Gonzalez-Carrasco, & Lopez-Cuadrado, 2018) cuyo autor lo es también de la tesis.

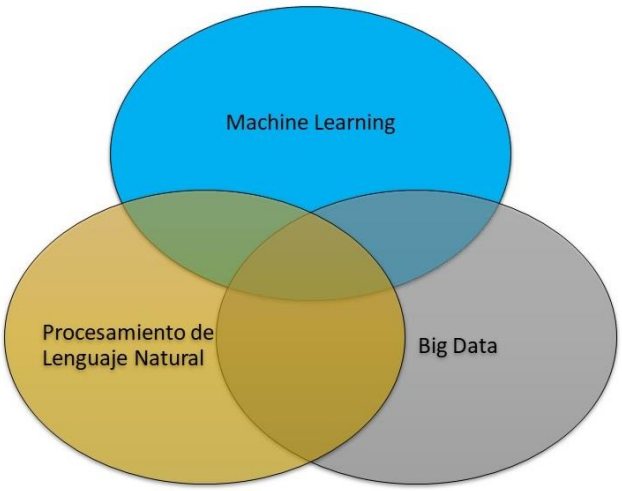

Figura 2.1 Áreas de investigación que se abarcan en la tesis

A lo largo de este capítulo se exploran estas áreas tanto de forma individual como de forma conjunta, además se presenta la relación que estas guardan con la tesis. De manera general se menciona la importancia que cada una tiene para la investigación:

- el ML es la disciplina científica que aporta un grupo de algoritmos que analizan patrones, predice escenarios o clasifican información en base a un conjunto de datos de entrenamiento.
- el PLN proporciona un conjunto de técnicas que permiten preparar un conjunto de expresiones dadas en lenguaje natural para que estos puedan ser leídos y procesados por un ordenador, además de estudiar las relaciones entre el lenguaje humano y los ordenadores.
- el BD aporta los medios para procesar, gestionar y almacenar grandes conjuntos de información, además de realizar análisis sobre estos para establecer predicciones o tendencias.

En la Figura 2.2 se presenta un esquema donde se presenta que esta tesis se encuentra relacionada de forma transversal con las áreas mencionadas. Es importante resaltar que cada una de estas áreas tiene gran profundidad en lo que compete a su alcance en el estado de la investigación actual. Dentro de cada área pueden desarrollarse innumerables investigaciones con diversos fines y para diversas áreas, no sólo la informática. Por lo que al reunir no sólo dos, sino las tres áreas, el campo potencial de aplicación resulta ser bastante extenso. Sin embargo, esto a su vez representa el reto de encontrar un espacio en el estado del arte donde poder llevar cabo la investigación debido a la diversidad de trabajos existentes que ya se han llevado a cabo empleando estas tres áreas. En el Capítulo 4 se demuestra cómo esta tesis contribuye a ese espacio.

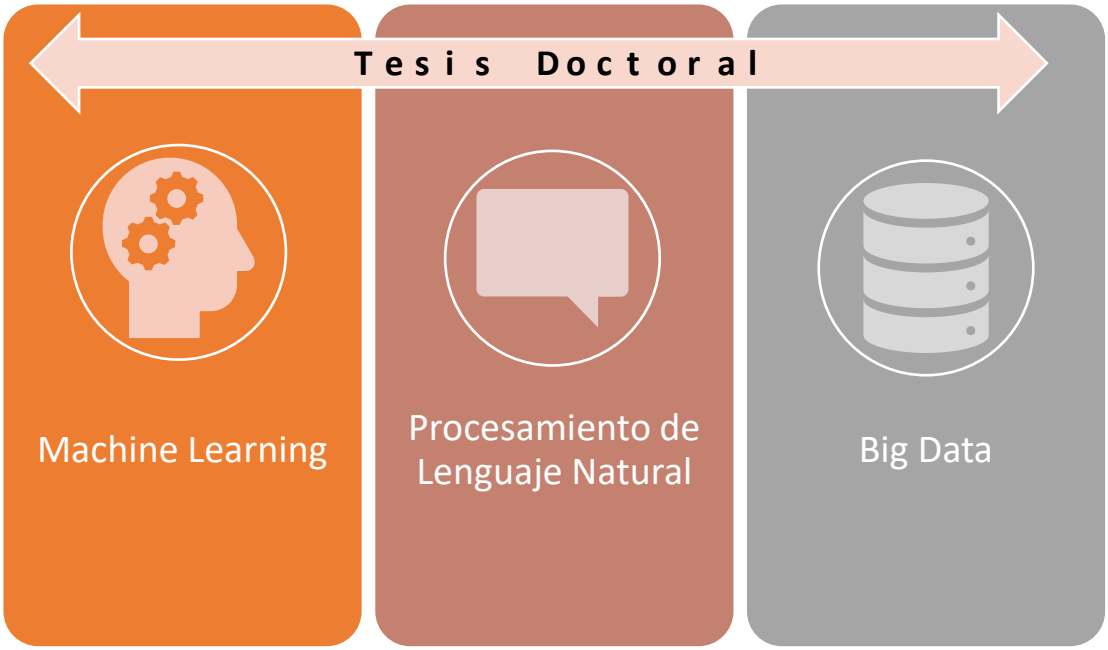

Figura 2.2 Relación transversal de la tesis con las áreas de investigación

También es importante mencionar respecto a la Figura 2.2 que, debido a la profundidad de cada una de las áreas, no se entra a fondo en estas durante la investigación debido a que existen estudios que realizan esto de forma individual, en su lugar, las áreas se conjuntan y exploran de forma transversal durante esta tesis.

2.2.1. Machine Learning

El ML es una subárea de la IA y se diferencia de esta en que tiene un mayor acceso a datos y los algoritmos de ML aprenden de estos. El ML es un área de rápida evolución, el cual su principal actividad es el diseño y análisis de algoritmos que le permitan a las computadoras aprender (Watt, Borhani, & Katsaggelos, 2016). Por su parte, Mueller & Massaron (2016) también concuerdan en el hecho de que ML se basa en emplear algoritmos para analizar grandes conjuntos de datos y además agregan que el potencial del ML es que puede realizar analítica predictiva mucho más rápido que cualquier humano, permitiéndole entonces trabajar de forma más eficaz.

Los algoritmos de ML se distinguen de otros paradigmas de programación tradicionales como la programación funcional o la programación orientada a objetos en el sentido de que los últimos están más orientados a tener una interacción con el usuario, o bien, leer datos de entrada de algún puerto periférico para ejecutar una serie de acciones que se suceden. Mientras que en el ML no se contempla tener interacción de ningún tipo, por el contrario, mientras que en los paradigmas tradicionales se busca llegar a obtener resultados, en este modelo se parte de resultados para generar un modelo de predicción o

clasificación, el cual pueda después predecir resultados a partir de nuevos datos de entrenamiento.

Comúnmente un algoritmo de ML se compone de las fases de entrenamiento y prueba, donde en el entrenamiento se proporcionan datos que pueden o no estar etiquetados, lo cual en primera instancia determina el tipo de aprendizaje que se emplea (supervisado, no supervisado, etc.), así como los clasificadores que habrán de aplicarse. Mientras que en la fase de prueba se trabaja con datos que el clasificador no había "visto" para generar predicciones sobre estos datos a partir del modelo construido en la fase previa. Algunas de las aplicaciones típicas del ML incluyen: reconocimiento de imágenes, minería de datos o robótica

2.2.2. Procesamiento de Lenguaje Natural

Como se ha descrito anteriormente, el PLN consiste en una serie de métodos y procedimientos computacionales cuyo fin principal es tomar como entrada el lenguaje humano, el cual puede haber sido voz, texto o imágenes, para que puedan ser leídos y procesados por un ordenador o cualquier dispositivo digital que tenga las funciones de un procesador de computadora, y entonces cumplir una función en particular. El PLN siempre ha estado unido a la relación humano-computador, ya sea porque necesitan "hablarse" o "entenderse" entre ambos (Yue, Di, Yu, Wang, & Shi, 2012). Se trata de una disciplina de la informática siempre en evolución y que se vuelve cada vez más compleja debido a las mayores exigencias del sector.

El PLN comenzó en la década de 1950 como la intersección de la inteligencia artificial y la lingüística (Jiayin Qi, Zhang, Jeon, & Zhou, 2016). Desde entonces diversas teorías han ayudado a la evolución del PLN, como la notación Backus-Naur Form (BNF) y herramientas como analizadores léxico y sintáctico. El estado del arte revela que el estudio del PLN está involucrado con la conjunción de técnicas como: autómatas de estado finito y de árbol, gramática libre del contexto y etiquetado de parte del discurso (Maletti, 2016), por mencionar algunos. Una de las áreas donde el PLN está colaborando y siendo más útil en la ciencia es en Medicina y Biomedicina.

Una investigación del término PLN en localizadores de bases de datos científicas conduce a encontrar numerosos artículos publicados en revistas no relacionadas a ciencias de la computación, lo cual habla sobre la enorme participación del PLN en estos campos. Otras áreas de aplicación se pueden encontrar por ejemplo en los sistemas de información

geográfica (Calì, Condorelli, Papa, Rata, & Zagarella, 2011) o entrevistas motivacionales (Tanana, Hallgren, Imel, Atkins, & Srikumar, 2016)[2].

2.2.3. Big Data

En el contexto de la investigación (Capítulo 1) se ha dado una definición de BD, dado que este término ha recibido diversas definiciones, no se profundiza en cuál es la correcta o la más acertada, dado que este debate se encuentra ya en el estado del arte. En su lugar, se exploran dos áreas de interés del BD para esta investigación como son: el BDA y el BDP. Como se ha mencionado, las industrias y organizaciones de todo el mundo están obteniendo información valiosa de las cantidades masivas de información que obtienen a través de la aplicación de técnicas de alto nivel de BD. Estas técnicas se refieren a BDA y consisten en un conjunto de algoritmos, estadística avanzada y análisis aplicado. Iqbal, Doctor, More, Mahmud, & Yousuf (2016) lo definen como: "(BDA) se refiere a las técnicas utilizadas para examinar y procesar BD de modo que los patrones subyacentes ocultos se revelan, las relaciones se identifican y otros conocimientos sobre el contexto de la aplicación bajo investigación están expuestos".

Debido al alto valor económico que tiene para las organizaciones y a la poderosa capacidad de analizar datos gigantescos, hoy en día BDA está siendo utilizado en sectores tan diversos como: gestión de la cadena de suministros (Wang, Gunasekaran, Ngai, & Papadopoulos, 2016), mercadotecnia (Erevelles, Fukawa, & Swayne, 2016) y apoyo al proceso de toma de decisiones (Elgendy & Elragal, 2016), por mencionar algunos. Las redes sociales son un gran recurso para aplicar BDA (Tan, Blake, Saleh, & Dustdar, 2013), por ejemplo: comprender las preferencias del usuario, saber las tendencias diarias, comprender el comportamiento de los usuarios con afinidades relacionadas, analizar los nuevos hábitos de la población, etc. En el futuro cercano, posteriores investigaciones y regulaciones oficiales tendrán que establecer los límites de BDA (Baruh & Popescu, 2017).

Por su parte, BDP constituye un nivel superior al de BDA en el sentido de que el primero puede hacer predicciones a partir de los resultados del análisis de datos. BDP está ayudando a las organizaciones a lograr algo que siempre han estado buscando: tomar mejores decisiones. Esto se está logrando mediante el procesamiento de grandes cantidades de datos a través de diversas técnicas (Perrons & McAuley, 2015) con el propósito de hallar patrones ocultos en tales datos y descubrir correlaciones desconocidas.

Las habilidades requeridas para desarrollar un estudio del tipo BDP son diversas, entre estas se encuentran: ML, análisis estadístico y cuantitativo (Shah, Irani, & Sharif, 2017),

[2] Este apartado es la adaptación del artículo de (Jimenez-Marquez, Gonzalez-Carrasco, & Lopez-Cuadrado, 2018), Sección 2.2, cuyo autor de este artículo es también autor de la tesis.

minería de datos (Saravana-Kumar, Eswari, Sampath, & Lavanya, 2015), visualización de datos y habilidades de programación. BDP no es un proyecto experimental o una tecnología emergente, es una realidad con éxito probado en diferentes campos, que van desde Recursos Humanos (Shah et al., 2017) a ciencias de la salud (Saravana-Kumar et al., 2015), campo en el que los autores estiman que BDP tendrá amplia aplicación en el corto plazo gracias a la gran cantidad de datos que se almacenan generan diariamente[3].

2.2.4. Análisis de sentimientos

El análisis de textos es una tarea compleja, la misma interacción verbal entre humanos es frecuentemente un tanto compleja al sucederse situaciones en las que hablante y escucha no se comprenden del todo, por lo que es aún más complejo plantear soluciones que resuelvan problemas como la ironía el sarcasmo, el doble sentido, etc. En lo que respecta al tema de tesis propuesto en el que se propone analizar reseñas o críticas de usuarios provenientes de redes sociales, citadas en ocasiones a lo largo del documento como review o reviews, puede establecerse que en la review que hace un usuario, pueden encontrarse situaciones en las que se expresa en un sentido, pero la calificación es opuesta. Es decir, la polaridad del sentimiento es presuntamente positiva, pero la valoración es negativa o la polaridad del sentimiento es presuntamente negativa, pero la valoración es positiva.

Otro de los problemas que se presentan es la cantidad de texto escrita, ya que es más complejo el poder comprender y analizar los procesos del individuo mediante el análisis de unas cuantas palabras que no llegan a formar incluso una frase coherente en ocasiones. Por otro lado, aunque pudiera pensarse que las revie*ws* que tienen una gran longitud incluyendo varias frases bien formadas llevan a un fácil análisis no es así, ya que el usuario puede estar cambiando el sentido de positivo-negativo-positivo o negativo-positivo-negativo en los casos más sencillos, los casos más complejos pueden presentar constantes cambios de sentido, haciendo más difícil el trabajo de valorar esta revie*w*.

Para justificar el por qué se incluye el AS en esta investigación, puede mencionarse el hecho de que para poder realizar un análisis de este tipo se involucran técnicas de ML y PLN, las cuáles se han referido anteriormente. Aunque los resultados que se obtienen de un estudio de este tipo pueden llegar a ser muy variados, el que se pretende efectuar en la tesis es un análisis predictivo sobre la polaridad del sentimiento que además es una de las tareas del AS como se refiere más adelante. Por otra parte, es una forma de medir la efectividad de las técnicas que se pretenden emplear, con lo cual se busca además

[3] Este apartado es una adaptación del artículo de (Jimenez-Marquez, Gonzalez-Carrasco, & Lopez-Cuadrado, 2018), Secciones 2.1.1 y 2.1.2, cuyo autor lo es también de esta tesis.

satisfacer las hipótesis de partida. Además de ser un área en constante crecimiento dentro de la investigación científica en informática como se describe a continuación:

El análisis de sentimientos (AS) es un área de reciente creación y estudio en las ciencias computacionales (Sun, Luo, & Chen, 2016) y está despertando cada vez mayor interés en la comunidad científica al ser estudiada y analizada por diversos cuerpos de investigación que aportan, cada uno desde su enfoque una interesante pieza al acervo científico del cual cada vez se desprenden más estudios. De acuerdo a Pang & Lee (2004) "el análisis de sentimientos busca identificar los puntos de vista que subyacen en un espacio de texto". Por su parte, Duric & Song (2012) señalan que el AS es un área que puede ser enmarcada como una tarea de clasificación donde las categorías son las polaridades positivas y negativas, pero el AS realiza otras tareas como: subjetividad, afecto, emoción y puntos de vista que describen o modifican a las entidades.

Como se ha mencionado, existen una serie de estudios que se han dedicado al tema del AS desde diversos enfoques, tantos que a su vez se cuenta con trabajos que clasifican y analizan el trabajo que se ha hecho en esta área. Como en el trabajo de Serrano-Guerrero, Olivas, Romero, & Herrera-Viedma (2015), en el que los autores hacen una revisión de los servicios web que llevan a cabo alguna función relacionada a AS. En este trabajo se destacan las tareas que están ligadas a AS, entre las que se encuentran: clasificación del sentimiento, clasificación de la subjetividad, resumen de opiniones, recuperación de opiniones, sarcasmo e ironía, y otras. Además, se destacan las técnicas que se han aplicado para realizar dichas tareas, entre estas se encuentran: enfoques de ML, enfoques basados en el léxico, PLN y recuperación de la información.

Dentro de este mismo enfoque, el de la revisión de los trabajo hechos sobre AS, destaca al igual que el mencionado anteriormente, el profundo estudio llevado a cabo por Ravi & Ravi (2015), en el cual los autores se dedican a la tarea de analizar 251 trabajos publicados sobre esta materia en el periodo 2002-2015. Las tareas y las técnicas referidas por Serrano-Guerrero et al. (2015) se encuentran comprendidas también en este trabajo en el cual se integra el análisis y la síntesis de 108 trabajos relacionados con el AS, además de que se refieren enfoques híbridos en cuanto a las técnicas empleadas, como ejemplo de estos se tiene el llevado a cabo por Ghiassi, Skinner, & Zimbra (2013) en el que se emplea información de Twitter para realizar AS sobre una marca en particular, en este caso, un cantante.

El área de AS es tan grande que el estudio de Ravi & Ravi refiere que se tuvo que estructurar a diferentes niveles, entre estos: documento, palabra, aspecto, sentencia, concepto, frase, enlace, cláusula y sentido. De hecho también se analizaron y clasificaron los trabajos tomando en cuenta las tareas y aplicaciones del AS, los cuáles se mencionan a continuación: 1) clasificación de subjetividad, 2) determinación de polaridad, 3)

vaguedad en texto opinado, 4) AS plurilingüe y multilingüe, 5) AS de dominio cruzado, 6) medición de utilidad de revisión, 7) detección de spam de opinión, 8) creación de corpus lexica, 9) palabra de opinión, extracción de aspectos y opinión, reconocimiento de entidades, desambiguación de nombres y 10) aplicaciones de AS.

Como se ha comentado anteriormente, Ravi & Ravi realizan la división de AS a diferente niveles entre los que se encuentra el nivel de aspecto, que es precisamente la temática principal del trabajo presentado por Schouten & Frasincar (2016). En el trabajo referido los autores se enfocan en investigar una serie de trabajos relacionados con el AS pero enfocado es la detección de aspectos. De tal manera que el AS al nivel de aspectos es un nivel muy fino y particular que permite identificar el análisis sobre una serie de aspectos que se están expresando en un documento, por ejemplo, en una estancia vacacional se podría obtener la polaridad del sentimiento sobre el estado de la piscina, el estado del aire acondicionado y el sabor de la comida. Schouten & Frasincar presentan los enfoques en el área a través de una taxonomía propia que les permite dividir los trabajos en aquellos centrados en AS, detección de aspectos o enfoques conjuntos.

Por su parte, Piryani, Madhavi, & Singh (2017) presentan un extenso estudio donde realizan un mapeo analítico de más de 300 artículos sobre las áreas de AS y minería de opiniones. En este trabajo, se presenta la evolución que ha tenido en cuanto a su desarrollo y la gran atracción que ha generado en el periodo de estudio del año 2000 al 2015. Se presentan además los autores más citados en el área y las revistas donde más se publican este tipo de trabajos. Por otra parte se realiza un análisis de densidad para ubicar gráficamente cuáles son las áreas (distinguieron 60) donde se concentra más la investigación en estas. La conclusión más relevante de estos autores para la tesis es la siguiente: se está empleando más el enfoque basado en técnicas de ML que en técnicas basadas en léxico.

Un enfoque distinto al trabajo antes citado es el realizado por Yadollahi, Shahraki, & Zaiane (2017) en el que los autores dividen el AS dos áreas diferentes: la minería de opiniones y la minería de emociones. La primera se refiere a cómo determinar la actitud de un escritor hacia un tema, mientras que la segunda está involucrada en la detección y clasificación de las emociones de los escritores hacia eventos o temas y para ambas tareas destacan el empleo de técnicas de PLN y ML. Dentro de las tareas de minería de opiniones destacan las siguientes: detección de subjetividad, clasificación de polaridad de opinión, detección de spam de opinión, resumen de opinión y detección de expresión de argumento. Mientras que para la minería de sentimientos resaltan las siguientes: detección de emociones, clasificación de polaridad de emociones, clasificación de emociones y detección de causa de emociones.

En un enfoque diferente, pero dentro de la línea de los trabajos que se han elaborado para hacer una revisión del estado del arte en el área de AS se encuentra el estudio hecho por Mäntylä, Graziotin, & Kuutila (2018). A diferencia de los trabajos citados donde los autores comúnmente hacen una revisión de los trabajos realizados en los últimos años, estos autores desarrollan un estudio sobre la evolución histórica del AS, descubriendo así un cúmulo de 6996 trabajos conteniendo este término. Según los autores, el primer trabajo en el área fue publicado en 1940, sin embargo, el 99% de los trabajos han sido publicados desde el 2004 a raíz de la expresión de textos subjetivos en la web. Por lo que estos autores (a diferencia de otros) deciden acotar el escenario de su análisis limitándose a los 20 trabajos más citados en el área. Dentro de los hallazgos relevantes se encuentran que hay concordancia con todos los trabajos citados en que las técnicas comúnmente empleadas en estas áreas son las técnicas de ML y PLN.

El deep learning ha constituido una evolución de las técnicas del ML y la IA, en diversos campos y áreas del quehacer científico tanto en la informática como en otras disciplinas científicas (Ain et al., 2017). Por lo que Zhang, Wang, & Liu (2018) realizan un estudio donde presentan las técnicas particulares del deep learning que se han aplicado en tareas de AS, entre estas: word embeddings, red neuronal autoencoder, CNNs, redes neuronales recursivas, red de memoria a corto y largo plazo, redes neuronales de memoria, o redes neuronales recurrentes. Muchas de estas técnicas de aprendizaje profundo han mostrado resultados de vanguardia para varias tareas de AS (las cuáles concuerdan con lo que otros autores han resaltado en los trabajos citados anteriormente), por lo que se espera en el futuro cercano que la aplicación de estas técnicas aporten importantes descubrimientos en la investigación de AS.

Sin embargo, la investigación realizada sobre AS no solamente ha contemplado el estudio de texto (a cualquier nivel) mediante las técnicas de ML, PLN o deep learning, sino que también puede desarrollars sobre medios que combinan texto, voz-audio (discursos), video (visual) o multimodal (fusión de las anteriores). Sobre esta última Soleymani et al. (2017) presentan una investigación sobre el AS multimodal, en la cual refieren tanto de forma individual como de manera conjunta parte de la investigación que se ha llevado a cabo en AS en cada uno de los tipos de medios referidos. De igual manera, presentan las técnicas que se han empleado para cada estudio referido y los retos y perspectivas que se tienen para la investigación del AS multimodal.

AS es un área que ha ganado tanta atención en el mundo científico que desde hace algunos años se celebra anualmente la competencia “International Workshop on Semantic Evaluations” o SemEval en donde diversos equipos de centros de investigación y universidades de todo el mundo compiten en diversas categorías para trabajar en tareas de predicción y análisis de tweets. En este apartado se realiza un resumen de las

competiciones de 2016 y 2017 (Nakov, Ritter, Rosenthal, Sebastiani, & Stoyanov, 2016; Rosenthal, Farra, & Nakov, 2017).

Como se ha mencionado, existen diversas categorías dentro del concurso SemEval, en las cuáles se tiene una tarea particular de predicción: definir la polaridad de un sentimiento (positivo, neutral, negativo), clasificar el tweet en una escala de dos puntos (positivo, negativo), clasificar el tweet en una escala de uno a cinco, estimar la distribución de los tweets entre uno o cinco, y estimar la distribución de los tweets en una escala de uno a cinco. Dentro de las técnicas empleadas por los equipos en ambos años destaca el uso de diversos clasificadores de ML como SVM, LR, NB o métodos de Deep Learning (CNN y c), además de emplear diversas técnicas de PLN como aquellas basadas en el léxico (Ahmad, Aftab, Muhammad, & Waheed, 2017).

Finalmente, como casos prácticos de la aplicación del AS en la hotelería que puedan servir como punto de referencia para esta tesis, se tienen los trabajos realizados por Geetha, Singha, & Sinha, 2017; y Valdivia, Luzón, & Herrera (2017; 2017) donde se analizan datos de medios sociales referentes a esta industria. En el estudio de Geetha et al. se busca establecer la relación entre los sentimientos del cliente y las calificaciones dadas a los hoteles respecto a una ciudad en la India, entre las técnicas empleadas se encuentran clasificadores de ML. Mientras que en el estudio presentado por Valdivia et al. se analizaron datos de TripAdvisor para investigar el sentimiento en opiniones encontradas de viajeros que habían visitado monumentos importantes en España; los autores concluyen que AS es un área que continuará evolucionando, ya que los retos que presenta la subjetividad en el texto están aún por superarse.

2.2.5. Sistemas de calificación de redes sociales

Un sistema de calificación o rating system puede referirse a cualquier tipo de calificación asociada a un dominio de aplicación específico, por ejemplo: películas, programas de televisión o videojuegos. Los sistemas de calificación de redes sociales (SCRS) resultan ser una derivación de los anteriores, en el sentido de que se emplea la Web 2.0 para crear plataformas sociales donde se asignan calificaciones en línea. La forma en que una comunidad de usuarios construye la reputación de una empresa a través de los SCRS puede desempeñar un papel clave en la decisión de un cliente potencial para comprar el bien o contratar algún servicio. Nema & Tang (2017) refieren que la teoría referente a esta área explica los comportamientos en SCRS “como Amazon, Yelp y Stackoverflow, donde el consenso (cantidad) y la reputación (calidad) influyen claramente en los resultados”.

La Web 2.0 es un término que involucra a los sitios web que facilitan la interoperabilidad y la colaboración entre sus usuarios. Existen diversos tipos de redes sociales donde los usuarios pueden interactuar o compartir sus experiencias de diversas formas. Aprovechando las posibilidades ilimitadas de Web 2.0 surgieron varias startups para desarrollar plataformas que permitieran a los usuarios evaluar y comentar los productos y servicios (Jimenez-Marquez, Gonzalez-Carrasco, & Lopez-Cuadrado, 2018). Los usuarios de las SCRS comúnmente dan una calificación a "algo" y de igual manera pueden hablar de su experiencia. También expresan sus sentimientos sobre la base del grado de satisfacción que obtuvieron. La relación que existe entre lo que el usuario expresa textualmente y la calificación no siempre es clara: se puede expresar textualmente que su experiencia fue de mala a terrible y en una escala de 1 a 5 estrellas pueden dar un puntaje de 4 o 5, esto también puede suceder de la manera inversa.

Allahbakhsh et al. (2014) describen los escenarios donde la reputación en los SCRS es injusta debido a la colaboración de varios usuarios, este trabajo también propone métodos para descubrir tales patrones. El grado de reputación de los usuarios es una manera de recompensar a alguien por su actividad y ayudar a otros miembros de la comunidad, lo cual aumenta su popularidad. Por otra parte, existen estudios alrededor de los SCRS sobre cómo estos ayudan a construir reputaciones sostenibles (Ekmekci, 2011) o clasificar a un usuario (Liu, Guo, Hou, Cheng, & Liu, 2015)[4].

Por otra parte, Akram, Raheman, Jagadeesh, Teja, & Krishna (2018) proponen un enfoque mediante el cual se exploran los comportamientos mostrados por los usuarios en los SCRS para predecir la calificación del servicio al usuario. Mientras que en el modelo propuesto por Chen, Fu, Yue, Liu, & Liu (2016) se emplean las evaluaciones encontradas en ciertos SCRS para transferirlas a una clasificación de las preferencias de los usuarios para los servicios mediante un cálculo elaborado con este fin. De igual manera, los autores proponen un algoritmo basado en la teoría de la elección social para obtener el ranking de servicios.

En la investigación realizada por Glenski, Stoddard, Resnick, & Weninger (2018) se creó una especia de juego denominado GuessTheKarma, que mediante una encuesta les permite explorar el desempeño de algunos medios de SCRS para analizar si los indicadores más comunes de estos medios pueden predecir verdaderamente la mayoría de opinión; este es un estudio interesante porque busca abstraer la parte social de los SCRS para descubrir realmente qué tan popular es un artículo o tema en una SCRS. Por otra parte, Kim, Han, Lee, & Park (2016) emplean los SCRS para describir escenarios futuros

[4] Algunas partes de este apartado se adaptaron del artículo de (Jimenez-Marquez, Gonzalez-Carrasco, & Lopez-Cuadrado, 2018), Sección 2.3, cuyo autor lo es también de la tesis

que permitan conducir el estado de la innovación, o la creación de nuevos productos en base a la minería de opiniones en redes sociales de este tipo.

Dentro de los trabajos que se están produciendo alrededor de los SCRS se encuentra el trabajo realizado por Singh et al. (2017) en el que los autores desarrollaron modelos basados en ML que pueden predecir la utilidad de una crítica tomando en cuenta varios atributos del mismo como: polaridad, subjetividad, entropía y facilidad de lectura. Este tipo de investigaciones resulta de utilidad ya que obtener los comentarios más útiles es una tarea que puede ayudar a otros usuarios a guiar mejor su decisión de compra de un artículo o selección de un servicio.

En un trabajo posterior del equipo de investigación anteriormente citado, Saumya, Singh, Baabdullah, Rana, & Dwivedi (2018) han tomado el SCRS de los dos sitios más populares de comercio electrónico en la India para generar un rank*ing* (clasificación) de las opiniones de consumidores en línea. En este trabajo emplean además del valor que indica la utilidad del comentario, otros atributos como la descripción del producto para realizar un listado ordenado de las críticas más útiles. En este trabajo se apoyan en técnicas de ML como el clasificador random-forest y el regresor de aumento de gradiente. De acuerdo a los autores, el introducir nuevos atributos ha llevado a mejorar su indicador de la utilidad de las críticas.

En un contexto alterno al explorado en esta tesis, Midgett, Bendickson, Muldoon, & Solomon (2017) apuntan sobre cómo uno de los SCRS más conocidos en la actualidad, Airbnb (un sistema empleado para la denominada economía compartida), puede servir para evaluar tanto a compradores y vendedores en cuanto a sus hábitos hacia la sostenibilidad del medio ambiente así como sus hábitos personales; este tipo de estudios comúnmente refieren situaciones de la sociedad moderna que han ido cambiando a través de este tipo de modelos que son posibles gracias a la proliferación de los SCRS en diversos medios, en este caso el hospedaje, pero también en casos conocidos como el de la transportación (Cabify, Uber o Blablacar entre otros).

En otro orden de ideas, los SCRS están siendo objeto de gran investigación debido a su importancia para un gran número de compañías. Entre estos se tiene la problemática de que está siendo objeto de usuarios malintencionados los cuáles buscan obtener algún provecho a partir de una mala crítica, o bien, cambiar la imagen que se tiene de la compañía a través de este tipo de prácticas. En este sentido la investigación realizada por Cao, Sun, Wang, & Li (2016) se enfoca en investigar estos comportamientos en los SCRS a través de un modelo que busca analizar y encontrar este tipo de críticas a través de alinear y comparar patrones que distinguen buenos de malos comportamientos.

2.3. Comparativa y análisis de las áreas de investigación

2.3.1. Machine Learning y Procesamiento de Lenguaje Natural

Se presentan en este apartado una serie de estudios relacionados con las áreas que involucran las áreas de ML y PLN, como se aprecia en la Figura 2.3, se resalta el área donde estas dos áreas se intersecan. Como se observa en la figura, se encontró que en la unión de estas áreas se hallaron estudios que resuelven problemas o tratan temas de SCRS y AS, no obstante, se prefirieron estas dos áreas sobre otras que se encontraron debido a que son áreas predominantes que se han considerado en la construcción de esta tesis.

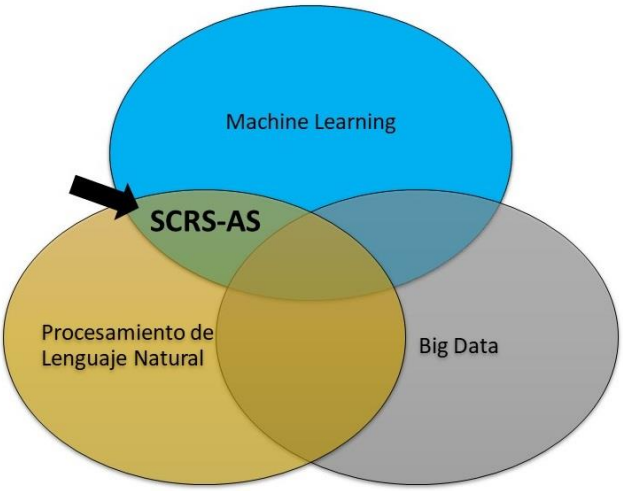

Figura 2.3 Intersección de las áreas de investigación ML y PLN

En este apartado además se exploran otras áreas afines a las citadas con el fin de explorar aquellos trabajos existentes en el área que han resuelto situaciones prácticas o propuesto nuevos modelos o enfoques dentro de las áreas de SCRS y AS, este puede referirse como el criterio de búsqueda y acotación de los resultados en base a los cuáles se citan estos trabajos. En primer lugar, se presenta el trabajo de Nadkarni, Ohno-Machado, & Chapman (2011) en el cual se realiza un estudio exploratorio hacia el área del PLN presentando un extracto sobre su historia e introduciendo algunos algoritmos de ML que han sido empleados para resolver problemas relacionados con PLN.

El estado del arte también indica que el PLN está involucrado con la conjunción de técnicas como: autómatas de estado finito y de árbol, gramática libre del contexto y etiquetado de parte del discurso, como es empleado por Maletti (2016) en tareas de análisis de sentimiento. Por otra parte, Tanana et al. (2016) emplean la conjunción de estas dos áreas para automatizar la tarea de codificar entrevistas motivacionales; en dicho

estudio se lleva a cabo una comparación de diversos métodos de PLN para lograr dicho objetivo.

En el estudio desarrollado por Felbermayr & Nanopoulos (2016), los autores se enfocan principalmente en analizar el rol de las emociones en las revie*ws* de los clientes. Además, se ocupan de dos aspectos notables en las revie*ws*: primero tomaron en cuenta el rol de las emociones y cómo éstas pueden influenciar a una crítica. Segundo, exploraron y analizaron conjuntos de datos donde los revisores también son votados por otros revisores y cómo este elemento constituye un factor cualitativo a tomar en cuenta. Los expertos del área de mercadotecnia emplean el estudio para analizar el efecto de las revie*ws* en línea. En este estudio se emplearon los clasificadores random forest.

En la investigación realizada por Gross & Murthy (2014) se emplearon técnicas de ML como SVM y la asignación latente de Dirichlet (LDA) en una investigación que involucra BD. Los autores demostraron en su estudio que el uso de LDA fue mejor que las SVM, aunque denotan que eso depende de la naturaleza del material fuente que se emplee para efectuar estudios que integren PLN. En otro sentido, Ngo-Ye & Sinha (2014) revelan que las nuevas críticas que se hacen en una plataforma social tienen pocas posibilidades de ser votadas como útiles, cuando algún producto o servicio tiene otras revie*ws* que han sido más votadas anteriormente. Dado que en su estudio desarrollan un modelo basado en las dimensiones frescura, frecuencia y valor monetario del revisor, ayudan a identificar nuevas críticas que sean de utilidad.

En un enfoque más reciente y a la vez más complejo Zhang, Du, Yoshida, & Wang (2018) emplearon RNA Convolucionales para identificar aquellas opiniones o comentarios engañosos o de calidad dudosa. Por su parte, Al-Smadi, Qawasmeh, Al-Ayyoub, Jararweh, & Gupta (2018) comparan el funcionamiento de RNA Profundas contra SVM para analizar el contenido de reviews tomadas de hoteles árabes con el fin de realizar análisis de sentimiento mediante el enfoque basado en aspectos.

Como se aprecia, cada vez más se nota la importancia de emplear RNA en investigaciones que involucran técnicas de PLN. En este sentido se puede citar el estudio de Ou, Huynh, & Sriboonchitta (2018) en el cual los autores buscan obtener atributos atractivos de los revie*ws* a partir de técnicas de RNA, habiendo empleado primero técnicas de PLN para posteriormente comparar estos métodos a su vez contra modelos estadísticos. Un ejemplo de cómo se puede generar investigación que aporte beneficios tangibles directos a la sociedad es el llevado a cabo por Zhou, Zeng, Liu, & Zou (2018) en el cual demuestran cómo empleando ML, PLN y otras disciplinas científicas como IoT se pueden ofrecer experiencias integradoras a usuarios de dispositivos electrónicos conectados a medios sociales.

Lee & Choeh (2014) por su parte, pudieron comprobar mediante el uso de las RNA la utilidad que pueden llegar a tener los comentarios, es decir, que mediante esta redes predicen qué tan útiles serán dichos comentarios. Mientras que Chang, Ku, & Chen (2017) realizan un extenso e interesante estudio en el que analizan los comentarios hechos en TripAdvisor tomando en cuenta tanto la parte cuantitativa (puntaje) como cualitativa (texto) de la crítica efectuada; además de plantear una arquitectura que permite el análisis de este tipo de información, se extraen aspectos y se detectan categorías en los datos.

En la Tabla 2.1 se presenta un resumen sobre estos trabajos donde se integran las áreas de ML y PLN, además son vistos desde un enfoque donde pudo haberse empleado para el estudio Contenido generado por el usuario (CGU) u Opiniones o calificaciones del usuario donde se evalúan productos o servicios (OCU). Se señalan de forma intuitiva las áreas que toca cada artículo (✔ o ✘).

Métodos empleados	CGU	OCU	Autores
SVM, Modelos de Markov, Campos aleatorios condicionales, N-Gramas	✘	✘	(Nadkarni et al., 2011)
Tecnología de estados finitos, Técnicas supervisadas y no supervisadas	✘	✘	(Maletti, 2016)
Modelo de frase discreta, Red Neuronal Recursiva	✘	✘	(Tanana et al., 2016)
Clasificador Random Forest	✔	✔	(Felbermayr & Nanopoulos, 2016)
Asignación latente de Dirichlet	✔	✘	(Gross & Murthy, 2014)
Bolsa de palabras, Modelos de regresión textual	✔	✔	(Ngo-Ye & Sinha, 2014)
RNA Convolucionales	✔	✘	(Zhang, Du, Yoshida, & Wang, 2018)
RNA Profundas, SVM	✔	✔	(Al-Smadi et al., 2018)
Modelo Kano, RNA	✔	✔	(Ou et al., 2018)
Deep learning, Recuperación de textos, IoT Social-Cognitivo	✔	✔	(Zhou, Zeng, Liu, & Zou, 2018)

RNA	✗	✔	(Lee & Choeh, 2014)
PLN, SVM, SVR, Convolution tree kernel, BOW, DAV	✔	✔	(Chang et al., 2017)

Tabla 2.1 Enfoques sobre áreas relacionadas a ML y PLN

(adaptada de (Jimenez-Marquez et al., 2018))

Los resultados mostrados en la Tabla 2.1 llevan a hallazgos interesantes: primero, se destaca que mediante la aplicación de ML y PLN se han desarrollado trabajos de sumo interés para diversos ámbitos en áreas que involucran CGU y OCU. Por otra parte, es de destacar que pueda llevarse a cabo investigación en estas áreas sin involucrar las técnicas de BD, por lo que sería interesante conocer los resultados de estos trabajos de investigación si se llegasen a escalar los datos empleados a niveles superiores[5].

2.3.2. Machine Learning y Big Data

En este apartado se presenta el panorama de los trabajos encontrados en el estado del arte que combinan las disciplinas de ML y BD (sin requerir de las técnicas de PLN), como se representa en la Figura 2.4. Como se observa en la figura, se encontró que en la unión de estas áreas se hallaron estudios que resuelven problemas o extienden la investigación sobre temas relacionados a los SCRS, no obstante, se prefirió esta área sobre otras existentes debido a que es una de las áreas predominantes que se han considerado en esta tesis.

Existen diversos estudios sobre el área en el estado del arte que se han realizado ya sea con la finalidad de comparar los diversos frameworks que existen, o bien, se presentan aplicaciones reales de la unión de estas áreas. Por lo que en la Tabla 2.2 se presentan algunos de los trabajos en los que los autores han profundizado en estas disciplinas. El criterio de consulta para referir estos trabajos fue el siguiente: se consultaron todos aquellos estudios recientes que incluyeran temas relativos a ML y BD, pero que no trataran temas de PLN o bien, que las tareas de PLN no hubieran sido incluidas para llevar a cabo la investigación. También se consideraron aquellos trabajos que hicieran una revisión metodológica del estado del arte refiriéndose a las áreas ya mencionadas.

[5] Algunas partes de este apartado se adaptaron del artículo de (Jimenez-Marquez, Gonzalez-Carrasco, & Lopez-Cuadrado, 2018), Secciones III y IV, cuyo autor lo es también de la tesis

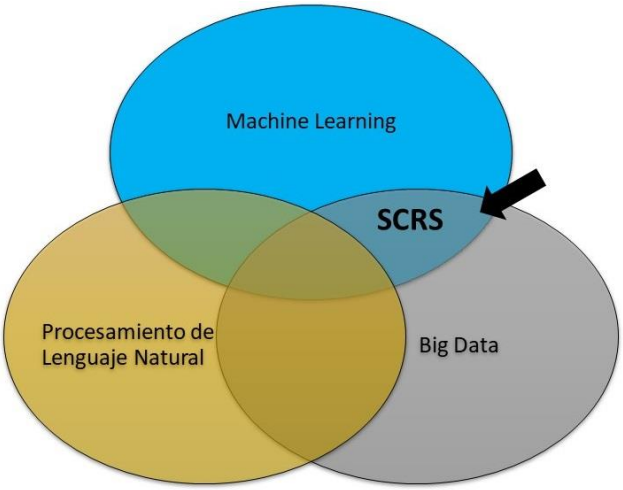

Figura 2.4 Intersección de las áreas de investigación ML y BD

Autores	Áreas del estudio	Contribución al Estado del Arte
(Pääkkönen & Pakkala, 2015)	-BD -ML -Arquitecturas de BD	Presenta una arquitectura de referencia independiente de tecnologías, para sistemas de BD basada en el análisis de casos de uso de arquitecturas similares.
(Bello-Orgaz, Jung, & Camacho, 2016)	-BD Social -Minería de datos -Redes sociales -Frameworks tecnológicos	Revisión de metodologías diseñada para la minería de datos y la fusión de información de las redes sociales y de las aplicaciones y frameworks nuevos hasta el momento de su publicación.
(Oussous, Benjelloun, Lahcen, & Belfkih, 2017)	-BD -Hadoop -BDA -ML	Hace una revisión que analiza las tecnologías desarrolladas para BD, apoyando en la selección y adopción correcta de tecnologías BD de acuerdo con sus necesidades tecnológicas y los requisitos.
(Shadroo & Rahmani, 2018)	-IoT -Minería de Datos -BD	Presenta un estudio sobre las investigaciones realizadas empleando IoT y BD, empleando además métodos de minería de datos para identificar

	-ML	líneas de investigación que satisfagan necesidades actuales y futuras.
(Elshawi, Sakr, Talia, & Trunfio, 2018)	-BD -Data Science -BD Science -ML	Presenta el área llamada Big Data Science, en el que se hace un estudio donde se busca intersecar las técnicas y herramientas que combinan BD y Data Science.
(Zhou, Pan, Wang, & Vasilakos, 2017)	-ML -BD -Procesamiento de datos	Se realiza un estudio donde se potencian las posibilidades que se tienen en investigación al emplear ML sobre arquitecturas de BD.
(Rahman, Esmailpour, & Zhao, 2016)	-RNA -BD -Hadoop -Mapreduce	Se demuestra cómo a través del uso de ML y BDA, se puede predecir el futuro consumo de energía y cuánto se deberá generar en próximos años.
(Habib, Chang, Batool, & Ying, 2016)	-BDA -Reducción de datos -ML	Presenta un framework centrado en BD para la reducción temprana de los datos que le permitan realizar objetivos tanto técnicos como comerciales.

Tabla 2.2 Resumen de trabajos en el estado del arte que integran BD y ML

Como se aprecia en la tabla Tabla 2.2, existen una serie de estudios que se han dedicado a la tarea de obtener un panorama sobre trabajos realizados en las áreas descritas de ML y BD. Gran parte de los trabajos citados se han abocado a la tarea de presentar un estudio donde se exploren estas áreas con el fin de dar a conocer las posibilidades que dichas áreas tienen para investigaciones que se desarrollan en el presente y lo que se espera para el futuro de estas tecnologías. A continuación se reseña cada trabajo en lo que refiere en su contribución a la tesis:

El trabajo de Pääkkönen & Pakkala (2015) permite conocer las arquitecturas de BD en el mundo real para identificar procesos que sean comunes a la tesis. Mientras que Bello-Orgaz et al. (2016) realizan una referencia tecnológica de los frameworks de ML para BD; además de que permiten tener la perspectiva de cómo se han integrado las redes sociales en los estudios que reseñan. Oussous et al. (2017) por su parte, presentan un escenario integral de las diferentes tecnologías desarrolladas alrededor de Hadoop; este

trabajo también contribuye a la tesis al explorar las tecnologías de BD que emplean ML de acuerdo a las subcategorías creadas con este fin.

Shadroo & Rahmani (2018) comparan y contrastan estudios que emplean BD sobre el área del IoT para dar a conocer los frameworks empleados en estos, de igual manera presentan los estudios que han involucrado clasificadores de ML en su construcción. Por otro lado, Elshawi et al. (2018) analizan en primera instancia cuáles son los bloques que dan soporte a lo que denominan 'ciencia BD' para dar a conocer la clasificación de los frameworks de BD Analítico, además de explorar los casos de usos reales que han empleado ML y BD.

Con respecto al trabajo de Zhou et al. (2017) los autores aportan un framework donde se buscan integrar ML sobre un stack tecnológico de BD. En contraste, Rahman et al. (2016) presentan una perspectiva de la aplicabilidad de las tecnologías de ML y BDA; los autores también exploran cómo mediante el empleo de estas dos disciplinas se puede predecir eficazmente sobre el consumo de energía eléctrica. Por último, Habib et al. (2016) presentan un framework robusto en cuanto a los pasos mediante los cuales se detalla la construcción del mismo, además de que se ha tomado como referencia este trabajo durante la elaboración de la tesis en el apartado referente a la preparación de datos.

2.3.3. Big Data y Procesamiento de Lenguaje Natural

En este apartado se consideran aquellos trabajos donde se apliquen exclusivamente técnicas de BD y PLN, como se aprecia en la Figura 2.5 donde estas áreas se intersecan. Como se observa en la figura, se encontró que en la unión de estas áreas se hallaron estudios que resuelven problemas o extienden la investigación sobre temas de AS, no obstante, se prefirió esta área sobre otras existentes, debido a que es una de las áreas predominantes que se han considerado en esta tesis.

Para realizar la consulta en el estado del arte respecto a estas dos áreas, se consideraron trabajos que abarquen exclusivamente estas dos áreas, lo cual acota significativamente los trabajos a incluir debido al auge que tiene el ML en el estado actual de la investigación, el cual aporta mayor poder de análisis de los problemas que involucran PLN. No obstante, se presentan los trabajos que emplean estas dos disciplinas en la resolución de problemas en la ciencia o en la técnica.

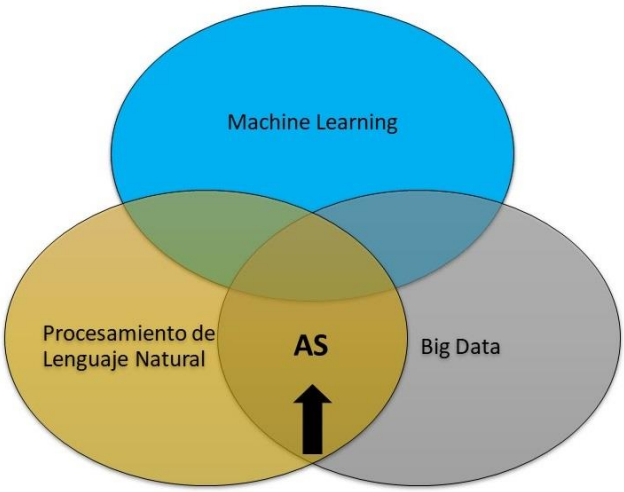

Figura 2.5 Intersección de las áreas de investigación PLN y BD

Primeramente, en el estudio de Nesi, Pantaleo, & Sanesi (2015) se presenta una arquitectura que permite la integración de BD, acoplando las funciones Mapreduce e integrándolas con la aplicación GATE (una plataforma de PLN), esto con el fin de generar un rastreador distribuido de páginas web. Por su parte, en el trabajo realizado por Marine-Roig & Anton-Clavé (2015) se presenta una metodología utilizada para extraer inteligencias comerciales en contenido generado por los usuarios, como reseñas de viajes o blogs turísticos, esta investigación destaca por la utilidad que se pudo obtener de BDA para dar soporte a destinos inteligentes[6].

Se resalta nuevamente la importancia de conocer el grado de satisfacción de los usuarios de servicios turísticos. Un enfoque de interés para esta investigación es el desarrollado por Liu et al. (2017) en el que se obtuvieron los atributos de mayor interés para los usuarios de hoteles de diversas nacionalidades que visitaron China y se obtuvieron hasta 412,784 críticas del sitio TripAdvisor. En dicho estudio se aplicaron herramientas estadísticas como R y de análisis de textos para obtener tales dimensiones. Los resultados son de particular interés ya que demostraron que los gustos y preferencias de los usuarios son diferentes entre turistas nacionales e internacionales.

Artola, Beloki, & Soroa (2014) realizaron un estudio en el cual se desarrollan técnicas de PLN para analizar en tiempo real los contenidos de noticias para detectar personajes, eventos y circunstancias que están ocurriendo dentro de un entorno de noticias. Estas técnicas a su vez se escalaron hacia el framework de BD Storm para “evaluar su eficacia y eficiencia cuando se procesan documentos en mediana y larga escala”. En contraste, Amith, He, Bian, Lossio-Ventura, & Tao (2018) presentan un trabajo en el cual se abarca

[6] Este párrafo se adaptó del artículo de (Jimenez-Marquez, Gonzalez-Carrasco, & Lopez-Cuadrado, 2018), Sección III, cuyo autor lo es también de esta tesis

el tema de cómo se ha extendido el uso de las ontologías en el área biomédica, resaltando la importancia que este tipo de estudios tienen en su integración con BD.

El empleo de las técnicas de BD y PLN puede ser empleado en aplicaciones prácticas en aplicaciones como la distribución de energía eléctrica, como se desarrolla en (Guerrero, García, Personal, Luque, & León, 2017), donde los autores llevaron a cabo un estudio donde se integran fuentes heterogéneas para ecosistemas de redes inteligentes basados en minería de metadatos. Se resalta también el trabajo realizado por Kwon, Park, & Geum (2018) en el cual los autores aplican Wikipedia para llevar a cabo análisis morfológico y construir una serie de métodos que den soporte a investigaciones futuras que puedan emplear este tipo de análisis.

Finalmente se destaca el estudio realizado por Xiang, Schwartz, Gerdes, & Uysal (2015) debido a la gran convergencia de temas en común que tiene este estudio con los objetivos planteados por la tesis. En este trabajo se emplean las técnicas de BD y PLN para analizar datos de tipo texto sobre hoteles y a partir del análisis de estos con las técnicas referidas, se extraen una serie de indicadores sobre el nivel de satisfacción que tiene el cliente a partir de los servicios recibidos durante su estancia. Este estudio además presenta importantes descubrimientos al detectar y relacionar variables en la información, así como detectar patrones sobre el comportamiento en los huéspedes del hotel.

Como se observa, existen trabajos que conjuntan estas dos áreas, sin embargo, son escasos en el estado del arte debido a que el emplear técnicas de ML contribuye a una investigación más profunda y avanzada debido al poder de análisis y clasificación que los algoritmos de ML realizan sobre los datos de tipo texto. En algunos de los trabajos citados en este capítulo donde se emplea ML y PLN con o sin técnicas de BD, el PLN es empleado comúnmente en una fase preparatoria de los datos, y en otros se emplea para dar un mayor peso a la investigación en función de los objetivos de la propia investigación.

2.3.4. Machine Learning, Big Data y Procesamiento de Lenguaje Natural

Se han presentado anteriormente la intersección de dos de las tres grandes áreas que conciernen a esta investigación y algunos de los trabajos en el estado del arte que son de importancia para esta tesis. En la Figura 2.6 se representa en el diagrama de Venn la intersección de las tres disciplinas y como se aprecia, justo en la intersección, se encuentra esta tesis doctoral.

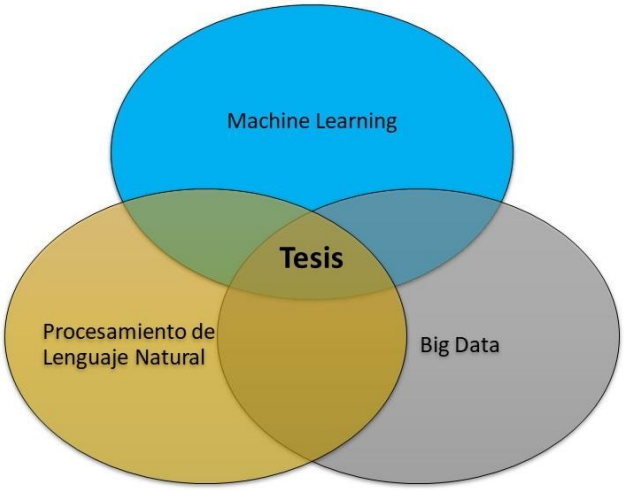

Figura 2.6 Intersección de las tres áreas de investigación respecto a la tesis

A continuación, en la Tabla 2.3 se destacan investigaciones encontradas en el estado del arte que combinan las tres áreas principales: ML, BD y PLN. El criterio de consulta para referir estos trabajos fue el siguiente: se consultaron todos aquellos estudios recientes que incluyeran temas relativos a ML, BD y PLN. Sin embargo, se acotaron los trabajos encontrados para considerar sólo aquellos que hubieran considerado resolver alguna situación real, ya sea una necesidad científica o tecnológica, empleando técnicas de BD que son utilizadas además para tareas de ML. También se consideraron aquellos trabajos que hicieran una revisión metodológica del estado del arte en cuestión refiriéndose a las áreas ya mencionadas.

Autores	Áreas del estudio	Contribución al Estado del Arte
(Qi, Zhang, Jeon, & Zhou, 2016)	SVM, Análisis Conjunto, KANO, Reviews en línea	Establecen que las revisiones en línea podrían ser la fuente de ideas para que los fabricantes diseñen nuevos productos, más adecuados para los clientes.
(Elgendy & Elragal, 2016)	BDA, B-DAD, Toma de decisiones	Busca investigar como BDA puede ser integrado en el proceso de toma de decisiones.
(Guo, Barnes, & Jia, 2017)	LDA, Satisfacción del turista, Minería de datos	Crearon un sistema que permite trabajar con GVDNE de TripAdvisor, donde se da mayor prevalencia al uso de dimensiones para trabajar

		directamente con la satisfacción de los visitantes.
(Agerri, Artola, Beloki, Rigau, & Soroa, 2015)	PLN, BD, Storm	Proponen un sistema en el que utilizan varias máquinas virtuales para ejecutar sus propios módulos de PLN: las herramientas IXA.
(Gudivada et al., 2015)	BD, PLN, Modelos Estadísticos	Estudio a fondo sobre la importancia y la necesidad de integrar más estudios que incluyan PLN y BD.
(Bilal et al., 2016)	BDA, ML, Ingeniería de BD, Industria de la construcción	Se exploran las áreas de BD y ML para analizar las aplicaciones que se pueden tener para la industria de la construcción, incorporando PLN.
(Wong, Lu, & Chao, 2016)	RNA Recursiva, Traducción máquina, Selección de datos	Proporcionar un método para generar traducciones del chino al inglés mediante una RNA recursiva para la traducción automática estadística
(Zhang, Wu, Bu, Jiang, & Cao, 2018)	Medios sociales, Análisis de sentimientos, Weibo	Se emplearon las tres áreas para analizar tweets de Weibo en tiempo real, además de emplear su arquitectura para minería de textos.
(Li et al., 2018)	Cómputo de alto rendimiento, Clasificación de textos cortos, Geocodificación	A partir de datos históricos del registro de empresas en China, se presenta un análisis espacio-tiempo empleando BD, ML y PLN.
(Das, Behera, kumar, & Rath, 2018)	RNA Recurrente, Análisis de Sentimientos, Spark, Flume	Aportan un framework que realiza predicciones de la bolsa de valores a partir del análisis de tweets en tiempo real.
(García-Pablos, Cuadros, & Rigau, 2018)	Minería de opiniones, Análisis de sentimientos hacia aspectos	Estudio en que se lleva a cabo AS basado en aspectos aplicado a diversos dominios e idiomas.

(Dessì, Fenu, Marras, & Recupero, 2018)	Cómputo cognitivo, BD, Analítica de aprendizaje	Tomando como referencia una serie de cursos en línea se emplean las tres áreas para realizar trascripciones de videos educativos.
(Qiu, Wu, Ding, Xu, & Feng, 2016)	ML, BD, minería de datos	Estudio de la literatura sobre avances en la investigación sobre ML para el procesamiento de BD.
(Müller, Junglas, vom Brocke, & Debortoli, 2016)	BDA, sistemas de información, metodologías	En el estudio se hace énfasis de cómo se puede emplear el área de BDA en la investigación concerniente a sistemas de información.
(Salehan & Kim, 2016)	BDA, PLN, Minería de sentimientos	En el estudio se analizan críticas de clientes para realizar minería de sentimientos, buscando un enfoque hacia BDA.
(Giatsoglou et al., 2017)	ML, PLN, Comentarios online de usuarios	Se propone un framework para el AS en el que se aprovechan las emociones y las palabras incrustadas.
(Yala et al., 2017)	ML, PLN, Patología mamaria	Mediante técnicas de análisis de textos se analizan reportes médicos de patologías mamarias para identificar y extraer información relevante
(Jiang, Luo, Xuan, & Xu, 2017)	BD de medios sociales, minería de textos	Se lleva a cabo AS para eventos de noticias basado en Big Data de medios sociales para extraer semántica y emociones de la información.
(Sohangir, Wang, Pomeranets, & Khoshgoftaar, 2018)	Deep learning, BD, AS, recuperación de información	Se emplean técnicas de deep learning, BD y PLN, para realizar AS en información financiera.

Tabla 2.3 Resumen de trabajos en el estado del arte que integran ML, BD y PLN

Como se refleja en la Tabla 2.3, existen diversos trabajos en el estado del arte que mediante la combinación de las técnicas de ML, BD y PLN han resuelto problemas reales con una aplicación no sólo científica sino también práctica, a la vez de haber contribuido con métodos y/o frameworks que contribuyen a realizar investigaciones más avanzadas en diversos campos. Es de resaltar la importancia que están tomando las RNA (recurrentes

o recursivas) en el desarrollo de este tipo de estudios, lo cual se debe en parte al poder de procesamiento con que se cuenta actualmente, lo cual lleva a terminar procesos más complejos en menor tiempo.

Se puede afirmar que la tesis contribuye al estado del arte al aportar un framework que está basado en áreas científicas del área de ciencias de la computación que tienen no sólo usabilidad en el mundo real, sino que han sido empleadas conjuntamente en diversos estudios científicos para resolver problemas con aplicaciones totalmente científicas o tecnológicas. A continuación se hace una reseña de los trabajos presentados en la Tabla 2.3 en lo que refiere en su contribución a la tesis:

El trabajo de Qi et al. (2016) es uno de los primeros estudios que orientaron el curso de esta investigación para descubrir qué aplicaciones prácticas se podían tener al combinar las tres áreas, indicando que las revisiones en línea podrían ser la fuente de ideas para que los fabricantes diseñen nuevos productos, más adecuados para los clientes. Por otra parte, Elgendy & Elragal (2016) presentan un framework que combina diversas arquitecturas de ML, BD y PLN orientadas a dar soporte al proceso de toma de decisiones en ambientes gerenciales. Mientras que Guo et al. (2017) proponen un framework alterno al descrito en el Capítulo 4, en el que se obtiene el grado de satisfacción del turista en base a las dimensiones obtenidas de las opiniones expresadas en los comentarios.

En contraparte, Agerri et al. (2015) dan a conocer los frameworks tecnológicos existentes para el procesamiento de tareas de PLN en entornos de BD y su integración con ML. En el trabajo de Gudivada et al. (2015) se tratan las tareas y aplicaciones de PLN para posteriormente presentar algunas fuentes de datos para la investigación en PLN. Por otro lado, Bilal et al. (2016) exploran aplicaciones hacia otros dominios que se pueden tener al aplicar las técnicas de BD, ML y PLN, además de tener un marco teórico de referencia sobre el escenario actual que se tiene en la construcción[7].

Wong et al. (2016) identifican estudios en el estado del arte que emplean las tres áreas referidas para aplicaciones prácticas que puedan servir de referencia en trabajos futuros. Mientras que Zhang et al. (2018) relacionan estudios que trabajan con datos de medios sociales similares a los empleados en la tesis, además de descubrir sistemas que analizan en tiempo real los eventos y tópicos de los que hablan los usuarios. El importante estudio llevado a cabo por Li et al. (2018) aporta al framework propuesto sobre la capa de visualización referida en el Capítulo 4, en el aspecto de las capacidades avanzadas que se pueden lograr al combinar estas tres disciplinas.

[7] Algunas partes de este apartado se adaptaron del artículo de (Jimenez-Marquez, Gonzalez-Carrasco, & Lopez-Cuadrado, 2018), Secciones III y IV, cuyo autor lo es también de la tesis

Con referencia al trabajo de Das et al. (2018), los autores permiten distinguir frameworks que, combinando las tres áreas referidas en este capítulo, realizan procesamiento en tiempo real de datos de medios sociales con aplicaciones prácticas. El estudio presentado por García-Pablos et al. (2018) permitió encuadrar el marco de esta investigación respecto a estudios más extensos en cuanto a su alcance, para identificar su aplicabilidad. Por último, la investigación llevada a cabo por Dessì et al. (2018) permite evaluar la aplicabilidad de estas tecnologías en entornos educativos, al emplearlos en entornos BD que combinan audio y video para la clasificación automática de estos materiales.

Qiu et al. (2016) señalan que los métodos empleados en el área de aprendizaje de representación han tenido aplicaciones reales en el munco real como reconocimiento de voz, PLN y vehículos inteligentes, en los cuáles se han resuelto problemas de BD empleando técnicas de ML, además de deep learning. Por su parte, Müller et al. (2016) realizan un estudio en el que presentan cómo puede emplearse BDA en la investigación de sistemas de información en la que se exploran las áreas de ML y PLN, para lo cual presentan también un caso de uso real en el que se analizan 1.3 millones de críticas en línea tomadas de Amazon.com para evaluar su utilidad.

El trabajo de Salehan & Kim (2016) presenta un enfoque diferente sobre cómo realizar la minería de sentimientos de un conjunto de datos de Amazon, en el cual se presenta una interesante propuesta hacia la extracción de características de las opiniones que permitan aportar un enfoque que pueda servir en entornos de BDA. Sobre una línea similar de investigación Giatsoglou et al. (2017) presentan un framework para el análisis de sentimientos el cual puede podría ser idóneo para su aplicación futura en la investigación que involucra técnicas de BD; este framework fue probado con éxito sobre críticas en línea de corpus en dos idiomas: inglés y griego.

Se ha hecho referencia anteriormente de las aplicaciones que las tres áreas mencionadas en este apartado están teniendo, una de estas es en investigaciones biomédicas como es el caso de análisis de patologías mamarias (Yala et al., 2017). Por otra parte, Jiang et al. (2017) desarrollaron un estudio en el cual se analiza información de BD proveniente de medios sociales para llevar a cabo AS que pueda relacionarse con eventos que se han obtenido de noticias a través de dos procedimientos: cálculo y refinamiento de la emoción en las palabras. En un área diferente como lo es las finanzas, Sohangir et al. (2018) aplicaron técnicas de deep learning, BD y PLN, mediante las cuáles explotaron información de redes sociales relativas a información financiera, para realizar AS que pudiera aprovecharse con el fin de predecir los movimientos futuros del mercado de la bolsa de valores.

2.4. Discusión

En gran parte de este capítulo se ha hablado del trabajo con datos procedentes de fuentes de medios sociales, pero eso es sólo una parte de todo lo que se entiende como BD. Los resultados de Rahman et al. (2016) demuestran que otros tipos de datos pueden tratarse eficazmente en entornos analítico-predictivos de BD. Dado que este tipo de estudios no implican el conocimiento del PLN, se aconseja una sólida base en estadística y ML para crear sistemas inteligentes de pronóstico. Los diversos autores en los artículos presentados han involucrado técnicas variadas de minería de textos para la recolección de datos y entonces llevar a cabo el proceso de experimentación. Con las nuevas tecnologías de generación de datos como el IoT y la telefonía móvil 5G, las redes sociales no serán la principal fuente de BD, y fenómenos sociales como el consumo de energía, las nuevas enfermedades o el humor social podrían ser mejor analizados y predichos con la conjunción de estas tecnologías[8].

Como se comentó en el apartado de AS, tanto Ravi & Ravi (2015) como Serrano-Guerrero et al. (2015) coinciden que en las técnicas empleadas para llevar a cabo las tareas de AS se pueden emplear técnicas de ML y PLN (entre otras), por lo que en su momento, aún no se contaba con la popularidad de BD para explorar esta disciplina; de tal forma que puede replantearse hacer un nuevo estudio que abarque las tres áreas con los estudios que se hayan hecho en común.

El rápido tiempo de desarrollo y el corto tiempo para salir al mercado que tienen las empresas en la actualidad, limitan el tiempo que los científicos de datos tienen para realizar el análisis de la información. Sin embargo, cada proyecto de análisis de datos debe seguir una metodología para garantizar resultados de óptima calidad. Existe un interés creciente sobre la creación de modelos que permitan gestionar grandes volúmenes de información, los centros de investigación no sólo están interesados en recopilar datos digitales y almacenarlos sin mayor explotación, sino que además existe el interés de poder transformar sus datos para obtener resultados significativos.

Un ejemplo de estos modelos es el propuesto por Vajirakachorn & Chongwatpol (2017) donde los autores proponen un modelo para un festival local de alimentos en Tailandia. Los pasos que estos autores proponen en su modelo comparten diversas similitudes con el framework que se plantea en el Capítulo 4: establecer objetivos, recopilar datos, analizar datos mediante técnicas de ML, etc. Sin embargo, este modelo presenta las siguientes desventajas:

[8] Este párrafo se adaptó del artículo de (Jimenez-Marquez, Gonzalez-Carrasco, & Lopez-Cuadrado, 2018), Secciones IV y V, cuyo autor lo es también de esta tesis

1. No considera el análisis de datos de texto.
2. No aplica técnicas de PLN.
3. No aplica técnicas de BD.
4. Las técnicas de gestión de datos propuestas están relacionadas con herramientas de inteligencia de negocios y gestión de bases de datos, mientras que el framework propuesto considera integrar herramientas y técnicas de BD.
5. Su estructura está diseñada para trabajar sólo con datos estructurados mientras que el framework propuesto puede analizar datos estructurados y no estructurados.

Por otra parte, el enfoque para integrar diferentes fuentes de datos en el framework propuesto es más simple y el conjunto de algoritmos ML más manejable, ya que está abierto a otros algoritmos ML, siempre que estos se adapten al conjunto de datos a analizar. Cabe aclarar que este trabajo no se considera dentro de la comparativa anteriormente presentada debido a que, de las tres áreas revisadas en este capítulo, sólo implica el uso de técnicas de ML. No obstante, se le considera en este apartado, pues es un framework que guarda cierta similitud con los objetivos del framework propuesto.

El modelo de BD propuesto por Habib et al. (2016) tiene como objetivo principal la "reducción de BD en el extremo del cliente". Aunque este modelo considera más elementos relacionados con la infraestructura y metodología de BD con respecto al framework propuesto, el primero se enfoca particularmente en la reducción de datos para disminuir los costos cuando se trata de BD y transacciones en Cloud Computing. Por lo que el estudio referido no considera la integración de ML y PLN para el análisis de datos con el fin de resolver desafíos específicos cuando se trata de datos supervisados. Por otra parte, el modelo propuesto por los autores sólo considera la creación de valor en el nivel de BD, además que consideran un conjunto más amplio de pasos para crear dicho valor a partir de los datos.

En el estudio presentado por Yuan, Xu, Qian, & Li (2016) se considera la información turística de los blogs de viajes, posteriormente se implementa el método frecuente de minería de patrones para identificar las ubicaciones populares de una ciudad mediante la creación de vectores de palabras para construir un red de palabras. Este estudio presenta un modelo que funciona con información turística extraída de reviews, por lo tanto, se limita a este dominio; mientras que en el framework que se propone en esta tesis se plantea reducir dicha complejidad ya que es capaz de trabajar con varios dominios. Este framework no se presentó anteriormente debido a que de las tres áreas en revisión durante este capítulo, sólo han empleado PLN (además de otras técnicas); no obstante se compara este framework con el propuesto al estar el primero enfocado en el turismo.

En el trabajo realizado por Chang et al. (2017), los autores presentan una arquitectura para el análisis de sentimiento basado en aspectos, de hecho van más allá del análisis que

se presenta en esta tesis, para extraer aspectos y detectar categorías de datos de TripAdvisor. Los autores han demostrado que su modelo de clasificación de núcleo de árbol de convolución supera a otros métodos de ML para la clasificación de sentimiento. Aunque por una parte el framework planteado en esta investigación no considera el análisis de aspectos, una complejidad que se simplificó en la metodología propuesta en la tesis, es que se encuentra abierta a todos los algoritmos de ML para el análisis de datos. Otra complejidad que resuelve el framework planteado es que el estudio de los autores citados anteriormente no considera cómo integrar estas técnicas en un entorno de BD (Jimenez-Marquez, Gonzalez-Carrasco, Lopez-Cuadrado, & Ruiz-Mezcua, 2019).

En la arquitectura presentada por Zhang et al. (2018) se propone un sistema de detección y análisis de temas basado en patrones de la plataforma Weibo que es similar al Twitter pero en China. Este modelo recoge datos mediante web crawlers, los almacena mediante herramientas de BD y posteriormente hace el preprocesado de los datos; posteriormente emplean sus técnicas desarrolladas para la extracción de aspectos donde se aplican algunos métodos de PLN y entonces se procede a una etapa donde se presenta el análisis de los datos a partir de representaciones visuales, o mediante texto normal. Sin embargo, esta arquitectura no plantea el análisis de información cuantitativa (ratings), además de estar orientado totalmente a entornos de BD, a diferencia del framework planteado en la tesis que puede ser implantado en entornos no BD.

Por su parte, Li et al. (2018) presentan un framework de cómputo para la imputación de datos de registros incompletos de grandes empresas en China. Dicho framework está basado totalmente en arquitecturas BD, particularmente Spark, el cual a través del uso del ML, PLN y el BD puede procesar grandes volúmenes de información de empresas asentadas en China y a través de esto presentar la información a través del análisis espacio-temporal y permitirse presentar la información de manera visual para presentar la evolución histórica en las diversas regiones de China. Este framework sin embargo, está totalmente orientado a trabajar con este tipo de datos, por lo que no está planteado para el análisis de datos de redes sociales o SCRS. Por otra parte, el framework no está construido para implantarse en arquitecturas no BD. Las dos cuestiones planteadas, son propuestas en el framework de la tesis tanto en el aspecto de los datos a analizar, como de las infraestructuras empleadas.

En cuanto al framework creado por Das et al. (2018) para la predicción en la bolsa de valores, éste lleva a cabo AS en tiempo real de datos de Twitter para producir esta función. Su arquitectura está basada en herramientas de BD para el análisis de datos fundamentalmente en tiempo real, cuenta con una capa de PLN y ML que le permite analizar el lenguaje en los tweet*s* para establecer las predicciones. Aunque este framework analiza datos de medios sociales mediante las tres áreas propuestas, además de realizar

AS, no está planteado para arquitecturas no BD, a diferencia del framework planteado en la tesis.

El modelo elaborado por García-Pablos et al. (2018) puede analizar datos de medios sociales y SCRS en un entorno no BD, estos datos pueden ser de cualquier dominio además de estar en cuatro idiomas; cabe mencionar que el framework propuesto tiene la restricción actual de no poder efectuar clasificación multilingüe. Este modelo además de realizar las tareas de AS y clasificación de polaridad de sentimientos que son llevadas a cabo por el framework, es capaz de clasificar categorías de aspectos y hacer la separación de palabras-opiniones. Si bien es un framework muy completo, no se tiene previsto el llevar a cabo estas tareas a mayor escala en un entorno de BD, por otra parte, no se especifica una fase o etapa en la cual se puedan evaluar diversos algoritmos de ML para evaluar el modelo en general, estas últimas cuestiones se plantean y solucionan en el framework del Capítulo 4.

En el esquema planteado por Dessì et al. (2018) se establece un modelo de trabajo para clasificar videos para la educación, y a través de las técnicas de ML, BD y PLN, se extraen aspectos relevantes de los videos para realizar su clasificación. Este esquema combina las áreas planteadas durante este capítulo, sin embargo, como se observa está más enfocado en el dominio específico de los videos educativos, además que está planteado para su desarrollo sobre arquitecturas de BD. Como se refiere en el Capítulo 4, el framework propuesto está abierto al análisis de datos de medios sociales de cualquier dominio tanto en arquitecturas BD como no BD.

Qi et al. (2016) plantean un modelo para obtener los requerimientos principales de los clientes a través del análisis de las reviews en línea. El framework ha sido planteado para analizar el dominio de dispositivos móviles en el idioma chino, de esta forma se extraen todas las opiniones que puedan ser consideradas útiles para proporcionar los resultados de este análisis para que los fabricantes puedan fabricar mejores modelos. Aunque el modelo realiza varias de las tareas propuestas en las dos fases del framework presentado en el Capítulo 4, éste no contempla que puedan emplearse varios dominios, además de estar implantado sobre arquitecturas de BD.

Por su parte, Elgendy & Elragal (2016) presentan un framework que está planteado desde la perspectiva de lo que el BDA puede aportar al proceso de toma de decisiones, por lo cual, es de reconocer que aunque los investigadores han fundamentado apropiadamente su framework (B-DAD) en la parte técnica y se han considerado las técnicas de las áreas propuestas en este capítulo entre otras, este framework no propone una solución práctica aplicando las tres áreas directamente, a diferencia del framework propuesto en el Capítulo 4 de esta tesis.

Un ejemplo de un modelo enfocado en el análisis de la satisfacción del turista es el elaborado por Guo et al. (2017) en el cual los autores emplean LDA para extraer las dimensiones que pueden ser empleadas por los hoteles para controlar la interacción con sus visitantes. Un aspecto notable de este modelo es el hecho de que parte del análisis de la información tanto cuantitativa cómo cualitativa como base de su análisis de datos de forma separada. No obstante que este modelo plantea algunas similitudes con las fases del framework propuesto, este modelo no contempla su implantación en un entorno alterno a BD, ya que desde el inicio parte de la premisa de trabajar con datos de gran volumen.

Qiu et al. (2016) presentan un framework jerárquico de lo que denominan 'ML eficiente' para el procesamiento de BD. El framework como tal puede ser concebido como una aproximación teórica de las distintas categorías que existen para distintos tipos de aprendizaje y cómo estos pueden ser empleados por tecnologías que pueden habilitar o hacer posible tales aplicaciones. En lo que respecta al área de BDA, se refleja cómo esta área puede impactar a disciplinas ya establecidas en la informática como se destaca en el trabajo de Müller et al. (2016) sobre la relación del BDA con los sistemas de información.

La convergencia de las tres áreas exploradas en este capítulo (ML, BD y PLN) ha sido estudiada, analizada y empleada para diversas aplicaciones como se ha reseñado a lo largo del mismo, entre estas aplicaciones se tienen: análisis de informes de patología mamaria (Yala et al., 2017), predecir el desempeño de los comentarios en línea de consumidores (Salehan & Kim, 2016), cálculo del sentimiento para eventos de noticias basado en BD de redes sociales (Jiang et al., 2017) o AS (Giatsoglou et al., 2017). No obstante, el acelerado crecimiento de la investigación en deep learning ha fomentado la conjunción de esta disciplina junto con el BD y el ML en aplicaciones reales (Sohangir et al., 2018), por lo que nuevos frameworks que exploren el potencial del deep learning serán objeto de futuras investigaciones.

Como se ha comentado anteriormente, la próxima irrupción de la tecnología 5G en el quehacer diario de la población, generará gran incremento del volumen de datos originado por los dispositivos móviles, el IoT y las redes sociales. Para poder administrar, analizar y gestionar los grandes volúmenes de información generados por los medios anteriormente mencionados, se requerirá entonces de las técnicas de ML, BD y PLN (así como otras áreas que escapan del alcance de esta investigación) para poder hacer frente a las nuevas demandas científicas y tecnológicas. Dado que BD es una línea de investigación de reciente creación, se espera que las publicaciones que combinen las áreas de ML, BD y PLN sean mayores en el futuro cercano.

De tal manera que esta tesis busca contribuir a ese espacio dentro del estado del arte donde se conjuntan las tres áreas para aportar un framework que pueda reunir las tres

áreas descritas con el fin de poder aportar valor a partir fundamentalmente del análisis de datos en dos ambientes de cómputo diferentes, pero a la vez relacionados por las técnicas de ML y PLN. Por otra parte, debido a lo expuesto al final de este capítulo, se puede afirmar que en función de los trabajos encontrados en el estado del arte aún existen áreas a cubrir en las que es preciso aportar modelos de trabajo que permitan orquestar las actividades entre los diversos paradigmas científicos expuestos. A lo largo de este apartado se han presentado una serie de estudios que han propuesto ya sea un framework (conceptual o tecnológico) o una metodología de trabajo relacionados con, al menos, una de las tres áreas discutidas en este capítulo.

Por lo que a continuación se presenta un resumen de los trabajos más relevantes presentados a fin de resaltar las áreas en las que se puede contribuir. En primer lugar se hace mención de los trabajos que emplean las áreas de ML, BD y PLN, en los que: Zhang et al. (2018); Li et al. (2018); Das et al. (2018); Qi et al. (2016); Guo et al. (2017) y Dessì et al. (2018) lograron conjugar las tres áreas para dar solución real a un problema existente mediante la aplicación de estas técnicas, sin embargo, estas soluciones están planteadas para su ejecución dentro de un entorno de BD, lo cual de alguna manera conlleva tener datos con estas características para ser analizados mediante este tipo de técnicas. Además no se plantea el cómo llevar estos modelos a una escala menor cuando los datos son pequeños (small data) mediante una arquitectura no BD; estos escenarios son analizados y resueltos con el planteamiento de esta tesis.

En contraparte a lo expuesto en el párrafo anterior se tiene que Chang et al. (2017) y García-Pablos et al. (2018) realizaron trabajos donde se conjuntan las áreas de ML y PLN para análisis de datos de medios sociales con un fin similar al perseguido por el framework propuesto en la tesis. Sin embargo, estos trabajos emplean datos de características no BD y no están planteados para poder escalar a una arquitectura de BD, situación que está prevista y resuelta en el marco de esta tesis. Por su parte, Elgendy & Elragal (2016) presentan un framework similar al propuesto desde la perspectiva de lo que se puede obtener a partir del procesamiento y análisis de los datos mediante la perspectiva exclusiva de BDA. En el framework introducido en el Capítulo 4, se presenta un planteamiento más extenso para el aprovechamiento de los datos desde dos plataformas diferentes de cómputo.

Por último, se hace mención de estudios que solamente emplean una de las tres áreas presentadas en este capítulo y que conllevan cierta similitud con los objetivos perseguidos en esta tesis, las cuáles se han discutido anteriormente. El estudio de Vajirakachorn & Chongwatpol (2017) comparte con la tesis intereses similares al procesamiento de los datos pero de las tres áreas exploradas sólo emplea ML. Por su parte Habib et al. (2016) presentan un trabajo que se involucra más con cuestiones de BD por lo que no se involucran las áreas de ML ni PLN. En lo que respecta a Yuan, Xu, Qian, & Li (2016)

dicho estudio emplea en su planteamiento sólo PLN referido a las tres áreas de interés en este capítulo. Por lo que como se observa, el llevar a cabo estudios que empleen las áreas de ML, BD y PLN es una actividad investigativa aún en ciernes.

En función de lo expuesto a lo largo de este apartado ninguno de los trabajos que se han localizado contempla una metodología que pueda lograr resultados empleando técnicas de ML y PLN tanto en un entorno no BD como en uno BD para el análisis de datos cuantitativos y cualitativos, por lo que es un área en investigación en el que la presente tesis puede contribuir. En el Capítulo 4 se presenta el framework propuesto para lograr los objetivos planteados, mientras que en el Capítulo 5 se realiza la evaluación y validación que sustenta los argumentos.

2.5. Sumario

El propósito de este capítulo ha sido el de enmarcar al framework dentro de un conjunto de teorías sobre las cuáles está basado y que son importantes que el lector las conozca para ubicar en qué áreas está circunscrita la investigación. Como se ha observado a lo largo del presente capítulo, el framework está relacionado con diversas áreas que, aunque pudiera pensarse que no están relacionadas, lo expuesto en la parte final reafirma que efectivamente las áreas están relacionadas y que es necesario el seguir desarrollando investigación sobre estas áreas temáticas ya que el estado de la técnica avanza vertiginosamente y deben existir modelos en la ciencia que aporten a este conocimiento.

De igual manera se presentaron una serie de estudios realizados sobre las áreas expuestas a lo largo de este capítulo para acercar al lector en la investigación que se ha hecho sobre las áreas de BD, ML y PLN, además de hacer una integración de todos los temas para poder ser vistos como un único modelo. Aunque el BD sigue siendo considerado por algunos expertos de la industria como una moda temporal, en lo que respecta a los trabajos citados a lo largo del capítulo el BD abre un nuevo horizonte a los académicos y profesionales para los desarrollos tecnológicos del mañana.

Capítulo 3. Estado de la Técnica

Debido a que esta tesis doctoral aborda diferentes áreas de investigación que se relacionan con las técnicas de machine learning, big data y procesamiento de lenguaje natural, se incluye en este capítulo una visión transversal de las principales técnicas para poder aportar un perspectiva que atraviesa los diversos ámbitos que forman parte de esta tesis doctoral.

Este capítulo lleva a cabo un estudio sobre los diferentes elementos que conforman el framework propuesto en el Capítulo 4. En primer lugar, se explora el tema de la inteligencia artificial y su relación con el aprendizaje automático, pasando por una reseña de los clasificadores que se emplearon en esta tesis para conocer su trasfondo teórico. Posteriormente se aborda el tema del procesamiento del lenguaje natural y los procesos relativos a esta área que se emplearon en la tesis.

A continuación, se revisa el área de big data para explorar elementos como los sistemas de big data, los administradores de clusters, el aprendizaje automático y la visualización. Finalmente se abordan temas como recuperación de información y modelado de datos para dar a conocer sobre la secuenciación que tienen los datos en el framework.

3.1. La inteligencia artificial

La IA es la inteligencia de las máquinas creada a partir de procesos establecidos por el hombre, y es además la rama en Ciencias de la Computación que impulsa su creación y avance (Swarup, 2012). Fue John McCarthy, quien en 1956 acuñó el término de IA y lo definió como "la ciencia y la ingeniería de hacer máquinas inteligentes". Este campo científico fue basado en el hecho de que como la inteligencia del ser humano puede ser descrita con detenimiento, de esta misma manera este razonamiento puede ser aplicado a las máquinas. Lo anterior, ha dado lugar a diversos debates filosóficos y éticos sobre si las máquinas y los robots podrán eventualmente sustituir al hombre en los trabajos o incluso si pudieran llegar a dominar a la raza humana, situaciones descritas en numerosas novelas de ficción o películas. Para efectos de este trabajo de tesis se fundamenta la inclusión de la IA como el antecesor directo del ML, del cual se habla más adelante.

El hombre ha logrado significativos avances tecnológicos a lo largo de la historia en diversas áreas de la ciencia. Una de las metas forjadas desde los siglos XVII y XVIII era el poder automatizar los procesos de cálculo matemático a través de máquinas que

pudieran realizar operaciones de distinto orden. Conforme transcurrió el tiempo, fue en el siglo XIX cuando Charles Babbage creó lo que hasta ese entonces se le consideraba lo más cercano a una computadora: la máquina Diferencial y Analítica, la cual ejecutaba cálculos complejos empleando restas. Por lo anterior, Babbage nombró a esta forma de cálculo el método de las diferencias de ahí el nombre de máquina diferencial (Dodd, Grant, & Seruwagi, 2011).

Entrando al siglo XX, Warren McCulloch y Walter Pitts presentaron en 1943 el que es considerado como el primer trabajo en el campo de la IA: el primer modelo de redes de neuronas (Russell & Norvig, 2013). Otro de los principales iniciadores de la era de la IA y que además es considerado uno de los más destacados científicos de este siglo fue Alan Turing. Turing propuso el conocido "test de Turing" el cual consistía en que una computadora aprobaba el test si un interrogador humano después de realizar ciertas preguntas no podía indicar si estas provenían de una persona o de una computadora (Russell & Norvig, 2013). Es así como inicia su camino la IA en la era moderna habiéndose establecido ya leyes formales aprobadas por científicos del área, y de igual manera al contar con dispositivos de cómputo cada vez más avanzados que permitían a investigadores probar los métodos desarrollados que dieran lugar a nuevas unidades de inteligencia.

Actualmente, podemos comprobar que la IA se encuentra presente en diversos elementos que están en contacto con el ser humano como robots, y que ha dejado de ser parte de un reducido grupo de expertos en el área. Los sistemas de IA del futuro deberán no sólo crecer y aprender, sino que su desarrollo y proceso de aprendizaje debe ser creado por el diseñador más poderoso de la complejidad adaptativa: la selección natural (Spector, 2006).

3.2. Algoritmos y elementos de Machine Learning

3.2.1. Redes Neuronales Artificiales

Las Redes Neuronales Artificiales (RNA) tienen su origen en el funcionamiento del cerebro humano en el cual la neurona es el elemento fundamental (González-Carrasco, 2010). Los primeros trabajos enfocados en establecer las bases del sistema nervioso fueron los realizados por Ramón y Cajal, el cual demostró los mecanismos que gobiernan la morfología y los procesos conectivos de las células nerviosas, estudios que a la postre le permitieron ganar el premio Nobel, premio que compartió con Camillo Golgi. De acuerdo a sus hipótesis, el cerebro recibe múltiples señales de entrada las cuáles son procesadas para generar una salida (Ramon y Cajal, 1894).

El cerebro está formado por millones de neuronas que se encuentran interconectadas para formar redes neuronales las cuáles ejecutan millones de instrucciones. Como parte de estas redes, las neuronas son las células que componen la corteza cerebral de los seres vivos las cuáles a su vez están formadas por: el núcleo, el cuerpo, el axón y las dendritas. Como se representa en la Figura 3.1, las dendritas son las terminales que rodean a la neurona y son las que reciben estímulos ya que fungen como receptoras de impulsos nerviosos provenientes del axón de otra neurona. La capacidad que tienen las neuronas para establecer comunicación entre ellas es lo que las diferencia del resto de células vivas, para lo cual emplean impulsos eléctricos y reacciones químicas.

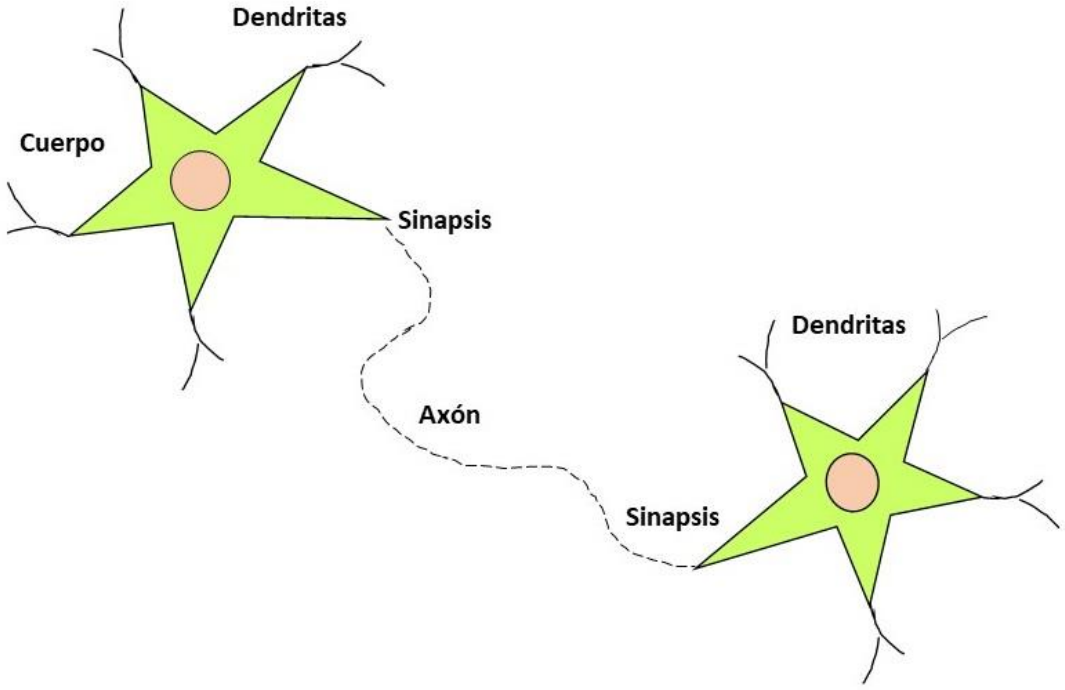

Figura 3.1 Representación simple de la neurona biológica

Mediante la abstracción del funcionamiento de las estructuras neuronales biológicas del sistema nervioso central es como se crearon las RNA (Arbib, 1995). En las cuales se plantean una serie de entradas (capa de entrada) que se encuentran conectadas con otras una capa interior denominada capa oculta. En función del diseño de la RNA puede darse el caso de que se encuentren más capas ocultas. La o las capas ocultas al final deben proveer una salida la cual se genera en la capa de salida

3.2.2. Perceptrón Multicapa

El perceptrón (MLP) es un algoritmo para el aprendizaje supervisado de clasificadores binarios, las cuáles son funciones que deciden si una entrada representada por un vector de números pertenece a una clase específica (Freund & Schapire, 1999). Es una especie de clasificador lineal en el sentido de que el algoritmo encontrará la solución que logre clasificar las salidas pertenecientes ya sea a una u otra clase. En la Figura 3.2 se presenta una red neuronal artificial con una capa de entrada de tres neuronas, una capa oculta de cuatro neuronas y una capa de salida de una neurona, las neuronas por capa pueden aumentar, así como las capas ocultas.

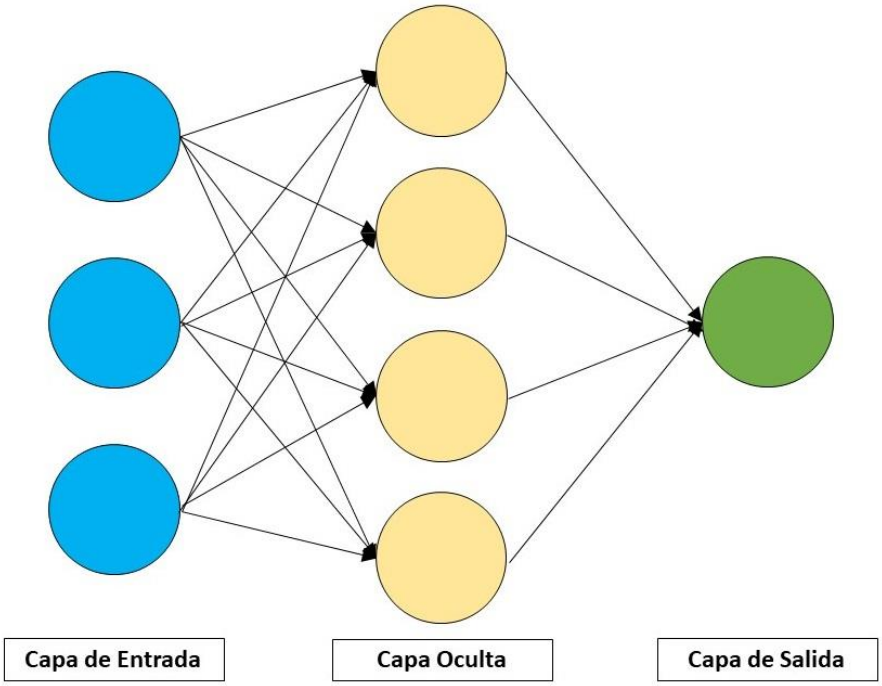

Figura 3.2 Red Neuronal Artificial Simple

Como se aprecia en la figura anterior se tiene una capa de salida, la función del clasificador es entonces hallar la función que logre clasificar cuál de una serie de probables salidas pertenecen a una categoría u otra, como ejemplos de estas salidas pueden ser: cierto o falso, un valor u otro o distinguir entre dos variables categóricas (hombre/mujer). Aquella función que separe mejor ambos conjuntos y, por ende, clasifique mejor las salidas, es considerada la función óptima y es entonces que puede generalizarse a futuros problemas similares. En la Figura 3.3 se aprecia un ejemplo gráfico de un clasificador lineal que clasifica y separa dos grupos distintos de un conjunto de datos.

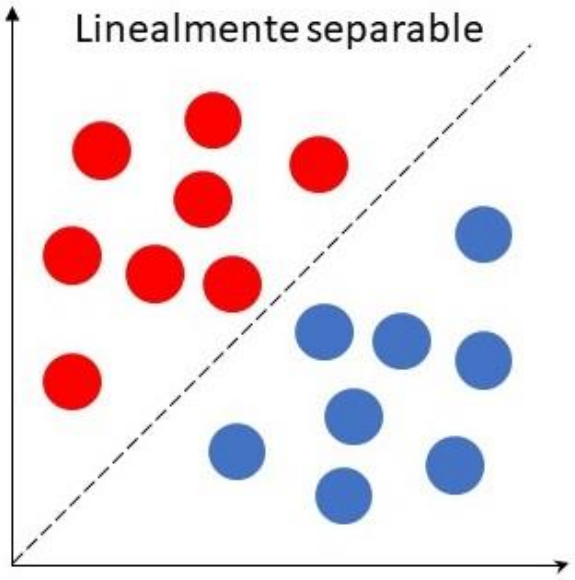

Figura 3.3 Clasificador lineal que separa dos grupos de datos

Entrando a detalle en el funcionamiento de este algoritmo, la forma en la que se transmiten los valores o conexiones de la red es a través de pesos. Estos pesos se establecen para cada nodo de la red de acuerdo a la configuración establecida para la red misma. De tal manera que, si un nodo de entrada tiene el valor 10 y su valor de activación es 0,5, entonces el valor que recibe el nodo de la capa siguiente es 10*0,5 = 5.

En los capítulos posteriores se describen y referencian una serie de parámetros configurados con diversos valores los cuáles buscan dar solución al problema descrito en el Capítulo 1. No todos los parámetros fueron ajustados ni se realizaron pruebas con todos los valores posibles ya que no era el objetivo único de la tesis sino una de las alternativas posibles. Por lo que a continuación se describen los parámetros más importantes que fueron empleados para configurar el perceptrón multicapa tanto para comprender su utilidad, como para justificar el ajuste de los mismos (lo cual se explica en el Capítulo 5).

Función de activación: la función de activación de un nodo define la salida de un nodo dada una entrada o un conjunto de entradas. Es decir, que la salida del nodo podría ser un 1 o un 0 en función de las entradas que el nodo reciba; aunque esta salida binaria es sólo una manera de representarlo. De tal manera que esta función realiza transformaciones u operaciones con el valor de *x* que se le indique. Para ilustrar estas funciones, en la Tabla 3.1 se presenta la relación y descripción de estas:

Nombre corto	Nombre largo	Descripción
Logistic	Función logística sigmoide	Esta es una curva en forma de S (sigmoide), con salida en el rango (0,1)
Hyperbolic Tan	Función hiperbólica tan	Es definida como el cociente entre el seno hiperbólico y el coseno hiperbólico de x, es decir: $\frac{\sinh x}{\cosh x}$
Rectified Linear Unit	Función rectificada de la unidad lineal (ReLU)	Es una función donde x es la entrada de la neurona, es más usada para redes de aprendizaje profundo *(*deep learning).

Tabla 3.1 Funciones de Activación del Perceptrón Multicapa

Solucionador: El solucionador o *solver* es una pieza de software, ya sea un programa o una librería que "resuelve" un problema matemático; en el caso del Perceptrón Multicapa, el solucionador es empleado para optimizar los pesos, es decir, que busca encontrar los pesos de los valores optimizados de la red neuronal para minimizar la función objetivo. En particular se emplearon dos solucionadores: 'Adam' y 'LBFGS'.

Adam (Kingma & Ba, 2015) según lo descrito por sus autores es "un algoritmo para la optimización basada en el gradiente de primer orden de las funciones estocásticas objetivas, basadas en estimaciones adaptativas de momentos de orden inferior". Mientras que LBFGS es un algoritmo denominado en la familia de los métodos quasi-Newton que tiene como característica, emplear una cantidad limitada de memoria de la computadora (Byrd, Lu, Nocedal, & Zhu, 1995). Cabe mencionar en el caso de Adam, que el máximo número de iteraciones establecida por el Perceptrón empleado denota en sí el número de *epochs*, es decir, cuantas veces se empleará cada punto de datos (Scikit-Learn, 2018).

Tasa de aprendizaje: Es un hiper-parámetro que controla cuanto se están ajustando los pesos de la red con respecto a la pérdida gradiente. Mientras más bajo el valor, se vuelve más lento el descenso por la pendiente, dicho de otra forma, se hace más lento el llegar al punto de convergencia. En la Figura 3.4 se presenta un ejemplo gráfico del descenso por el gradiente, en el que conforme se van actualizando los pesos se va descendiendo poco a poco hasta llegar al mínimo global.

Epochs: Son las iteraciones que realiza la red neuronal.

Término de regularización: También llamado parámetro de penalización, es un término que penaliza los detalles del modelo de complejidad innecesaria, se enfoca en las características más relevantes y evita el sobreajuste de los datos (Demir-Kavuk, Kamada,

Akutsu, & Knapp, 2011). La regularización L1 agrega la suma de los valores absolutos de los parámetros del modelo a la función objetivo mientras que la regularización L2 agrega la suma de los cuadrados de ellos. L2 es la regularización conocida como regresión de cresta, la cual es empleada de forma opcional por el Perceptrón Multicapa de la implementación elegida.

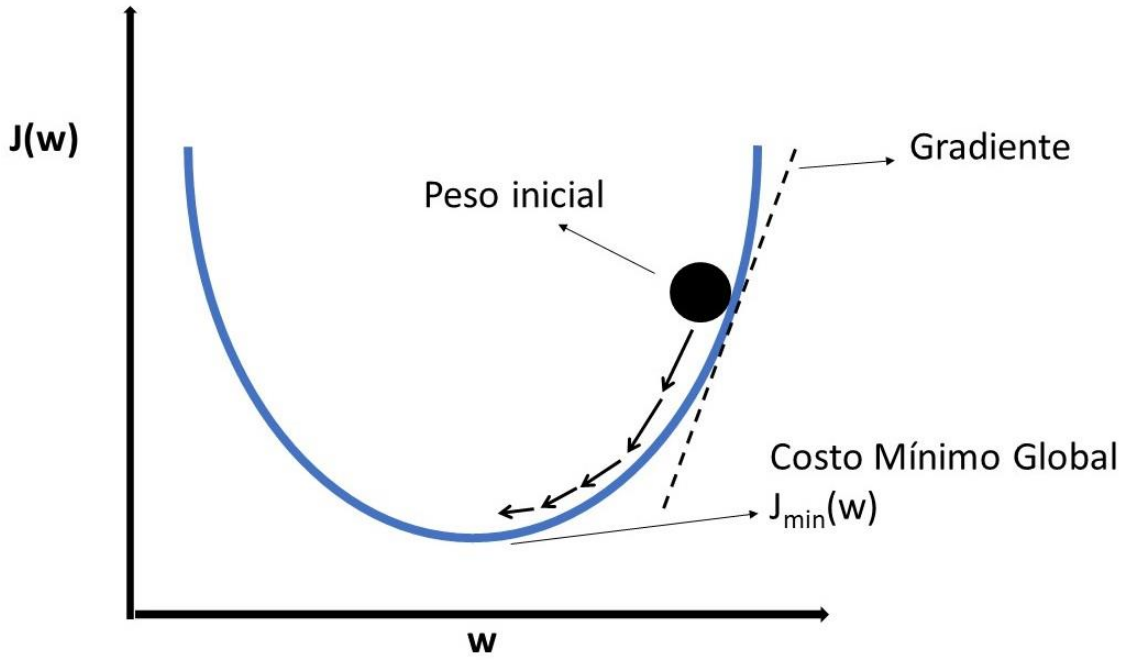

Figura 3.4 Función de costo con un solo coeficiente de peso

3.2.3. Máquinas de vectores de soporte

Las máquinas de vectores de soporte o SVM fueron introducidas por Cortes & Vapnik (1995). Este algoritmo es ampliamente usado en problemas de clasificación y regresión, siendo una de sus aplicaciones prácticas, en problemas referentes a clasificación de textos (Ramesh & Sathiaseelan, 2015; Sebastiani, 2002; Shafiabady et al., 2016). En contraparte, sus desventajas son que debido a las matemáticas complejas que tiene que realizar le es difícil escalar a entornos de BD, por lo que se ve limitado a problemas que no tengan demasiadas características y no sean muy numerosos. Como se ha mencionado, una de las razones por las que le es difícil escalar hacia el número de casos dados es el hecho de que debe de efectuar un número de cálculos proporcional a estos pero elevados a la segunda o tercera potencia, por lo que sobrepasar este límite requiere que el tiempo para desarrollar la solución sea muy extenso (Mueller & Massaron, 2016). Como se muestra en la Figura 3.5, las SVM buscan separar y clasificar los conjuntos de datos en dos clases, esta figura presenta el ejemplo básico.

En la Figura 3.5, se ilustra que el algoritmo busca obtener los vectores más internos o próximos para alrededor de estos construir la solución, es justamente a estos puntos o vectores que se les conoce como “de soporte”, lo que da nombre al algoritmo. Una vez teniendo identificados los vectores de soporte se traza un hiperplano óptimo o hiperplano separador que divide a los vectores de soporte, esta división es llamada los márgenes y es precisamente que la solución óptima es encontrada cuando el algoritmo encuentra el hiperplano que tiene la mayor distancia entre los puntos cerca de este hiperplano.

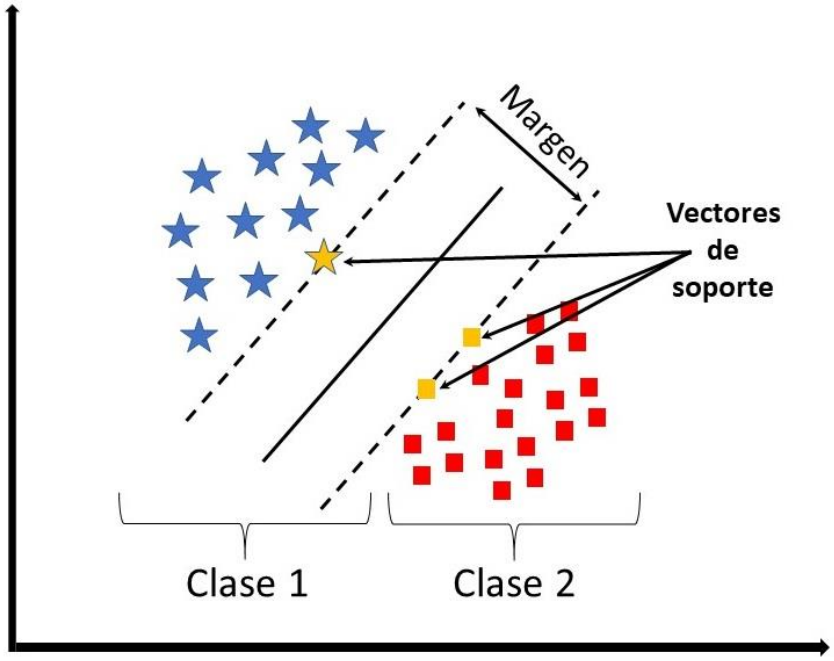

Figura 3.5 Representación de Máquina de vectores de soporte (SVM)

En el caso particular de la tesis se emplearon dos variantes de las SVM: la SVM lineal y el SVC o clasificación de vectores de soporte tipo C. Como en el caso del Perceptrón Multicapa, se procede a explicar aquellos valores que fueron relevantes para la construcción de la solución y que, por lo tanto, es importante conocer para entender los ajustes hechos en la implementación utilizada.

Kernel: También llamado el truco kernel, se refiere a que normalmente el espacio dimensional en el que es proyectado la solución no puede ser ajustada. Por lo anterior, existen alternativas mediante las cuales se puede proyectar la información a un espacio de mayor dimensión, lo cual hace que a su vez se incremente el poder computacional de este tipo de clasificadores. Cada kernel tiene ventajas y desventajas, las características de uno podrían no ser las deseadas para otro (Smits & Jordaan, 2002), los kernel empleados se presentan en la Tabla 3.2:

Kernel	Función	Descripción
Linear	$K(x_i,x_j) = 1 + x_i^T x_j$	Kernel lineal. Es una función simple denotada por basada en el parámetro de penalización C (Sangeetha & Kalpana, 2011), dado que ya dicho parámetro controla el intercambio entre la frecuencia de error c y la complejidad de la regla de decisión (Cortes & Vapnik, 1995).
Polynomial	$K(x_i,x_j) = (1 + x_i^T x_j)^p$	Kernel polinomial o global. Realiza la estimación de kernel no estocástica con dos parámetros C y grado polinomial p. Proporciona una clasificación acertada cuando se tiene un número mínimo de vectores de soporte y una tasa baja de error en la clasificación (Sangeetha & Kalpana, 2011).
Radial basis	$K(x_i,x_j) = exp(-\gamma\|\|x_i - x_j\|\|^2)$	Función de base radial. Es equivalente a transformar los datos hacia un espacio dimensional Hilbert infinito (Sangeetha & Kalpana, 2011).
Sigmoid	$K(x_i,x_j) = \tanh(kx_i^T x_{j-\delta})$	No es tan eficiente como otras funciones porque carece de la condición necesaria de ser un kernel válido (Sangeetha & Kalpana, 2011).

Tabla 3.2 Relación de Kernel empleados en SVM

Es conveniente destacar que la implementación de SVM realiza un número ilimitado de iteraciones el cual se establece en -1 para que el algoritmo de forma libre realice las iteraciones que sean necesarias. No obstante, dicho valor puede ajustarse a un número finito de iteraciones.

C: El parámetro de regularización C establece el margen que separa a los vectores de soporte. De tal manera que un número pequeño de C constituye que el hiperplano tenga un margen con una mayor separación. Mientras que un número grande de C selecciona un hiperplano con un margen menor.

Máximo de iteraciones: El número máximo de iteraciones a ejecutar.

Gamma: Coeficiente del kernel para "rbf", "poly" y "sigmoid". El valor predeterminado actual es ‘auto’ que utiliza 1 / n_features (número de características).

Multiclase: Determina la estrategia multiclase si Y contiene más de dos clases. Ovr entrena n clases de clasificadores del tipo uno contra el resto, mientras que Crammer_singer optimiza un objetivo conjunto en todas las clases (Scikit-Learn, 2018).

3.2.4. Regresión logística

Tomada de la ciencia Estadística, la Regresión Logística (LR) es un modelo comúnmente aplicado a una variable binaria dependiente. Se le puede ver como una función en forma de S (la cual es ajustable en función de los parámetros) donde, visto de una forma práctica como en la Figura 3.6, se aprecia que la función traza una secuencia entre 0 y 1 (falso y verdadero, por ejemplo) en la cual dichos valores nunca llegan a tocarse. Esta peculiaridad es uno de los elementos que distinguen a este algoritmo dado que los problemas que resuelve tienen esta característica: se tiene una progresión de una serie de valores los cuáles han de ser clasificados mediante esta función.

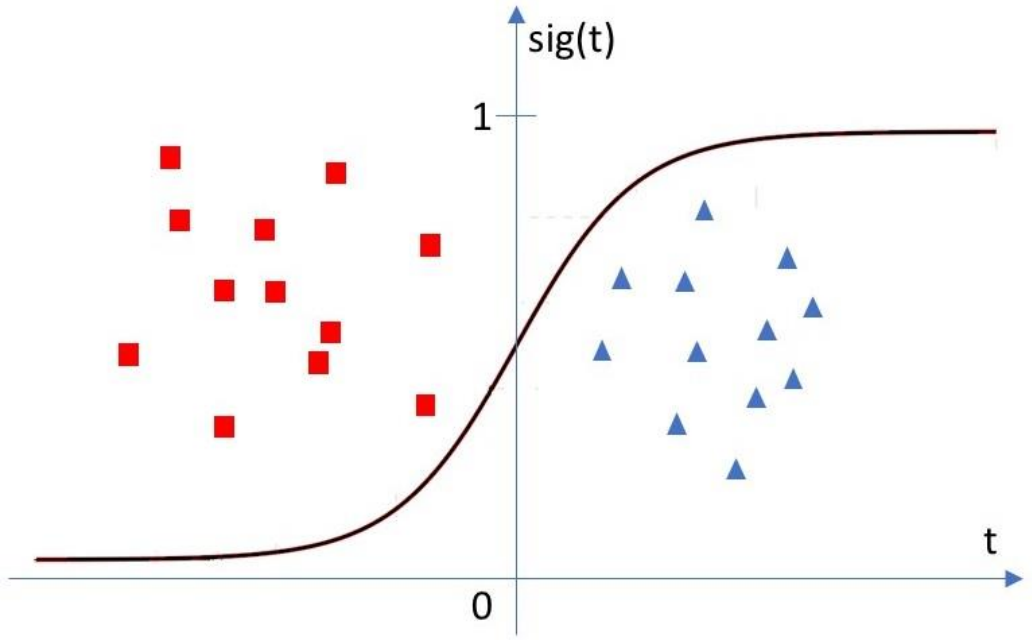

Figura 3.6 Representación de una función de Regresión Logística

Los métodos de regresión han sido parte integral de cualquier análisis de datos interesado en describir la relación entre una variable de respuesta y una o más variables explicativas (Hosmer & Lemeshow, 2000). De acuerdo con los autores citados, el objetivo de un análisis que emplee este método es el mismo que cualquier técnica empleada en

estadística: encontrar el modelo que mejor se ajuste y que además sea capaz de describir la relación entre una variable de salida y un conjunto de variables independientes. Continuando con estos autores, lo que distingue a un modelo de LR de un modelo de regresión lineal es que la variable de salida es binaria o dicotómica (es decir, que toma uno de dos posibles valores).

De acuerdo a Brink, Richards, & Fetherolf (2017), algunas propiedades de la LR son:

- El algoritmo es relativamente simple de entender, en comparación con los algoritmos más complejos. También es computacionalmente simple, por lo que es escalable para grandes conjuntos de datos.
- El rendimiento se degrada si el límite de decisión que separa las clases necesita ser altamente no lineal.
- Los algoritmos de LR a veces pueden sobreajustar los datos, y a menudo es necesario utilizar una técnica llamada regularización que limita este riesgo.

A continuación se presentan los parámetros que fueron ajustados en el modelo de LR empleado a través de la implementación descrita en (Scikit-Learn, 2018).

Penalidad: Se emplea para especificar la norma utilizada en la penalización.

C: Inverso de la fuerza de regularización. De acuerdo a la documentación de Scikit-Learn (2018), el establecer valores más pequeños en este parámetro, especifican una regularización más fuerte. La regularización consiste en aplicar una penalización para aumentar la magnitud de los valores de los parámetros a fin de reducir el sobreajuste. Esto es, dado que se están seleccionando los mejores parámetros que se adapten a los datos, se minimiza el error entre lo que el modelo predice para la variable dependiente en función de los datos comparado contra el valor real de la variable dependiente. En el Capítulo 5 se presentan las diversas pruebas y los resultados que se obtuvieron al ajustar este parámetro.

Solver: Algoritmo empleado en el problema de optimización. *Liblinear* es una librería desarrollada por Fan, Chang, Hsieh, Wang, & Lin (2008) para la clasificación lineal a gran escala que puede ser empleada tanto para LR como para SVMs. *LBFGS* es un algoritmo denominado en la familia de los métodos quasi-Newton que tiene como característica, emplear una cantidad limitada de memoria de la computadora (Byrd et al., 1995).

Máximo de iteraciones: Número máximo de iteraciones a ejecutar para que los solucionadores converjan.

3.2.5. Naïve Bayes

El clasificador Naïve Bayes (NB) es un algoritmo que fue creado para su uso en clasificación de textos (Brink et al., 2017). Es un clasificador probabilístico fundamentado en el teorema de Bayes (Russell & Norvig, 2013) el cual estableció la siguiente ecuación:

$$P(b|a) = \frac{P(a|b)P(b)}{P(a)}$$

El cual es interpretado como la probabilidad de que ocurra *b* siendo de que *a* es verdadero es igual a la probabilidad de que ocurra *a* siendo de que *b* es verdadero multiplicado por la probabilidad de *b*, ambos divididos sobre la probabilidad de *a*. Es de notarse el uso de la palabra inglesa *naive* que significa "ingenuo" debido a que su distribución de probabilidad es empleada a menudo en casos donde las variables de "efecto" no son condicionalmente independientes dada la variable de causa. Dicho de otra forma, la ecuación se aplica a los datos con suposiciones muy ingenuas sobre su independencia.

No obstante que el clasificador NB fue creado originalmente para su uso en el dominio de la recuperación de información y minería de datos, es precisamente la independencia que se crea sobre los textos lo que constituye una de sus desventajas, siendo la clasificación de documentos una de sus mejores aplicaciones debido a que el texto subyacente no representa mayor complejidad para este algoritmo. Una justificación para la presunción de la independencia condicional es que, si se sabe que un documento es acerca de un cierto tema o tópico, esto es un buen indicativo de otros tipos de palabras que se encuentran en el documento (Barber, 2017).

Se ilustra a manera de ejemplo práctico en la Figura 3.7 cómo se realiza la clasificación de documentos empleando NB. Como se observa existen dos personas que hablan acerca de tres temas en particular, uno con mayor frecuencia que los otros dos, de tal manera que cada tema tiene una mayor probabilidad o representatividad. A través de la fórmula indicada anteriormente y una posterior serie de operaciones se obtiene la probabilidad sobre el tópico que se está se decidió emplear tratando en un documento o, en este caso, una persona en particular.

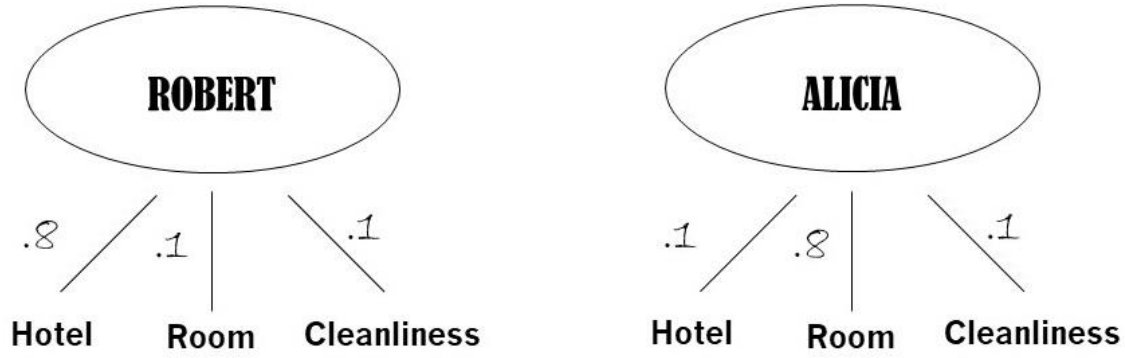

Figura 3.7 Ejemplo de probabilidades para Naïve Bayes

La herramienta seleccionada emplea el clasificador MNB, el cual de acuerdo a la documentación de Scikit-Learn (2018) es adecuado para la clasificación con características discretas como el conteo de palabras para la clasificación de texto. La distribución multinomial referida es una generalización hecha de la distribución binomial, empleada en probabilidad. En el caso de este clasificador son tan sólo tres parámetros los que se pueden configurar para probar el clasificador, los cuáles se explican a continuación:

Alpha: Parámetro de suavizado aditivo. Existen ocasiones en las que un término no existe en un conjunto de palabras, pero sí existe en otro que, a la inversa, tiene una palabra que no existe en el conjunto anterior, esto es visto desde un razonamiento sencillo, ya que esto podría generalizarse a ejemplos más extensos, pero la base de este razonamiento es el descrito. Ante esta situación y con el sistema de estimación de probabilidades podría asignarse un 0 como probabilidad al término 'único' lo cual puede ser erróneo porque se están descartando elementos que pueden existir en otros conjuntos de palabras y viceversa. Para resolver este problema se aplica el llamado 'suavizado sumar uno' o *add-one smoothing*.

Esta técnica también conocida como suavizado aditivo o suavizado de Laplace (Coelho & Richert, 2015) es una sencilla técnica que suma uno a todas las ocurrencias de características (palabras). Esto tiene la suposición implícita de que incluso si no se ha visto una palabra dada en todo el corpus, aún existe la posibilidad de que sólo sea que la información que se está analizando no incluya esa palabra. Se ilustra esto con el siguiente ejemplo: teniendo las clases X y Y con palabras: hotel, habitación y avión, de la siguiente manera:

X: hotel=3, habitación=1, avión=0

(En esta clase, hotel aparece tres veces y habitación sólo una, avión ninguna).

Y: hotel=0, habitación=1, avión=3

(En esta clase, avión aparece tres veces y habitación sólo una, hotel ninguna).

Si se descartaran los términos que aparecen 0 veces en cualquier otra clase quedarían de la siguiente forma:

X: hotel=3, habitación=1

Y: habitación=1, avión=3

Lo cual llevaría a estar perdiendo información importante ya que futuros términos o palabras serían descartados del conjunto de datos. Por lo que, cómo se ha explicado antes, se suma un uno a las probabilidades dadas para considerar todas las palabras existentes.

fit_prior: Si se aprenderán las probabilidades de clases anteriores o no (Scikit-Learn, 2018).

class_prior: Probabilidades previas de las clases. Si se especifican, las probabilidades previas no se ajustan de acuerdo a los datos (Scikit-Learn, 2018).

3.2.6. Support Vector Machine con entrenamiento SGD

La herramienta con la cual se desarrolló la parte de experimentación proporciona un método con el cual poder realizar pruebas con clasificadores lineales (por ejemplo, SVM o LR) con entrenamiento de Gradiente Descendente Estocástico (SGD). El método permite desarrollar pruebas empleando diferentes clasificadores, no obstante, se decidió emplear el valor default el cual permite usar una SVM lineal. Se preserva el concepto original de cómo funciona el método SGD en el que se va reduciendo una serie de pesos en la función hasta alcanzar un costo global mínimo a lo largo de un gradiente. De acuerdo a Do & Tran-Nguyen (2016) el algoritmo SVM-SGD converge rápidamente a la solución óptima debido a que el problema no restringido (descrito en su trabajo) es un problema de optimización convexa en conjuntos de datos muy grandes.

Se describen a continuación los parámetros que fueron empleados en este clasificador (lo cual fue tomado de (Scikit-Learn, 2018)):

Tasa de aprendizaje: Se refiere al plan o programa que se utiliza para ajustar el valor que viene establecido por el parámetro **Eta0** que es el valor inicial de la tasa de aprendizaje.

Penalty: La penalidad (también conocida como término de regularización) que se utiliza, el valor predeterminado es 'l2', que es el regularizador estándar para modelos SVM lineales, aunque también se empleó ‘l1’. De la cual ya se comentó al respecto en el perceptrón multicapa.

L1 ratio: Método de regresión regularizada a emplear: 'red elástica' o *'elastic net mixing parameter'*.

Iteraciones: El número de pasos sobre los datos de entrenamiento (también llamados epochs).

3.2.7. Tipos de Aprendizaje

Se habló en el punto anterior sobre la capacidad que tienen los algoritmos de ML de aprender sobre los datos. Ahora bien, es necesario resaltar que, de acuerdo a la información disponible, se puede clasificar el aprendizaje en supervisado, no supervisado y semi- supervisado. Esta diferencia en los conjuntos de datos es la que permite elegir los algoritmos de ML que mejor se adapten a la información y entonces, poder obtener los mejores resultados posibles en cuanto a la predicción hecha sobre los datos.

El aprendizaje supervisado ocurre cuando un algoritmo aprende de datos de ejemplo y estos tienen asociadas respuestas objetivo las cuáles podrían consistir de valores numéricos o cadena como clases o etiquetas; esto con el fin de poder predecir a futuro la respuesta correcta cuando se presentan nuevos ejemplos (Mueller & Massaron, 2016). En la Figura 3.8 se presenta un ejemplo gráfico de este tipo de aprendizaje. Como ejemplo, tómese el caso de la Tabla 3.3 en la que se presentan una serie de opiniones respecto a una unidad de hospedaje.

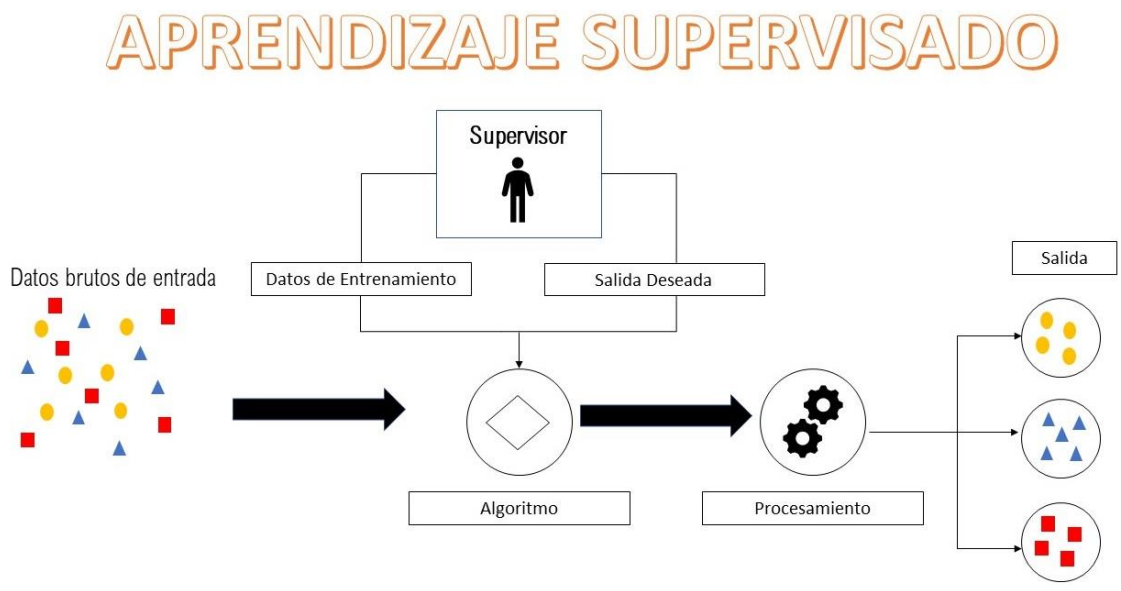

Figura 3.8 Representación del aprendizaje supervisado

Descripción de la crítica (cualitativo)	Valoración (cuantitativo)
Habitación normal, cama cómoda, el agua de la ducha salía caliente. Muy bien calidad-precio. Cerca de la estación de tren. Habitación limpia. En el centro estas en unos 15 min andando.	5
Fui por motivos de un examen dos noches. Está en una buena zona para aparcar porque hay un aparcamiento cerca del puerto. La habitación estaba bien. Prácticamente solo la utilicé para dormir y el precio estuvo acertado.	4
Buen servicio; la habitación está bastante bien y está bien situado. Está muy cerca del centro y de la estación de trenes. Recomendable para cuando vayáis de visita turística y estar unos días.	3
El hostal está al lado de las estaciones de autobús y tren, por lo que en ese sentido es recomendable. Pero solo en ese, porque el hotel no pasa de correcto. Las habitaciones son viejas, no especialmente agradables y tampoco es excesivamente barato para lo que ofrece.	2
Lo único positivo de este hostal es su ubicación y la limpieza. La verdad es que no sé cómo se atreven a poner estos precios con el servicio que dan. Las habitaciones son antiquísimas, y muy pequeñas para 3 camas.	1

Tabla 3.3 Ejemplo de datos para aprendizaje supervisado

El aprendizaje no supervisado, como se ejemplifica en la Figura 3.9, sucede cuando un algoritmo aprende de ejemplos simples que no tienen asociada ninguna respuesta, permitiendo al algoritmo determinar los patrones de los datos por su cuenta. Un ejemplo práctico del uso de este tipo de algoritmos es en los sistemas de recomendaciones empleados en la web en la que se proporcionan sugerencias basadas de compra basado en lo que se ha comprado en el pasado (Mueller & Massaron, 2016).

El aprendizaje semi-supervisado es un paradigma que combina el aprendizaje supervisado y el no supervisado, de tal manera que los algoritmos de ML aprenden tanto de datos que están etiquetados, como no etiquetados. En ciertos estudios (Zhu & Goldberg, 2009) se ha comprobado que aplicar esta técnica conlleva ciertas ventajas ya que la parte no supervisada llega a aprender de los datos supervisados. Esto representa una ventaja ya que es más común recolectar datos no supervisados que supervisados.

Figura 3.9 Representación del aprendizaje no supervisado

El aprendizaje reforzado como se representa en la Figura 3.10, se da cuando se le presentan ejemplos al algoritmo que no tienen etiquetas (como en el aprendizaje no supervisado), pero se puede acompañar de un ejemplo con retroalimentación positiva o negativa de acuerdo a la solución que el algoritmo proponga,. Está asociado al aprendizaje humano a prueba y error en el que se puede aprender a prueba y error, en el que los errores están asociados a penalidades o costos.

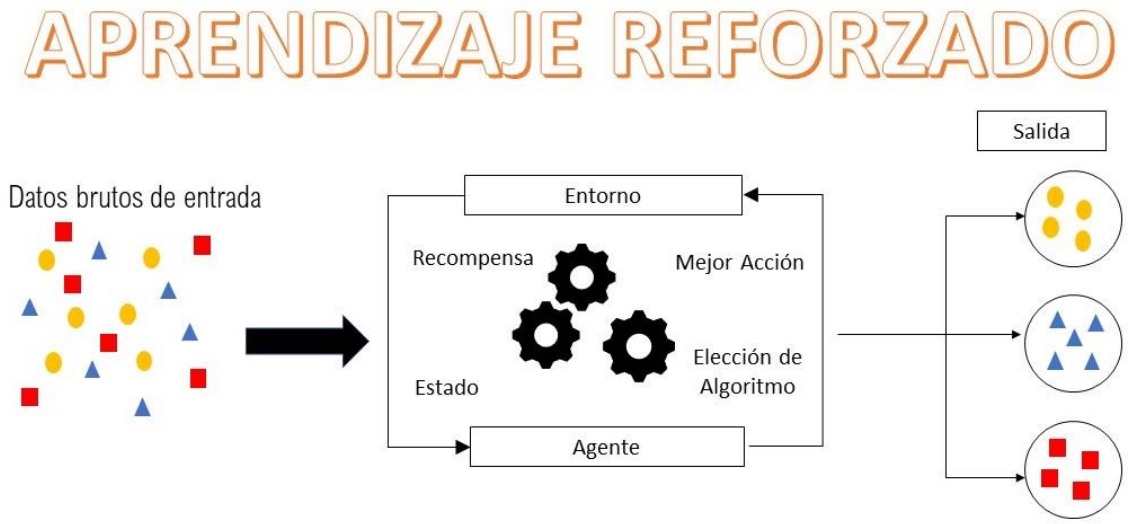

Figura 3.10 Representación del aprendizaje reforzado

Existe una gran diversidad de algoritmos de ML, el querer mencionarlos a todos está fuera del alcance de este trabajo de tesis además de que, debido a la rápida evolución del campo, no se podrían abarcar todos los existentes. Por lo que a continuación después de explicar las fases de entrenamiento y prueba se describen los algoritmos de ML que fueron más explorados durante el trabajo de tesis doctoral. Es conveniente denotar que

esta acotación se ha realizado en función del desempeño que los mencionados algoritmos presentaron en la solución propuesta, descrita en el Capítulo 4.

3.2.8. Fases de entrenamiento y prueba

La programación tradicional o estructurada se compone de una serie de instrucciones, ciclos y estructuras lógicas que están enfocadas en solucionar un problema en particular. Conforme la solución construida tiene que resolver más casos especiales, son mayores los casos que se tienen que ir atendiendo. Este modelo ha servido durante muchos años y sigue sirviendo para una gran variedad de modelos computacionales, sin embargo, en el estado actual en que las comunicaciones permiten generar millones de datos por segundo, este modelo de programación resulta obsoleto e incluso poco eficiente ya que no se pueden estar generando programas por cada situación particular que surge.

Por lo anterior, el ML ha servido como la alternativa para dar respuesta al procesamiento de tales volúmenes de datos. Una vez construido el MLM se procede a evaluar la efectividad del mismo al aplicarlo al conjunto de datos disponible, sin embargo, ha de tenerse en cuenta que no es una práctica común el efectuar la validación del MLM en todo el conjunto de datos, por lo que ha de tomarse una parte de los datos para realizar el entrenamiento y otra parte para la prueba del MLM. Un enfoque es emplear el 70% para el entrenamiento y 30% para la prueba, o bien 80% y 20% respectivamente. Estos valores dependen del diseñador del MLM y el volumen de datos disponible; incluso se puede manejar un tercer subconjunto de datos denominado de validación el cual puede ser de 20% dejando 70% para el entrenamiento y 10% para la prueba.

3.2.9. Métricas de evaluación

Las métricas de evaluación son un conjunto de indicadores que miden diversos elementos sobre el desempeño de un clasificador. No todas las métricas son empleadas con la misma importancia e incluso, dependiendo de la herramienta empleada, no todas está disponibles. Algunas de las métricas más empleadas son *Accuracy, Precision* y *Recall* como se describe en esta sección. De acuerdo a Hossin & Sulaiman (2015), muchos clasificadores emplean *accuracy* como una medida para discriminar la solución óptima durante el entrenamiento de clasificación. Existen diversos tipos de clasificación como la clasificación binaria o la clasificación multiclase, en lo particular se hace mención de la primera.

En la clasificación binaria se generan diversos indicadores de acuerdo con la siguiente tabla:

	Valores Reales Positivos	**Valores Reales Negativos**
Predicción Positiva	Verdadero positivo (vp)	Falso negativo (fn)
Predicción Negativa	Falso positivo (fp)	Verdadero negativo (vn)

Tabla 3.4 Matriz de confusión

La situación de estos indicadores se explica a continuación:

- **Verdadero positivo (vp) –** Es una predicción indicada como positiva que fue clasificada correctamente por el algoritmo como tal. Por ejemplo, una reseña de una película que indica que es positiva, fue predicha como tal.
- **Falso negativo (fn) -** Es una predicción indicada como negativa, pero realmente es positiva. Por ejemplo, la detección de cierta enfermedad que es existente en un paciente ha sido diagnosticada como no existente.
- **Falso positivo (fp) -** Es una predicción indicada como positiva, pero realmente es negativa. Como ejemplo podría ser un sistema que detectara que un coche está óptimo para salir a carretera cuando en realidad no lo está.
- **Verdadero negativo (vn)** - Es una predicción indicada como negativa que fue clasificada correctamente por el algoritmo como tal. Como un clasificador de imágenes que detectó correctamente que una fruta no es una pera sino una manzana.

En la Tabla 3.5 se presentan las fórmulas mediante las cuales se calculan estas métricas:

Métrica	**Fórmula**	**Descripción**
Accuracy	$\frac{vp + vn}{vp + fp + vn + fn}$	Mide la tasa de predicciones correctas sobre el total de instancias evaluadas
Precision	$\frac{vp}{vp + fp}$	Se emplea para medir los patrones positivos que están pronosticados correctamente de los patrones totales predichos en una clase positiva.
Recall	$\frac{vp}{vp + vn}$	Es empleado para medir la fracción de patrones positivos que están clasificados correctamente

Tabla 3.5 Métricas de evaluación [adaptado de (Hossin & Sulaiman, 2015)]

En este trabajo de tesis se exploraron las métricas expuestas en la Tabla 3.5 para evaluar el desempeño de los clasificadores, no obstante, en el Capítulo 5 se presentan los resultados correspondientes a *accuracy.* Como se describe en el trabajo desarrollado por Hossin, Sulaiman, Mustapha, Mustapha, & Rahmat (2011), se establece que *accuracy* es una métrica comúnmente empleada para comparar el rendimiento de clasificadores.

En el apartado anterior se hacía mención de dejar ciertos porcentajes de los datos para llevar a cabo las fases de entrenamiento y prueba con los datos. No obstante, el emplear un porcentaje de los datos para el entrenamiento y otro tanto para la prueba, no siempre es el enfoque más ideal ya que no se alcanza a comprobar la efectividad del modelo en la totalidad de los datos, por lo que existen otras alternativas a emplear solamente las fases de entrenamiento y prueba. Una de estas es el modelo *KFold cross validation* o validación cruzada de K iteraciones en el que los datos de muestra se dividen en K subconjuntos. Cada K subconjunto o *fold* se va probando hasta completar todos los *folds* en que se dividió el conjunto original de datos, resultando como una de sus ventajas la estimación precisa del rendimiento (Refaeilzadeh, Tang, & Liu, 2009). En la Figura 3.11 se presenta una representación de cómo se dividirían los conjuntos de entrenamiento y prueba para una serie de 10 iteraciones.

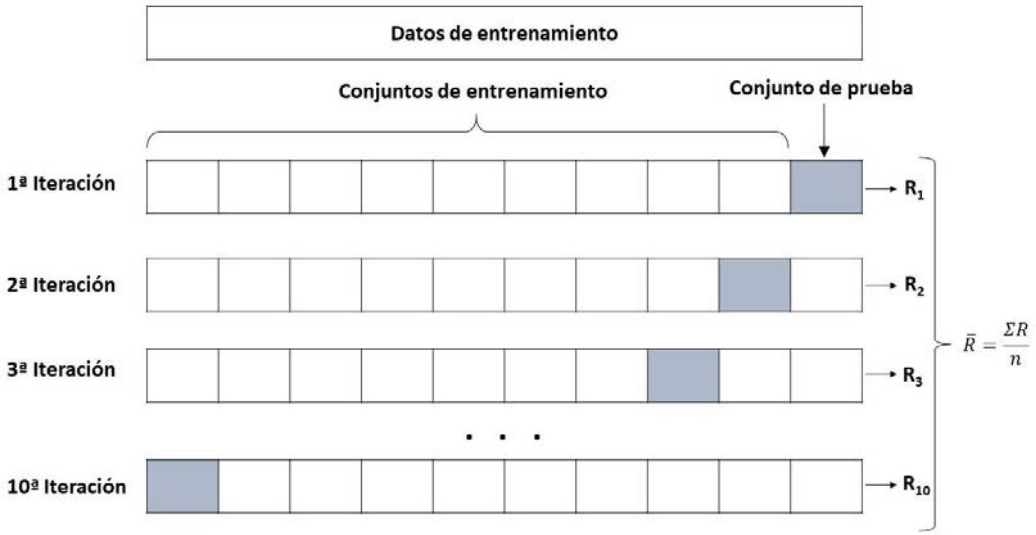

Figura 3.11 Validación cruzada

3.2.10. Sobreajuste

El sobreajuste u *overfitting* se caracteriza por haber obtenido muy alta *accuracy* para un clasificador cuando se le evaluó en el conjunto de entrenamiento pero, por el contrario, se consiguió baja *accuracy* cuando se evaluó dicho clasificador en un conjunto de prueba separado (Subramanian & Simon, 2013). Dicho de otra forma, normalmente los clasificadores reciben un conjunto de vectores los cuáles han sido procesados a partir de la entrada original en una etapa anterior.

Dicha etapa de procesamiento puede haberse enfocado en ajustarse lo más posible al conjunto de datos de entrada, de tal manera que cuando se le prueba en uno o más clasificadores que también pueden estar adaptados a dichas entradas, entonces se obtienen

valores altos en accuracy pero cuando se intenta probar esta solución con nuevos datos de prueba, los resultados resultan ser bajos al no adaptarse al diseño original de la solución. Por lo anterior, una solución óptima debe enfocarse en conseguir una *accuracy* que sea lo más alta posible sin ajustarse demasiado a un conjunto particular de datos.

3.3. Procesamiento de Lenguaje Natural

3.3.1. Introducción

El PLN siempre ha estado unido a la relación humano-computador, ya sea porque necesitan "hablarse" o "entenderse" entre ambos (Yue et al., 2012). Se trata de una disciplina de la informática siempre en evolución y que se vuelve cada vez más compleja debido a las mayores exigencias del sector (Jimenez-Marquez et al., 2018). El área de PLN inició en la década de 1950 como la intersección de la inteligencia artificial y la lingüística (Nadkarni et al., 2011). Desde entonces diversas teorías han ayudado a la evolución del PLN, como la notación Backus-Naur Form y herramientas como los analizadores léxico y sintáctico.

El PLN ha constituido dentro de la informática, uno de los mayores retos para los investigadores, y se sigue proyectando como una de las áreas de mayor interés para la ciencia y la tecnología en el futuro a mediano y largo plazo. Una de las áreas donde PLN está siendo más útil en la ciencia es en la Medicina y la Biomedicina. Otras áreas de aplicación donde se puede encontrar la conjunción del PLN con otras áreas es, por ejemplo, en los sistemas de información geográfica (Calì et al., 2011) o en entrevistas motivacionales (Tanana et al., 2016).

Dentro del PLN y dependiendo del enfoque del proyecto de investigación se distinguen varias etapas a seguir dependiendo del nivel del estudio o la profundidad a la que se pretende llegar, entre estas se encuentran: análisis léxico, análisis sintáctico, análisis semántico, reconocimiento de entidades (NER), polaridad del sentimiento, etc. Para cada una de las etapas de PLN se pueden construir analizadores propios que realicen el trabajo deseado, no obstante, existen diversas herramientas desarrolladas para llevar a cabo todas, o la mayoría de las etapas mencionadas.

3.3.2. Tokenización

El análisis léxico visto desde un enfoque simple es el proceso mediante el cual se transforma una oración o un texto en una secuencia de caracteres. Estos caracteres son reconocidos mediante los elementos que los constituyen y entonces de acuerdo con el nivel (habilidad para reconocer símbolos especiales) en que este haya sido construido se forman los denominados *tokens* (elementos, partes, trozos), es decir, cuando se encuentra un punto, una coma o un espacio, por mencionar los más comunes. De tal manera que la *tokeniz*acion es el proceso con el que se van formando los *tokens* que pertenecen a una unidad de texto, que podría ser un libro, un capítulo, un párrafo o una oración.

De acuerdo a Prado-Daud & Costa-Ribeiro (2010) el token es un conjunto de caracteres que forman una unidad lingüística con significado. Estos autores además señalan que: "una de las primeras tareas que se realizan en el preprocesamiento de un texto es la *tokeniz*ación, que identifica y separa las expresiones que aparecen en el texto separadas por espacios, comas, puntos, etc.". Por ejemplo, al nivel de palabra no es muy claro el cómo tratar las siguientes cadenas del idioma inglés: "won't", "14,30", "Los Angeles", o "so-called". De tal manera que el analizador léxico debe poder identificar las partes donde una palabra termina y otra comienza. En la Figura 3.12 se presenta de manera gráfica el proceso de *tokeniz*ación para una expresión dada.

Figura 3.12 Ejemplo del proceso de Tokenización

3.3.3. Los N-gramas

El modelo de n-gramas no es más que una secuencia contigua de n elementos (Nagalavi & Hanumanthappa, 2016). Por lo que el modelo es empleado para predecir las palabras provenientes de una secuencia dada de texto. Está basado en el modelo de lenguaje probabilístico donde la palabra anterior se predice en la forma de un orden (n-l) del modelo de Markov. La fórmula que define estas probabilidades se define así:

$$p(W) = \prod_{i=1}^{n} p(w_i|w_1, w_2, \dots, w_{i-1})$$

De tal manera que se estiman las probabilidades de las palabras antecesoras dentro de un contexto dado. Cuando n=1 es un unigrama, si n=2 es un bigrama, si n=3 es un trigrama, comúnmente son más empleados los bigramas y trigramas.

En lo que respecta al particular contexto dentro de esta tesis, los n-gramas también son empleados puesto que después del proceso de *tokeniz*ación, se emplean los tokens para construir tales n-gramas. Por lo que los n-gramas son de gran utilidad puesto que muchas veces son empleados para identificar pequeñas frases dentro de un texto que son comunes pero que de otra forma podrían no ser identificables, ejemplos: "coche de alquiler", "habitación compartida", "aire acondicionado", etc. En la Figura 3.13 se detalla de forma gráfica tres ejemplos de N-gramas para N=1, 2 y 3.

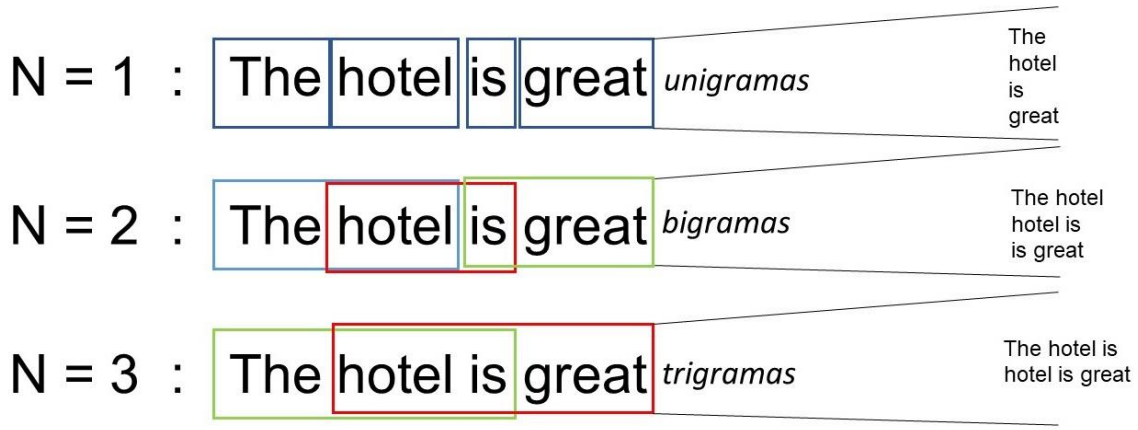

Figura 3.13 Ejemplo de N-gramas

3.3.4. Stop Words

Las *stop words* o palabras de parada/alto son palabras que abundan en el texto que se está analizando debido a que son palabras auxiliares como artículos o proposiciones que aportan muy poco al estudio del texto en general. Se denominan *stop* (alto) debido a que cuando el analizador las encuentra se detiene al encontrar una de estas, descartándola como candidata de ser incluida (Sadeghi & Vegas, 2014) en el índice general de palabras recuperadas o features. Al eliminar estas palabras se reduce el número de palabras que conforman el diccionario de términos con lo cual se hace más efectivo el proceso de recuperación de la información.

Para iniciar el proceso de descartar tales palabras, comúnmente se recurre a un diccionario ya construido el cual contiene los términos que han sido identificados previamente como *stop words*. Sin embargo, en función de la herramienta empleada, puede ser que dicho diccionario se complemente con un diccionario personalizado que podría a su vez contener términos adicionales que se repiten para un contexto muy particular, los cuáles a gusto del diseñador, no deberían ser tomados en cuenta.

De manera general, esta técnica es efectiva pues permite a etapas posteriores enfocarse en términos que son de mayor relevancia para el estudio que se está desarrollando, que por el contrario, analizar términos repetitivos que nada agregan a contexto general de una oración o un párrafo. Las posibles desventajas de remover las *stop words* podría ser el que sea eliminada una palabra que junto con otra forman un término establecido, por ejemplo “de facto”, podría conservarse “facto” dejando a un lado “de”.

3.3.5. Stemming

Comúnmente cuando se escribe, se emplean palabras que derivan de la misma raíz, por ejemplo:

transforma, transformé, transformación, transformado, etc ➔ transform.

Pero este es el ejemplo más sencillo que puede encontrarse, las complicaciones surgen cuando se encuentran términos comunes como:

soy, seré, eres, fuiste, era, etc. ➔ ser.

El criterio de reducir los términos anteriores a la palabra ser puede ser criterio de algún desarrollador o científico del área. De hecho, puede tenerse que remitir a las bases mismas del lenguaje a analizar para comprender hacia donde se pueden reducir tales términos, de tal manera que esto puede abarcar áreas de la morfología y la lingüística. De manera que son ciertos algoritmos los que realicen durante el proceso de PLN el proceso de convertir las palabras a su raíz o *stemming*.

Uno de los primeros algoritmos propuestos en esta área y además uno de los más referidos dentro del área es el propuesto por Porter (1980), el cual se basa en diversos elementos como generar una secuencia de reglas de eliminación de sufijos. Como menciona Willett (2006) quien en su trabajo realiza una comparativa del algoritmo de Porter hasta las mejoras y nuevas propuestas que se han hecho en este campo, en lugar de reglas basadas en el número de caracteres que quedan después de la eliminación, Porter emplea una longitud mínima basada en el número de cadenas consonante-cadena-consonante tras remover un sufijo.

En lo que respecta al presente trabajo de tesis, el emplear *stemming* en la preparación del corpus tiene las siguientes ventajas:

1. Ayuda de forma importante a reducir el número de features que se analizan en etapas posteriores.
2. Reduce la tabla de términos que es construida durante la fase de TF-IDF y el Modelo de Espacio Vectorial. Esto se explica en los siguientes apartados.
3. Al reducirse la tabla, pueden integrarse features que tienen mayor relevancia para el estudio de los elementos que conforman el corpus.

3.4. Big Data

La cantidad de información digital que se produce actualmente a través de diversos medios y plataformas de hardware y software es ingente; incluso las tendencias establecen que esta información seguirá incrementándose aceleradamente lo cual requiere de técnicas computacionales que permitan el almacenaje, tratamiento y análisis de tales volúmenes de datos. Esta información es variada y puede encontrarse en diversas fuentes, sólo por citar algunas de estas:

- El IoT recoge información de cada sensor implantado en casas, automóviles, fábricas o plantas de generación de energía de cualquier tipo.
- Los videos producidos y subidos a plataformas digitales como Youtube o Facebook
- La información que es emitida por millones de dispositivos de Geoposicionamiento (GPS) que informan sobre los desplazamientos humanos
- Las diversas redes sociales en las que los usuarios comparten sus experiencias y emociones.
- Plataformas digitales de experiencias de experiencias de usuario sobre diversos tópicos generalmente orientadas a la evaluación de los servicios recibidos.

Tradicionalmente las empresas almacenaban sus datos en Sistemas de Bases de Datos Relacionales (RDBMS) (Leavitt, 2010), pero este esquema aunque no puede considerarse obsoleto, no puede cubrir las necesidades actuales de las organizaciones. Algunos de los impedimentos para almacenar los datos que se generan actualmente en tiempo real en las RDBMS son: las limitaciones propias para el almacenamiento de los datos, ya a sea a nivel tabla o campo, la imposibilidad de crecer a entornos distribuidos o virtualizados (Ji Qi, Qian, & Luo, 2009), la velocidad con la que se procesan las consultas, la rapidez con que se devuelven los conjuntos de información deseados, etc.

Debido a lo anterior surgió el paradigma BD el cual está causando un gran impacto en la comunidad académica (Wamba, Akter, Edwards, Chopin, & Gnanzou, 2015) debido a que BD tiene el potencial de transformar los datos disponibles en información de valor que antes era más difícil de encontrar a través de los procesos comúnmente establecidos. No obstante, es importante entender qué se quiere obtener de los datos partiendo de aquellos se ya se han recolectado o se plantean recolectar, deben establecerse los objetivos a los cuáles se quiere llegar a través del análisis de los datos ya que BD no es una estrategia que debe ser vista como un fin, sino un medio para llegar a obtener el valor esperado en los datos.

Existen diversas soluciones tecnológicas para BD que gestionan los GVDNE y en determinadas arquitecturas conviven dos o más de estas soluciones dependiendo del problema al que se quiere dar solución y del volumen de datos que se está manejando. Por lo que a continuación, se ofrece un panorama general de los componentes principales que integran una solución de BD.

3.4.1. Sistemas de Archivos Distribuidos

El enfoque tradicional de almacenaje de archivos comprendía el uso del sistema local del ordenador. Aunque esto ha servido durante mucho tiempo en otro tipo de proyectos, e incluso, es potencial su uso en proyectos de GVDNE, no es lo más factible ya que al interactuar constantemente con los datos contenidos en los GVDNE, el acceso a disco puede volverse lento e ineficiente.

Por otro lado, la Nube o Cloud Computing se ha convertido en los últimos años en el modelo a seguir para proyectos que involucran el concepto serverless, es decir, no estar condicionado al uso de cierta infraestructura física, particularmente de servidores de cualquier tipo. En este enfoque de cómputo, la gestión de los servidores es gestionado por un tercero, lo cual permite a los equipos científicos y de desarrollo poder enfocar sus esfuerzos en sus objetivos principales, pudiendo disminuir o anular la carga de trabajo que involucra el administrar los servidores necesarios (por ejemplo: de base de datos, de red o de archivos). De tal forma que el acceder a los datos en entornos de nube también constituye una estrategia adecuada, aunque con los inconvenientes del coste que puede conllevar hacer uso de la propia nube, así como depender siempre del enlace de red.

El manejo de la información digital siempre ha recibido especial atención por investigadores, académicos y empresas de tecnología. Desde el inicio de la era informática diversos medios fueron empleados para almacenar la información, no se reproduce en este espacio toda la historia relacionada con estos aspectos ya que el material daría para redactar varias tesis, además que esto no es el objeto del presente estudio. La

descripción de esta primera etapa en la segunda fase comienza en la era informática en la que existían ya los primeros servidores de archivos. Cuando los usuarios de estos servidores querían tener acceso a algún archivo tenían que conocer la ruta física del recurso, lo cual llevaba a diversos problemas ya sea de seguridad o de búsqueda del mismo.

Las técnicas de manejo de la información y versiones más avanzadas (para su época) de servidores de archivos permitieron el desarrollo de los Sistemas de Archivos Distribuidos (DFS por Distributed File System). Los DFS son sistemas que permiten ocultar la ubicación física de recursos en la red para que los usuarios de esta le encuentren mediante su nombre lógico, esto además permite evitar la redundancia que existe cuando se encuentran los recursos en diversas ubicaciones, generando con esto la repetición innecesaria de la información o un manejo de versiones erróneo.

Diversas implementaciones de DFS se han propuesto e implementado en diversos sistemas, en (Donnelley, 1995) ya se llevaban a cabo comparaciones de los enfoques posibles para hacer el manejo de los recursos de forma más transparente y sobre todo eficaz para el usuario final. Mientras que Machanick (2005) afirma que el manejo de DFS además de presentar ventajas también conlleva diversas consideraciones de seguridad, diseño y respuesta que han de tomarse en cuenta al momento de emplear cualquier DFS.

Por otro lado, cuando se habla de temas de computación distribuida se habla también del procesamiento en paralelo. En este caso, se está refiriendo no sólo al hecho de que el sistema a cargo almacena los archivos en diversas ubicaciones físicas, sino que también, se lleva a cabo el procesamiento de las tareas en diversos nodos de la red con lo cual se busca aprovechar de cada nodo el espacio de almacenamiento o la capacidad de procesamiento, entre otras prestaciones de hardware. Estos sistemas fueron creciendo para dar respuesta a las exigencias computacionales del momento, las cuáles incluso podían llegan a rebasar a la capacidad instalada y a la arquitectura propuesta.

De particular interés resulta el trabajo desarrollado por Shen & Choudhary (2004) en el cual se afirmaba desde entonces que “la capacidad de cálculo se acelera más rápido que la capacidad para manejar y visualizar los datos resultantes". Los autores reflejaban en su trabajo la preocupación por contar con sistemas de alto desempeño que pudieran llevar a cabo cómputo distribuido y paralelo, para lo cual construyeron un modelo propio. El trabajo mencionado es significativamente anterior (considerando el tiempo en que evoluciona el cómputo) a la era del BD la cual planteó mayores necesidades de procesamiento y manejo de archivos.

En Cho, Jin, Lee, & Schwan (2014) se menciona que en los sistemas tradicionales de procesamiento en paralelo el almacenamiento de los datos y el almacenamiento eran ejecutados en distintos nodos los cuales estaban dedicados a uno u otro propósito, de manera que era fácil hasta cierto punto el repartir las tareas de cómputo al estar fácilmente estas diferenciadas entre los nodos de la arquitectura. Pero como el tamaño de los datos creció drásticamente nuevas arquitecturas y modelos de programación fueron propuestos dando espacio a modelos como MapReduce, el cual es explicado en los aspectos de implementación.

Por otra parte, se destaca el hecho de que para poder hacer uso del DFS no sólo se cuenta con las capacidades que la infraestructura propia del cómputo local puede aportar, ya que el paradigma del *Cloud Computing* (cómputo en la nube) tiene mucho que aportar al framework propuesto. Este paradigma empezó a tomar fuerza en ámbitos académicos y comerciales alrededor del año 2010 siendo entonces una forma alterna de almacenar no sólo los ficheros locales sino también grandes volúmenes de información de universidades o centros de investigación. A día de hoy, el Cloud Computing es mucho más que un simple medio alterno para el almacenamiento de datos, ya que se ha convertido en la solución tecnológica a seguir tanto por investigadores y académicos, como por empresas privadas.

Cloud Computing surge de la combinación tradicional de tecnologías de cómputo y redes, tales como: computo en malla (Grid computing), cómputo distribuido, cómputo paralelo y la virtualización (Liu & Dong, 2012). Cloud Computing es entonces también parte del modelo propuesto tanto como el sustituto de DFS como en otras etapas del modelo que se comentan en su momento. Las ventajas de emplear cloud computing son diversas y aunque constantemente se habla de estas en el aspecto comercial, lo cierto es que cuando aumenta la demanda de capacidad de procesamiento para aplicarlo a los datos, el cloud provee las técnicas necesarias (desarrolladas para el cloud) para hacer frente a estas necesidades tanto en hardware como en software.

Uno de los inconvenientes que ha encontrado esta tecnología para su adopción lo representan los costos que se llegan a manejar, los cuáles pueden llegar a ser prohibitivos para grupos de investigación con escasos fondos o pequeñas organizaciones. La anterior barrera puede ser solventada ya que existen diversos esquemas para ayudar a este grupo de interés por lo que la adopción de la misma puede, en el caso de los proyectos de investigación, ayudar a que estos estudios se lleven a cabo empleando un poder de cómputo más potente.

3.4.2. Hadoop Distributed File System

El sistema de archivos distribuidos de Hadoop (Hadoop Distributed File System o HDFS) es un sistema que está diseñado para trabajar en hardware común y ser resistente a las fallas de los nodos a la vez que proporciona un alto rendimiento de datos (Karau, Konwinski, Wendell, & Zaharia, 2015). Hadoop es un marco de código abierto basado en el entorno de Java, que contribuye al procesamiento de grandes conjuntos de datos en un entorno de cómputo distribuido. HDFS provee almacenamiento redundante para BD almacenando los datos a través de un clúster sencillo (Shvachko, Kuang, Radia, & Chansler, 2010), extendiendo así la cantidad disponible de almacenamiento qué una sola máquina puede tener. Pero debido a la naturaleza de red conectada de un sistema de archivos distribuido, HDFS es más complejo que los tradicionales sistemas de archivos (Bengfort & Kim, 2016).

Las características de HDFS son:

- Provee acceso de alto rendimiento a los bloques de datos. Cuando los datos no estructurados son montados en este sistema, son convertidos en bloques de datos de tamaño fijo. Los datos son cortados en bloques de tal manera que sea compatible con el almacenamiento del hardware común.
- HDFS provee una interfaz limitada que ayuda a administrar el sistema de archivos y permitirle que escale. Esto asegura que se puedan ampliar o reducir los recursos en el clúster.
- Implementa el patrón “escribir una vez, lee varias”. No se permiten escrituras o agregaciones aleatorias a los archivos.
- Está optimizado para una lectura grande y continua de archivos, no realiza lecturas o selecciones aleatorias de datos.

3.4.3. Spark

Apache Spark es una plataforma de computación en clúster diseñada para ser rápida y de uso general (Karau et al., 2015). Spark extiende el conocido modelo MapReduce para soportar más formas de cómputo, incluyendo consultas interactivas y procesamiento de flujos. Spark además está diseñado para cubrir un amplio rango de cargas de trabajo que de otra forma requeriría sistemas distribuidos separados, entre estas: aplicaciones en lote, algoritmos y consultas interactivas. Spark es altamente flexible ya que cuenta con APIs en Scala (el lenguaje en que Spark está escrito), Python, Java y R, así como numerosas librerías preconstruidas.

Spark puede alcanzar baja latencia en las cargas de trabajo BD, para esto emplea la memoria para el almacenamiento de datos (Yadav, 2017). Spark se ejecuta sobre una variedad de administradores de clusters, incluido YARN y Mesos (de los cuáles se habla más adelante en este capítulo) y el propio administrador de clúster de Spark que se ejecuta en modo independiente, es decir, en un nodo único. Spark puede ser desplegado en diversos sistemas operativos, como Windows, Linux y MacOS; además puede emplear tanto en el cloud público como privado los servicios en nube de Amazon EC2. Spark puede acceder a los datos de una amplia variedad de repositorios de datos, entre los cuáles se incluyen: HDFS, Apache Cassandra, Hbase, Hive, etc (Thottuvaikkatumana, 2016).

El modelo de ejecución interno de Spark es el Grafo Acíclico Dirigido (DAG), el cual tiene múltiples niveles que forman una estructura de árbol. DAG puede tener múltiples niveles que forman una estructura de árbol. Por otra parte, DAG es más rápido que el paradigma MapReduce al no escribir los datos intermedios a disco. En la Figura 3.14 se presenta la representación de este modelo.

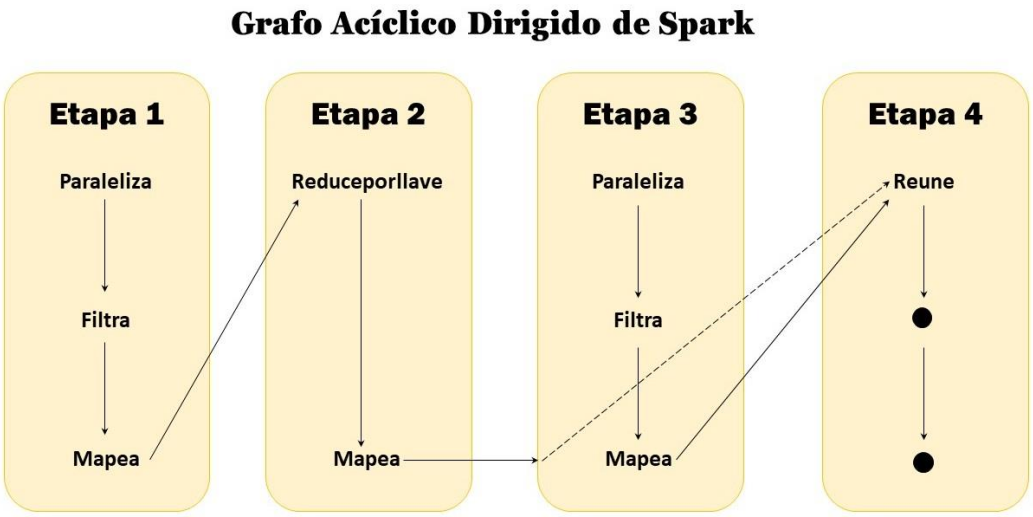

Figura 3.14 Grafo Acíclico Dirigido de Spark

3.4.4. Cassandra

Apache Cassandra es una base de datos de código abierto, distribuida, descentralizada, elásticamente escalable, altamente disponible, tolerante a fallos, consistente y orientada a filas que basa su diseño de distribución en Dynamo de Amazon y su modelo de datos en Bigtable de Google; se creó en Facebook y ahora se emplea en algunos de los sitios más populares en la Web (Carpenter & Hewitt, 2016). El proyecto Cassandra, creado originalmente para resolver problemas de mensajería de la aplicación de Facebook, fue dado a conocer por Lakshman & Malik (2010).

Las aplicaciones en la nube requieren sistemas de almacenamiento altamente escalables que sean capaces de cargas de trabajo masivas y distribuidas en clusters de servidores. Estas aplicaciones requieren rápido acceso a datos para satisfacer el uso interactivo de los almacenes de datos por varias aplicaciones, así como consultas personalizadas. Cassandra está diseñada expresamente para aplicaciones en la nube de gran volumen y baja latencia (Alapati, 2018). En la Figura *3.15* se presenta la estructura de un nodo de Cassandra, el cual se explica a continuación: cuando se realiza una operación de escritura esta es escrita en un registro de transacciones, la cual no se registra como exitosa hasta que se haya completado la operación.

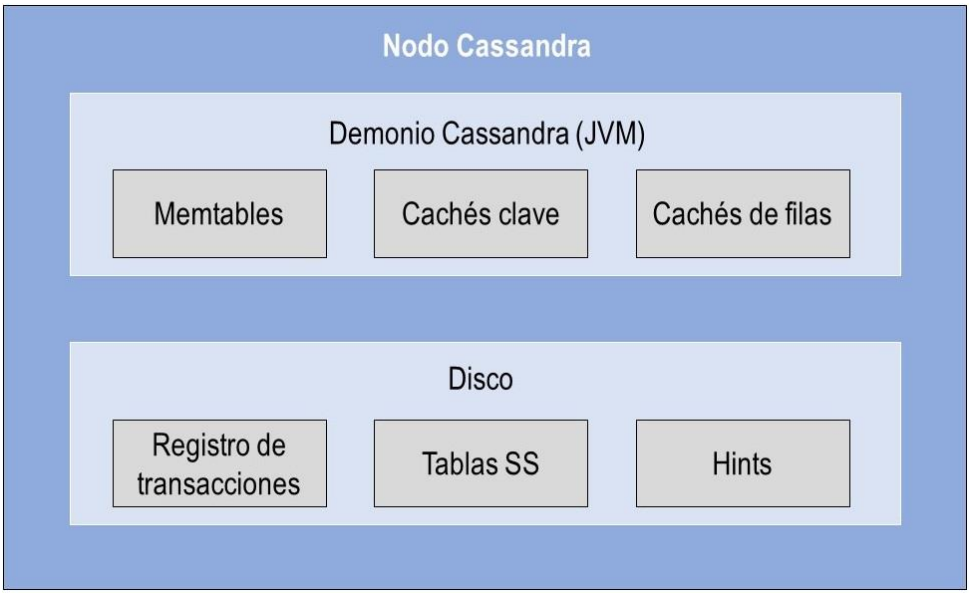

Figura 3.15 Estructura de un nodo de Cassandra
(adaptada de (Carpenter & Hewitt, 2016))

Después de ser escrito a disco, el valor se escribe en una estructura de memoria llamada *memtable*, cada memtable contiene datos para una tabla específica. Cuando el número de objetos almacenados en un memtable alcanza un límite, sus contenidos son vertidos a disco en un fichero llamado Tabla SS (*SSTable*), con lo que se crea una nueva memtable. Los caches son salvados a disco periódicamente para tenerlos disponibles más rápidamente cuando un nodo se reinicia. Los *Hints* es un tipo de recordatorio para volver a guardar datos en un nodo cuando este no se encuentra disponible.

La estructura de Cassandra permite el almacenamiento de datos a gran escala, lo que la vuelve una herramienta confiable para desarrollar estudios donde se requieren características de este tipo, como el trabajo desarrollado por Carcillo et al. (2018) en el que se emplearon Cassandra, Spark y Kafka (Kreps, Narkhede, & Rao, 2011) para desarrollar un framework escalable para la detección de fraude en tarjetas de crédito. Por otra parte Zheng et al. (2018) diseñaron el sistema de almacenamiento distribuido J-TEXT

empleando Cassandra como su motor de almacenamiento, lo cual también confirma su robustez para administrar grandes volúmenes de datos.

3.4.5. Hive

Hive es un componente del ecosistema de Apache Hadoop desarrollado por Facebook para consultar los datos almacenados en HDFS. Hive proporciona un modelo de consulta similar a SQL llamado *Hive Query Language* (HQL) para acceder a y analizar BD. Esto también se ha denominado marco de almacenamiento de datos de Hadoop y proporciona varias funciones analíticas, como ventanas y particiones (Bansal, Chauhan, & Mehrotra, 2016).

Como lo señalan sus creadores (Thusoo et al., 2010), el modelo de programación MapReduce establecido en Hadoop es de muy bajo nivel y requiere que se desarrollen programas personalizados que son complejos de reusar y mantener, por lo que señalan que Hive y su lenguaje HiveQL son una mejora para el manejo de cantidades ingentes de datos, siendo capaz de manejar esta información en poco tiempo (dependiendo del volumen de datos a tratar).

3.4.6. Pig

Pig proporciona un motor para ejecutar flujos de datos en paralelo en Hadoop, además incluye el lenguaje *Pig Latin* para expresar estos flujos de datos (Gates, 2011). Pig Latin incluye operadores para las operaciones más comunes de datos tradicionales, de igual forma brinda a los usuarios la capacidad para desarrollar sus propias funciones de lectura, procesamiento y escritura de datos (Olston, Reed, Srivastava, Kumar, & Tomkins, 2008). Pig Latin es a su vez un lenguaje de flujo de datos que sigue un proceso paso a paso para analizar los datos (Vaddeman, 2016).

3.4.7. Alternativas Cloud

Como se ha podido apreciar en los anteriores apartados, existen diversas alternativas para el manejo de grandes volúmenes de datos, por lo que escoger la opción más adecuada puede depender del fin que se tenga o del propósito que se quiera resolver. Sin embargo, cuando las cargas de trabajo son críticas y se requiere dar una respuesta inmediata debido a que se cuenta con servicios a los que se tiene que dar respuesta en tiempo real, entonces el montar alguna de estas arquitecturas en la red de computo local puede que no sea la solución más optima debido al tiempo de instalación por cada nodo si lo que se requiere es contar con una red de cómputo distribuido que soporte altas cargas de trabajo.

Por lo que existen diversas alternativas que pueden dar solución a los problemas de la organización, unas de estas son las soluciones de cómputo en la nube. Las plataformas de cómputo en la nube brindan todos los servicios de BD que se han expuesto en este capítulo, así como otros servicios que escapan del alcance de este trabajo.

Actualmente los líderes del mercado incluyen a Amazon Web Services (AWS) y Google Cloud Platform (GCP). De acuerdo a Saif & Wazir (2018), GCP ofrece un conjunto de herramientas poderosas para diversos propósitos que van desde bases de datos y almacenamiento hasta *BD Analytics* y *BD warehouse.* Algunos de los servicios más populares de GCP para el procesamiento de datos a gran escala son Google Cloud Dataproc y Google BigQuery (Krishnan & Ugia, 2015).

En lo que respecta a AWS destaca el servicio de Amazon Elastic Compute Cloud (EC2), la infraestructura como servicio (IaaS) de Amazon que además es uno de los servicios de este tipo más populares que ofrecen recursos de cómputo en Internet (Leong & Chamberlin, 2010). El rendimiento en la red es uno de los factores a considerar si se desea hacer uso de este servicio, las máquinas virtuales que son creadas en este tipo de arquitecturas pueden ver afectado su rendimiento en función del tamaño de paquetes que se envían y reciben. De manera general tanto AWS y GCP como otras tecnologías de nube (IBM, Microsoft, etc.) tienen la ventaja de que pueden conectarse a infraestructuras locales (o viceversa) para ampliar la arquitectura de datos en función del procesamiento que se desea realizar y de la carga que es necesaria llevar a la nube.

3.4.8. Administradores de Clusters

Un clúster de computación consiste en un conjunto de computadoras interconectadas para funcionar como un supercomputador o una macrocomputadora, de manera que esta unión permite aumentar tanto las capacidades de procesamiento como de almacenamiento. La necesidad de BD de tener que ejecutarse en un ambiente de cómputo distribuido vuelve idóneo el poder contar con un clúster de computación. Se ha comentado a lo largo de este trabajo de tesis que en los tiempos actuales se generan grandes cantidades de datos, por lo que en ocasiones es necesario recurrir a este tipo de arquitecturas para que se pueda compartir el hardware que soporte el almacenamiento de los datos.

Los administradores de clusters que se exploran en este apartado corresponden a aquellos que han sido diseñados para administrar clusters de Spark. En el Capítulo 4 se explica por qué se ha seleccionado Spark como la infraestructura local para el procesamiento de BD.

3.4.9. Hadoop Yarn

YARN (que en inglés significa Yet Another Resource Negotiator o Aún Otro Negociador de Recursos) es una plataforma genérica de recursos para administrar estos en un clúster típico. Se convirtió en subproyecto de Hadoop en 2012 por lo que se le conoce también como MapReduce 2.0. Esto debido a que Apache Hadoop MapReduce ha sido re-arquitecturizado desde el principio hacia Apache Hadoop YARN.

YARN permite a múltiples aplicaciones ejecutarse simultáneamente en el mismo clúster compartido y permite a las aplicaciones negociar recursos basados en la necesidad. Por lo anterior, la asignación y administración de recursos es central a YARN (Fasale & Kumar, 2015). YARN bifurca la funcionalidad de administrador de recursos y administrador de tareas hacia diferentes demonios. El plan es obtener un administrador global de recursos (RM) y un Maestro de Aplicaciones (AM) por aplicación.

3.4.10. Apache Mesos

Mesos es un administrador de clúster que mejora la utilización de los recursos mediante la compartición dinámica de dichos recursos a través de diferentes entornos. Fue desarrollado por la Universidad de California, Berkeley en 2009 y está en producción en muchas compañías, como Twitter y Airbnb. Mesos comparte la capacidad disponible de los nodos entre las tareas de diferente naturaleza. Mesos puede ser visto como un sistema operativo para el centro de datos ya que provee una vista unificada de los recursos en todos los nodos y sin problemas para acceder a tales recursos de la misma forma que el kernel de un solo ordenador haría (Kakadia, 2015).

Mesos provee un núcleo para construir aplicaciones en centros de datos y su principal componente es un planificador de dos fases. La API de Mesos permite expresar una amplia gama de aplicaciones sin llevar información específica del dominio hacia el núcleo de Mesos. Manteniéndose enfocado en el núcleo, Mesos evita problemas que se ven con planificadores monolíticos.

3.4.11. Machine Learning en Big Data

Se ha comentado a lo largo de este capítulo sobre la gran velocidad a la cual se generan enormes volúmenes de datos, los cuáles requieren ser analizados para dar pronta respuesta a las necesidades actuales y satisfacer las metas a corto plazo. Las técnicas de ML para BD ofrecen la posibilidad de realizar diversos tipos de análisis como BD Analítico, BD Predictivo o BD Prescriptivo, todo esto en función del objetivo de la investigación o las

necesidades de la empresa. Lo anterior se ejemplifica en la Figura 3.16, como se puede apreciar a medida que se incrementa la dificultad del análisis, es mayor el valor que se obtiene de la información.

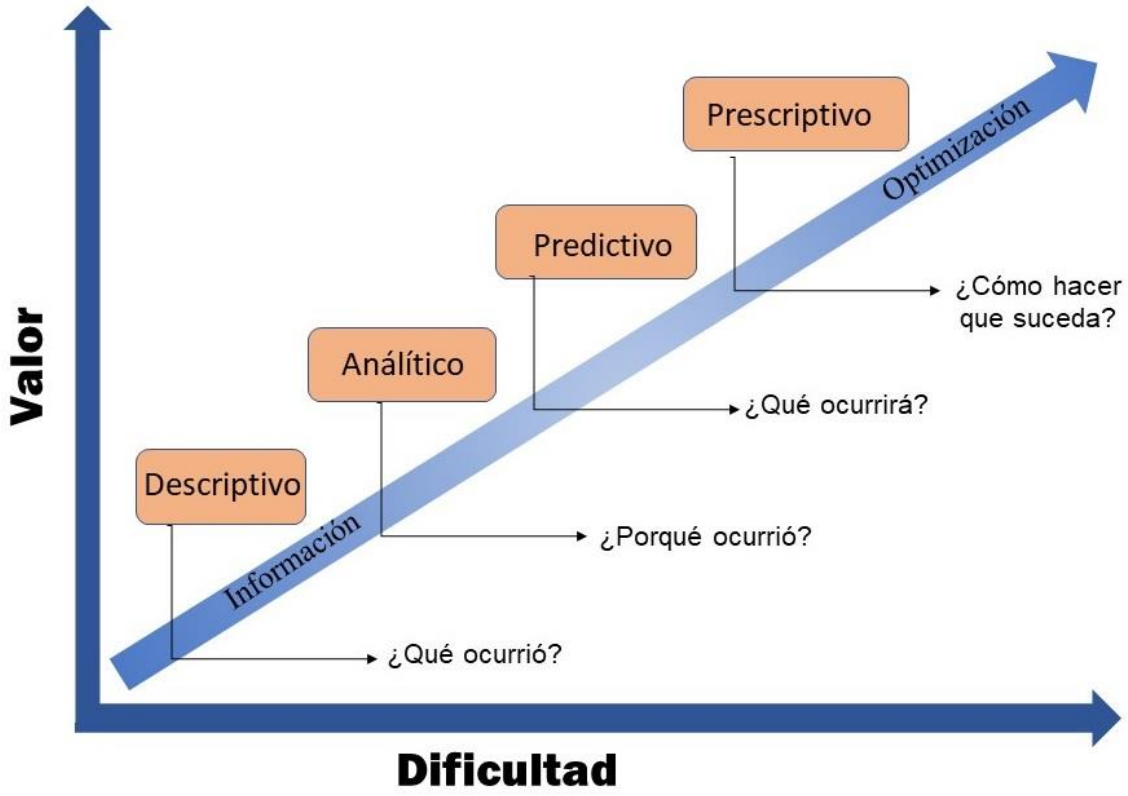

Figura 3.16 Progresión de los tipos de análisis en función del valor y dificultad

Ahora bien, como se ha expuesto anteriormente, existen demasiadas soluciones que dan respuesta a estas necesidad y, muchas de las veces el escoger alguna solución en particular puede ser una tarea compleja. Uno de los primeros estudios llevados a cabo donde se abordan estas necesidades es el llevado a cabo por Landset, Khoshgoftaar, Richter, & Hasanin (2015) en el cual se exploran una serie de soluciones tecnológicas, todas alrededor del ecosistema Hadoop para realizar ML en BD.

Mientras que el trabajo propuesto por Elshawi et al. (2018) expande el anterior estudio a través de un enfoque más amplio abarcando no sólo el ecosistema Hadoop sino otros ecosistemas existentes y las actuales alternativas para la nube. En este apartado se exploran algunas librerías empleadas para desplegar ML en BD, sin dejar de hacer mención de otras como: AzureML (Team, 2016) o el ya referido Pig (Gates et al., 2009).

3.4.12. Spark MLlib

El procesamiento de datos en BD es significativamente distinto, y en diversos aspectos, como se ha expuesto anteriormente. El modelo de cómputo distribuido vuelve un tanto más complejo el ejecutar ML al dividir el procesamiento en diversos nodos. MLlib ha sido creado nativamente para realizar ML en Spark (Meng et al., 2015), por lo que se acopla bien a esta plataforma al estar integrado dentro de su arquitectura. MLlib da soporte a un amplio conjunto de clasificadores como algoritmos de clasificación, regresión y agrupamiento (clusters). Se da soporta también a métodos para la extracción, transformación y selección de características (features), muy útiles en el procesamiento de información tipo texto.

De igual manera, MLlib ofrece una serie de evaluadores (múltiple, binario, etc.) para medir el desempeño del MLM. De manera general, el modelo de trabajo en MLlib es a través de *pipelines* (procesos) las cuáles norman u orquestan los diversos componentes que se integran al modelo. Actualmente MLlib ofrece una API para su desarrollo en Scala, Java, Python o R, siendo Scala donde se encuentra el mayor soporte a las librerías al ser este el lenguaje en que está construido Spark. Algunas de las desventajas de MLLib son:

- No se tienen librerías para todos los clasificadores existentes, pero cada cierta versión se agregan nuevos clasificadores a la lista.
- Los métodos empleados en los clasificadores no dan soporte a todos los parámetros y atributos que se encuentran en otras herramientas como Scikit-Learn.
- Como se ha mencionado, está integrado en Spark, lo que puede ser complejo si se está trabajando en un ambiente multiplataforma.

3.4.13. Spark ML

El paquete MLLib presentado en el apartado anterior opera estrictamente sobre conjuntos de datos distribuidos resilientes (RDDs), sin embargo Spark cuenta con un segundo paquete el cual opera estrictamente con *DataFrames* (Drabas & Lee, 2017) conocido como ML o Spark ML. De acuerdo a los autores citados anteriormente, esta es la principal librería oficial para realizar ML en PySpark (Python para Spark), lo cual también es reforzado por Deshpande & Kumar (2018). Este paquete provee tres clases abstractas en lo más alto de su interfaz: *Transformer*, *Estimator* y *Pipeline*, los cuáles se describen a continuación:

- ***Transformer:*** es una clase que transforma los datos mediante la anexión de una nueva columna al *DataFrame* empleado.

- ***Estimator:*** pueden ser vistos como unos modelos estadísticos que necesitan ser estimados para generar predicciones o clasificar las observaciones.
- ***Pipeline:*** es un concepto que maneja el proceso de estimación-transformación punto a punto que toma ciertos datos en un estado puro (*DataFrame*), realiza ciertos acciones sobre los datos (*Transformer*) y al final estima un modelo estadístico (*Estimator*).

Como se indica en (Duvvuri & Singhal, 2016) la diferencia fundamental entre Spark MLLib y Spark ML es que MLlib funciona sobre RDDs, mientras que el paquete ML funciona sobre DataFrames y Pipelines de Spark ML. Actualmente, ambos paquetes son compatibles con Spark, pero los autores recomiendan usar el paquete spark.ml.

3.4.14. Mahout

Mahout es un proyecto de la Fundación Apache que permite analizar mediante ML grandes conjuntos de datos, una de sus ventajas es que puede realizar dicho análisis tras haberlo descompuesto en diversas tareas paralelas. Mahout se inicia en 2008 como un subproyecto del proyecto Lucene, el cual proporciona implementaciones avanzadas de búsqueda, extracción de texto y técnicas de recuperación de información. Algunas partes del trabajo de Lucene que cayeron más en áreas de ML se convirtieron en su propio subproyecto. Poco después, Mahout absorbió el proyecto de filtrado colaborativo de código abierto Taste (Owen, Anil, Dunning, & Friedman, 2011).

Aunque Mahout es un proyecto abierto a implementaciones de todo tipo de técnicas de ML, en la práctica es un proyecto que se centra en tres áreas clave: motores de recomendación (filtrado colaborativo), agrupamiento (clustering) y clasificación. Por lo anterior, Mahout ha sido empleado en diversos estudios que tratan el manejo y análisis de información a gran escala, entre los que pueden mencionarse: evaluación del rendimiento y calidad de las medidas de similitud (Bagchi, 2015) o el análisis de datos provenientes de Twitter (Cunha, Silva, & Antunes, 2015).

3.4.15. Tensorflow

Tensorflow es una librería de código abierto creada por Google, que fue creada originalmente para realizar tareas con cálculos numéricos complejos. Su principal uso es en tareas de ML y redes neuronales profundas. Dado a que ha sido creado en C/C++, Tensorflow puede ejecutarse más rápido que el código de Python. Por otra parte, ofrece una API en C++ y Python, siendo esta la más completa y de uso más general. Tensorflow es más rápido de compilar tareas de aprendizaje profundo comparado con librerías similares, además de poder procesarse en CPUs, GPUs y clusters (Abadi et al., 2016).

El modo de computo está definido como un grafo de flujo de datos el cual está definido por:

- Nodos: Realizan las operaciones matemáticas
- Aristas: Son matrices multidimensionales (Tensores)

Tensorflow tiene una arquitectura flexible que permite desplegar tareas de cómputo en uno o más CPUs o GPUs, así como en un ordenador común, un servidor o hasta en un dispositivo móvil, como se indica en la Figura 3.17, todo lo anterior mediante el uso de una simple API. De tal forma que no se necesita un hardware especial para poder escalar las aplicaciones de ML que sean desarrolladas. Tensorflow está comúnmente asociado a estudios avanzados que requieren la aplicación de redes neuronales convolucionales (CNN) para el reconocimiento de imágenes. No obstante, se están empezando a publicar los primeros trabajos en el área de análisis de sentimientos empleando CNN (Kim, 2014) mediante Tensorflow como el llevado a cabo por Yoo, Song, & Jeong (2018) en el que se analizan datos de Twitter para realizar la tarea mencionada.

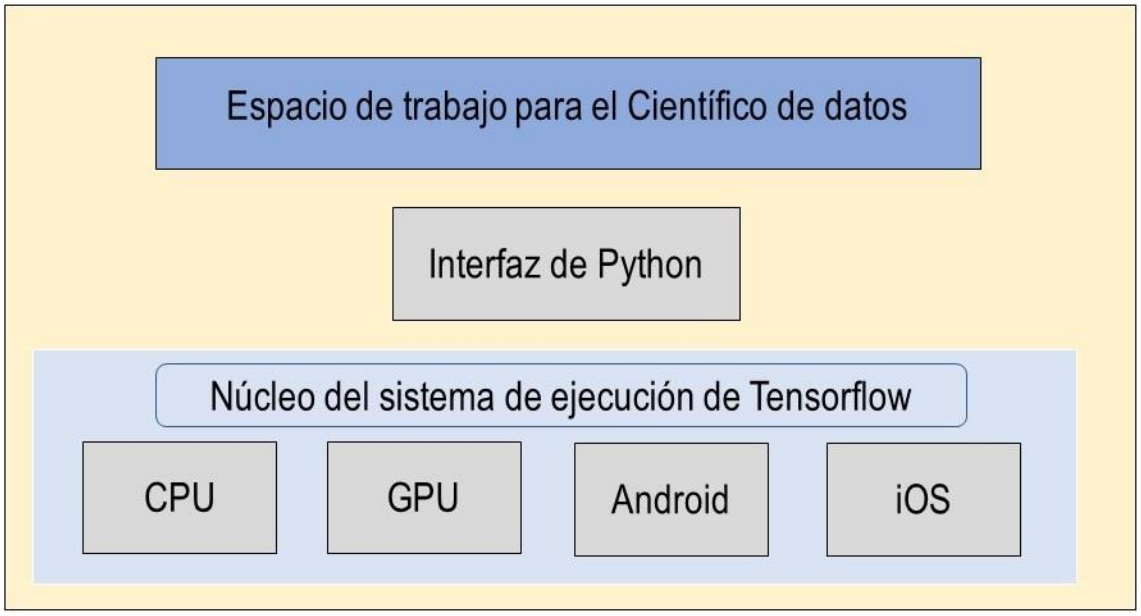

Figura 3.17 Esquema de la arquitectura de Tensorflow

3.4.16. Visualización de Datos

Dice una frase muy conocida que “una imagen vale más que mil palabras” y en el mundo de BD y el ML esto no es la excepción. Y es que todo este universo de información generada de la que se ha hablado bastante a lo largo de esta tesis debe poder ser representada de alguna manera (Cybulski, Keller, Nguyen, & Saundage, 2015). Es decir que, aunque anteriormente se ha expuesto que existen una serie de métricas para poder

medir los resultados y comprobar la eficacia del modelo construido, lo cierto es que esto es efectivamente útil, pero más para personas especializadas como investigadores, científicos o ingenieros. En la Figura 3.18 se presentan algunos ejemplos de visualización de la información, entre estos: nube de palabras, gráfica de barras, mapa de calor y gráfica de pastel.

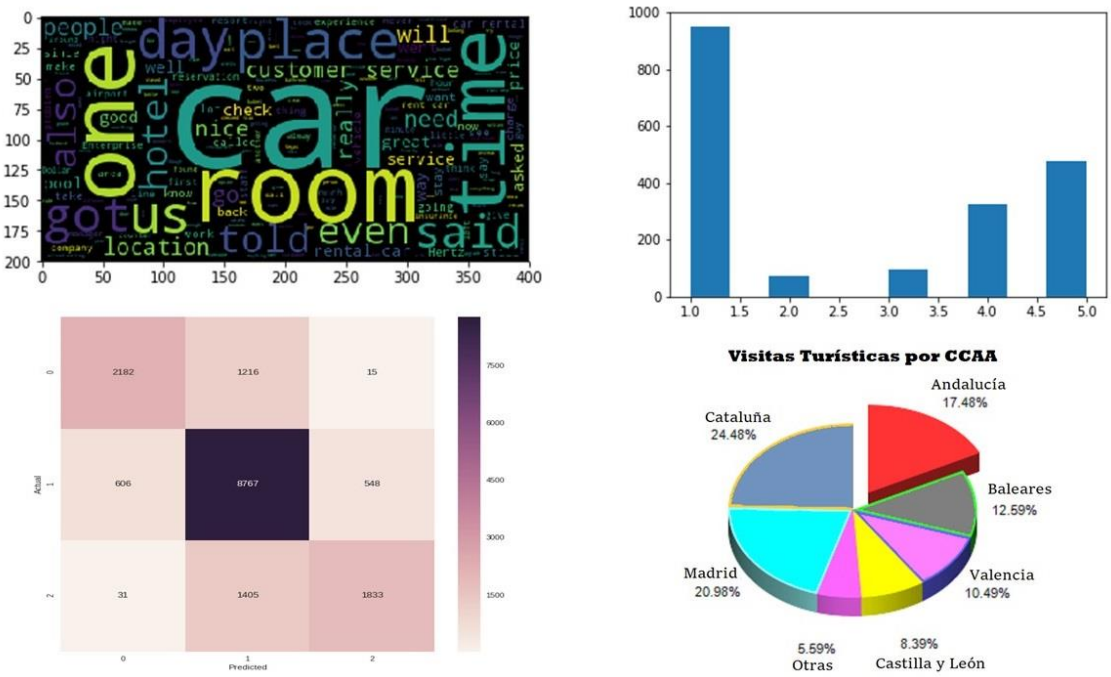

Figura 3.18 Ejemplos de visualización de la información

Sin embargo, una de las premisas de la ciencia en general, es que esta debe ser accesible al menos, para la mayoría de la población. Por lo que esta situación trasladada al área que compete a BD debe establecerse como todos aquellos métodos y procedimientos que son capaces de visualizar grandes volúmenes de información a un público selecto (Evelson & Yuhanna, 2012). De tal manera que esto permite visualizar GVDNE de una forma intuitiva y sencilla a la vez. De nada sirve contar con el mejor modelo o haber montado una gran arquitectura de BD y ML si los resultados no se pueden transmitir a gente no experta en el área, y esto es posible gracias a la visualización.

3.4.17. Técnicas de visualización

Existen diversos tipos de visualizaciones en BD, a saber: de barras, de líneas, nube de palabras, *boxplot*, etc. pero lo importante es saber a quién se le va a presentar la información y cuál es la información que se considera útil. En el área de BD y ML esto puede realizarse a través de varias técnicas, desde herramientas de Business Intelligence

como QLik, Tableau y SAS (L Zhang et al., 2012), herramientas de código abierto como Apache Zeppelin, frameworks de desarrollo como Python, R o librerías como Matplotlib, Plotly, etc. En el trabajo desarrollado por Piazza & Davcheva (2015) se explora un ejemplo práctico de cómo, mediante la aplicación de técnicas de visualización se pueden obtener resultados en el contenido *oculto* de la información. Al ser un tema que recae más en el área técnica, la opción seleccionada para realizar la visualización se explora en los capítulos 4 y 5.

3.5. Recuperación de Información

3.5.1. Introducción

También conocido como Búsqueda y Recuperación de Información (Information Retrieval en inglés), según Manning, Raghavan, & Schutze (2008) es "encontrar material de naturaleza no estructurada que satisface una necesidad de información de grandes colecciones". Esta definición puede ser un tanto general ya que no contempla (por el momento) el uso de medios digitales, refiriéndose a una actividad que pudo incluso llevarse a cabo antes de la era de los ordenadores, como es el ejemplo de abogados o bibliotecarios.

En la era digital, este término es ahora ampliamente empleado para una gran categoría de actividades relacionadas con buscar alguna información en la web (o en repositorios digitales) y su obtención para fines específicos. Como ejemplos de lo anterior pueden mencionarse:

- Buscar un correo particular dentro de un cliente de correo.
- Rastrear un pedido proveniente desde otro continente a través del proveedor de un servicio de paquetería internacional.
- Investigar un término particular en un motor de búsqueda por Internet.
- Emplear un dispositivo móvil para buscar un contacto dentro de una determinada red social.

Como se aprecia, el término referido contempla una amplia serie de actividades. El realizar la búsqueda de la información contempla una serie de diversos elementos tanto de software como de hardware para poder ser llevada a cabo. Los elementos de hardware podrían ir desde la memoria local de un ordenador o un dispositivo móvil hasta buscar en un clúster de ordenadores o en la nube. Mientras que el software comúnmente involucra consultas a bases de datos y algoritmos de ML para optimizar la búsqueda y encontrar lo

más cercano a lo que el usuario ha solicitado. En esta sección se presentan los aspectos relevantes al área de Búsqueda y Recuperación de Información que competen a este trabajo doctoral.

3.5.2. Frecuencia de términos

En el ámbito de la Recuperación de Información intervienen diversos factores, uno de estos es el asociado a los elementos que son recuperados dentro de un contexto de búsqueda. De tal manera que se considera "texto" o "colección" a el conjunto total de información recuperada y "documento" a cada una de las colecciones individuales que componen el texto. Es decir que, al recuperar un conjunto de información, por mencionar el caso particular críticas de usuarios, a cada una de esas críticas se le llama documento.

Ahora bien, se tiene la situación de que se requieren buscar los términos más importantes dentro de ese conjunto de documentos, por ejemplo, para buscar cual es la película más criticada o la canción más comentada por los usuarios. Es decir, no solamente se necesita buscar en la colección un término (feature) en particular, sino que además es necesario conocer cuáles son los más comunes o importantes. Uno de los enfoques empleados para esto es el de Frecuencia de términos y ponderación (Manning et al., 2008) en el cual se asigna a cada término dentro de un documento un peso para ese término, que depende del número de ocurrencias de ese término dentro del documento. Lo que se espera es calcular un puntaje entre un término de búsqueda *t* y un documento *d* basado en el valor de *t* en *d*.

De acuerdo con los autores citados, el enfoque más simple es que el peso sea igual al número de ocurrencias de *t* en *d*. Esto se denota Frecuencia de término o Term Frequency y se denota como $tf_{t,d}$, los subíndices denotan el término y el documento. Este modelo es conocido como el modelo Bolsa de Palabras (Bag of words o BOW por sus siglas en inglés) en el cual sólo se tiene información acerca del número de ocurrencias de cada término, por lo que no se toman en cuenta el orden en que aparecen los términos en un documento.

3.5.3. Frecuencia inversa de documentos

El enfoque presentado en el apartado anterior tiene el inconveniente de que todos los términos son igualmente importantes cuando se trata se evaluar la relevancia en una consulta. Por ejemplo, en un contexto en el que la colección de documentos versa sobre turismo, la palabra hotel va a aparecer casi en la totalidad de los documentos, haciendo que, bajo el enfoque anterior, tenga el mayor peso asociado. Esto puede ser un claro

inconveniente ya que es de esperar que este término sea altamente recuperado de tal colección.

Por lo anterior, un enfoque que puede resolver dicho inconveniente es emplear la frecuencia del documento df_t el cual está definido como el número de documentos en la colección que contienen un término *t* (Manning et al., 2008). Dicho indicador es más conveniente pues con el propósito de asignar una puntuación es preferible este que está basado en una estadística a nivel del documento que el tener solamente una cifra sobre el total de términos en una colección. Por lo que, según los autores referidos, se define la frecuencia inversa del documento (*idf*) de un término *t* de la siguiente manera:

$$idf_t = \log \frac{N}{df_t}$$

Por lo tanto, el valor idf de un término poco común es alto mientras que un término frecuente tiene un valor más bajo.

3.5.4. Term Frequency – Inverse Document Frequency

En este apartado se unen ahora las definiciones de frecuencia de término TF y frecuencia de documento inverso IDF (TF-IDF) para generar un peso compuesto para cada término en cada documento. Según Manning et al. (2008) la fórmula que define este peso está definido por:

$$\text{tf-idf}_{t,d} = \text{tf}_{t,d} \text{ x } \text{idf}_t$$

La fórmula anterior asigna a *t* un peso en *d* que es:

1. Más alto cuando *t* ocurra muchas veces dentro de un pequeño número de documentos.
2. Más bajo cuando el término ocurra menos veces en el documento o aparezca en demasiados documentos.
3. Mucho muy bajo cuando el término aparezca en casi el el 100% de los documentos.

3.5.5. Modelo de Espacio Vectorial

Una vez que se han obtenido los pesos TF-IDF para los términos a considerar dentro del modelo BOW se requiere ahora un modelo mediante el cual se puedan organizar un conjunto de operaciones para poder clasificar los documentos, agruparlos o evaluarlos, dicho modelo es conocido como el modelo de espacio vectorial (Manning et al., 2008).

Este modelo propone convertir cada término en un vector y proyectar cada uno hacia un espacio vectorial. En este espacio se mide la distancia que existe entre un determinado documento y los diversos vectores.

Lo que se busca es obtener una matriz en la cual en primer plano se haga una relación de los términos que se encontraron en los documentos, y que para cada intersección documento/término se almacenen los pesos de estos términos. En la Tabla 3.6 se presenta un ejemplo de esta representación.

	Hotel	Habitación	Limpieza	Comida
Documento 1	0	0.71	0.70	0.41
Documento 2	0.88	0.57	0.41	0.10
Documento 3	0.46	0	0.07	0.81

Tabla 3.6 Ejemplo del Modelo de Espacio Vectorial

3.5.6. Preparación de Datos

Los datos recolectados a través de diversas técnicas son conocidos como datos en bruto. Tales datos comúnmente presentan inconsistencias que deben ser consideradas y analizadas. De forma general, algunas de las inconsistencias que pueden presentarse son:

- Valores nulos.
- Valores en blanco (espacios).
- Existe información que proviene de distintos idiomas.
- Algunos valores en los datos se encuentran muy por encima o muy por debajo del rango en que está el resto.
- Cuando se trata de valores numéricos, hay diferencias en el formato empleado: se puede emplear punto o coma para los decimales, diferencias en los signos monetarios empleados, etc.
- Existen entradas del tipo *SQL injection.*
- Los archivos recuperados no están bien construidos: pueden estar en formato XML o JSON pero no cumplen con el estándar al querer ser leídos.
- Se encuentran inconsistencias en la codificación del fichero (ANSI, UTF-8, etc.) por provenir de un sistema operativo distinto al empleado.
- Si son datos recuperados de formularios Web, la información aún presenta etiquetas de HTML que no han sido removidas.
- Los datos de texto como opiniones o comentarios presentan deformaciones del lenguaje, como en: “lol, mi gf and i went to the store 2 buy cand1es 4 u” o símbolos, como en: “yo ♥ la ♪♫ que me pone ☺”.

- Error de conversión entre sistemas de medición: si se transforman erróneamente datos de pulgadas a metros o kilos a libras.

Por lo que todo lo anterior y demás factores pueden producir o generar datos sucios.

3.5.7. ¿Por qué se ensucian los datos?

De acuerdo a Tessari (2018), los principales factores que generan datos sucios son:

1. **Error humano:** es la generación de datos sucios por la introducción manual en hojas de cálculo o error al capturar.
2. **Sistemas diferentes:** Se tiene datos que se encuentran en sistemas CRM (Customer Relationship Management), ERP (Enterprise Resource Planning), marketing digital, datawarehouse, sistemas de logística, etc. Por lo que los datos terminan cumpliendo con diferentes requisitos, algunos son datos estructurados, otros relacionales, otros de grafos, hay diferencias en los nombres de los campos, etc.
3. **Cambios en los requisitos:** Los datos se cambian por decisiones del director de datos de la empresa (CDO), las organizaciones cambian, algunos campos se quedan obsoletos, por lo que es posible que los analistas no estén al tanto de cambiar los datos hacia una nueva plataforma tecnológica.

3.5.8. Problemas y soluciones de la preparación de datos

Como apunta Tessari (2018), los siguientes son los problemas encontrados al momento de preparar los datos y sus posibles soluciones:

Problema 1: Los procesos son poco flexibles y tardan mucho tiempo en dar respuesta, generando como consecuencia que no sigan el ritmo de la demanda debido a que los datos son “viejos” y ya no non útiles o a que los usuarios hacen sus análisis con múltiples versiones de datos similares, los cuáles generan confusión y se almacenan datos que son inútiles o inconsistentes.

Soluciones: Desarrollar procesos ágiles con las herramientas correctas para soportarlos. Compartir todo el proceso de preparación de datos y democratizar el acceso a los datos. Que la generación de los datos en la organización se vea de una manera holística donde todos los departamentos se relacionen.

Problema 2: La preparación de datos requiere un amplio conocimiento de los datos de la organización. Este elemento de descubrimiento es crucial, se debe conocer el origen de los datos. Los altos cargos de las organizaciones comúnmente se realizan las siguientes

preguntas respecto a los datos: ¿Qué datos hay disponibles? ¿Dónde están ubicados? ¿Cómo están definidos? Es decir, estos elementos no tienen una visión completa de los datos finales, desconocen la granularidad en ellos, por lo que ralentizan el proceso.

Solución: Establecer estándares para las definiciones de los datos en toda la empresa.

Problema 3: El concepto de datos limpios es subjetivo. Los departamentos tienen diferentes visiones sobre qué son los datos limpios. El área encargada de inteligencia de negocios puede estar más interesada en cómo se podrán almacenar los datos.

Solución: Poner el poder de los datos en manos de los expertos. Que un solo departamento especializado en gestión de los datos se concentre en la limpieza de estos.

Problema 4: Hay una realidad oculta en los almacenes de preparación de datos. La gente extrae los datos de forma independiente de alguna aplicación de ofimática y como resultado los datos tienen una nueva estructura que satisface un requisito específico muy *adhoc* a esa persona o departamento. Lo anterior conlleva a que una organización no conoce los datos reales con que se cuenta.

Soluciones: Debe crearse un diccionario de datos centralizado. Diseñar un proceso de preparación de datos coherente y colaborativo. Debe ser una tarea compartida entre el departamento de TI y de negocios en el que realicen la preparación de datos conjuntamente. En estas situaciones se necesita la experiencia de los *stakeholders* (especialistas), quienes pueden ver un informe y ser capaces de reconocer datos anómalos o extraños, situación que la gente del área de datos es incapaz de realizar.

3.5.9. Etapas de la Preparación de Datos

El dato puede ser comparado en cierta forma con el petróleo: estando en bruto sus usos son limitados; mediante el refinamiento se pueden obtener derivados del petróleo, mientras que a través de un proceso similar el dato se vuelve útil (Forbes, 2017), por lo que es de suma importancia preparar y refinar los datos. Según lo propuesto por Habib et al. (2016) las operaciones involucradas en el preprocesamiento tienen como propósito:

- **Reducción de ruido:** Los datos provenientes de redes sociales o sensores en el Internet de las cosas introducen cantidades masivas de ruido en la información.
- **Detectar valores atípicos:** La presencia de valores atípicos (*outliers*) degrada la calidad de los patrones del conocimiento, según se observa en la Figura 3.19.
- **Eliminar anomalías:** La presencia de valores irregulares o inusuales impacta en la calidad del conocimiento que se llegue a tener de los datos.

- **Extraer características:** Los métodos de extracción de características se emplean para separar los datos útiles de los datos crudos.
- **Fusionar flujos de datos de múltiples fuentes de datos:** Se requiere de operaciones de fusión de datos que sean capaces de integrar datos provenientes de diversas fuentes.
- **Crear conjuntos de datos uniformes:** Los datos se presentan en forma no estructurada, por lo que deben establecer los métodos que contribuyan a convertir estos a una forma estructurada para un mejor análisis
- **Dimensiones reductoras:** Los datos que se encuentran en GVDNE contienen de miles a millones de dimensiones que pueden ser muy difíciles de analizar, por lo que se tienen que aplicar métodos que permiten reducir las dimensiones para producir grandes conjuntos de datos que puedan ser analizados.
- **Manejo de valores perdidos:** Se deben generar criterios que permitan establecer qué hacer cuando se encuentran valores nulos como, por ejemplo: sustituir ese valor con la media, la mediana o la moda del resto de valores en esa columna si se tratase de un valor numérico. Otra solución podría ser el eliminar todas las columnas con valores nulos y realizar un nuevo análisis con el dataset resultante de datos completos.

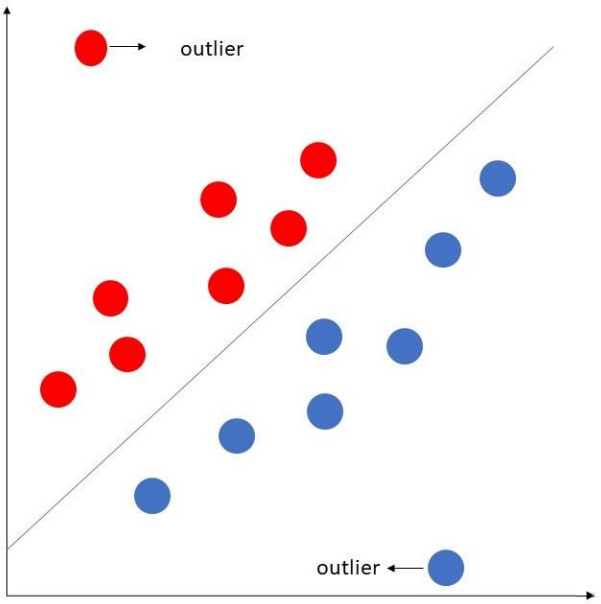

Figura 3.19 Ejemplos de los valores atípicos (outliers)

Por su parte, Lara, Lizcano, Martínez, & Pazos (2014) señalan que no obstante es complejo establecer una lista de tareas de preparación de datos, las más comunes son las siguientes:

1. **Recopilación e integración de datos:** Su objetivo es recolectar datos de diferentes fuentes, además de representar, codificar e integrar los datos de diferentes tablas para homogeneizar la información (Detours, Dumont, Bersini, & Maenhaut, 2003).
2. **Limpieza de datos:** Resuelve los conflictos en los datos, remueve los valores inválidos y, busca y resuelve problemas de ruido, valores perdidos, etc (Kim, Choi, Hong, Kim, & Lee, 2003; Pyle, 1999).
3. **Transformación de datos:** Transforma o consolida los datos para la extracción de información (Lin, 2002).
4. **Reducción de datos:** Selecciona datos relevantes para la minería de datos. Las formas que se tienen para realizar esta tarea son diversas, entre las que se incluyen: selección de features (Liu & Motoda, 1998), selección de instancias (Reinartz, 2002) o discretización (Liu, Hussain, Tan, & Dash, 2002).

3.6. Modelado de datos

3.6.1. Introducción

Desde la creación de los primeros sistemas relacionales de bases de datos se establecieron los procedimientos mediante los cuales se pudiera hacer un abstracción de los datos almacenados para que reflejaran elementos como: las relaciones entre las diversas entidades y la estructura de los datos desde un punto de vista lógico. Uno de los primeros modelos propuestos fue el modelo entidad relación del cual se habla en el siguiente apartado. El modelado de datos tradicional fue establecido en un momento de la informática en qué los datos eran todos estructurados, es decir había unanimidad en cuanto a que todos estaban organizados en tablas y tenían campos, índices o llaves y tipos de datos en común.

Pero en la época actual, en la que ahora lo que se impone son los datos no estructurados en gran volumen, es más complejo hablar de cómo modelar los datos con qué se cuentan. Uno de los motivos principales es que los datos no son propios de la empresa u organización, sino que han sido obtenidos mediante técnicas de *Web Scraping* o adquiridos a través de terceros que son empresas que realizan estás técnicas de recolección de información. Por lo que es necesario crear nuevos modelos que permitan organizar los volúmenes ingentes de datos provenientes de diversas fuentes para ilustrar las relaciones entre estos y entender mejor su compleja estructura.

De tal manera que en el estado actual de la ciencia de datos están surgiendo nuevos modelos para el mundo No SQL, es decir, aquellos sistemas de datos no estructurados que realizan la gestión de los datos no a través del estándar SQL. Uno de estos ejemplos es el propuesto por Atzeni, Bugiotti, Cabibbo, & Torlone (2016) en el que se propone una metodología de diseño de bases datos basada en el modelo abstracto NoSQL (NoAM). En este trabajo de tesis doctoral se siguen las mejores prácticas empleadas en el modelado de datos tradicional ante la ausencia actual de un estándar *de facto* para el modelado de GVDNE.

3.6.2. Modelo Entidad Relación

El modelo Entidad-Relación fue desarrollado para el diseño de bases de datos por Chen (1976). Este modelo ve los dominios de negocio en términos de entidades que tienen atributos y participan en relaciones (Halpin & Morgan, 2008). Por ejemplo, el hecho de que un alumno tiene un número de matrícula es modelado por el atributo NoMatricula del tipo de entidad Alumno, donde el hecho de que un alumno estudie en un grupo de una universidad es modelado como una relación entre ellos. Este enfoque es un tanto intuitivo y aunque el Lenguaje unificado de modelado (UML) es una alternativa idónea para modelar la estructura de una base de datos, lo cierto es que el modelo Entidad-Relación sigue siendo el modelo de datos más popular para aplicaciones de bases de datos.

El modelo propuesto por Chen no permitía expresar otras características que se fueron encontrando posteriormente para el diseño de bases de datos, como carecer la habilidad de establecer una multiplicidad mínima para los roles dados. Por lo que subsecuentes mejoras al modelo fueron planteadas llegando al punto de no contar con un estándar al existir tantas versiones del modelo. Un modelo que se planteó como mejora al modelo de Chen es el propuesto por Barker (1990), que fue posteriormente adoptado por Oracle para sus herramientas CASE (*Computer Aided Software Engineering*, Ingeniería de Software Asistida por Computadora).

El modelo de Barker está basado en atributos, lo cual facilita el diseño lógico que a su vez permite diagramas compactos que representan directamente la implementación de las estructuras de datos, por ejemplo, las tablas (Halpin & Morgan, 2008). De acuerdo al ejemplo presentado anteriormente de la relación alumno – grupo una representación de esto se ilustra en la Figura 3.20.

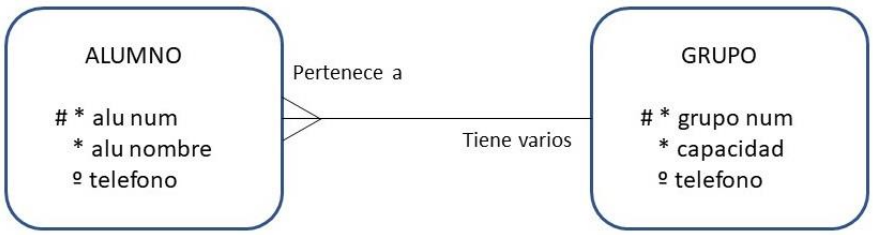

Figura 3.20 Modelo Entidad-Relación de Barker

El diseño de bases de datos relacionales es una materia comúnmente impartida en carreras del área de Informática. Dado que esta tesis doctoral no contempla el estudio de bases de datos como tal, no se ahonda en el tema salvo para justificar la especificación empleada en el Capítulo 4, particularmente en el apartado correspondiente a estructura y modelado, sin embargo las bases de esta especificación se han tomado de (Silberschatz, Korth, & Sudarshan, 2006), el cual se puede referir para una mejor comprensión de todo el proceso de diseño de bases de datos relacionales.

Una tabla tiene dos elementos: nombre de la tabla y atributos. Los atributos de la tabla denotan precisamente características inherentes a una 'entidad', 'cosa' o 'sujeto'. Al atributo se le conoce también como columna o campo. Por ejemplo, un Alumno además del número de matrícula tiene atributos como su nombre, sus apellidos, su edad, su sexo, el semestre actual, etc. El conjunto de estos atributos por cada alumno se denota como un registro o fila. Mientras que un Grupo universitario también tiene diversos atributos como: La carrera a la que pertenece, el aula designada, el turno (horario) y los alumnos que están suscritos a este grupo. Por otra parte, se pueden definir *relaciones* para estas tablas las cuáles establecen la cardinalidad entre dos tablas, siendo las más comunes: 1 a 1, 1 a Muchos y Muchos a Muchos.

En el diagrama de la Figura 3.21 se presenta gráficamente lo descrito hasta esta parte. Se aprecia en la parte superior de las tablas sus nombres y por debajo, los atributos. Una línea simple denota una relación 1 a 1, mientras que las 3 líneas al final de una línea denotan que ahí pueden relacionarse varios registros. De esta manera en la Figura 3.21 se aprecia que grupo tiene una relación con varios alumnos. Dicho de otra manera, un grupo contiene a varios alumnos; aunque podría establecerse una relación muchos a muchos, esto es sólo un ejemplo, recordando que en la vida real se dan diversas situaciones respecto a este o muchos otros ejemplos.

Por consiguiente, se define de esta manera la notación empleada en el siguiente capítulo para el apartado correspondiente al modelado de los datos. No se pretende afirmar que esta pudiera ser una especificación a considerar para sistemas NoSQL que, como ya se indicó, existen trabajos que buscan aportar a este hueco en la ciencia. Más bien lo que se

pretende, es indicar de la manera más sencilla pero entendible a la vez la relación entre volúmenes de datos que son complejos debido a su origen y volumen.

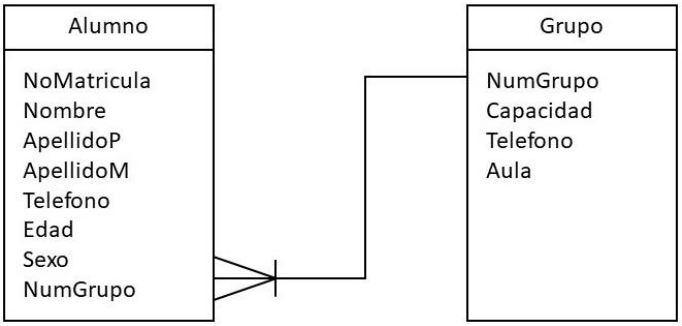

Figura 3.21 Ejemplo del Modelo Entidad-Relación

3.7. Sumario

Debido a que esta tesis doctoral aborda diferentes áreas de investigación que se relacionan con las técnicas de machine learning, big data y procesamiento de lenguaje natural, se incluye en este capítulo una visión transversal de las principales técnicas para poder aportar un perspectiva que atraviesa los diversos ámbitos que forman parte de esta tesis doctoral. El capítulo está estructurado de forma que se prioriza comenzar por la inteligencia artificial y el ML, aunando a estos hablar del PLN para tener una noción temprana de los elementos primordiales en la primera fase del framework. Otras fases preparatorias del *corpus* son también incluidas después de las mencionadas. Luego, se incluye lo que podría verse como la parte media de este capítulo siendo BD y otras técnicas como clusters, Cloud Computing, ML en BD y visualización de datos los temas primordiales.

Al iniciar el capítulo se ha hablado sobre las generalidades de la inteligencia artificial y el ML para conocer sobre ambos, pero también para establecer sus diferencias. Se abordaron temas relacionados con los clasificadores empleados en el modelo, así como generalidades propias del tema como: tipos de aprendizaje, fases de entrenamiento y prueba, las métricas de evaluación y, el sobreajuste. El área en sí es muy extensa, tan sólo hablar de las redes de neuronas artificiales lleva a muchas derivaciones, por lo que se entra poco a detalle en los clasificadores empleados al no ser este el elemento principal de la tesis, pero sí el más importante.

Posteriormente se abordaron temas como: procesamiento del lenguaje natural, recuperación de información, preparación de datos y estructura y modelado de datos, los cuáles se enmarcan en la primera fase del framework propuesto como se detalla en el Capítulo 4. El tema de BD es otra área que, aunque es más reciente que el ML, también abarca demasiadas temáticas, por lo que se abordan los áreas que se relacionan directamente con el framework, entre estas: los sistemas BD, los clúster BD, ML en BD y visualización de datos.

El estudio de datos de medios sociales es un área en constante evolución debido al potencial económico que grandes empresas tecnológicas están encontrando en el valor oculto de la información. Como se ha explorado a lo largo de este capítulo, para poder realizar investigación en esta área se deben integrar diversas disciplinas que acerquen los resultados al objetivo final. Se ha destacado la importancia que el ML tiene en este tipo de estudios ya que es en este dónde reside el núcleo o el centro del análisis de los datos.

Pero para poder llegar a esta etapa es necesario haber pasado por un conjunto de etapas previas que se encargan desde la recolección y obtención de los datos hasta la limpieza y preparación de los mismos (Hu, Wen, Chua, & Li, 2014). Se han expuesto en los diversos apartados una serie de técnicas para poder acercarse al problema de limpieza y preparación de datos. Se dice que se del total del tiempo de un proyecto de análisis de datos se invierte entre un 60% y un 70% del tiempo en preparar y limpiar los datos, por lo que esta se necesita aún mayor investigación en el área para que se disminuya este tiempo y se destine más a el análisis de los datos.

Otra de las áreas abordadas es la de PLN en la cual se expusieron una serie de técnicas relacionadas con la preparación del texto o corpus para su análisis. Algunas de estas técnicas se llevan a cabo en la etapa de carga o lectura de la información como: Stop-Words, o *tokeniz*ación. Mientras que en la parte de generar el modelo de espacio vectorial se aplican las técnicas de N-gramas y stemming. Cabe hacer mención que estos son meramente los fundamentos teóricos que subyace a la tecnología y técnicas abordadas, ya que, como se ve en los capítulos 4 y 5, el realizar estas tareas, requiere de implementaciones muy diferentes.

Capítulo 4. Solución Propuesta

En este capítulo se presentan las partes que integran el framework que da solución al problema establecido en el Capítulo I. Estos elementos provienen de diversas disciplinas las cuales son empleadas en conjunto, dando lugar a un framework novedoso en su área, como lo establecen las publicaciones realizadas que se refieren al final de esta tesis. En primera instancia, se introduce el planteamiento de partida que sirvió de motivación para proponer esta tesis doctoral, especificando las consideraciones que se tuvieron en el momento de proponer los elementos que integran la solución.

Partiendo de los recursos computacionales con los que se contempló contar durante la investigación fue como se agregaron o descartaron elementos que se consideraron no viables de incluir en el modelo propuesto; las razones se explican en los apartados correspondientes. Posteriormente se van presentando los elementos que conforman el framework, indicando su función dentro del mismo y describiendo la interacción que pueden llegar a tener con otros elementos de este.

4.1. Introducción

El objetivo principal de la investigación es definir un framework de trabajo que combine el uso de técnicas empleadas en el ML, el procesamiento del lenguaje natural y el análisis predictivo de GVDNE, que sea capaz de generar predicciones basadas en el análisis y la comparación de datos cualitativos contra datos cuantitativos. El framework tradicional de análisis de datos mediante el uso de un Sistema de Gestión de Bases de Datos (SGBD) se vuelve obsoleto ante la ingente cantidad de datos con que se cuenta actualmente ante el advenimiento de Internet.

No obstante, se han publicado diversos trabajos que involucran el uso de los SGBD para aplicar diversas técnicas para el análisis de datos, por mencionar uno de los más relevantes es el llevado a cabo por Fayyad, Piatetsky-Shapiro, & Smyth (1996) en el que ya se proponía un modelo que integraba diversas técnicas para la minería de datos. Los autores propusieron que el modelo puede ser empleado para el análisis de bases de datos y establecer relaciones entre estos, este modelo puede considerarse como un antecedente del propuesto en esta tesis.

La analítica de datos no es una rama nueva de la ciencia, tampoco puede pensarse que es algo que haya surgido con la revolución de BD o el crecimiento de la Web 2.0, de hecho, es un área que se ha servido de la estadística (una ciencia anterior a la era informática), las matemáticas y los primeros sistemas de gestión de datos digitales (década de 1960). Ertemel (2015) establece que lo que ahora cambia es que existe una gran variedad de datos para el análisis por lo que se requieren nuevas técnicas para el procesamiento y análisis de los mismos. Este autor también establece cómo se ha integrado el análisis de sentimientos en los modelos que implican BD y el provecho sustancial que los investigadores pueden obtener al combinar ambas técnicas.

Diversos autores han abordado la importancia que tiene el BD para realizar predicciones en el turismo. Uno de estos es el de Song & Liu (2017), una investigación teórica que habla en un primer plano sobre la naturaleza y necesidad de la unión entre el BD y el turismo, a la vez que se presenta un framework en el cual se propone la conjunción de una serie de métodos para llevar a cabo dicho procesamiento. Dicho modelo tiene ciertas similitudes con el framework aquí propuesto, con la salvedad que dicho modelo se especializa en la predicción del turismo empleando diversos métodos que pueden combinarse con técnicas de BD. En el trabajo referido se presentan una serie de etapas y requisitos que se deben alcanzar para lograr los objetivos predictivos establecidos.

De tal forma que igualmente a como se ha presentado en el Estado del Arte, en los trabajos anteriormente citados, no se encontró un modelo que aglutine las tres áreas de interés para el framework: BD, ML y PLN.

4.1.1. Framework para el Análisis Predictivo de Datos no Estructurados

El framework propuesto en esta tesis permite en primera instancia, la evaluación de un subconjunto de GVDNE que contiene datos concernientes a las apreciaciones hechas por los usuarios de diversos servicios relacionados al sector turístico, esto como mero caso de estudio (como se detalla más adelante), ya que como se ha comentado anteriormente, el conjunto de datos se ha establecido que puede ser multidominio. Dicho conjunto de datos contiene además una serie de puntuaciones numéricas que van del uno al cinco con el cual se evalúa cuantitativamente el servicio recibido. Esencialmente, el framework se compone de dos fases: en la primera se propone crear un MLM que sea capaz de obtener información oculta en los datos, además de evaluar los clasificadores más adecuados para el conjunto de datos disponible a la vez que se emplean técnicas de PLN; así como obtener resultados a partir de una infraestructura no BD para el análisis de datos que no reúnan el volumen de datos similares a los de GVDNE.

En la segunda fase se pretende aprovechar los algoritmos de ML que se emplearon en el MLM de la primera fase para hallar la relación entre opiniones y calificaciones para aplicarlos dentro del proceso de análisis de GVDNE. Una serie de técnicas predictivas son aplicadas en esta fase para que la salida demuestre que se obtuvieron resultados similares o incluso se optimizaron los hallados en la primera etapa. Los resultados obtenidos gracias a las técnicas empleadas en el framework permiten demostrar la validez de los métodos de análisis y procesamiento de la información (grande y pequeña).

El dominio del estudio de esta investigación (turismo) no constituye una limitante para poder extrapolarlo hacia otros dominios, siempre que los datos de entrenamiento cumplan con las características establecidas por el aprendizaje supervisado (Flath & Stein, 2018), es decir, que los datos de entrada estén asociados a etiquetas ya establecidas denominadas como categóricas. De igual manera, el framework propuesto se encuentra abierto a poder emplear cualquier técnica para comprobar la efectividad de sus resultados.

En la Figura 4.1 se presenta el conjunto de elementos que conforman el framework. En la parte inferior se distingue una fase inicial que es la preparación de datos la cual comprende desde la entrada de los datos, hasta obtener el MLM. En la parte superior se aprecia la fase que corresponde al tratamiento y análisis de los GDVNE, además que se presenta la propuesta de una arquitectura dinámica que pueda interactuar con las diversas soluciones de nube para BD.

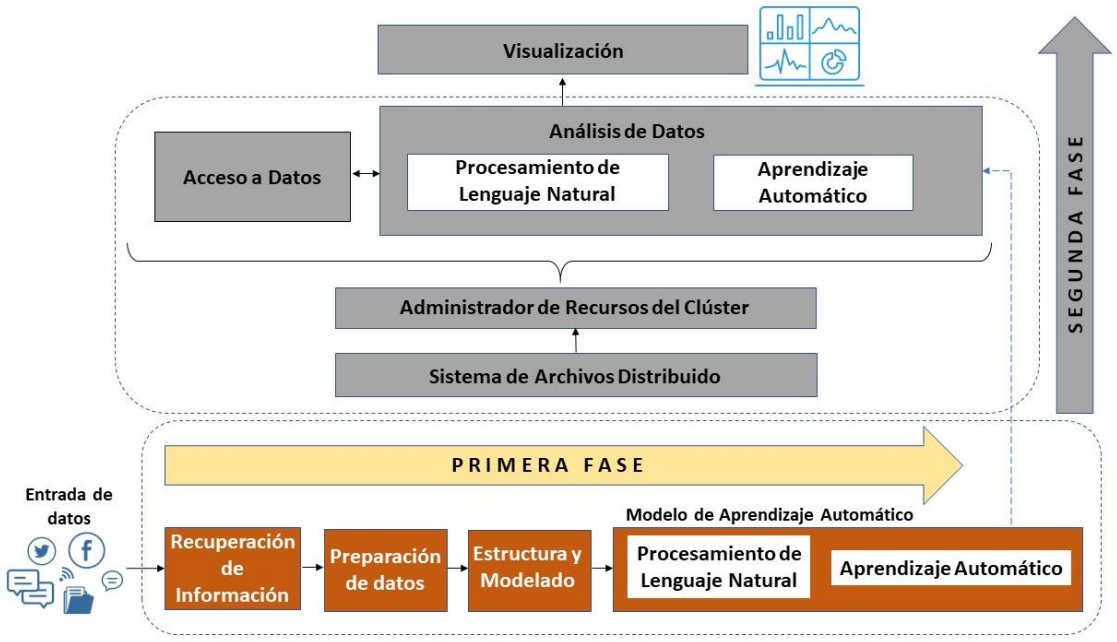

Figura 4.1 Fases y etapas que integran el framework para el análisis predictivo

Visto de una manera general, la aportación del framework es establecer una metodología que determine las bases para crear un MLM que pueda emplearse tanto en sistemas BD como fuera de estos. Lo anterior viene a constituir el núcleo principal del modelo, pero no se limita sólo a eso ya que a través de este modelo se puede llevar a cabo el análisis de información etiquetada, característica requerida en los datos analizados mediante técnicas de aprendizaje supervisado. Los resultados de esta investigación se emplean para encontrar patrones en los datos que den valor agregado a los mismos. A continuación, se presenta la descripción de las fases que integran el framework.

4.2. Elementos de la primera fase

En este apartado se detallan las etapas que integran la primera fase de la solución propuesta. Cabe recordar que esta fase está integrada por métodos y técnicas externas al entorno de BD el cual es propuesto en la segunda fase. Se explica cómo se llegó a obtener el MLM habiendo partido desde la obtención de datos.

4.2.1. Recuperación de Información

Aspectos Teóricos

Esta es la etapa inicial del framework la cual comprende llevar cabo las acciones necesarias para hacer la recolección de los datos empleados en la investigación. Esto se encuentra en función de la investigación que se lleve a cabo y de la infraestructura técnica instalada o que se pueda contratar dependiendo del alcance. Para obtener el máximo valor, las empresas recogen la mayor cantidad de información que les es posible: críticas de clientes, datos de sensores y otras operaciones relevantes para el estudio. El proceso de obtención de datos debe ser lo suficientemente óptimo para evitar la recolección de datos irrelevantes.

Las estrategias óptimas para la recolección de datos ayudan a disminuir el coste económico y facilitan la carga computacional y de almacenaje de los centros de datos (Habib et al., 2016). En la Figura 4.2 se muestran de forma emblemática algunas de las fuentes de datos de donde extraer contenido para BD y cómo el volumen de estos va aumentando en función del origen de los mismos. BD es una tendencia actual en el campo científico, tecnológico y empresarial, a través del macro estudio de los datos, se puede obtener información interesante y en ocasiones vital para equipos de investigación y la alta gerencia en las empresas.

De tal manera que, visto desde una perspectiva científica, pero sobre todo empresarial se menciona que los datos son el oro del siglo XXI, si esto se lleva al ámbito de la ciencia se puede constatar con certeza ya que los datos forman parte del acervo más valioso con que puede contar un investigador para realizar la labor investigativa. Las empresas líderes del sector como Google, Amazon, Facebook, YouTube y eBay son ejemplos de que a través de la integración de equipos de investigación dedicados al análisis de BD se pueden obtener resultados que son un factor determinante para el establecimiento de ventajas competitivas (Özköse et al., 2015).

Figura 4.2 El efecto multiplicador del dato

(Adaptado de Andersson, B.; Beals (2018))

El proceso de obtención de datos mediante la simple descarga de los mismos no debe verse como un proceso infructuoso. La iniciativa mundial Open Data (Thorsby, Stowers, Wolslegel, & Tumbuan, 2017) ha logrado que cada vez más gobiernos y agencias del orden público, den a conocer sus datos a través de portales de transparencia, por lo que sí existen recursos para realizar trabajos similares a BD. Por otro lado, empresas privadas como Amazon también han dispuesto grandes *datasets* (conjuntos de datos) para que puedan ser estudiados o analizados.

En lo que respecta al modelo propuesto, las técnicas de obtención de datos contribuyen a obtener por métodos diversos la mayor cantidad de información posible para alimentar los almacenes de BD. Dichas técnicas pueden ser preconstruidas por el equipo de investigación (por ejemplo, web scrapers) o bien adquirir el producto de un tercero. De acuerdo a Olmedilla, Martínez-Torres, & Toral (2016) la recolección de datos puede ser hecha empleando las APIs provistas por sitios de redes sociales como YouTube, Flickr o Amazon, también puede recurrirse a lenguajes de programación como Python, Java, Ruby o PHP. El trabajo mencionado es de particular relevancia debido a que se propone una metodología para obtener datos directamente de sitios web con contenido generado por el usuario.

Las actividades de Recuperación de Información deben asegurar que se ha recolectado la información necesaria para llevar a cabo el estudio, lo cual implica que no existan vacíos en la información obtenida, por ejemplo, alguna tabla o haber obtenido los datos de un proveedor adicional. Una medida para evitar lo anterior es recurrir a las variadas técnicas de la ingeniería de requisitos en las concernientes al acopio de la información. Aunque en un estudio que implique el tratamiento de GVDNE siempre puede regresarse a esta etapa desde alguna de las que le suceden, esto conlleva a un impacto en el tiempo de finalización del estudio. La salida de esta etapa alimenta de forma natural a la posterior, Preparación de datos, ya que se encarga de proveer los datos que son normalizados en esta última.

Aspectos de Implementación

Se ha hablado ya de que BD está presente en cualquier sector o actividad ya sea industrial o comercial que genere enormes cantidades de información. Para el enfoque de esta tesis se estableció inicialmente que el sector iba a ser comercial y dirigido al turismo, esto al ser una actividad que se encuentra en constante evolución y ha sido abordada en diversas publicaciones arbitradas y artículos científicos, pero no totalmente con el enfoque que se propone en este trabajo. El turismo abarca una gran gama de servicios, los cuáles no se mencionan en este espacio por estar fuera del ámbito de interés. Inicialmente se pensó en obtener datos de empresas dedicadas a la movilidad urbana, en concreto empresas como BlaBlaCar[9], pero después de varias conversaciones, se nos negó el acceso a tales datos.

Posteriormente se estableció como meta poder contactar tanto con personal de Booking.com y Foursquare, pero el resultado fue el mismo: no ceden el acceso de sus datos a terceros. Al estar revisando el estado del arte, se observó que un grupo de investigadores estuvo realizando una serie de investigaciones con datos tomados de la

[9] https://www.blablacar.es/

página de TripAdvisor [10]mediante la técnica de web scraping (Li, Ott, & Varadarajan, 2013). Debido a esto, se decidió investigar en este sitio el cómo poder acceder a sus conjuntos de datos a través de la API de TripAdvisor, pero para fines de investigación no se permite el uso de tal API.

En el proceso de la investigación por obtener datos significativos, se llegó a establecer contacto con la agencia de turismo de España, la cual tiene información pública sobre los desplazamientos tanto de los nacionales como los extranjeros en territorio español. Se examinó el contenido de estos datasets y finalmente se determinó que la información no contenía la cantidad ni el contenido necesario para poder realizar una investigación con datos de dimensiones similares a los GVDNE. Por ser de menor la relevancia que ocuparon durante este proceso, sólo se mencionan los siguientes sitios u organizaciones que también fueron consultados con el propósito de obtener datos: Enginuity[11], IAB Spain[12] y Str[13].

Continuando con la actividad de búsqueda de datasets que pudieran analizarse mediante entornos GVDNE, se encontraron los datasets Yelp. Yelp es una empresa norteamericana multinacional acuartelada en San Francisco, California, como tal desarrolla y soporta el sitio Yelp.com. En ambos, se publican y evalúan los servicios de gran cantidad de empresas que ofrecen sus servicios en gran número de ciudades de todo el mundo (Xu et al., 2016). Las mismas empresas buscan darse de alta en Yelp y estar posicionados en la cima de las búsquedas cuando los usuarios realizan una consulta general para conocer de nuevos lugares para el ocio y descanso.

Con fines de investigación y de análisis esta empresa publica semestralmente un grupo de datasets en idioma inglés que contienen información relacionada entre estos y que pertenecen a diversos subgéneros de actividades. Este fichero puede descargarse, previo registro en el enlace:

https://www.yelp.com/dataset/challenge[14]

A la vez Yelp convoca a un concurso denominado el reto Yelp en el que equipos de investigación tienen que demostrar los hallazgos que han encontrado en los datos mediante las técnicas que se hayan diseñado para lograrlo. De tal manera que en estos datasets podemos encontrar información tan variada como renta de autos, servicios religiosos o veterinarios. Pero en todas estas actividades se encuentra en gran medida información relacionada con turismo particularmente aquellos relacionados con

[10] https://www.tripadvisor.com/
[11] https://enginuity.freshdesk.com/
[12] https://iabspain.es/
[13] https://www.str.com/
[14] El enlace se visitó el 05/05/2016, por primera vez

hostelería y restauración, por lo que se decidió tomar este dataset para llevar a cabo el estudio. Otro de los factores que propiciaron el seleccionar este dataset fueron:

1) El dataset es de casi 2,5 GB y aunque no se le puede considerar lo suficientemente grande para llamarle BD, es de los mayores que se encontró en Internet para fines de esta investigación.
2) Se tiene un dataset de opiniones el cual contiene información que emplea el lenguaje natural de los usuarios para expresar quejas, sugerencias o recomendaciones entre otros. Esto aunado a que por cada registro se tiene una calificación o estrellas asociadas a la expresión empleada por los usuarios, se vio como un área de oportunidad para enfocar más el estudio dentro del ámbito de la ciencia informática.
3) Los datos están cifrados, es decir que se oculta la identidad de los usuarios mediante la asignación de claves aleatorias (hashing). No obstante, se cumple de forma parcial con la ley europea RGPD (Reglamento General de Protección de Datos o General Data Protection Regulation) debido a que esta empresa se encuentra asentada en San Francisco, CA, EE. UU.

El reto de analizar las reviews es uno que conlleva diversos aspectos a considerar. Por un lado, se tiene que un usuario puede hacer una extensa crítica sobre su experiencia mientras que otro usuario puede realizar una crítica más bien corta, ambos sobre el mismo negocio y servicio recibido. Por otro lado, no hay una relación directa entre el tamaño de la crítica y la calificación asignada, es decir, no se puede afirmar que a más información se le otorgue una mejor calificación o viceversa. Otro aspecto a considerar son los modificadores del texto, es decir se puede tener una crítica que casi hasta el final se consideraba como favorable, razón por la que se otorgaría la calificación más alta, pero al final una negación cambia el enfoque original que se tenía de la crítica.

Por otro lado, técnicas como web scraping son comúnmente usadas en entornos científicos para obtener información de diversas fuentes, con el inconveniente de poder llegar a saturar la red local o ser rechazado por servidores externos. Sin embargo, no deben desestimarse las fuentes públicas de información que ofrecen sus datasets a todo el público interesado. Mediante un adecuado estudio de la información disponible pueden aplicarse las herramientas de BD para realizar estudios que involucren información de diferentes fuentes, pero con características en común que lleguen a ofrecer hallazgos interesantes sobre una cierta entidad de estudio.

Dentro de las actividades de recuperación de información que a su vez emplean técnicas y herramientas de PLN se encuentra la tarea de: reconocimiento de entidades nombradas o extracción de entidades (Named Entity Recognition o NER por sus siglas en inglés). Esta tarea tiene como finalidad el reconocer entidades del texto que se está analizando

como: personas, organizaciones o medidas de tiempo, entre muchos otros. Esta actividad actualmente es muy estudiada en ámbitos científicos dado que se está expendiendo su aplicabilidad hacia esferas que no son solamente de usos comunes o comerciales, sino que se está empleando en dominios muy específicos como la medicina, la bioinformática, la química o la farmacéutica, sólo por mencionar algunos. Mientras que parte del discurso (Part Of Speech o POS por sus siglas en inglés) es otra tarea del PLN que reconoce y/o detecta las posiciones de los elementos de un texto para dar mayor o menor peso a los diversos componentes léxicos.

Sin embargo, estas tareas requieren a su vez de otras actividades que no están consideradas en esta tesis debido a que no forman parte del propósito inicial de la investigación, además de que no contribuyen de manera directa a comprobar ninguna de las hipótesis planteadas. No obstante, se reconoce que integrar estas tareas no sólo en esta tesis sino en cualquier estudio que tenga fines similares al presente, contribuye a un mejor y más completo análisis de información textual. Por lo que será importante la integración de NER y POS en estudios futuros que amplíen la presente investigación.

4.2.2. Preparación de datos

Aspectos Teóricos

En la etapa de obtención de datos las actividades a cargo tuvieron que ver con el acopio de GVDNE tomados estos de diversas fuentes, una vez que se cuenta con tales volúmenes de información estos deben ser ahora estudiados en la etapa de Preparación de datos, la cual es la más importante en un proyecto que involucra GVDNE, otras etapas asociadas son: el preprocesamiento de los datos y las operaciones de integración (Habib et al., 2016). Gran parte del tiempo involucrado actualmente por los equipos de ciencia de datos se invierte en la preparación de estos. Se estima que, del total de tiempo del proyecto, un 70% se invierte en la preparación de datos y un 30% en el análisis, esto puede denominarse como la “limpieza” de los datos.

Si se ha empleado la técnica de web scraping, durante el proceso de obtención se pudieron haber obtenido etiquetas HTML o de otro sistema de etiquetado además de información comercial y publicidad, toda esta información debe ser eliminada del dataset de estudio final. La extracción de características de la información es otra de las operaciones de preprocesamiento que debe ser llevada a cabo en esta etapa (Habib et al., 2016). Dentro de las actividades de esta etapa está la detección de valores anómalos o que no guardan una uniformidad respecto del resto de los datos. Por ejemplo, cuando se trabaja con valores numéricos estos normalmente deben estar dentro de un rango que

satisfaga las condiciones del estudio, pero comúnmente pueden encontrarse valores que sobresalen muy por encima o muy por debajo de los valores esperados.

Otro caso puede ser al tratar con valores de texto o cadenas, que estas sean nulas o incluso pudieran venir en diferentes idiomas con el consecuente cambio de los caracteres mayormente empleados. En cualquiera de los casos expuestos anteriormente, como en muchos otros casos similares, la etapa de preparación de datos debe comprender el empleo de diversas técnicas (mayormente computacionales) para dar algún tratamiento especial a eso registros, ya sea que se aparten del resto de la información o se incluyan en un caso especial. El no detectar y tratar con tales valores, lleva a una desviación significativa de los resultados del estudio.

La salida de esta etapa es el conjunto de la información en un estado que conserva ya una estructura homogénea y una serie de valores en los datos que se adecuan a los objetivos del estudio además de no contener valores que sobresalgan del resto. Esta etapa permite que la información sea leída por técnicas tanto de GVDNE como externas a este paradigma de forma natural, es decir, que no contenga errores de ningún tipo. Podría resumirse esta etapa mencionando que se "limpió" la información original y se entregó en dicho estado a la siguiente, la cual se encarga de analizar su estructura para generar un modelo esquematizado que represente lo más fehacientemente posible al conjunto global de datos.

Aspectos de Implementación

Para los datasets Yelp se comenzó por crear un método informático que pudiera recorrerlos, pero estos estaban mal construidos, es decir, toda la información estaba ahí pero no fueron creados aplicando cabalmente la notación de JSON, por lo que se creó un método adicional que reconstruyera los ficheros. Posteriormente se examinó el dataset en busca de valores perdidos y para remover posibles anomalías (Habib et al., 2016), respecto a esto se ha encontrado en versiones posteriores del fichero de opiniones, que ha sido generado con anomalías desde su origen.

Para este efecto, se nos informó por parte de Yelp que ellos no proporcionaban información alguna sobre las características de los datasets, que esto era parte del reto para los participantes, por lo que se tuvo que dar a la tarea de extraer las características a base de analizar sus contenidos. Los datasets son cinco: Negocio (Business), Usuario (User), Revisión o Crítica (Review), Consejos (Tip) y Registro (Check-in).

El dataset Registro tiene información del número de registros que se hicieron en un negocio, datos que son relevantes para los negocios y no se tomaron en cuenta para el

estudio. Otro dataset que no fue tomado en cuenta fue el de Usuario, no obstante, se recomienda explorar su contenido ya que puede revelar información relevante al presentar este los amigos con los que cuenta el usuario dentro de la red y podría analizarse si esta red de amigos influye en la credibilidad del usuario, por ejemplo. Se presenta un registro de este dataset como ejemplo:


```
{"yelping_since": "2004-10", "votes": {"funny": 1, "useful": 11, "cool": 5}, "review_count": 11, "name": "Ken", "user_id": "fHtTaujcyKvXglE33Z5yIw", "friends": ["18kPq7GPye-YQ3LyKyAZPw", "rpOyqD_893cqmDAtJLbdog", "i63u3SdbrLsP4FxiSKP0Zw", "uKgbjPhcrS2pi9hS39Az6A"], "fans": 2, "average_stars": 4.64, "type": "user", "compliments": {"cute": 2}, "elite": []}
```


Otro dataset que puede emplearse en otros ámbitos de estudio es el de Consejos debido al impacto que podría explorarse en la posible relación entre dichos consejos y la decisión posterior que ejercen los usuarios sobre sus hábitos o gustos en el consumo, este es el ejemplo de un registro:


```
{"user_id": "-6rEfobYjMxpUWLNxszaxQ", "text": "Don't waste your time.", "business_id": "cE27W9VPgO88Qxe4ol6y_g", "likes": 0, "date": "2013-04-18", "type": "tip"}
```


El dataset Negocio es empleado en el estudio pues contiene un dato que es fundamental: categorías. Este campo representa todas las etiquetas que los usuarios han asociado al negocio, es decir, que los mismos usuarios van indicando qué actividades comerciales tienen relación con dicho negocio. Por lo que las búsquedas por categoría deben realizarse con mucha precisión ya que puede obtenerse información cruzada, esto es, datos de otros negocios que no tengan mucha relación con lo que se está buscando, como se observa a continuación:


```
{"business_id": "zgy27FSnvwdINfk5cXBIyQ", "full_address": "520 North Bell Avenue Carnegie Carnegie, PA 15106", "hours": {"Monday": {"close": "00:00", "open": "00:00"}, "Tuesday": {"close": "00:00", "open": "00:00"}, "Friday": {"close": "00:00", "open": "00:00"}, "Wednesday": {"close": "00:00", "open": "00:00"}, "Thursday": {"close": "00:00", "open": "00:00"}, "Sunday": {"close": "00:00", "open": "00:00"}, "Saturday": {"close": "00:00", "open": "00:00"}}, "open": true, "categories": ["Hotels & Travel", "Event Planning & Services", "Hotels"], "city": "Carnegie", "review_count": 7, "name": "Extended Stay America - Pittsburgh - Carnegie", "neighborhoods": ["Carnegie"], "longitude": -80.088557, "state": "PA", "stars": 3.5, "latitude": 40.417419, "attributes": {"Accepts Credit Cards": true, "Wi-Fi": "free", "Price Range": 2}, "type": "business"}
```

El dataset de mayor importancia es el de Review ya que contiene la crítica o reseña que hacen los usuarios de los servicios recibidos. Esta información cualitativa se relaciona directamente con el dato cuantitativo de las estrellas: un puntaje de 1 a 5 que indica, de menor a mayor, el grado de satisfacción del cliente. En el ejemplo que sigue se aprecia, entre otros datos, que cada crítica viene asociada a un usuario y a un negocio:


```
{"votes": {"funny": 0, "useful": 5, "cool": 0}, "user_id":
"LWbYpcangjBMm4KPxZGOKg", "review_id": "6w6gMZ3iBLGcUM4RBIuifQ",
"stars": 5, "date": "2012-12-01", "text": "This place was DELICIOUS!!
My parents saw a recommendation to visit this place from Rick Sebak's
\"25 Things I Like About Pittsburgh\" and he's usually pretty accurate.",
"type": "review", "business_id": "mVHrayjG3uZ_RLHkLj-AMg"}
```


Todos los conjuntos de datos mencionados y explicados anteriormente han sido estudiados y analizados desde diferentes enfoques científicos y técnicos. La posibilidad de realizar estudios sobre estos datos es enorme al encontrarse relacionados los datasets y, de esta forma, explicar los fenómenos que acontecen en una plataforma de estas características.

4.2.3. Estructura y Modelado

Aspectos Teóricos

La etapa anterior se encargó de transformar la información original existente en volúmenes de datos en un estado integro o congruente para los objetivos del estudio, que además puedan ser leídos por diversas técnicas computacionales. Estos volúmenes de información son llevados ahora a la etapa en que se define la estructura que contienen estos y además establecer un modelo de datos que los represente.

En primera instancia si se trabaja con datos provenientes de una sola entidad, puede encontrarse que esta ha definido la estructura que guardan estos datos, por lo que la tarea está hecha. En caso contrario, se tiene que hacer una selección directa de una muestra de la información para identificar el tipo de información que se obtuvo, buscando entre otros elementos: posibles tipos de datos y la relación entre estos. Un factor que puede ayudar a esto es que comúnmente se conoce cuál es el propósito de estos datos, es decir, estos son obtenidos con un propósito anteriormente establecido por los fines de la investigación.

El proceso de estructura y modelado tiene grandes similitudes con el proceso similar del diseño de bases de datos relacionales. Es decir, el producto final al que se pretende llegar es a una estructura de tablas y campos para poder ver el conjunto global de datos desde una perspectiva de mayor organización. Cuando la información pertenece a la

propia organización, este modelo puede obtenerse más fácilmente, contrario al hecho de que se haya recabado la información a través de técnicas de web scraping en cuyo caso será necesario un mayor análisis de los datos obtenidos, para lo cual el conocer la plataforma o sitio de donde se recolectaron tales datos ayuda en el proceso de comprender el origen y naturaleza de estos.

El proceso de estructura y modelado para GVDNE puede volverse aún más complejo cuando se requiere conjuntar información de dos o más orígenes de datos. Un ejemplo práctico podría ser: se estudiará cuál es el efecto que tiene el clima en los compradores de cierto supermercado previo a la transmisión televisiva de un partido de futbol de la liga de España. En el ejemplo anterior se necesitarían datos de: meteorología, el supermercado y la televisora, dado que cada uno de los mencionados no conserva una relación directa con los otros dos, el proceso de análisis debe procurar encontrar las relaciones que existen entre tales orígenes de datos.

Habiendo definido una estructura al menos conceptual de los datos existentes, se procede a generar un modelo de los datos. Este modelo debe ser lo más flexible que se pueda dado que se está en el proceso de convertir los datos no estructurados en lo más estructurados que sea posible. Comúnmente los datos pueden apegarse más al modelo de filas y columnas establecido en bases de datos relacionales u hojas de cálculo, pero si se han recabado datos de redes sociales como Facebook o Twitter, estos pueden asemejarse más a un modelo orientado a grafos.

El resultado de esta etapa es el tener una mayor comprensión de los datos: su naturaleza, su estructura interna, las categorías de valores que pueden existir y las posibles relaciones que existen entre los diversos entes o tablas. La salida de esta etapa puede verse como una que genera una dimensión conceptual de la información la cual sirve para poder aplicar de forma certera las técnicas de ML y PLN al contar con información suficiente sobre la estructura de los datos, esto se trata en la siguiente etapa.

Aspectos de Implementación

Después de analizar el contenido de los datasets de Yelp, se realizó el diccionario de datos de cada uno, los cuales se presentan a continuación de la Tabla *4.1* a la Tabla *4.5*:

Business			
#	Nombre	Extensión	Descripción
1.	business_id		Id encriptado del negocio
2.	full_address		Dirección completa
3.	hours	(day_of_week): { close (HH:MM), open (HH:MM) }, ...	Horas en que apertura el negocio, es un campo con extensión variable
4.	open		True / False: corresponde a cerrado, no horas de oficina
5.	categories	[]	Nombres de las categorías localizadas
6.	city		Ciudad
7.	review_count		Conteo de opiniones
8.	name		Nombre del establecimiento
9.	neighborhoods	[]	Nombres de los barrios
10.	longitude		Longitud
11.	state		Acrónimo del estado
12.	stars		Valoración de estrellas, redondeado a media estrella
13.	latitude		Latitud
14.	attributes	{ (attribute_name): (attribute_value), ... },	Colección de valores de diversas longitudes con diversas cadenas. Se refieren a las características del lugar. Pueden a su vez tener sub-categorías.
15.	type	'business'	Valor fijo

Tabla 4.1 Contenido del dataset business

User			
#	Nombre	Extensión	Descripción
1.	yelping_since		Fecha formateada como '2012-03'
2.	votes		funny: #, useful: #, cool: #.
3.	review_count		Conteo de críticas
4.	name		Primer Nombre
5.	user_id		Id encriptado del usuario
6.	friends	[(friend user_ids)]	Ids encriptados de los amigos del usuario
7.	fans		Número de fans
8.	average_stars		Promedio en punto flotante, como 4.31
9.	type	'user'	Valor fijo

10.	compliments	{ compliment_type) :(num_complime nts_of_this_type), ... },	Colección de pares de valores tipo - descripción: conteo. Entre los cuales están: profile, cute funny, plain, writer, list, note, photos, hot, cool, more. Relativo a “cumplidos” que les envían a los usuarios.
11.	elite	[]	Años elite. [Cadena variable o nula de años]

Tabla 4.2 Contenido del dataset user

Review			
#	Nombre	Extensión	Descripción
1.	votes		funny: #, useful: #, cool: #. Donde # = 0-9
2.	user_id		Id encriptado del usuario
3.	review_id		Identificador de la crítica
4.	stars		Valoración de estrellas, redondeado a media estrella
5.	date		Fecha formato tipo '2012-03-14'. Del 2004 – 2016.
6.	text		Texto con diversos caracteres y tabuladores, ancho variable.
7.	type	'review'	Valor fijo
8.	business_id		Id encriptado del negocio

Tabla 4.3 Contenido del dataset review

Tip			
#	Nombre	Extensión	Descripción
1.	user_id		Id encriptado del usuario
2.	text		Texto con diversos caracteres y tabuladores, ancho variable.
3.	business_id		Id encriptado del negocio
4.	likes		conteo
5.	date		Fecha formato tipo '2012-03-14' Del 2009 – 2016.
6.	type	'tip'	Valor fijo

Tabla 4.4 Contenido del dataset tip

Check-in			
#	Nombre	Extensión	Descripción
1.	checkin_info	{ … }	{ 0-0 (número de registros desde 00:00 a 01:00 todos los domingos), 1-0 (número de registros desde 01:00 a 02:00 todos los domingos), 14-4 (número de registros desde 14:00 a 15:00 los jueves), 23-6 (número de registros desde 23:00 a 00:00 los sábados) }, # si no hay registro a una hora, no estará en el dataset
2.	type	'checkin'	Valor fijo
3.	business_id		Id encriptado del negocio

Tabla 4.5 Contenido del dataset check-in

Como se puede apreciar, se cuenta con una mezcla entre datos estructurados y no estructurados, es decir, se encuentran semiestructurados ya que se tiene una cierta estructura homogénea de la información que cada registro tiene, pero por otro lado el contenido de la misma no se apega a los valores normalmente establecidos para los datos en los SGBD relacionales. No obstante, se puede establecer un modelo relacional de base de datos entre estas tablas para comprender las relaciones entre las mismas, este modelo da soporte a las etapas posteriores del framework. En la Figura 4.3 se presenta el diagrama entidad-relación de los ficheros Yelp en el cual se aprecian las relaciones entre estos. Es importante resaltar que este diagrama se presenta con objetivos de análisis y comprensión de los datos, porque en el entorno BD este análisis no serviría del todo.

En algunos casos, la información se encuentra en forma de tuplas, por lo que se tienen que establecer mecanismos adicionales para poder estructurar y organizar esta parte de la información. Considerando que en esta etapa se cuenta con los metadatos de la información, esta puede exportarse hacia otros medios en los que sea más ágil realizar el proceso de consulta de la información, en particular, hacia sistemas de bases de datos relacionales. Por otra parte, en la Tabla 4.6 se presenta la descripción de las instancias que se encuentran en las tablas para generar una noción del volumen total que se tienen en los ficheros.

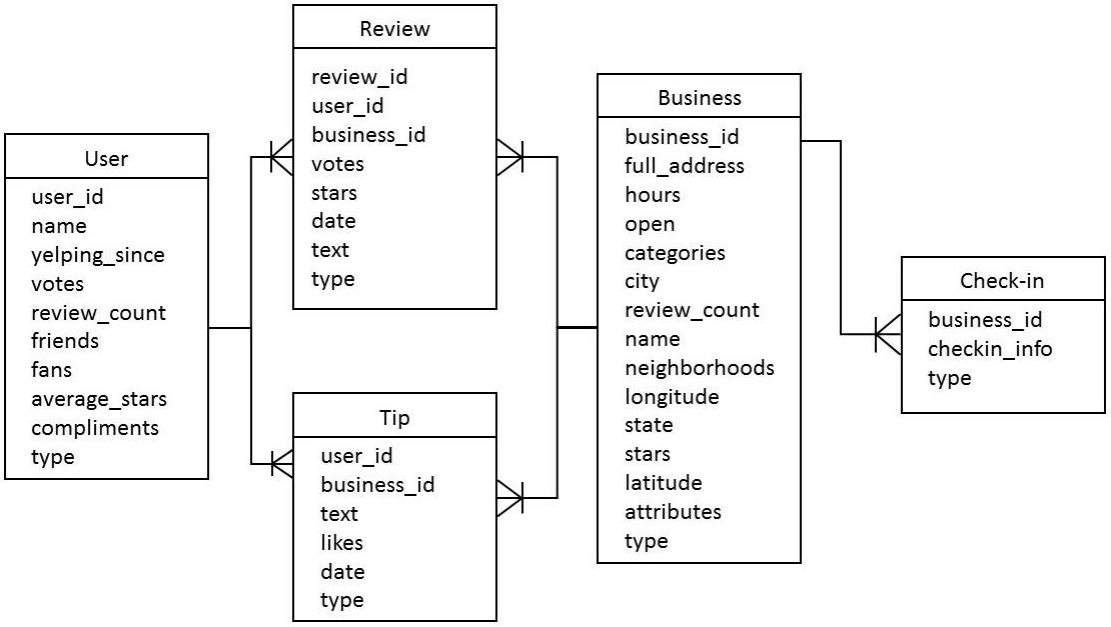

Figura 4.3 Diagrama entidad relación del dataset Yelp

Tabla	Número de atributos	Número de instancias
Review	8	2.225.213
Tip	6	591.864
User	11	552.339
Business	15	77.445
Check-in	3	55.569

Tabla 4.6 Descripción de atributos e instancias de las tablas del dataset Yelp

Dado que en esta fase no se contempla aún el uso de técnicas de GVDNE, se propone realizar este análisis directamente sobre los datos. Aunque Python es un lenguaje que está altamente integrado con JSON para operaciones de lectura y escritura, se determinó que para ahorrar tiempo en la creación de métodos que iban a efectuar tareas que ya están hechas por los SGBD relacionales, se llevaran los datasets a un SGBD. Estos datos fueron cargados entonces en el SGBD empleando los servicios de transformación de datos propios del programa, con lo cual pudieron recuperarse los datos de la base de datos de forma tradicional empleando el lenguaje SQL.

Una vez que se contó con la información en un servidor de bases de datos, se consultaron mediante SQL los registros que tuvieran información relacionada a hostelería y restauración. Esta consulta se tuvo que estar refinando puesto que, como se comentó anteriormente, los negocios están vinculados a diversas categorías. Además de buscar las críticas dadas a los negocios correspondientes, de las categorías resultantes se obtenían las estrellas o puntaje relacionado con cada registro. Los resultados obtenidos se guardaban en un fichero para analizarlos en la siguiente etapa del modelo.

Es importante resaltar que el haber llevado los datos a un SGBD es sólo el método que se escogió en su momento para poder explorar y relacionar los datos que en este se encontraban. Este método es puramente opcional ya que pudieron haberse empleado otros lenguajes de programación que recorran archivos JSON, que prácticamente todos los lenguajes modernos cuentan con tales librerías. Por otra parte, no ha de confundirse que se esté mezclando el paradigma BD con el de bases de datos, dado que en esta fase aún no están contempladas las técnicas de BD.

4.2.4. Modelo de Aprendizaje Automático

Aspectos Teóricos

La etapa anterior se enfocó en analizar los conjuntos de información existentes para generar un modelo de datos que represente, de la manera más exacta y concisa posible, la estructura y relaciones que existen entre los datos. Lo anterior con el fin de que se puedan aplicar métodos y técnicas computacionales que permitan un mayor estudio y tratamiento de los GVDNE y de estos, los que interesan en esta etapa del framework son los algoritmos de ML y las técnicas de PLN. El análisis de GVDNE puede ser llevado a cabo mediante diversas técnicas no solamente aquellas relacionadas con ML. De hecho, en diversos estudios se ha preferido la aplicación de técnicas estadísticas para realizar análisis de BD.

Podría considerarse que para el análisis de BD la mayoría de las técnicas empleadas provienen del ML, de la estadística o bien una combinación de estas, no obstante, se han realizado otras investigaciones sin emplear los métodos anteriores, como ejemplo, métodos matemáticos. En el estudio en cuestión se consideró la aplicación de técnicas de ML a diferencia de las estadísticas primero, por ser el método más aceptado y probado por la comunidad informática internacional, segundo al existir una comunidad científica y tecnológica que da soporte y sustento a lo que se pretende desarrollar (Xiang, Du, Ma, & Fan, 2017) y tercero, al ser esta tesis de naturaleza informática.

Ravi & Ravi (2015) realizaron un extenso estudio sobre las diversas técnicas y los distintos enfoques que se han aplicado en estudios comprendidos entre 2002 y 2015 que involucran la minería de opiniones y el análisis de sentimientos. En dicho trabajo se indica que los estudios se pueden dividir en dos categorías, tomando como base, el enfoque mediante el cual se abordó el estudio para su resolución: 1) Basados o no en Ontologías y 2) Basados en el léxico, el ML o ambos (híbrido). De los enfoques enunciados anteriormente se escogió el de ML debido no sólo a la alta fiabilidad de sus resultados, sino a que ha sido probado con éxito en corpus extraídos de medios sociales.

Uno de los puntos a tener en cuenta en esta parte es el nivel del análisis, basado en la granularidad del análisis del sentimiento con el cual se hace el estudio, de acuerdo a la distribución hecha por Ravi & Ravi (2015) se distinguen los siguientes niveles: documento, palabra, aspecto, oración, concepto, frase, vínculo y clausula; mencionados cada uno por el número de trabajos que se han elaborado de ellos en orden de mayor a menor.

Documento se refiere a revisiones de servicios y productos, blogs, foros, biografías, noticias, comentarios de noticias, tweets, comentarios de Facebook, etc. El nivel **palabra** identifica la polaridad de las palabras. **Aspecto** se centra en las características precisas del objetivo del sentimiento. **Oración** analiza el sentimiento general de una oración. El **concepto** se refiere a una clase o categoría en ingeniería ontológica. **Frase** es una combinación especial de dos o más palabras. Los estudios basados en **vínculos** se llevan a cabo en la extracción de opinión de las redes sociales, mientras que los estudios a nivel de **cláusula** tratan con oraciones condicionales.

De tal manera que se integra PLN al estudio debido a que reúne una serie de técnicas que permiten el preprocesamiento del corpus, mismo que es aprovechado posteriormente por técnicas de minería de datos y ML. Cabe mencionar que cuando se habla de que se emplea PLN como parte de del estudio o framework, se está haciendo referencia al hecho de que sólo algunas técnicas, así como ciertas herramientas de PLN son las utilizadas en esta etapa. PLN como área de la computación es una disciplina que abarca demasiadas actividades, todas en relación con el estudio del lenguaje natural, sin embargo, el profundizar en cada una de estas no forma parte de los objetivos o hipótesis de esta tesis.

En lo que respecta al objetivo de esta tesis en particular, se integran las técnicas de ML y PLN para analizar un conjunto de datos que son analizados mediante las particulares técnicas de aprendizaje supervisado. Es decir, que el fin específico para el que se han seguido las etapas anteriores es el de preparar un conjunto de información que a su vez cuente con un descriptor de esos datos, que idealmente contendrá información relativa a un criterio de clasificación. Este criterio de clasificación conocido como ‘variable

categórica', puede describir diversos estados: [cierto o falso], [bueno, regular y malo], [un valor entre 1 y 5] o etiquetas como [deportes, política, clima], etc.

Por lo que se persiguen dos objetivos: un general y un específico. El general es establecer un framework que contribuya al análisis de la información referida en un ambiente técnico no BD. Mientras que el específico es analizar un conjunto de datos del ámbito turístico para establecer deducciones a partir de la información contenida en este como: la polaridad del sentimiento u obtener cuales son las palabras o términos más empleados por los usuarios en función de la polaridad (términos encontrados como más positivos y negativos). La última etapa de esta fase genera un método que combina técnicas de ML y PLN centradas en el análisis de datos de texto, dicho método constituye la base de un modelo de ML.

Por otra parte, el ML lo está revolucionando todo, desde reconocimiento de imágenes hasta la conducción autónoma de vehículos, de tal manera que su integración en proyectos de investigación que involucran PLN es esencial de diversas formas. Por otra parte, diversos estudios han tratado en cierta medida el análisis de diversos textos provenientes de medios sociales. Varios de estos estudios tienen que ver con la utilidad del comentario que es analizado a través de diversas métricas que caracterizan el material fuente (Colace, De Santo, Greco, Moscato, & Picariello, 2015).

Uno de los primeros trabajos que fueron realizados empleando ML fue llevado a cabo por Pang, Lee, & Vaithyanathan (2002) en el cual emplearon un dataset con críticas de películas y en el cual se emplearon para su análisis tres métodos de ML: NB, Maximum Entropy Classification y SVM. Los resultados obtenidos indicaban en su momento que estos métodos no eran tan efectivos en la clasificación de sentimientos como si lo eran en la clasificación del tópico o tema tratado. De acuerdo con sus conclusiones, NB, resultaba ser el método con el peor desempeño, mientras que SVM fue el mejor durante las pruebas.

Siguiendo con los trabajos presentados por Pang & Lee, en un trabajo posterior al mencionado anteriormente (2004), se explora el hecho de encontrar en un conjunto distinto de datos sobre críticas de películas, la polaridad del sentimiento. En este trabajo se extrae la subjetividad del sentimiento mediante una combinación del método de NB como un extractor de la subjetividad aplicado en el nivel de análisis de documento, esto es posteriormente comparado empleando SVM. Los resultados de la predicción, aunque son muy similares, se inclinan un tanto por el uso de NB.

Diversos estudios han considerado el uso de SVM o NB para el análisis de sentimientos, ya sea considerando las características del revisor, del texto o ambos, entre muchos otros. Las RNA son otro método de amplio uso en la comunidad científica cuando

se explora la resolución de problemas mediante métodos de ML. Sin embargo, hasta hace relativamente poco tiempo estas redes no habían sido empleadas en el análisis de documentos para la extracción y análisis de sentimientos. Moraes, Valiati, & Gaviao Neto (2013) presentan un estudio que compara las SVM y las ANN con el fin de presentar a las ANN como una alternativa más eficiente (y en algunos casos al menos similar). Sus resultados demuestran que, en términos de clasificación, las ANN resultan ser mejores mientras que las SVM resultan menos afectadas por términos 'ruidosos' en el dataset.

En lo que respecta a la parte de integrar ML con PLN, esto es un paso que puede pensarse es un tanto natural dado que el proceso de preparación de la información o preprocesado del texto involucran algunas técnicas de PLN. El hecho de querer aplicar sólo PLN en un estudio de esta naturaleza resulta inviable ya que las técnicas actuales (entre estas ML, estadísticas y matemáticas) que por su capacidad de análisis y procesamiento sobrepasarían la mayoría de las veces los resultados obtenidos si únicamente se aplicara PLN. Existen diversos ejemplos que ilustran la integración de las técnicas de ML y PLN para el análisis de información, algunas de las más relevantes son: análisis de sentimiento, sistemas de recomendación o análisis de opiniones de usuarios (Alahmadi & Zeng, 2015; Araque, Corcuera-Platas, Sánchez-Rada, & Iglesias, 2017; Chen, Yan, & Wang, 2017).

Es de notar entonces que tanto los lenguajes de programación como los algoritmos ML que están orientados al análisis de textos, cuentan con diversas funciones que se realizan en herramientas de terceros especialmente dedicados a PLN. En el estudio propuesto se prefirió integrar un paquete tecnológico más a la arquitectura para incrementar las capacidades de preprocesamiento y procesamiento del lenguaje. Sin embargo, cada investigación futura deberá analizar cual enfoque o técnica de análisis de PLN conviene mejor para los objetivos del estudio propuesto.

Este MLM tiene las siguientes consideraciones:

I. El modelo se puede aplicar para realizar el análisis de los datos existentes para obtener la información oculta en ellos, por ejemplo: gustos y aversiones del usuario, extracción de características o análisis de sentimientos. La precisión del modelo en la predicción de resultados podría variar de acuerdo con las técnicas de ML y PLN utilizadas.
II. El modelo está diseñado para ser ejecutado en una arquitectura que no sea BD, lo que significa que datos de menor dimensión (small data) podrían analizarse sin tener recursos de alto rendimiento para efectuar esta tarea. Esto también tiene la ventaja de que se pueden obtener resultados confiables con mayor agilidad al eliminar la barrera de tener que contar con una infraestructura informática muy sofisticada.

III. Este modelo es la base de la etapa ML en la segunda fase, ya que esta involucra el análisis mediante BD. Por lo tanto, el modelo de ML podría aplicarse total o parcialmente al análisis de BD.

Este modelo pretende hacer una correlación entre datos cuantitativos y cualitativos a través de la comparación entre texto y valoraciones numéricas.

Procesamiento de Lenguaje Natural

En esta etapa, algunas de las técnicas de PLN son incorporadas al framework para realizar cierto preprocesamiento del corpus disponible, es decir, que el corpus como tal en su estado 'inicial' debe ser transformado hacia otro que permita un procesamiento más adecuado por técnicas relacionadas con el ML. De tal manera que en este trabajo no se plantea innovar en alguna área del PLN o modificar lo ya establecido ya que eso se encuentra totalmente fuera del alcance de este trabajo. Por el contrario, se le incluye en este trabajo por estar totalmente relacionado con disciplinas como análisis de sentimientos donde entran en conjunto técnicas tanto de PLN como de ML. Por lo tanto, PLN viene a completar esta etapa del framework dando un mayor sentido a las posteriores tareas de ML (como se explica más adelante en este mismo apartado).

Las herramientas computacionales empleadas en esta etapa como el lenguaje Python y a su vez el paquete NLTK (Natural Language Toolkit) integran en su estructura los métodos necesarios para el tratamiento de textos de diversa índole, es decir, que ya cuentan con los procedimientos necesarios para llevar a cabo el PLN. De hecho, Python es ampliamente reconocido en el ámbito de la investigación por la forma en que facilita la productividad, calidad y mantenibilidad del software (Bird, Klein, & Loper, 2009). De tal manera que, aunque se reconoce que la gran mayoría de las herramientas computacionales llevan a cabo el manejo de cadenas de texto, Python ha destacado recientemente en ciencia de datos como el lenguaje más empleado.

En esta etapa del framework, Python y NLTK brindan capacidades de PLN que ya están integradas en estas herramientas. Durante la etapa de construcción del modelo de ML se emplean las herramientas mencionadas pues tienen métodos preconstruidos para dar soporte a dicho modelo, con lo cual se consigue emplear menos tiempo en realizar los pasos preparatorios del texto y enfocarse más en la preparación del modelo.

Aspectos de Implementación

Como se ha mencionado, uno de los principales objetivos de esta tesis es estudiar las reviews relacionadas con los servicios turísticos, específicamente hoteles. Para distinguir

qué reviews estaban relacionadas con los hoteles, se tomó una muestra de la tabla Reviews la cual se relacionó con la tabla Business mediante lenguaje SQL. Luego, de esta muestra, solo los campos 'estrellas' y 'texto' fueron considerados para la etapa de ML. Estos campos son de gran importancia ya que reflejan cómo el usuario califica los servicios tanto en aspectos cuantitativos como cualitativos.

Para una misma unidad de negocios los usuarios expresan sus sentimientos de diferentes maneras, lo que los convierte de gran interés para analizar cómo las palabras y expresiones utilizadas están relacionadas con las 'estrellas' asignadas. En esta etapa era necesario tener la información de Reviews (estrellas, texto) en un único archivo, sin embargo, estos datos necesitaban una limpieza adicional ya que había muchos elementos encontrados en formularios web (caracteres especiales, espacios en blanco, nulos), por lo tanto, la limpieza de datos se realizó ejecutando una instrucción SQL para eliminar datos ‘ruidosos’.

El ML se integró al estudio en virtud de querer encontrar una técnica computacional que contribuyera a relacionar los elementos cuantitativos (estrellas) contra los cualitativos (críticas) empleando no sólo el PLN, esto debido a que aplicando sólo técnicas de PLN resultaría más complejo y extenso el llevar a cabo este estudio. El MLM que se describe debe analizar el texto para identificar los términos subyacentes que puedan relacionarse con la calificación dada por el mismo revisor. Los proyectos que involucran ML pueden llevarse a cabo mediante lenguajes como Python o R, para lo cual se empleó Python.

Para poder desarrollar las tareas de ML es necesario recurrir a algún paquete tecnológico que permita el acceso a tales métodos y procesos, en este caso se seleccionó Scikit-learn, un proyecto de código abierto que es ampliamente usado en la academia y la industria (Mueller & Guido, 2016), además de ser considerada la herramienta más popular y prominente para proyectos de Python que involucran ML. Para la parte de PLN y el preprocesado de datos se usaron paquetes como: NLTK y Pandas (manipulación y análisis de datos). Las stopwords y signos de puntuación se eliminaron de las reviews que además se transformaron en minúsculas.

Después de haber ejecutado modelos preliminares de ML empleando los datos referidos, se descubrió que la distribución de la muestra estaba sesgada hacia reviews que tenían una o cuatro a cinco estrellas, haciendo que las revisiones con dos y tres estrellas fueran las menos presentadas en la muestra. Tal sesgo estaba causando que el modelo de ML se centrara más en los bordes y menos en el centro, por lo que se decidió elegir una muestra equilibrada de reviews. Esto hizo que el conjunto final de datos consistiera en 66,410 reviews, teniendo 13,282 reviews para cada estrella.

Sin embargo, cuando se estaba en la etapa de probar los diversos algoritmos de ML que se seleccionaron para la clasificación con cinco estrellas, los porcentajes de predicción que se obtenían eran muy bajos, por mencionar alguno, 58% con LR (el mejor en su caso). Luego, se reorganizaron las etiquetas de los datos para establecer en 1 todas las valoraciones que tuvieran un 2, y en 5 todas las que tuvieran un 4; dejando de este modo las etiquetas en 1, 3 y 5. Sin embargo, esta restructuración de las etiquetas tampoco llevó a porcentajes de predicción muy altos pues estos rondaban el 73%, por lo que esto tampoco fue una alternativa de solución.

Posteriormente, después de considerar el estado de la investigación previa donde los autores usaron conjuntos de datos similares al de Yelp (Chang et al., 2017; Ghaddar & Naoum-Sawaya, 2018; Jiangtao Qiu, Liu, Li, & Lin, 2018), se observó que el método que utilizaron para etiquetar los datos fue empleando la clasificación binaria. Por lo que se decidió realizar dos pruebas usando: (i) 1 y 2 estrellas para la clase negativa; 3, 4 y 5 para el positivo, o (ii) 1, 2 y 3 estrellas para la clase negativa; 4 y 5 para el positivo, con los resultados mejorando para la primera combinación de datos (Jimenez-Marquez et al., 2019).

Para ilustrar este proceso, en la Figura 4.4 se presenta un ejemplo de dos críticas, una con la más baja y otra con la más alta puntuación para una misma unidad de servicio, y cómo a partir del conjunto de datos de Yelp se dividieron y etiquetaron los datos en dos clases: positivos y negativos.

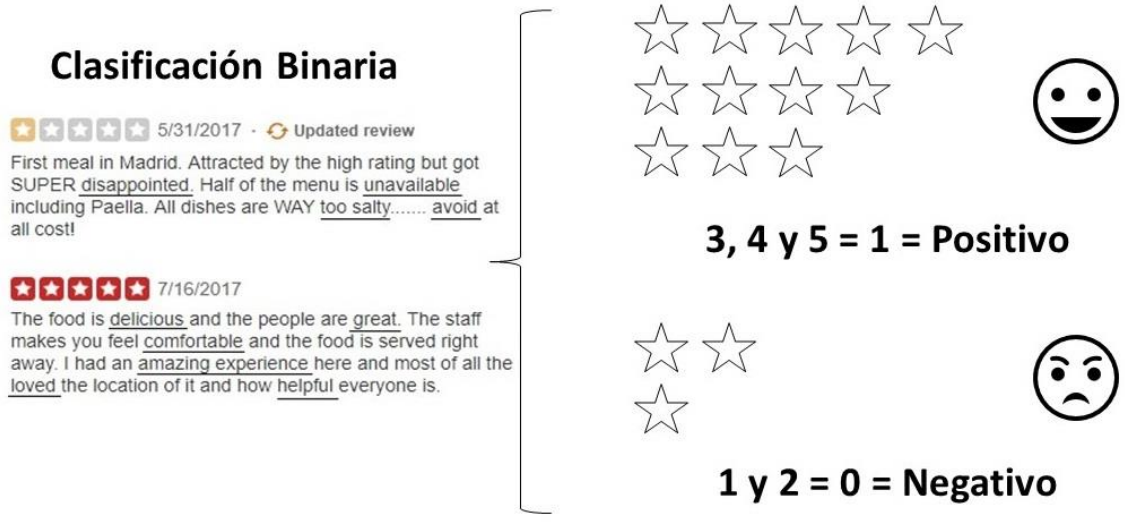

Figura 4.4 Representación de clasificación binaria hecha a partir de los datos de Yelp

Como se aprecia, los retos que se presentan son importantes ya que no existe una clara distinción de qué términos pueden ser empleados tanto en una valoración positiva como

en una negativa. En la crítica negativa presentada en la Figura 4.4 se emplean términos como *high* y *super* que normalmente podrían emplearse en puntuaciones más altas. Otros problemas que pueden presentarse se dan cuando casi todas las oraciones parecen ir en un sentido, pero la oración final tiene un sentido contrario y, en consecuencia, asigna una puntuación diferente a la calculada. También se debe considerar el tamaño de la crítica ya que una con más contenido puede contribuir en mayor medida a establecer el gusto del consumidor.

De acuerdo a la clasificación hecha por Ravi & Ravi (2015), el nivel de análisis con el cual se realiza el estudio en esta parte es a Nivel Documento, esto basado en el hecho de que los estudios a este nivel buscan en todos los elementos que hablan del documento, en este caso las críticas. No obstante, se consideran en la construcción del modelo, los otros niveles de análisis con el fin de lograr un mayor porcentaje de predicción del algoritmo de ML.

Antes de empezar a describir el modelo, es importante comprender el concepto del Modelo de espacio vectorial (Amine, Elberrichi, & Simonet, 2010). Este modelo es ampliamente empleado en el ámbito de recuperación de la información (información retrieval) y se emplea para tareas de filtrado, recuperación, indexado y cálculo de relevancia de información. De manera muy básica podemos plantearlo como una matriz donde la primera columna está formada por todos los documentos que conforman el corpus, y el resto de las columnas están encabezadas por los términos relativos a los documentos y la frecuencia con que estos aparecen. Se presenta un ejemplo básico de este modelo en la Tabla 4.7:

	Término 1	**Término 2**	**Término 3**	**Término 4**
Documento 1	0	1	1	0
Documento 2	1	0	1	0
Documento 3	1	0	0	0
Documento 4	1	1	1	1

Tabla 4.7 Ejemplo del Modelo de espacio vectorial

Como se aprecia en la tabla anterior, para cada documento existente en el corpus se hace una relación entre los términos que se van reconociendo y su frecuencia a lo largo de los documentos. Por ejemplo, el término 1 sólo está presente en los documentos 2, 3 y 4 mientras que está ausente del documento 1, el cual a su vez sólo tiene los términos 2 y 3, pero no el 1 y el 4. Entonces, dado que los algoritmos de ML trabajan particularmente con representaciones de vectores y matrices, se requiere de una técnica que permita transformar el corpus y los términos en un orden matricial para su posterior análisis

mediante ML. Llevando esto al corpus particular que se analizó, puede pensarse en una representación similar como se muestra en la Tabla 4.8:

	Room	**Surprise**	**Hotel**	**Stay**
They even upgraded me to a larger **room** at no charge which was a nice **surprise**	1	1	0	0
We got the smallest **room**, and I was **surprised** at how tiny it was. There was barely any **room** to store our luggage or walk around the **room**.	3	1	0	0
The **rooms** are rather tiny but just what you need for a night or two in the city. Pretty solid **hotel**.	1	0	1	0
The **room** was great, and we would ABSOLUTELY **stay** there again.	1	0	0	1

Tabla 4.8 Representación del Modelo de espacio vectorial del Corpus

La tabla anterior presenta un abstracto de lo que serían las reviews en la primera columna de la tabla, mientras que las siguientes columnas hacen una relación de los términos encontrados por documento y cada documento qué términos tiene de los 30,000 en total que se consideraron (la cifra se explica más adelante). Como se ha explicado antes, se necesita de una técnica que tome el corpus y lo transforme en la tabla anterior para que pueda ser analizada mediante procesos de ML, tal técnica es conocida como el vectorizador o Vectorizer.

El primer aspecto a considerar en el modelo fue qué vectorizador (trabaja en base al modelo del espacio vectorial) se aplicaba a los datos. Teniendo CountVectorizer y TfIdfVectorizer como las principales opciones, ambos fueron probados, encontrando que este último se desempeñó mejor para los datos. Como se comentó en el Estado del Arte, existen diferencias entre emplear vectorizadores como CountVectorizer y TFIDFVectorizer. Mientras que el primero sólo realiza el conteo de los términos o palabras a lo largo de un documento, el segundo lleva a cabo el mismo conteo de palabras, pero además mide la frecuencia con que estos términos ocurren a lo largo del documento.

De tal manera que las palabras que aparecen con mayor frecuencia tienen un índice TFIDF menor, y las palabras que aparecen con menos frecuencia tengan un índice TFIDF mayor. Estas situaciones deben ser tomadas con particular consideración dependiendo del problema que se esté pretendiendo resolver o de la información que se quiera analizar, así como el correspondiente valor que se pretenda extraer de los datos. Para el corpus particular que se tiene, aplicar CountVectorizer no satisfacía las necesidades del estudio

debido a que contar los términos o palabras no es de relevancia especial. Mientras que TFIDFVectorizer obtiene la frecuencia inversa con que cada término ocurre en los documentos.

Para ilustrarlo mejor: es más relevante saber con qué frecuencia se menciona la palabra 'cleanliness' (limpieza) en las reviews (documentos) que saber si hubo 10.000 menciones de esta. Por otro lado, esto no debe confundirse con las stopwords o palabras vacías, las cuales ya fueron previamente removidas del corpus. En este caso, interesan los términos que, si bien aparecen con mucha frecuencia, tampoco presenten una frecuencia tan alta que constituya de tal manera un término muy empleado en el dominio particular a analizar, en este caso, hotelería y/o turismo; por lo que este tipo de términos fueron descartados.

Pero tampoco se necesitan términos que aparezcan tan pocas veces que, aunque presenten un valor TFIDF muy alto, sean términos pocos comunes, ejemplo: encontrar hasta cinco veces la ocurrencia de palabras como 'soooooo', 'gooooood'; por lo que términos de esta tipo también fueron descartados del corpus para el procesamiento posterior. El siguiente paso fue cómo configurar el TfIdfVectorizer, la Tabla 4.9 muestra los parámetros empleados, su descripción y valor:

Parámetro	**Descripción** (información adaptada de la documentación oficial (Scikit-Learn, 2018))	**Valor**
use idf	Habilita la reponderación inversa de documentos de frecuencia (inverse-document-frequency).	True (default)
Tokenizer	Anula el paso de *tokeniz*ación de cadenas mientras se conservan los pasos de generación de preprocesamiento y n-gramas.	tokenize(). Método propio
ngram range	[tupla (min_n, max_n)]. El límite inferior y superior del rango de n valores para diferentes n-gramas que se extraerán. Todos los valores de n tales que min_n <= n <= max_n se usarán.	(1, 2)
min df	Al construir el vocabulario, ignora los términos que tienen una frecuencia de documento estrictamente más baja que el umbral dado. El parámetro representa una proporción de documentos, conteos absolutos enteros.	5
max df	Flotante en el rango [0.0, 1.0]. Al construir el vocabulario, ignora los términos que tienen una frecuencia de	0,5

	documento estrictamente más alta que el umbral dado (palabras vacías o *stop words* específicas del corpus). El parámetro representa una proporción de documentos, conteos absolutos enteros.	
max features	Crea un vocabulario que solo considera las máximas características (o términos) ordenadas por frecuencia de términos en todo el corpus.	30.000
stop words	[cadena {'english'}, lista, o Ninguno] 'English' es actualmente el único valor de cadena soportado. Si se incluye una lista, se supone que esa lista contiene *stop words*, todas las cuales se eliminan de los tokens resultantes.	English

Tabla 4.9 Parámetros empleados en la configuración de TFIDFVectorizer

Para ilustrar de forma gráfica el proceso anterior, en la Figura 4.5 se presenta una representación de cómo transforma TFIDFVectorizer el corpus original. Como se observa en dicha figura, el corpus se conforma de diversos documentos o reviews, los cuáles conforman una serie de elementos en la figura. En el proceso se eliminan las stopwords o palabras repetidas que no son necesarias ya que no brindan información adicional sobre el texto analizado. Este proceso también extrae las features que están formadas por palabras simples y combinadas en pares, lo cual está expresado por los ngramas.

De los ngramas se establecen los límites mínimos y máximos a considerar como se explica en la Figura 4.5, esto conforma el modelado del corpus que se pasa a la etapa de ML. En la parte inferior de la figura se ilustra nuevamente una representación del modelo del espacio vectorial, como se aprecia, sólo se considera un máximo de 30,000 features. También se presentan los encabezados de los features, los cuáles como se indica, son reducidos a su forma raíz para conjuntar los términos que expresan lo mismo y así enriquecer el número de features que se incluyen en el análisis.

Para evitar el overfitting o sobreajuste al momento de entrenar los algoritmos de ML, se establecieron de dicha forma los parámetros mostrados en la Tabla 4.9, es decir, haberlos ajustado más podría no servir para un corpus distinto, si acaso el parámetro max_features podría adaptarse a un valor distinto en función del tamaño total del corpus. A continuación, se explican algunos parámetros de la Tabla 4.9: el método predefinido tokenize para separar las cadenas del texto y obtener los tokens, emplea una herramienta del lenguaje natural de Python, llamada SnowballStemmer, el cual convierte también cada cadena a minúsculas y convierte cada pablara a su forma raíz. Es decir, palabras como complicated, complication o complicating serían reducidas a complicate, lo cual

contribuye a reducir el tamaño del corpus (features) y enfocarlo en concentrar los términos comunes más habituales.

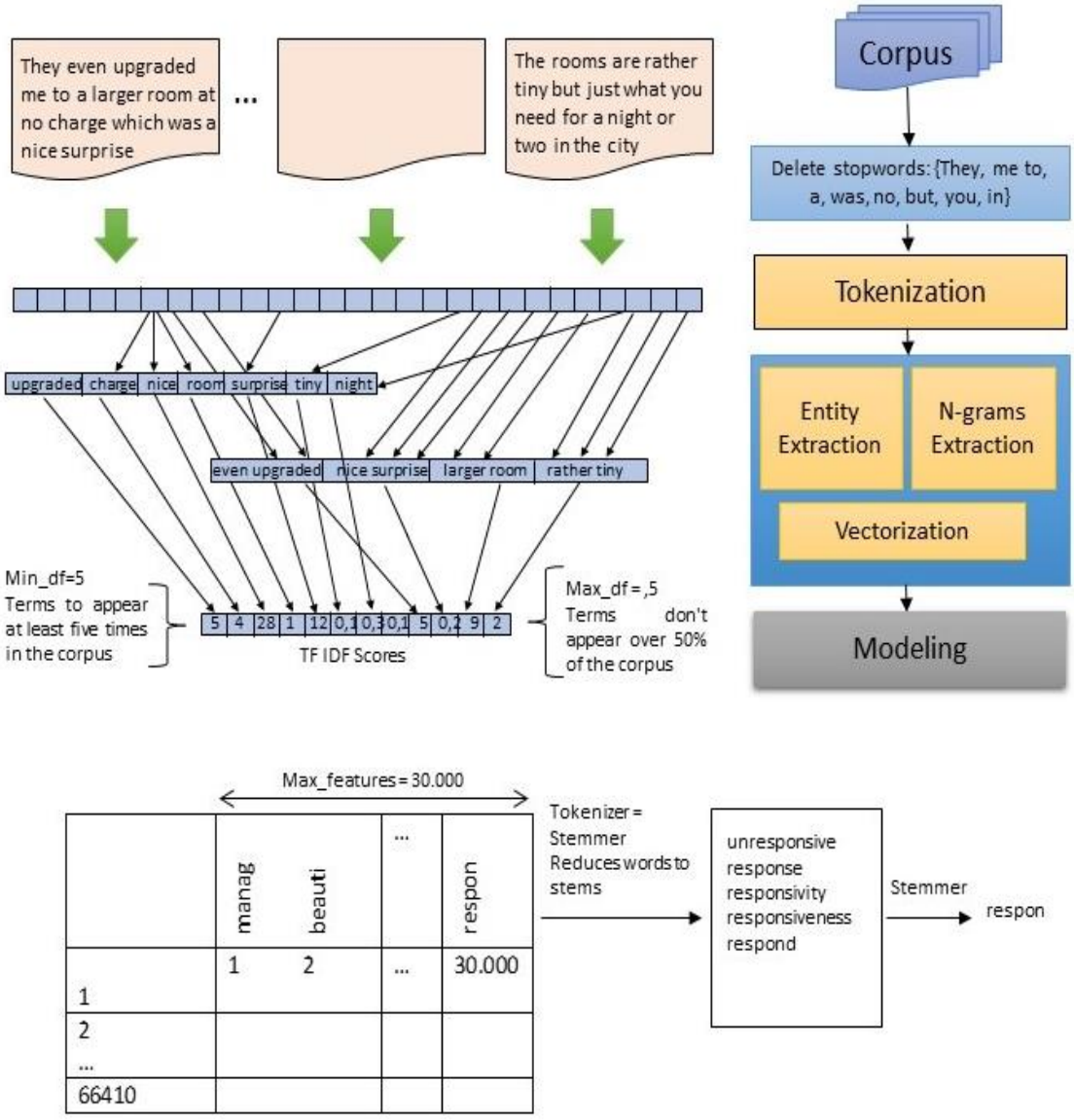

Figura 4.5 Proceso realizado por TFIDFVectorizer para construir la tabla de features

adaptado de (Ojeda, Bilbro, & Bengfort, 2018)

El rango de los ngramas se refiere a la amplitud o el espectro de palabras combinadas a considerar como parte del corpus. Es decir, se mencionó anteriormente que se obtuvieron los términos o tokens del corpus, por otro lado, existen términos agrupados que se repiten con habitual frecuencia a lo largo del corpus. Se realizaron algunas pruebas con este parámetro para comparar el rendimiento de este, como se ilustra mejor en el siguiente capítulo. Ejemplo de los n-grams más comunes serían:

(1,2): ‘too expensive’, ‘very nice’, ‘really dirty’.

(1,3): ‘way too high’, ‘ideal for kids’, ‘was so happy’, ‘a joyful resort’.

Donde: el primer valor indica el menor número de términos a considerar y el segundo el máximo. Se encontró que la combinación de (1,2) era la que encontraba mejor los términos comunes, se desempeñaba mejor al momento de la predicción y mantenía un número de elementos (features) no muy elevado en el corpus. La frecuencia mínima de documentos (min df) se estableció en 5 debido a que se ignoran los términos encontrados en menos de cinco documentos. Es decir, qué términos que aparecen en el rango de 1 a 4 documentos son excluidos del corpus por considerarse poco comunes o frecuentes; estos términos no sirven para el estudio debido a que sería muy extenso abarcar una magnitud mayor de términos, además que, como se ha mencionado, muchos de estos términos son palabras mal escritas (baaad, incredibli).

La frecuencia máxima de documentos (max df) se estableció en ,5 debido a que se ignoran los términos encontrados en más del 50% de los documentos. Es decir, qué términos aparecen en más de la mitad de los documentos. Aunque el valor pueda parecer alto se determinó este porcentaje al considerar que un mayor número de apariciones pueden ser también stop words, las cuales ya han sido filtradas previamente mediante el parámetro stop_words. Se hace esto para asegurar que haya una alta diversidad de términos, pero a la vez filtrar que no sean demasiado repetitivos. En el Capítulo 5 se presentan los resultados alcanzados al haber variado este parámetro y los valores de predicción a priori que se fueron obteniendo.

En lo que respecta al parámetro max_features, considerar 30,000 como el número máximo de características o palabras, fue por dos razones: primero, el número bruto de características era superior a 100.000, lo que resultaba en bajo rendimiento y predicción del clasificador ML, segundo, porque era el valor óptimo que se ajustaba a los datos. Estas consideraciones en el vectorizador y el preprocesamiento de datos previos conducen a evitar el sobreajuste ya que muchas pruebas se ejecutaron con diferentes clasificadores y combinaciones de datos y no fue necesario ajustar un nuevo vectorizador para cada prueba.

Los clasificadores utilizados para las pruebas fueron: Perceptrón multicapa (MLP), Clasificador de vectores de soporte - C (SVC), Regresión logística (LR), Clasificador lineal de vectores de soporte (LSVC), Clasificador lineal con entrenamiento de Gradiente estocástico (SVM-SGD) y Naïve Bayes Multinomial (MNB). También se probaron otros clasificadores durante las pruebas preliminares, pero estos presentaron baja precisión en los resultados, por lo tanto, no se mencionan en este apartado. En el Capítulo 5 se entra a fondo sobre la función de cada clasificador, su relación con la clasificación de textos y

los parámetros empleados; mientras que en el Apartado 5.3.5 se explican los motivos de haber seleccionado estos algoritmos de ML para realizar las pruebas.

El preprocesado del texto empleando PLN y la puesta a punto de los algoritmos es la base del modelo de ML propuesto para la resolución del tipo de problemas descrito a lo largo de este capítulo. Como se ha mencionado, se pretende que empleando las técnicas utilizadas por el modelo se agilice el proceso de creación de un modelo similar para un ambiente BD. No obstante, se hace hincapié en que la base tecnológica que impulsa ambos ambientes (BD y no BD) no va necesariamente de la mano de ambos dado que BD es una filosofía completamente distinta de los sistemas convencionales de procesamiento de datos.

4.3. Elementos de la segunda fase

En esta fase se plantea el diseño de una arquitectura de BD que dé soporte a la etapa de análisis predictivo de GVDNE. BD aunque es una técnica relativamente nueva en el mercado (poco más de un lustro) ha contado desde su inicio con el soporte de diversos proveedores que han propuesto desde su perspectiva tecnológica particular, diversas soluciones a la gestión de los GVDNE. De tal manera que en esta fase se lleva a cabo la propuesta de una arquitectura que integra diversos paquetes tecnológicos para establecer una metodología que permita el análisis de GVDNE que en su mayoría contengan datos con lenguaje natural para establecer ciertas predicciones basadas en el análisis mediante la integración y adaptación del modelo de ML obtenido en la fase anterior.

Uno de los principales objetivos de esta fase es establecer una metodología que permita el análisis de datos principalmente de medios sociales mediante técnicas de ML, PLN y BD. Esta propuesta se establece para repositorios de datos de grandes dimensiones, no se abarca el estudio de datos generados en tiempo real, lo cual puede ser objeto de un proyecto de investigación alterno. Por otra parte, BD puede llegar a ser considerado en ocasiones como un esfuerzo inútil tanto en el aspecto tecnológico, como un gasto sin retorno en el económico. Por lo que es necesario establecer nuevas metodologías que permitan obtener un valor real o un significado del conjunto de datos en estudio.

Oussous et al. (2017) establecen que han sido creados diversos modelos para obtener el conocimiento de los datos, así como la infraestructura tecnológica que dé el soporte necesario a esta demanda, pero reconocen que el estar seleccionando las técnicas adecuadas puede consumir bastante tiempo además del coste futuro que esto pueda acarrear. Aunque se puede recurrir a la asesoría de un proveedor particular, el contar con

un modelo que sólo contempla una alternativa tecnológica puede no ser siempre la mejor opción para una organización. Deben también ser tomadas en cuenta las necesidades futuras de la organización para evaluar si la alternativa de solución cumple con parámetros como: compatibilidad, seguridad, eficiencia, desempeño, confiabilidad y seguridad, entre otras.

De tal manera que, aunque en esta fase se propone una solución tecnológica que involucra diversas soluciones, se exploran también otras técnicas que permiten en mayor o menor medida realizar las mismas funciones de cada etapa. Se ha considerado también integrar en esta parte a la Nube (Cloud Computing) en el sentido de que cada vez más proveedores como Amazon o Google proponen el desarrollo de servicios basados en la nube para soluciones tecnológicas que requieren la integración de BD.

4.3.1. Sistema de Archivos Distribuido

Aspectos Teóricos

Esta etapa no genera como tal una salida, más bien debe verse como una base sobre la cual se busca establecer toda la arquitectura y el ambiente de BD. Esta base sirve a la etapa siguiente para asegurar cuáles son los servicios que son empleados y que a su vez potencian las posibilidades de las capas superiores del modelo, servicios que por cierto ya están en su mayoría predefinidos al haberse establecido determinado sistema de archivos distribuido (DFS). Como se ha visto, la justificación y a la vez la importancia de emplear un DFS en el framework propuesto se deriva del hecho de que para poder procesar GVDNE se necesita ir más allá de las capacidades que un solo ordenador puede prestar, lo cual se puede lograr mediante la integración de un DFS.

Aspectos de Implementación

Para realizar el procesamiento distribuido de la información se decide adoptar HDFS (Hadoop Distributed FileSystem) un sistema de archivos distribuido basado en Hadoop, el cual es un entorno abierto diseñado desde un principio para que permitiera el almacenamiento distribuido de datos en un clúster de cómputo y llevar a cabo el procesamiento de los datos que estén repartidos a lo largo de dicho clúster. HDFS ha sido ampliamente adoptado por diversos fabricantes de entornos de BD ya que su filosofía es similar a la de un DFS, es decir, que el conjunto total de los datos se distribuye entre todos los equipos que se hayan dispuesto para este fin.

Lo anterior conlleva significativos ahorros en el tiempo de acceso ya que se distribuye el almacenamiento entre todos los nodos, lo cual es adecuado para la solución propuesta

ya que se pueden escalar los datos en cualquier medida sin verse afectado el desempeño de las aplicaciones. MapReduce es el modelo genérico de procesamiento de datos que acompaña a Hadoop, este procesamiento se hace en dos pasos uno llamado Map el cual se encarga de dividir los datos de entrada en un número de pequeñas partes de manera que puedan ser procesados independientemente. Cuando termina *Map*, se consolida la salida y el resultado final es generado en la fase de *Reduce*.

El modelo computacional para procesamiento BD, Hadoop – MapReduce, fue de los primeros propuestos para el tratamiento de GVDNE. Este modelo ha sido probado y puesto en práctica en diversos proyectos de la comunidad científica y tecnológica, sin embargo y ante el auge cada vez mayor de necesidades de almacenaje, tratamiento y análisis de tales volúmenes de datos, se han ido proponiendo nuevas técnicas que resuelvan de forma más ágil y eficaz las cuestiones citadas. Una de las herramientas que se han propuesto y se está convirtiendo en el modelo a seguir es Apache Spark, la cual se seleccionó para el diseño del sistema de archivos distribuido.

Apache Spark es un entorno abierto de procesamiento distribuido creado en el AMPLab de la Universidad de California Berkeley. Spark es un motor escalable de procesamiento de datos distribuidos basado en Java Virtual Machine (JVM), el motivo de haber seleccionado Spark como motor central de la arquitectura de BD es que es más rápido comparado con otros frameworks (entornos) de procesamiento de datos (Thottuvaikkatumana, 2016). Como se explica por uno de sus creadores, Spark fue desarrollado al haber identificado trabajos que se ejecutaban pobremente en Hadoop y que requerían procesamiento de algoritmos de ML. En la Figura 4.6 se presenta el esquema de la división de tareas en base a la pila de Spark.

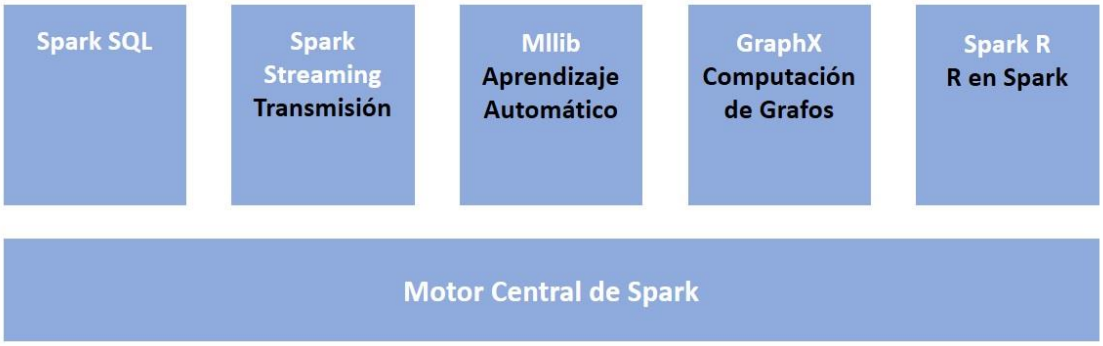

Figura 4.6 División de las tareas en base a la pila de Spark
adaptado de (Scott, 2015)

Precisamente uno de los puntos más fuertes de Spark es su motor de procesamiento el Grafo acíclico dirigido o DAG (Directed Acyclic Graph) que se basa en el procesamiento

de las tareas más en memoria que en disco, que lo hace hasta 100 veces más rápido que MapReduce si se ejecuta en memoria o 10 veces si es en disco. Esto se debe principalmente a que realiza el procesamiento en la memoria principal de los nodos principales y no en disco. Por cada trabajo en Spark se crea un grafo DAG de tareas que es ejecutado por el motor mencionado.

Se ha mencionado que se estableció HDFS como el sistema de almacenamiento distribuido de disco para Spark, no obstante es importante mencionar que debido a la versatilidad que ofrece, Spark puede ser también integrado con otros subsistemas populares de almacenamiento entre los cuales están: HBase, Cassandra, MapR-DB, MongoDB, S3 de Amazon (Scott, 2015), Google Cloud Storage y Apache Couch DB. Si bien los sistemas antes mencionados deben cubrir características particulares de rendimiento que deben ser tomadas en cuenta por el diseñador de la arquitectura particular. Entre otras, las características a cubrir son: la velocidad de ejecución en memoria o el tiempo que tardan en procesar las operaciones.

Una evaluación del desempeño de sistemas NoSQL como los mencionados anteriormente es descrita en (Talha & Kara, 2016) donde se hace en primera instancia, una distinción del tipo de sistemas NoSQL que existen en base a diferentes optimizaciones: almacenamiento llave-valor, almacenamiento de documentos, familia de columnas y base de datos de grafos, por lo que antes de realizar una evaluación de desempeño, deben homologarse antes los criterios para que estos sean uniformes (evaluar sistemas de la misma categoría). Las aplicaciones más críticas, que demanden una respuesta casi en tiempo real deben recurrir a evaluar sistemas NoSQL que permitan el acopio, análisis y procesamiento continuo de datos

Cabe destacar los enfoques orientados a la Nube de Amazon Web Services (Amazon S3) y Google (Google Cloud Dataproc) los cuales dan soporte a un gran conjunto de servicios de Spark, no obstante, su uso está bajado en el concepto “paga por lo que uses” (pay as you go) el cual puede ser un tanto limitante para aquellas organizaciones que hagan uso intensivo de estos servidores. Mucho es lo que se ha comentado en el estado del arte sobre el modelo de la Nube, sus ventajas y características, por lo que respecta a este apartado se agrega que el considerar esta alternativa como base de la arquitectura aporta diversas ventajas al disminuir la carga de realizar el soporte y mantenimiento del hardware y software base.

4.3.2. Administrador de Recursos del Clúster

Aspectos Teóricos

La etapa anterior estableció la base del DFS a utilizar en el estudio, ya sea que esta sea implementada en la infraestructura local de cómputo o que sea a través de Cloud Computing. En el entendido de que el DFS sea implementado en el ámbito local, se requiere un administrador de recursos del clúster que administre la ejecución de las tareas que se reparten entre los nodos. Luego, ¿por qué debe emplearse un clúster? Para responder lo anterior se puede dar una definición de clúster, el cual es una colección de computadoras conectadas en una red que trabajan juntas para ejecutar varias tareas en paralelo (Dar, 2016).

De acuerdo con el autor citado anteriormente, los clusters permiten conectar un gran número de computadoras y emplear todo su poder de procesamiento para resolver un problema, además de que aseguran la tolerancia a fallos ya que los nodos con mal funcionamiento pueden ser fácilmente reemplazados por otros. Por lo anterior, se propone entonces emplear un clúster para elevar la capacidad de procesamiento de las tareas, ya que no solamente es necesario para dar respuesta a futuras demandas, sino que las herramientas de BD están especialmente adaptadas para permitir el cómputo paralelo y distribuido a lo largo de clusters de computadoras.

Un clúster de computadoras necesita de un elemento central que orqueste las diversas tareas que se realizan en los nodos, así como unificar sus resultados, este elemento es el Administrador de Recursos del Clúster. Dicho administrador es un elemento de software que actúa como un orquestador de las actividades a ejecutar y que monitorea la operación y el rendimiento de los nodos. Esta etapa al igual que la anterior, sirve como una base en el modelo propuesto ya que el administrador de recursos del clúster depende mucho del DFS que se haya escogido. Por lo que no produce en sí una salida sino más bien funge en las capas inferiores del modelo como un elemento que da soporte y coordina las acciones que se llevan a cabo en las etapas superiores.

Aspectos de Implementación

Como se comentaba en el punto anterior, la necesidad de analizar GVDNE puede llegar a requerir la integración de un clúster (local o en la nube) para aumentar las capacidades que ofrece Spark. Spark puede ser instalado en modo autónomo en un solo ordenador, siendo esta la forma más simple de instalar Spark y empezar el trabajo de forma casi inmediata. Los otros modos de despliegue del clúster son a través de Hadoop YARN y Apache Mesos, los cuáles se presentan y exponen en la Tabla 4.10:

Modo	Manejador	Trabajador	Ejecutor	Maestro
Local	Se ejecuta en una JVM	Se ejecuta en la misma JVM del manejador	Se ejecuta en la misma JVM del manejador	Se ejecuta en un solo anfitrión
Autónomo	Puede ejecutarse en cualquier nodo del clúster	Se ejecuta en su propia JVM de cada nodo	Cada trabajador en el clúster lanza su propia JVM	Puede ser ubicada arbitrariamente donde el maestro haya iniciado
Yarn (cliente)	En un cliente, no en parte del clúster	YARN NodeManager	El contenedor de YARN NodeManager	El administrador de recursos de YARN trabaja con la aplicación Maestra
Yarn (clúster)	Se ejecuta dentro de la aplicación Maestra de YARN	La misma que YARN en modo cliente	La misma que YARN en modo cliente	La misma que YARN en modo cliente
Mesos (cliente)	Se ejecuta en un cliente, no es parte del clúster Mesos	Se ejecuta en el esclavo Mesos	Contenedor dentro del esclavo Mesos	Maestro Mesos
Mesos (clúster)	Se ejecuta dentro de un Maestro Mesos	La misma que modo cliente	La misma que modo cliente	Maestro Mesos

Tabla 4.10 Modos de ejecución en Spark

Comparativo entre YARN y Mesos

Es un tanto complejo hasta cierto punto el tener que definir entre YARN y Mesos cuál de ellos es el mejor administrador de recursos para el clúster a emplear en el framework. Detrás de ambos está el respaldo de centros de investigación que avalan la eficacia de ambas herramientas para la integración de BD en un clúster y que uno de estos sea su administrador de recursos. En el trabajo desarrollado por Reuther et al. (2018) se analiza y compara el rendimiento de Yarn, Mesos, Slurm y Grid Engine como los más populares administradores de recursos tanto para el Cómputo de Alto Desempeño (HPC) como para BD. Después de haber sometido a estos administradores a diversas pruebas de rendimiento tales como la medición en la latencia del administrador y validar su modelo

con un conjunto de mediciones de tiempo, se encontraron con que Hadoop YARN se desempeñó peor que los otros tres.

Por lo que tanto por los resultados arrojados por este estudio como por el hecho de que Mesos proviene del mismo grupo de estudio que Spark, se establece que Apache Mesos es la opción óptima para ser el administrador de recursos del clúster BD para el framework propuesto. No obstante, para efectos del estudio llevado a cabo las tareas de Spark se llevaron a cabo mediante el modo Local, debido a que se careció de una infraestructura de hardware en la que se pudieran probar soluciones alternativas. Por otro lado, cabe mencionar que esto no afectó los resultados finales del estudio debido a que el dataset de entrenamiento y prueba cabe perfectamente en la memoria del ordenador empleado.

4.3.3. Acceso a Datos

Aspectos Teóricos

En las etapas anteriores se definieron los componentes primarios de la segunda fase del modelo para contar con una infraestructura de cómputo capaz de dar respuesta a altas necesidades de procesamiento al contar con un clúster de computadoras que pueda dar respuesta a tales demandas. Al contar con estos elementos se procede a definir cómo se va a realizar el acceso a los datos que se cargaran posteriormente en los nodos. En este caso, esto está definido por consultas tipo NoSQL las cuales están diseñadas para tratar con GVDNE. Sin embargo, es válido emplear también SQL si los datos se encuentran estructurados.

Esta etapa sienta las bases de haber definido qué tipo de unión entre consultas tipo NoSQL o SQL son empleadas y el lenguaje de programación que se emplea para ejecutar las tareas de consulta y recuperación de datos. La salida de esta etapa son los datos en el formato interno que se maneje por parte del DFS los cuáles son sometidos a técnicas de análisis predictivo en la siguiente etapa.

Aspectos de Implementación

Spark cuenta con Spark SQL como la interfaz que permite acceder a datos tanto estructurados como semiestructurados. Spark es descrita por sus autores (Armbrust et al., 2015) como “un modelo que evolucionó a partir de su esfuerzo anterior *Shark*, brindando a los usuarios una API tanto procedural como relacional, permitiéndoles mezclar ambas interfaces”. Otra de las ventajas de Spark SQL es que permite la ejecución de algoritmos de ML gracias a su integración con la librería ML de Spark. Spark SQL ha demostrado

un mejor rendimiento en la ejecución de las consultas comparado contra otros motores de BD como Hive según se explica en (Talha & Kara, 2016).

En particular Spark SQL ofrece tres capacidades:

1. Puede cargar datos de una variedad de fuentes como JSON, Hive y Parquet.
2. Permite consultar los datos empleando SQL tanto dentro de un programa nativo Spark como desde fuentes externas que se conectan a través de conectores (JDBC/ODBC).
3. Spark SQL provee alta integración con código Java, Scala o Python.

En lo concerniente al estudio, el acceso a los datos fue llevado a cabo mediante Pyspark, (una variante de Python adaptada para su uso en Spark) así como NoSQL.

4.3.4. Análisis de datos

Aspectos Teóricos

La etapa anterior generó como salida el conjunto de datos que se tratan en esta etapa de ML, es decir, que provee en su estructura interna la información a la cual se aplican técnicas de análisis de datos mediante ML. Las herramientas de BD cuentan con métodos de ML que pueden ejecutar la mayoría de los algoritmos ML de forma paralela a través de los clusters. Sin embargo, existe una limitante sobre este conjunto de algoritmos ML y es que solamente algunos de estos algoritmos pueden ejecutarse sobre la arquitectura nativa de BD (Shirdastian, Laroche, & Richard, 2017). Para solventar esto se pueden ejecutar en los nodos del clúster los algoritmos ML que no estén cubiertos por la tecnología adoptada mediante la ejecución local en cada nodo de una subtarea que lleve a cabo las tareas particulares de un proceso de ML.

De tal manera que esta etapa a su vez se sirve de dos sub-etapas: una que emplea técnicas de PLN y otra de ML. Los algoritmos de ML utilizan técnicas de PLN para mejorar el proceso de preparación del conjunto de datos disponible. De tal manera que estas técnicas tienen que contribuir a mejorar la eficacia del método a construir para que de esta manera se obtengan los mejores resultados posibles. Se subraya el hecho de que no solamente mediante técnicas de ML y PLN es que se puede realizar el análisis de los datos, también existen un conjunto de sistemas estadísticos que han demostrado ampliamente su gran efectividad no sólo en la comunidad científica de estadística, sino en diversas áreas de la ciencia. Por lo que resolver el mismo problema empleando la estadística en esta etapa, dará lugar en los trabajos a futuro a soluciones similares.

Los enfoques que existen en el estado del arte tratando el tema de la predicción del turismo son diversos, uno de estos tiene que ver con analizar las consultas de búsqueda. En (Li, Pan, Law, & Huang, 2017) se estableció un framework y un procedimiento para crear un índice de búsqueda compuesto adoptado en un modelo de factor dinámico generalizado. Dicho estudio ahonda en el área de los motores de búsqueda para poder establecer predicciones a partir de las búsquedas más frecuentes.

Siguiendo esta línea de investigación se tiene el trabajo de Pan & Yang (2017) los cuáles también trabajan con datos provenientes de motores de búsqueda y datos del tráfico de los sitios webs relacionados precisamente con esos destinos para pronosticar la ocupación semanal de los hoteles en una ubicación particular. En este caso se emplearon varios modelos de series de tiempo que incorporaron fuentes de BD al ser los datos de diversa naturaleza.

Mientras que Dali & Yutaka (2015) emplearon algoritmos de ML como SVR y SVM para predecir el comportamiento de turistas de China que desean viajar a Japón. Los alcances obtenidos en esta investigación son de relevancia al haber encontrado aquellos aspectos que son más comentados en un foro de discusión sobre viajes. No obstante, los autores señalan que su trabajo tiene aún diversas limitaciones debido entre otros aspectos, al origen de datos con que trabajan y la escasez de métodos empleados.

Dentro de los trabajos que han empleado BD aplicado al sector del turismo se encuentra el realizado por Marine-Roig (2017) en el cual se propone un método basado en tecnologías de BD para analizar y medir la imagen del destino turístico a partir de la información contenida en los sitios web de reseñas de turismo. No obstante que el método desarrollado no está relacionado con ninguna técnica de ML, se destaca el hecho de que a partir del análisis de diversas fuentes de información se obtienen indicadores numéricos sobre los sentimientos de los visitantes a la zona de Île de France con los cuáles se consiguen importantes hallazgos sobre la imagen que tienen los turistas de los principales sitios de interés de esta zona de Paris.

El valor que revisten las críticas en medios digitales para los hoteles ha cobrado cada vez mayor importancia, los resultados del estudio llevado a cabo por Xie, Zhang, & Zhang (2014) demuestran que la calificación general, las calificaciones de atributo del valor de compra, la ubicación, la limpieza, la variedad y cantidad de las críticas recibidas, y el número de respuestas por parte de la gerencia, son factores que en general están asociados con el desempeño del hotel. La pregunta de investigación establecida en este trabajo: ¿cuál es el efecto de los comentarios del cliente en el desempeño del hotel? Es de particular interés en esta parte de la investigación al ser un trabajo que reúne datos de dos fuentes, una similar al dataset Yelp (TripAdvisor) y la otra directa, de una oficina local de ventas.

En varias investigaciones se ha abordado el problema de predecir el comportamiento del turista a partir de datos generados de redes sociales. Como el trabajo que desarrollaron Pantano, Priporas, & Stylos (2017) en el cual se analizaron datos de Tripadvisor para predecir la respuesta de los turistas a una atracción turística. Una desventaja de este trabajo es que no se aplican técnicas de PLN o que hagan un análisis del sentimiento. De acuerdo a Yüksel (2007) el realizar predicciones en la industria hotelera es sensitivo a las fluctuaciones en la demanda debido a la estructura del negocio. Según este autor, la hotelería es un sector que presenta escenarios de crisis comúnmente debido a las fluctuaciones que existen en la demanda de los servicios.

En el estudio llevado a cabo por Cankurt & Subasi (2015) se emplearon métodos de ML como RNA y Regresión de vectores de soporte para desarrollar modelos que contribuyan a la predicción de la demanda del turismo. Cabe destacar que este tipo de estudios involucra la toma de datos numéricos, en particular diversas series de tiempo con información de diversas variables sobre el comportamiento de los turistas en el período comprendido entre 1996 y 2013 que visitaron Turquía.

Tradicionalmente se han empleado modelos de series de tiempo para predecir y calcular las llegadas a los destinos turísticos según Claveria & Torra (2014) los cuales presentan un estudio en el que se compara la efectividad de estos modelos contra las RNAs. En su investigación se explora la efectividad que tiene realizar un estudio de predicción de la demanda del turismo empleando información de las llegadas mensuales y pernoctaciones a/en Cataluña. En sus resultados demuestran que el empleo de técnicas de series de tiempo sobrepasa ligeramente aquellos métodos que emplearon Redes Neuronales Artificiales. No obstante, se menciona que esto es relativo ya que dichas comparaciones se encuentran en función de la información disponible y de cómo se haya armado la Red Neuronal.

Procesamiento de Lenguaje Natural - Aspectos de Implementación

Como se ha comentado en el apartado 4.2.4, las técnicas de PLN se emplean para preprocesar el corpus de datos y transformar este hacia un estado que pueda ser empleado por los clasificadores de ML, es decir, hacia una matriz que contenga los documentos y features más importantes del corpus (como se trató a detalle en la primera fase). De forma similar a lo planteado en la primera fase, las técnicas de PLN se emplean como un medio en lugar de perseguir un fin *per se*; es decir, no se plantean nuevos avances en las áreas de NER o POS debido a que esto no es un objetivo a lograr ni por parte del framework, ni de la tesis; al respecto existe investigación en desarrollo que entra a profundidad en estas áreas.

En lo que respecta a los aspectos técnicos de esta parte del modelo, se emplearon las librerías de PySpark.ML, dado que son las que permiten llevar a cabo las tareas de

preprocesamiento del texto. No obstante, se cuenta con herramientas como Stanford CoreNLP que pueden ser incorporadas en esta plataforma para ejecutar tareas más específicas de PLN (Manning, Bauer, Finkel, & Bethard, 2014).

Machine Learning - Aspectos de Implementación

En la primera fase del modelo se abordó este tema desde un enfoque en el que se estaba en la búsqueda de los algoritmos que contribuyeran al análisis de la información en un entorno distinto a GVDNE. En esta fase se emplea la librería de Spark para tareas de ML: Pyspark.ML, la cual soporta una gran variedad de algoritmos de ML incluyendo MLP, LR, LSVC y MNB. Existe una segunda librería de Spark para realizar tareas de ML: MLlib, sin embargo esta no fue empleada durante las pruebas debido a que esta librería no aporta los métodos necesarios para evaluar los clasificadores que sí se tienen en Pyspark.ML, entre los que se tienen: un conjunto de métodos que permiten evaluar, medir y orquestar (*pipeline*) las actividades de ML. Spark ML ha demostrado ser más veloz que otros motores de procesamiento distribuido, como Flink (García-Gil, Ramírez-Gallego, García, & Herrera, 2017).

En el apartado 4.2.4 se explicó el proceso de cómo se convirtió el corpus existente a su representación matricial o BOW; dicho proceso corresponde a la implementación efectuada mediante el paquete Scikit-learn como se indica en la Tabla 4.9. Sin embargo en Spark ML este proceso es diferente al no contar con un método propio que haga el proceso de TF-IDF, en su lugar, se cuenta con un método para realizar TF (HashingTF) y otro método para IDF. Durante la adaptación del modelo de la primera etapa a esta se buscó que el vectorizador TF-IDF conservara las mismas características con el fin de que los clasificadores trabajaran con la misma BOW, no obstante, la versión actual de Spark ML impuso ciertas restricciones.

Por una parte, el método IDF de Spark ML cuenta con el parámetro min_df, pero no con el parámetro max_df, con lo cual no se limitan los términos que aparecen con demasiada frecuencia; no obstante, se eliminan las palabras vacías del corpus a través de otro parámetro. Otra función no encontrada en Spark ML es la construcción de n-gramas respecto a que en la implementación actual se obtienen sólo aquellos que se indiquen, ejemplo n=2 solamente regresa elementos conteniendo 2 palabras, en lugar de regresar 1-gramas y 2-gramas; lo cual se tuvo que solucionar localmente en el modelo. El resto de los parámetros en la Tabla 4.9 se preservaron en el modelo de la segunda fase.

Los clasificadores utilizados para las pruebas fueron: Perceptrón multicapa (MLP), Regresión logística (LR), Clasificador lineal de vectores de soporte (LSVC) y Naïve Bayes Multinomial (MNB). En lo que respecta al Clasificador de vectores de soporte – C (SVC) y el Clasificador lineal con entrenamiento de Gradiente estocástico (SVM-SGD),

estos no se emplearon durante las pruebas debido a cuestiones del paquete Spark ML que se explican en el Capítulo 5 en los apartados 5.5.3 y 5.5.6.

4.3.5. Visualización

Aspectos Teóricos

La etapa anterior genera como salida una serie de valores numéricos, los cuales deben tener algún valor o significado en función del estudio que se esté llevando a cabo. Para representar estos valores es necesario recurrir a técnicas de visualización de GVDNE que permitan el análisis de los datos a gran escala de una forma ágil y comprensible. La salida a generar son una serie de métodos creados para dar respuesta a las solicitudes de visualización de datos, es decir, que sirven de enlace entre técnicas que permiten ver los GVDNE representados de forma gráfica después de haber aplicado el procesamiento de las etapas anteriores, y las solicitudes de tal información visual, las cuáles pueden incluso provenir de interfaces de usuario externas.

Los analistas de datos necesitan de técnicas mediante las cuales se pueda difundir los descubrimientos que se obtuvieron a partir del valor oculto en los datos. Otro valor buscado en este aspecto es la manipulación de los datos a través de la variación de los datos de entrada y la posterior observación de cómo se comportan estos elementos gráficos. La visualización de datos no sólo sirve como una herramienta exploratoria para la comprensión de los mismos y obtener nuevos conocimientos, sino que además puede ser empleada como una herramienta de presentación con los propósitos de ilustración, explicación y comunicación de los resultados (Cybulski et al., 2015).

Esta etapa es la última tanto de la segunda fase, como de todo el framework, con lo cual se concluyen las actividades iniciadas desde la primera fase. Aunque en otros modelos propuestos en el estado del arte la visualización puede ser un paso intermedio, uno alterno o incluso no considerarlo del todo.

Aspectos de Implementación

Las herramientas de visualización de datos que existen actualmente para la representación de GVDNE son diversas. Dentro de estas, existen de hecho algunas herramientas de visualización que estas orientadas a determinadas áreas científicas las cuáles no se cubren en este trabajo al escapar del enfoque de estudio. Por lo que las herramientas de visualización de datos que se cubren en este apartado son aquellas que han sido particularmente adaptadas para dar respuesta a las necesidades de análisis de BD.

Diversos estudios se han llevado a cabo para establecer las características de estas herramientas de visualización, las cuales de hecho ya existían previo a la llegada del BD como lo establecen Zhang et al. (2012) quienes realizaron un estudio exploratorio sobre las prestaciones que las herramientas existentes podrían dar soporte a la representación gráfica de GVDNE. Por su parte, Pääkkönen & Pakkala (2015) presentaron un estudio sobre las herramientas que eran empleadas para BD, y en el aspecto de las herramientas de visualización de datos reportaron los siguientes productos: Karmasphere, Datameer, Platfora, Qlikview, SAS Visual Analytics, Streambase y Tableau.

Los productos anteriormente citados fueron analizados tomando como parámetros: las capacidades de su interfaz gráfica, el ambiente de ejecución, las fuentes de datos, la capacidad de análisis de datos y las interfaces externas de estadística. Aunque todas estas herramientas han sido empleadas por éxito por diversos equipos de análisis de datos, destacan Tableau y Qlik por su mayor penetración en el ambiente científico, así como por la versatilidad de estas herramientas para poder adaptarse a ambientes de GVDNE y permitir la integración tanto de fuentes estructuradas como no estructuradas.

4.4. Despliegue del modelo

La realización de un proyecto de ciencia de datos normalmente involucra actividades como (Habib et al., 2016; Olmedilla et al., 2016):

1. El acopio de la información.
2. La preparación y limpieza de los datos.
3. La búsqueda de los clasificadores más adecuados de acuerdo al tipo de análisis a realizar y el tipo de información disponible.
4. Construir el modelo de ML.
5. Analizar la información mediante el modelo de ML y obtener predicciones en base a este análisis.
6. Presentar los resultados.

Como se ha presentado a lo largo de este capítulo, las actividades enlistadas anteriormente se encuentran presentes en el framework propuesto, sin embargo, falta una actividad que es requerida por los equipos de producción de proyectos de este tipo: el despliegue del modelo. Ryza, Laserson, Owen, & Wills (2015) mencionan que si bien tener un MLM es útil, resolver un problema de ML del mundo real requiere más que correr un algoritmo, para lo cual hacen referencia de la necesidad de hacer el despliegue del modelo. El despliegue es referido como el método mediante el cual los equipos de desarrollo usan el MLM para aplicaciones generalmente comerciales, aunque también pueden ser para fines científicos o académicos.

Dichas aplicaciones pueden ser empleadas en dispositivos móviles, páginas web o software de escritorio, aunque con la evolución del IoT también pueden ser presentadas o mostradas en artefactos (gadgets) diversos. En lo que respecta a esta tesis se aborda esta actividad desde dos perspectivas tecnológicas distintas: un caso de uso general y el particular que compete a la tesis. En el primer caso, existen diversas soluciones tecnológicas que permiten realizar el despliegue de MLMs como: Anaconda Enterprise, IBM Watson Data Platform, Flask o Heroku. Cada uno de los mencionados ofrece diversas modalidades de despliegue: en un producto se pueden tener funcionalidades más visuales, mientras que otro puede requerir más la generación de código. Cabe mencionar que Anaconda es una distribución gratuita de código abierto de Python y R para propósitos de cómputo científico, comúnmente empleada en proyectos de ciencia de datos.

Por otro lado, todos los medios referidos ofrecen la posibilidad de invocar el MLM mediante el paso de parámetros para obtener el porcentaje de predicción de un caso particular. Mientras que otros pueden incluso acceder a la visualización generada desde el modelo. El despliegue también puede ser "montado" sobre contenedores para que el equipo de desarrollo lo integre en sus contenedores ya creados y genere una invocación directa del servicio (del MLM) para que puedan mostrar sus resultados al usuario desde donde esté realizando la invocación del servicio. Ryza et al. (2015) también enumeran una serie de tareas a tener en cuenta al momento de hacer el despliegue de un MLM.

En el ámbito referido, un contenedor es un método de virtualización de un servidor en el cual el núcleo de un sistema operativo gestiona la existencia de diversas instancias aisladas de espacios donde pueden ejecutarse diversos procesos del usuario. Mientras que Docker se refiere a uno de estos ejemplos de contenedores de software, el cual realiza las funciones referidas anteriormente, así como automatizar el despliegue de aplicaciones.

En lo que respecta al despliegue individual que se daría al MLM construido: en primer lugar, al haber sido construido sobre la distribución de Anaconda, se emplearía entonces la versión empresarial de esta plataforma para generar el despliegue en contenedores de Docker. Luego, se estima que la parte del proyecto donde se obtiene una predicción sobre el puntaje de sentimiento de los datos pueda ser empleada del lado de la empresa, considerando que necesitan saber en todo momento el desempeño actual de la unidad de negocio, por lo que esto podría ser integrado como parte de una plataforma digital donde se invocaría al contenedor descrito.

Mientras que la funcionalidad donde se obtienen las palabras (features*)* que describen los términos que valoran más negativa y positivamente los servicios recibidos, podrían ser utilizados tanto por usuarios de los hoteles, como por los empresarios. Por lo que, visto desde el caso de uso de los usuarios, podría construirse una aplicación móvil y un

sistema Web donde al buscar un hotel específico, se obtuvieran los términos que la gente está valorando como lo más y menos favorable de ese negocio. En cualquier caso, la llamada al servicio se realizaría de igual manera a través del contenedor Docker que a su vez se serviría del despliegue generado por Anaconda.

4.5. Sumario

El objetivo de este capítulo ha sido presentar las diversas etapas que componen el framework justificando cada una de estas desde una perspectiva teórica, para comprender su importancia y, en su caso, los trabajos relacionados en el estado del arte. De igual manera, en cada apartado se habla de la forma de llevar a cabo una posible implementación de las ideas que fueron indicadas de forma teórica, pero dentro de un dominio concreto. Esto da como resultado que se pueda evaluar la calidad del framework y así validar las hipótesis de partida. Se ha estructurado el capítulo de tal manera que se presentaron primero las etapas que comprenden la primera fase para posteriormente efectuar lo mismo con las etapas de la segunda fase. Como puede apreciarse, los elementos que componen la primera fase dan un soporte a la segunda fase en el sentido de que esta realiza un estudio exploratorio inicial sobre el estado de los datos.

Los aspectos más relevantes del framework son: *(i)* es un modelo de dos fases: la primera se centra principalmente en preparar los datos y encontrar el MLM más adecuado para analizar los datos; la segunda se refiere a la creación de una infraestructura de BD capaz de realizar tareas de análisis y visualización; *(ii)* las etapas en la primera fase se han reducido para obtener resultados en el menor tiempo posible al reducir la complejidad, lo que permitirá incluir nuevos dominios y algoritmos de ML; y *(iii)* el MLM no está relacionado con determinados algoritmos de ML, lo cual aumenta la flexibilidad de utilizar el framework.

La primera fase se puede llevar a cabo en un entorno no BD y, a diferencia de otros modelos, se resalta que se debe crear un modelo conceptual antes de llevar a cabo el análisis. Por otra parte, en el Apartado 4.2.1 se discutió acerca de haber trabajado directamente con datos obtenidos de Yelp; por lo que en la investigación futura se puede considerar obtener información de otras fuentes, así como variar las técnicas de recolección empleadas para este fin. En el siguiente capítulo se efectúa la evaluación y validación del framework mediante la comparación de los resultados obtenidos al final de la primera fase contra aquellos obtenidos al final de la segunda.

Capítulo 5. Evaluación y Validación

En virtud de que se ha presentado el framework para el análisis predictivo de datos no estructurados, habiéndose efectuado tanto su justificación teórica, metodológica y procedimental, queda como objetivo de este capítulo el evaluar el modelo propuesto empleando los datos relativos al turismo descritos en el apartado 4.2.1.

En primera instancia, se describe el diseño de experimentos a realizar con el cual se pretende evaluar de forma cuantitativa las dos fases del framework. En la primera fase del framework se evalúa el rendimiento del modelo mediante el análisis predictivo de los datos, empleando tareas de clasificación binaria para la prueba de un conjunto de clasificadores de machine learning seleccionados para acometer esta tarea.

Para llevar a cabo los experimentos en la segunda fase del framework se lleva a cabo un conjunto de pruebas similar al de la primera fase teniendo en cuenta la capacidad del entorno de big data. En esta fase se ejecutan más pruebas pero con un conjunto menor de clasificadores lo cual se explica en los apartados correspondientes. Por otra parte, tanto en la primera como en la segunda fase se valida el modelo mediante el análisis de las medidas de calidad de las alternativas empleadas.

Finalmente, se presenta el sumario del capítulo donde se resumen los resultados que se encontraron en ambas fases del framework y cómo estos comprueban las hipótesis planteadas al inicio de la investigación.

5.1. Introducción

El framework pretende ser una base de referencia para analizar datos no estructurados tanto de menor como de mayor volumen. La comprobación del modelo se realiza mediante el análisis de información del dominio turístico, pero el framework también puede ser empleado para analizar conjuntos de información que pertenezcan a otros dominios, que además tengan características similares, es decir, se cuente con un conjunto datos cualitativos y cuantitativos que se encuentren intrínsecamente relacionados.

El capítulo inicia con una descripción del análisis predictivo, en virtud de que este es el tipo de análisis que se lleva a cabo en el framework, estableciendo también una diferencia con el pronóstico o el establecimiento de pronósticos que es un tipo de estudios diferente al presentado en la tesis y que es mencionado en el Capítulo 6. Seguido de este

apartado se presenta el diseño de los experimentos mediante el cual se lleva a cabo la evaluación del framework para las dos fases del mismo, el cual establece las bases sobre las que se sustentan las pruebas desarrolladas.

Posteriormente, se presentan las pruebas que se llevan a cabo en la primera fase del framework, para lo cual se emplean seis clasificadores de ML para realizar la tarea de clasificación binaria del texto en función del etiquetado de los datos introducido en el apartado 4.2.4; en el Apartado 5.3.5 se presentan los algoritmos de ML empleados para realizar las pruebas y los motivos por los cuáles estos fueron seleccionados. Se presenta cómo se preprocesó el texto para poder ser empleado por los clasificadores, así como la comparativa de los resultados obtenidos por los clasificadores.

En el caso de los experimentos llevados a cabo en la segunda fase, se toma como base el modelo construido en la primera, donde se lleva a cabo el preprocesamiento de la información; así como la propuesta de una serie de clasificadores de ML mediante los cuales se realiza la tarea de clasificación binaria de la información. En esta fase se emplearon los siguientes clasificadores: Perceptrón multicapa (MLP), Regresión logística (LR), Clasificador lineal de vectores de soporte (LSVC) y Naïve Bayes Multinomial (MNB). Se emplean dos clasificadores menos que en la primera fase lo cual se debe a cuestiones técnicas del entorno de BD empleado lo cual se explica en los apartados 5.5.3 y 5.5.6. En ambas fases se analizan estadísticamente los resultados para validar el modelo presentado.

Como parte de la última etapa del modelo se presenta la etapa de visualización donde se representa de forma gráfica lo que se ha referido anteriormente como la información oculta en los datos que es factible obtener a través del empleo conjunto de las técnicas de BD, ML y PLN. Finalmente, en el sumario, se realiza el contraste entre las hipótesis planteadas al inicio de la investigación y cómo estas han sido comprobadas mediante los trabajos elaborados y presentados en este capítulo.

5.2. El análisis predictivo

En primera instancia se explora el concepto de análisis predictivo para enmarcarlo dentro del contexto de esta tesis y en particular en lo que refiere al framework. Según Siegel (2013) el análisis predictivo es “Tecnología que aprende de la experiencia (datos) para predecir el comportamiento futuro de las personas para poder tomar mejores decisiones”. Mientras que Finlay (2014) explica que a mediados de los 2000s se empleó este término a la par con el de ‘minería de datos’ para desarrollar herramientas para predecir el comportamiento de individuos o entidades.

Sin embargo, hay que entender bajo el concepto del análisis predictivo cuál es la diferencia entre predecir y pronosticar. Como el mismo Siegel lo indica: el análisis predictivo es algo completamente diferente de pronosticar ya que este último "hace predicciones agregadas en un nivel macroscópico". Visto de otra manera, el pronóstico habla de establecer tendencias futuras después de haber analizado datos presentes y futuros, mientras que el análisis predictivo se detiene más en realizar una serie de observaciones hechas sobre un conjunto de datos y seleccionar algunos de estos elementos basados en sus características a raíz de haber llevado a cabo el análisis.

Visto de otra manera con un claro ejemplo práctico: tomando en cuenta un muestreo de un grupo de consumidores de comida en Valencia, el análisis predictivo establecería quienes de estos individuos son los que con un cierto porcentaje de confianza comprarán paella; mientras que el pronóstico podría establecer indicadores como: ¿Cuántos kilos de paella se consumirán el próximo mes en Valencia? o ¿Cuántos euros gastará cada individuo en paella en un periodo determinado? Como se observa, el pronóstico y la predicción son conceptos totalmente distintos, cada uno enfocado en conseguir un resultado en función del estudio que se desee llevar a cabo.

De tal manera que el modelo de análisis predictivo se evalúa en función del porcentaje de acierto en la predicción que se haya llevado a cabo. Por otra parte, Finlay establece que los tipos de datos de texto son empleados para predecir el comportamiento de la unidad analizada mediante la aplicación de técnicas de análisis de sentimiento. Este autor también hace una distinción entre las aplicaciones de los modelos predictivos, resumiéndolas a dos: problemas de clasificación y los relacionados a obtener cantidades. En el presente capítulo se abordan los problemas de clasificación de un conjunto de datos en específico y cómo estos fueron resueltos a través de la utilización del framework.

5.3. Diseño de los experimentos

Con el fin de poder probar las hipótesis de partida establecidas en el Capítulo 1, se tiene que comprobar lo formulado por el framework planteado en el capítulo anterior. Para lo cual se realizaron una serie de experimentos que se explican a continuación.

5.3.1. Experimentos de la primera fase

Los pasos a seguir son los siguientes:

1. Se toma como entrada del modelo los datos que fueron procesados previamente y que han sido etiquetados como positivos (1) o negativos (0) para la

clasificación binaria.

2. En el entorno no BD se prueban seis clasificadores de ML haciendo variaciones en los parámetros de cada clasificador buscando encontrar la configuración óptima que obtenga los mejores resultados en cuanto a la predicción, de acuerdo a un conjunto de trabajos donde los autores hayan empleado cada algoritmo para la clasificación de textos.
3. Se efectúan diez pruebas por cada clasificador mediante la modificación de los parámetros de prueba. Las muestras de datos se seleccionaron mediante el método de validación cruzada, habiendo realizado en primera instancia una selección de k=10 particiones (10*10=100 ejecuciones en total por clasificador). Con este número de pruebas se pudo llevar a cabo la validación estadística que permitió establecer una comparación significativa sobre los resultados.
4. Por cada prueba con determinados parámetros se calcula el promedio de exactitud en la predicción (accuracy), mínimo de exactitud (MIN), máximo de exactitud (MAX), tiempo en segundos de la prueba y el promedio del error cuadrático medio (MSE); tomado este último de cada una de las pruebas.
5. Por cada clasificador se selecciona un conjunto de parámetros tomando en cuenta que con dichos parámetros se haya obtenido el valor más alto en cuanto al promedio de exactitud en la predicción obtenido en alguna de las diez pruebas realizadas.
6. Con los parámetros seleccionados en el punto anterior, se repiten las pruebas mediante validación cruzada ahora haciendo particiones con k=3 y k=5.
7. Se comparan los mejores resultados obtenidos por cada clasificador tomando en cuenta el porcentaje de predicción y el tiempo de ejecución.

En la Figura 5.1 se representa el anterior proceso de forma gráfica.

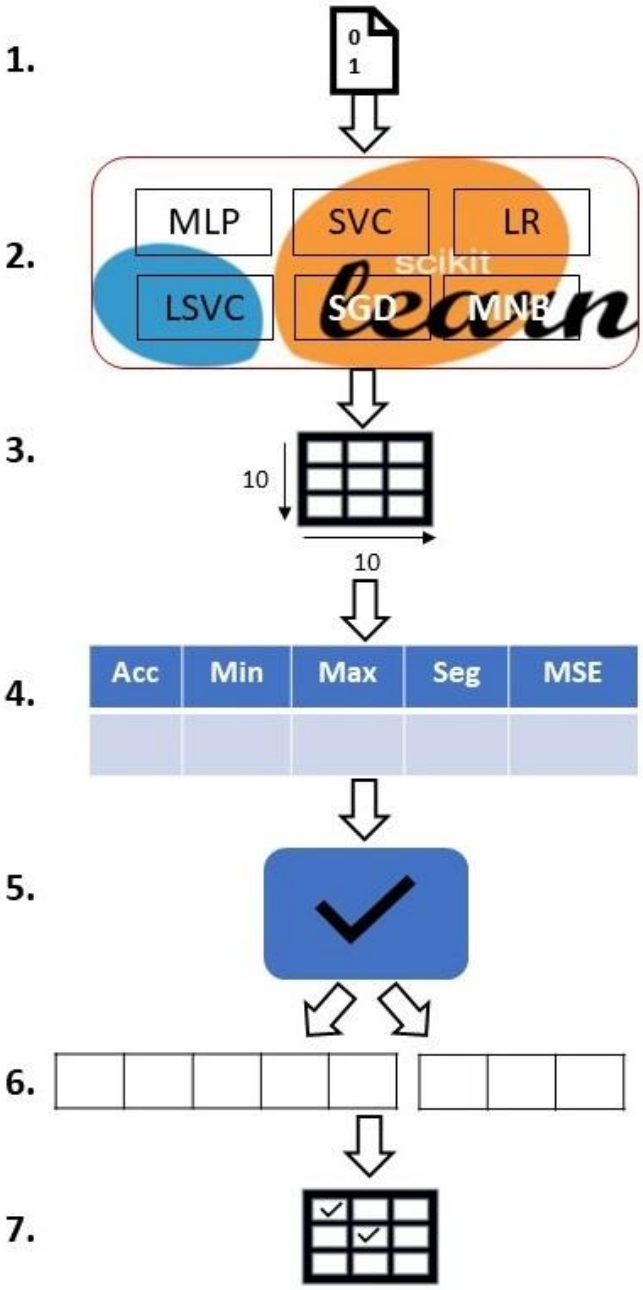

Figura 5.1 Representación del proceso de experimentación en la primera fase

5.3.2. Experimentos de la segunda fase

Los pasos a seguir son los siguientes:

1. Se toma como entrada del modelo los datos que fueron procesados previamente y que han sido etiquetados como positivos (1) o negativos (0) para la clasificación binaria.

2. En el entorno BD se prueban cuatro clasificadores de ML variando los parámetros de cada clasificador para encontrar la configuración óptima que obtenga los mejores resultados en cuanto a la predicción, lo anterior debe estar en concordancia con los experimentos hechos en la primera fase.
3. Se efectúan 15 pruebas por cada clasificador mediante la modificación de los parámetros de prueba. Las muestras de datos se seleccionaron mediante el método de validación cruzada, habiendo realizado en primera instancia una selección de k=10 particiones (15*10=150 ejecuciones en total por clasificador). Se aumentó en cinco el número de pruebas para que la validación estadística permitiera establecer una comparación significativa sobre los resultados.
4. Por cada prueba con determinados parámetros se calcula el promedio de exactitud en la predicción (accuracy), tiempo en segundos de la prueba y el error cuadrático medio (MSE); este último se toma a partir de la predicción hecha por el clasificador, dado que no se pueden obtener los MSE de cada prueba de forma individual. El mínimo y máximo de exactitud no se obtienen debido a que no son métricas que se puedan obtener, según la documentación del entorno tecnológico empleado.
5. Por cada clasificador se selecciona un conjunto de parámetros tomando en cuenta que con dichos parámetros se haya obtenido el valor más alto en cuanto al promedio de exactitud en la predicción obtenido en alguna de las quince pruebas realizadas.
6. Con los parámetros seleccionados en el punto anterior, se repiten las pruebas mediante validación cruzada ahora haciendo particiones con k=3 y k=5.
7. Se comparan los mejores resultados obtenidos por cada clasificador tomando en cuenta el porcentaje de predicción y el tiempo de ejecución.

En la Figura 5.2 se representa el anterior proceso de forma gráfica.

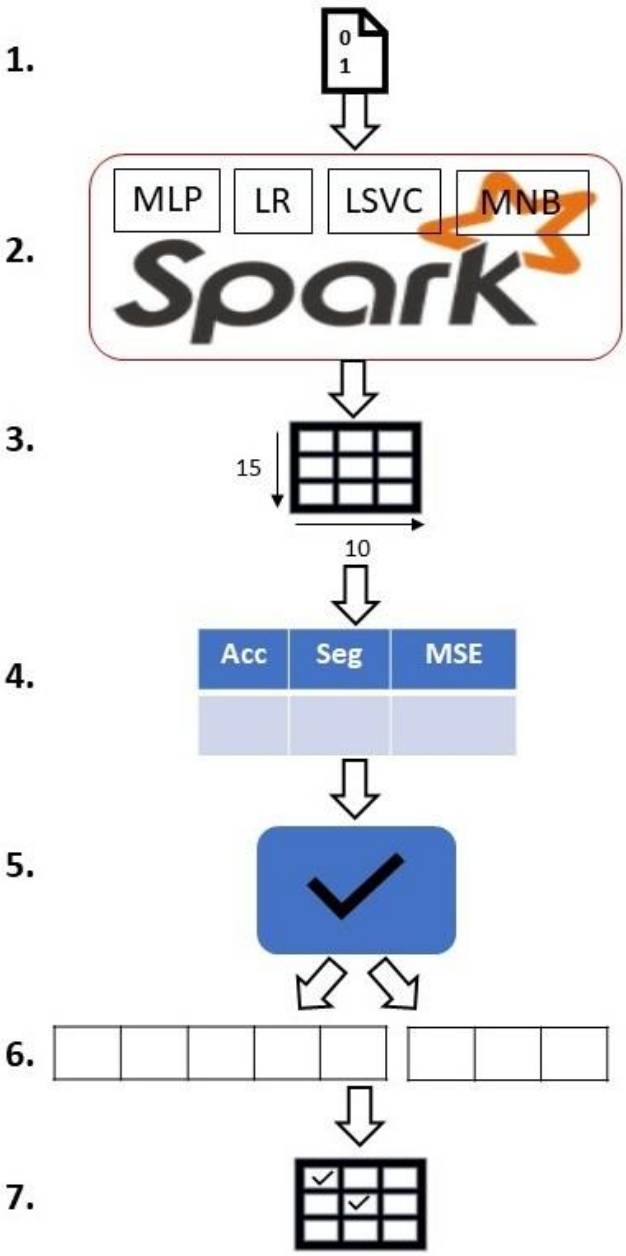

Figura 5.2 Representación del proceso de experimentación en la segunda fase

5.3.3. Métodos de validación cruzada empleados

Respecto a las particiones (folds) que se tomaron para hacer la división de los datos en las diversas pruebas se siguieron los procesos de experimentación llevados a cabo por otros autores en el estado del arte, todos referidos a estudios que tuvieron como tarea principal o secundaria la clasificación de textos, los cuáles se refieren a continuación:

- Se emplea el enfoque de 10 folds como en la investigaciones de Feng, Guo,

Jing, & Sun; Salles, Gonçalves, Rodrigues, & Rocha; y Tutkan, Can, & Akyokus (2015; 2018; 2016).

- En el caso de 5 folds, se sigue esta metodología en base a trabajos como los realizados por Jiang, Li, Wang, & Zhang; Liu, Liu, & Huang; y Silva, Almeida, & Yamakami (2016; 2017; 2017).
- Como ejemplo de estudios que emplean 3 folds se tienen los de Kang, Ahn, & Lee; Rocha et al.; y dos estudios de Zhang, Yoshida, & Tang (2018; 2013; 2008, 2011).

5.3.4. Relación con las hipótesis

El llevar a cabo los experimentos descritos anteriormente tiene como finalidad la validación de las hipótesis planteadas al inicio de la presente investigación. Por lo que se reproducen a continuación nuevamente las hipótesis planteadas en el apartado 1.5 así como una justificación del procedimiento con el cual se pretende validarlas.

1. **Es posible definir un framework en el que se puedan aplicar diversos indicadores de estudio independientemente del volumen de datos a analizar.** Como se describe en los experimentos a realizar en ambas fases, se emplean diversas métricas para medir la eficacia en cuanto a la predicción de los algoritmos de ML durante la fase uno. Por lo que se espera medir con un conjunto similar de métricas los resultados obtenidos en la fase dos que comprenden técnicas aplicables a GVDNE.
2. **Las técnicas de análisis de aprendizaje automático propuestas por el framework detectan patrones que permiten establecer predicciones sobre los datos de cualquier volumen.** Al haber empleado técnicas de ML en las dos fases del modelo, se obtienen porcentajes de predicción sobre los datos de cualquier escala, los cuáles serán posteriormente validados al comparar el desempeño de diversos clasificadores.
3. **Los modelos de aprendizaje automático definidos para trabajar en un entorno con un volumen reducido de información se pueden extrapolar a un entorno GVDNE para realizar el mismo tipo de análisis.** Se emplea el MLM evaluado en la primera fase, para aplicarlo en la segunda, empleando técnicas de BD con el objetivo de efectuar el mismo análisis sobre los datos.

En los apartados posteriores se hace nuevamente referencia a las hipótesis de manera particular para resaltar con qué parte de los experimentos y/o resultados se está validando cada una de dichas hipótesis. Cabe resaltar también que la validación de las hipótesis se sucede en diversas partes de la metodología que se está siguiendo de acuerdo al diseño de los experimentos de ambas fases, por lo que se resalta la validación cuando se ha concluido una secuencia ordenada de estos pasos que resulta en la validación de cada

hipótesis de una forma más general. Las hipótesis no están validadas de forma secuencial, por lo que las referencias a estas se suceden de forma alterna.

5.3.5. Clasificadores empleados y estudios relacionados

En la experimentación que tuvo lugar previamente a las pruebas de la primera fase, se realizó una investigación tanto en el estado de la técnica como en el estado del arte sobre los clasificadores empleados con más frecuencia en tareas de clasificación de textos. A partir de los resultados de esta investigación, se realizaron pruebas con un conjunto de diversos clasificadores de ML para evaluar de forma general los resultados que se obtenían al emplear el dataset original que contenía el mismo número de instancias para cada una de las cinco clases de reviews. A medida que se fueron obteniendo los primeros resultados de los porcentajes de acierto en este conjunto de clasificadores, se fueron descartando aquellos que habían obtenido los menores porcentajes, hasta considerar un total de seis clasificadores. A continuación, se presentan algunos clasificadores similares a los empleados, que se han empleado por otros autores en el estado del arte.

La investigación existente sobre la tarea de clasificación de textos empleando diversos algoritmos permite enmarcar el estudio realizado en esta investigación en su relación con otros trabajos sobre el área, a continuación se citan sólo algunos de estos a manera de complementar lo descrito en el Apartado 2.4. En primera instancia, Dhanalakshmi, Bino, & Saravanan (2016) presentan un estudio en el que llevan cabo minería de datos de opiniones de los alumnos para determinar la polaridad del sentimiento. En este caso los autores emplean los siguientes algoritmos: SVM, Naïve Bayes, K-Nearest Neighbor y RNA, para efectuar la tarea de clasificación.

En la investigación de Pranckevicius & Marcinkevičius (2017) se lleva a cabo una comparación de opiniones provenientes de Amazon empleando cinco clasificadores: MNB, LR, SVM (LSVC), Random Forest y Decision Tree en un entorno BD (Spark). Como se observa, de los cinco algoritmos seleccionados por estos autores, tres son empleados por el framework propuesto en al evaluación del mismo. Por otra parte, el estudio citado se enfoca en realizar clasificación multiclase de las opiniones empleando diversas combinaciones de n-gramas, por lo que no es posible comparar los resultados obtenidos contra este trabajo al tener enfoques distintos.

Li, Zhang, Peng, Yin, & Xu (2018) por su parte estudian cómo identificar a los usuarios pertenecientes a diversas redes sociales a través de su actividad y perfiles públicos en dichos medios. En concreto, los autores extraen características de particulares de dos redes sociales muy conocidas como Facebook y Twitter para establecer la similitud de que un usuario pueda tener una cuenta entre estas redes sociales. Para llevar a cabo el

estudio, los autores emplean los siguientes diez clasificadores: Bagging, MNB, Gaussian Naïve Bayes, LR, LR con validación cruzada interna, SVM, Decision Tree, Random Forest, GraBoosting y AdaBoost. Como se ha mencionado, este estudio tiene otro enfoque al framework presentado dado que en el primero se exploran otras dimensiones de la información de los usuarios.

Por último, en la investigación llevada a cabo por Silva, Alberto, Almeida, & Yamakami (2017), se presenta un framework que aborda el problema de detección de mensajes no deseados (spam) en diferentes medios digitales como mensajería instantánea (SMS), redes sociales o críticas de hoteles, aplicando técnicas de indexación semántica logrando obtener resultados óptimos en cuanto al porcentaje de predicción. Para poder llevar a cabo la evaluación de su modelo, los autores emplearon los siguientes algoritmos: MNB, Bernoulli naïve Bayes, MLP, Approximate Large Margin Algorithm (ALMA), Online Gradient Descent (OGD), SGD y Relaxed Online Maximum Margin Algorithm (ROMMA).

Como se puede observar en función de los trabajos presentados anteriormente, los seis clasificadores empleados en la primera fase: MLP, SVC, LR, LSVC, SVM-SGD y MNB, que además sirven como base para las pruebas llevadas a cabo en la segunda fase, son algoritmos de ML que han sido probados con éxito en otras investigaciones. Siendo el caso de los clasificadores SVC y SVM-SGD una variación de las SVM que se consideró probar durante la primera fase. En los apartados 5.5.3 y 5.5.6 se explica por qué no fueron empleados estos clasificadores en la segunda fase.

5.4. Pruebas y resultados de la primera fase

En este apartado se trata la evaluación concerniente al MLM planteado en la fase uno del modelo presentado en el apartado 4.1.1. Se retoma el framework a partir de la etapa Modelo de Aprendizaje Automático en función de que las etapas: Recuperación de información, Preparación de datos y Estructura y modelado, ya fueron en su momento presentadas y desarrolladas en el apartado 4.2, a las cuáles se les valida a partir de lo que estas entregan como resultado y que es, un conjunto de información que se analiza en las etapas que competen a este capítulo.

Como se mencionó en el Capítulo 4, las pruebas llevadas a cabo en esta fase se ejecutan empleando la biblioteca de aprendizaje automático Scikit-learn versión 0.19.2, la cual tiene entre sus principales ventajas su robustez, facilidad y rapidez de uso, así como el estar respaldada por una amplia comunidad de desarrolladores que dan soporte a la misma. La hipótesis que se quiere validar con las pruebas de esta fase es la 1 que indica: “Es posible definir un framework en el que se puedan aplicar diversos indicadores de

estudio independientemente del volumen de datos a analizar", lo cual se ilustra a lo largo del presente apartado.

5.4.1. Preprocesamiento del texto

En el capítulo anterior se presentó la metodología empleada para procesar el corpus de acuerdo a lo propuesto por el framework, que de manera general se resume en los siguientes pasos:

1. Limpieza del fichero original.
2. Identificación de campos para distinguir información a clasificar y etiquetas.
3. Selección de registros del área de hotelería en gestor de base de datos.
4. Transformación hacia fichero destino.
5. Etiquetado del corpus en positivos y negativos.
6. Unificación del texto (minúsculas) y eliminación de caracteres especiales.
7. *Tokeniz*ación del texto.
8. Vectorización del texto a través del uso de TfIdfVectorizer.

Tras los pasos citados se obtiene la matriz conocida como BOW. Posterior a esto, se procedió a llevar a cabo las pruebas con los diversos clasificadores que, como se ha mencionado, el procedimiento para preprocesar el texto fue el mismo en todos los experimentos, de forma que no se produjera un sesgo en los resultados por esta causa.

5.4.2. Experimentos con MLP

El perceptrón multicapa (MLP) es un algoritmo para el aprendizaje supervisado de clasificadores binarios, las cuáles son funciones que deciden si una entrada representada por un vector de números pertenece a una clase específica (Freund & Schapire, 1999). Debido a la naturaleza de la información de entrada, que como se ha explicado se conforma por dos clases, una positiva y otra negativa, se seleccionó este algoritmo para llevar a cabo la clasificación binaria. Para realizar las pruebas con el clasificador MLP, en primera instancia se buscaron trabajos en el estado del arte que emplearan este algoritmo de ML y que hubieran estado sujetos, pero no limitados, a tareas de clasificación de textos.

A partir de esta búsqueda se encontraron los parámetros y valores que los investigadores emplearon en su estudio para replicar dichos experimentos en el entorno tecnológico local construido para las pruebas de la primera fase. En la Tabla 5.1 se presentan los parámetros que se emplearon durante las pruebas, resaltando en negrita el conjunto de parámetros con los que se obtuvieron los mejores porcentajes de predicción; además, se presentan entre paréntesis los valores default del clasificador que no se

ajustaron en algunas de las pruebas.

Se empezó por emplear el solucionador Adam, el cual es utilizado también por Segura-Bedmar, Colón-Ruíz, Tejedor-Alonso, & Moro-Moro (2018) para tareas de clasificación de textos como registros médicos electrónicos o selección de oraciones, no obstante que este dominio es distinto al del turismo, este trabajo se tomó en cuenta debido a que además de emplear RNA como parte de la solución se emplearon técnicas diversas de PLN en la preparación del texto. La función de activación empleada fue relu y se variaron los valores de la tasa inicial de aprendizaje y los epochs; las configuraciones que fueron tomadas de este trabajo corresponden a las pruebas 1, 2 y 3 presentadas en la Tabla 5.1.

Otro solucionador empleado fue LBFGS, el cual fue usado en los experimentos llevados a cabo por Liu, Liu, & Huang (2017), cuyo trabajo se dedica a estudiar el cómo ampliar el resultado de una consulta a partir de acrónimos; este trabajo fue tomado como referencia porque en la industria del turismo se emplean diversos acrónimos los cuáles varían incluso por país, por lo que este trabajo podría emplearse a futuro en el dominio particular del turismo para comprobar sus resultados. La función de activación empleada fue tanh; la configuración que fue tomada de este trabajo corresponde a la prueba 4 presentada en la Tabla 5.1.

También se tomaron los valores de los parámetros empleados por Xia et al. (2018), que ajustaron los parámetros del solucionador, la tasa de aprendizaje inicial, las capas ocultas y el momentum; las configuraciones que fueron tomadas de este trabajo corresponden a las pruebas 5, 6 y 7 presentadas en la Tabla 5.1; en el caso de la prueba 8, se mezclaron los valores utilizados para observar los resultados. El trabajo de estos autores se incluyó dentro de este conjunto de pruebas debido a que emplean variantes más profundas de las RNA, concretamente en aplicaciones de deep learning, por lo que a través del modelo propuesto pudieron analizar características más complejas del texto para tareas como clasificación de preguntas o de secuencias. Este tipo de tareas más complejas se proponen llevar a cabo en la investigación futura dentro del dominio de turismo para realizar un estudio con fines similares a los propuestos por los autores.

Por otra parte, después de haber observado los resultados obtenidos empleando los valores de las investigaciones anteriormente referidas, se decidió probar dos configuraciones propias en las cuáles se puso a prueba la función de activación logistic; estas configuraciones corresponden a las pruebas 9 y 10 presentadas en la Tabla 5.1.

#	Solver	Tasa de apren-dizaje	Epochs	Función de Activación	Parámetros adicionales	Autores de Parámetros
1	Adam	0.001	50	Relu	No se aplicaron	(Segura-Bedmar et al., 2018)
2	Adam	0.001	100	Relu	No se aplicaron	
3	Adam	0.05	50	Relu	No se aplicaron	
4	LBFGS	(0.001)	(200)	Tanh	No se aplicaron	(Liu et al., 2017)
5	Adam	0.001	(200)	(Relu)	Dos capas ocultas de 100 neuronas cada una (El default es una capa oculta de 100 neuronas)	(Xia et al., 2018)
6	SGD	0.0003	(200)	(Relu)	Momentum = 0.98 (default 0.9)	
7	Adam	0.0001	(200)	(Relu)	No se aplicaron	
8	SGD	0.0001	(200)	(Relu)	Dos capas ocultas de 100 neuronas cada una	
9	**Adam**	**0.001**	**(200)**	**Logistic**	**Dos capas ocultas de 100 neuronas cada una** **Early stopping = True** **(default False)**	**Configuración propia**

10	Adam	0.001	(200)	Logistic	Dos capas ocultas de 100 neuronas cada una	Configuración propia

Tabla 5.1 Valores de los parámetros en las pruebas efectuadas con el clasificador MLP

La función de los parámetros empleados en este clasificador se explica a continuación: el solucionador (solver) es una pieza de software (programa o librería) que resuelve un problema matemático; también es empleado para optimizar los pesos. La tasa de aprendizaje es un hiper-parámetro que controla cuanto se están ajustando los pesos dentro de la red con respecto a la pérdida gradiente. Los *epochs* miden el número de veces que se emplean todos los vectores de entrenamiento una vez para actualizar los pesos. La función de activación de un nodo define la salida de un nodo dada una entrada o un conjunto de entradas, en la Tabla 3.1 se presentó la relación y descripción de las funciones de activación empleadas.

Respecto a los parámetros adicionales: momentum es el valor que se establece para actualizar la pendiente de gradiente. Early stopping es un parámetro que determina si se debe usar la detención temprana para finalizar el entrenamiento del clasificador cuando el puntaje de validación no está mejorando. Mención aparte merece la configuración de la red neuronal que realiza de forma implícita Scikit-learn: las unidades de entrada (nodos) se ajustan de forma automática de acuerdo a las features, en este caso serían 30,000; las capas ocultas establecen la profundidad de la red que, como se ha comentado anteriormente, el default es una capa oculta de 100 neuronas; y la capa de salida se ajusta también de forma automática ya que depende del número de etiquetas, por lo que esta es de dos nodos.

Resultados

Después de haber llevado a cabo las diez pruebas que se ejecutaron diez veces por cada configuración de parámetros, se observa que el conjunto óptimo de parámetros resultó ser el correspondiente a la prueba 9, como se concentra y detalla en la Tabla 5.2. De lo cual se deduce que, para el conjunto de datos de prueba disponible, la función de activación Logistic resultó ser la óptima ya que no sólo mejoró los datos en cuanto a predicción, sino también en cuanto al tiempo en que se completó la tarea de clasificación.

Prueba	Predicción Promedio	MIN	MAX	Segundos	MSE Promedio
1	82.56%	81.51%	83.77%	8329.01	2.80
2	82.56%	82.01%	82.98%	8617.80	2.79
3	84.04%	83.26%	84.58%	4495.53	2.52
4	84.04%	83.23%	85.50%	3799.21	2.54
5	83.82%	83.23%	84.79%	4447.40	2.58
6	87.16%	86.63%	87.61%	52515.33	2.06
7	82.73%	82.08%	83.26%	24592.47	2.77
8	60.00%	59.39%	60.31%	2572.39	6.40
9	**87.21%**	**86.42%**	**87.83%**	**1279.92**	**2.04**
10	82.44%	81.43%	83.09%	6310.82	2.83
Nota: Cada una de las pruebas referidas se realizó diez veces *(k=10-fold cross validation)*					

Tabla 5.2 Resultados del clasificador MLP para diez pruebas con 10 folds cada una

Una vez completadas las pruebas con 10 folds, se llevaron a cabo las pruebas con 3 y 5 folds (veces que se llevan a cabo las pruebas y en los que se dividen los datos en entrenamiento y prueba), empleando los parámetros que se obtuvieron en la prueba 9, como se presenta en la Tabla 5.3. Los resultados demuestran que el promedio obtenido en las pruebas de 3 y 5 folds es muy cercano al que emplea 10 folds (0.37% y 0.22% de diferencia respectivamente).

# Folds	Predicción Promedio	MIN	MAX	Segundos	MSE Promedio
3	86.84%	86.50%	87.17%	265.72	2.08
5	86.99%	86.26%	87.36%	470.49	2.08

Tabla 5.3 Resultados del clasificador MLP para prueba óptima con 3 y 5 folds

5.4.3. Experimentos con SVC

El clasificador SVC es una variante de las SVM, algoritmos ampliamente usado en problemas de clasificación y regresión, siendo una de sus aplicaciones prácticas, en problemas referentes a clasificación de textos (Ramesh & Sathiaseelan, 2015; Sebastiani, 2002; Shafiabady et al., 2016), razón por la cual fue seleccionado este clasificador. Por su parte SVC es la implementación de Scikit-learn para el clasificador SVM que emplea la librería LIBSVM la cual según Chang & Lin (2011) ha sido empleada en trabajos cuyo dominio es el PLN, entre otros. Para realizar las pruebas con el clasificador SVC, en primera instancia se buscaron trabajos en el estado del arte que emplearan este algoritmo de ML y que hubieran estado sujetos, pero no limitados, a tareas de clasificación de textos.

A partir de esta búsqueda se encontraron los parámetros y los valores que los investigadores emplearon en su estudio para replicar dichos experimentos en el entorno tecnológico local construido para las pruebas de la primera fase. En la Tabla 5.4 se presentan los parámetros que se emplearon durante las pruebas, resaltando en negrita el conjunto de parámetros con los que se obtuvieron los mejores porcentajes de predicción; además, se presentan entre paréntesis los valores default del clasificador que no se ajustaron en algunas de las pruebas.

En este clasificador se comenzó por probar todos los kernel disponibles del clasificador como en el trabajo realizado por Manochandar & Punniyamoorthy (2018), para verificar el kernel más apropiado para la clasificación del texto; las configuraciones que fueron tomadas de este trabajo corresponden a las pruebas 1 a 4 presentadas en la Tabla 5.4. Este trabajo fue tomado en cuenta en este apartado debido a que los autores presentan un método para aumentar las capacidades de selección de características de los datos específicamente para tareas de clasificación de opiniones, como se plantea llevar a cabo en el framework propuesto; también se considera este trabajo porque una de las fuentes de datos con que desarrollan las pruebas proviene de Yelp.

Otro conjunto de parámetros que se probaron fueron: el kernel, el número de iteraciones, el parámetro de penalización del término de error (C) y gamma, estos valores fueron establecidos de acuerdo a las pruebas realizadas en el estudio de Sarkar, Vinay, Raj, Maiti, & Mitra (2018); las configuraciones que fueron tomadas de este trabajo corresponden a las pruebas 5 a 8 presentadas en la Tabla 5.4. No obstante que el dominio de este trabajo es la predicción de accidentes en el trabajo, es uno que difiere del dominio del turismo, se consideró emplearlo como referencia en las pruebas debido a que presentan una serie de técnicas de ML enfocadas a mejorar las tareas de clasificación de algoritmos como SVM, y en particular el SVC.

Por otra parte, se establecieron también los valores de kernel y C de acuerdo a la investigación que realizaron García-Pedrajas & Ortiz-Boyer (2011); las configuraciones que fueron tomadas de este trabajo corresponden a las pruebas 9 y 10 presentadas en la Tabla 5.4. Este trabajo es tomado como referencia debido a que los autores emplean una serie de clasificadores binarios (como el problema a resolver por el clasificador) para tratar problemas que involucran la clasificación multiclase; diversos algoritmos son evaluados para medir su rendimiento en este tipo de tareas, uno de los cuáles es el clasificador presentado en esta sección.

#	Kernel	Máximo de Iteraciones	C	Gamma	Autores de Parámetros
1	Linear	(-1)	(1.0)	(auto)	(Manochandar & Punniyamoorthy, 2018)
2	Poly	(-1)	(1.0)	(auto)	
3	Rbf	(-1)	(1.0)	(auto)	
4	Sigmoid	(-1)	(1.0)	(auto)	
5	Sigmoid	500	0.25	128	(Sarkar et al., 2018)
6	Sigmoid	500	0.0078	2	
7	Sigmoid	500	1.1093	0.2474	
8	Sigmoid	500	1.3405	0.2257	
9	Linear	(-1)	0.1	(auto)	(García-Pedrajas & Ortiz-Boyer, 2011)
10	**Linear**	**(-1)**	**1**	**(auto)**	

Tabla 5.4 Valores de los parámetros en las pruebas efectuadas con el clasificador SVC

La función de los parámetros empleados en el clasificador SVC se explica a continuación: el kernel es la función que se emplea en el clasificador, los kernel son alternativas mediante las cuales se puede proyectar la información a un espacio de mayor dimensión, en la Tabla 3.2 se presentó la relación y descripción de los kernel empleados. Las iteraciones representan el número fijo de veces que se está iterando dentro del clasificador. El parámetro de regularización C establece el margen que separa a los vectores de soporte. Mientras que gamma es el valor del coeficiente del kernel, en su caso, aplica para sigmoid.

Resultados

Después de haber llevado a cabo las diez pruebas que se ejecutaron diez veces por cada configuración de parámetros, se observa que el conjunto óptimo de parámetros resultó ser el correspondiente a la prueba 10, como se concentra y detalla en la Tabla 5.5. De lo cual se deduce que, para el conjunto de datos de prueba disponible, el kernel Linear resultó ser el óptimo con un valor de C=1. Una de las características de este clasificador, es que es el que se lleva más tiempo en entregar los resultados. Cabe mencionar que se intentó repetir esta última prueba con el valor de C=10, pero el proceso se tuvo que detener tras varios días de ejecución sin terminarse.

Prueba	Predicción Promedio	MIN	MAX	Segundos	MSE Promedio
1	86.85%	86.10%	87.95%	26453.08	2.11
2	60.00%	59.27%	60.83%	21086.40	6.40
3	60.00%	58.62%	61.32%	21783.07	6.40
4	60.00%	59.34%	60.76%	21407.65	6.40
5	50.41%	48.00%	52.10%	536.69	7.89
6	63.87%	57.63%	68.86%	281.57	5.96
7	68.91%	65.01%	70.41%	358.54	4.99
8	68.41%	62.19%	70.29%	362.65	4.89
9	86.28%	85.27%	87.00%	18514.82	2.20
10	**86.94%**	**86.55%**	**87.40%**	**25832.11**	**2.11**
Nota: Cada una de las pruebas referidas se realizó diez veces *(k=10-fold cross validation)*					

Tabla 5.5 Resultados del clasificador SVC para diez pruebas con 10 folds cada una

Este clasificador de acuerdo a sus características comentadas en el estado de la técnica reviste mayor complejidad para generar los resultados, por las transformaciones matemáticas que se producen. Siendo de los seis clasificadores examinados, el que más tiempo tardó en terminar la tarea de clasificación, además que los valores de predicción tampoco fueron los más altos. Por lo que en un entorno real de producción se desaconseja el empleo de este clasificador para este tipo de tareas en particular.

Una vez completadas las pruebas con 10 folds, se llevaron a cabo las pruebas con 3 y 5 folds (veces que se llevan a cabo las pruebas y en los que se dividen los datos en entrenamiento y prueba), empleando los parámetros que se obtuvieron en la prueba 10, como se presenta en la Tabla 5.6. Los resultados demuestran que el promedio obtenido en las pruebas de 3 y 5 folds es muy cercano al que emplea 10 folds (0.32% y 0.15% de diferencia respectivamente).

# Folds	Predicción Promedio	MIN	MAX	Segundos	MSE Promedio
3	86.62%	86.38%	86.94%	4227.66	2.15
5	86.79%	86.62%	86.94%	9582.12	2.12

Tabla 5.6 Resultados del clasificador SVC para prueba óptima con 3 y 5 folds

5.4.4. Experimentos con LR

De acuerdo a Onan, Korukoğlu, & Bulut (2017) la LR es una generalización de la regresión lineal que modela la probabilidad de la ocurrencia de los eventos y puede ser empleada para predecir el valor de variables dependientes; en el caso de los autores, este clasificador fue apropiado para construir un método conjunto para la clasificación de textos, razón por la cual fue seleccionado este clasificador. Para realizar las pruebas con el clasificador LR, en primera instancia se buscaron trabajos en el estado del arte que emplearan este algoritmo de ML y que hubieran estado sujetos, pero no limitados, a tareas de clasificación de textos.

A partir de esta búsqueda se encontraron los parámetros y los valores que los investigadores emplearon en su estudio para replicar dichos experimentos en el entorno tecnológico local construido para las pruebas de la primera fase. En la Tabla 5.7 se presentan los parámetros que se emplearon durante las pruebas, resaltando en negrita el conjunto de parámetros con los que se obtuvieron los mejores porcentajes de predicción; además, se presentan entre paréntesis los valores default del clasificador que no se ajustaron en algunas de las pruebas.

En primera instancia, se modificaron los valores de los siguientes parámetros: norma de regularización (penalty), solucionador y número de iteraciones de acuerdo al trabajo

de Andrade, Tamura, & Tsuchida (2018); las configuraciones que fueron tomadas de este trabajo corresponden a la prueba 1, mientras que las pruebas 2 y 3 son variaciones que se hicieron al valor del número de iteraciones, las tres son presentadas en la Tabla 5.7. Este trabajo sirvió como referencia para las pruebas por emplear el método de LR para la clasificación de textos tomadas de críticas o noticias. Por otra parte este trabajo propone un nuevo método para emplear las incrustaciones (embeddings) como parte del estudio de las características (features) a obtener de la información; esto puede servir como referencia para investigación futura en el dominio del turismo.

Por otra parte, se tomaron los parámetros que emplearon Yang & Loog (2018) para realizar pruebas con valores de la norma de regularización), el solucionador y el inverso de la fuerza de regularización (C); las configuraciones que fueron tomadas de este trabajo corresponden a la prueba 4, mientras que las pruebas 5 y 6 son variaciones que se hicieron al valor del parámetro C, las tres son presentadas en la Tabla 5.7. Este estudio es considerado dentro de estas pruebas debido a que los autores presentan un método que puede mejorar el aprendizaje activo para ciertos clasificadores; como tal este método es probado en diferentes conjuntos de datos, siendo de particular interés los binarios para tareas de clasificación de textos, como en el caso de las pruebas planteadas.

También se consideró en las pruebas un valor alterno a la norma de regularización según Onan, Korukoğlu, & Bulut (2017); esta configuración corresponde a la prueba 7, la cual se presenta en la Tabla 5.7. Este trabajo se considera dentro de las pruebas al haber evaluado un conjunto de datos perteneciente a críticas de hoteles, como es el caso del conjunto de datos que se evalúa en el framework. Mientras que los valores para las pruebas 8 a 10 que se muestran en la Tabla 5.7, fueron tomados del estudio llevado a cabo por Carneiro et al. (2017). Aunque el dominio de este trabajo es la detección de fraude bancario, se tomó en cuenta para las pruebas por un lado por haber empleado LR para la clasificación de textos y por otro, por el método de minería de datos que los autores proponen para llegar a los resultados, el cual puede emplearse como referencia en la investigación futura.

#	Penalty	Solver	Máximo de Iteraciones	C	Autores de Parámetros
1	L2	LBFGS	20	(1.0)	(Andrade et al., 2018)
2	L2	LBFGS	10	(1.0)	
3	L2	LBFGS	30	(1.0)	
4	L2	Liblinear	(100)	100	(Yang & Loog, 2018)

5	L2	Liblinear	(100)	10	
6	**L2**	**Liblinear**	**(100)**	**1**	
7	L1	(Liblinear)	(100)	(1.0)	(Onan et al., 2017)
8	(L2)	LBFGS	(100)	100	
9	(L2)	LBFGS	(100)	3.16	(Carneiro et al., 2017)
10	(L2)	LBFGS	(100)	1	

Tabla 5.7 Valores de los parámetros en las pruebas efectuadas con el clasificador LR

La función de los parámetros empleados en el clasificador LR se explica a continuación: la técnica L1 de regularización es conocida como Regresión Lasso mientras que la técnica L2 de regularización se le conoce como Regresión Ridge, la diferencia principal entre ambas es el término de penalidad. La diferencia principal entre ambas técnicas es que Lasso reduce a cero el coeficiente del feature menos importante, eliminando entonces algunos features del conjunto de datos.

Por otra parte, el solucionador se refiere a un conjunto de algoritmos que se empleas en el problema de optimización, estos se utilizan en función del tipo de clases o el volumen de los datos, ya que de acuerdo a esto es como se maximiza su rendimiento. Max_iter es el número máximo de iteraciones que se establece para que los solucionadores converjan. C representa el inverso de la fuerza de regularización, la cual consiste en aplicar una penalización para aumentar la magnitud de los valores de los parámetros a fin de reducir el sobreajuste.

Resultados

Después de haber llevado a cabo las diez pruebas que se ejecutaron diez veces por cada configuración de parámetros, se observa que aunque el conjunto óptimo de parámetros resultó ser el correspondiente a la prueba 1, se decidió que el mejor era la prueba 6 debido a que apenas existe una variación del 0.03% entre estos, pero la prueba 6 se ejecutó en la mitad del tiempo de la prueba 1, como se concentra y detalla en la Tabla 5.8. Por otra parte, se aprecia que los valores de predicción se encuentran en valores superiores al 84%, habiendo apenas una diferencia del 2.8% entre la mejor y peor prueba.

Otro aspecto a destacar es que LR es uno de los clasificadores que se ejecutó en el

menor de los tiempos, tomando incluso en cuenta que cada prueba conlleva diez ejecuciones. Por lo que es uno de los mejores clasificadores para esta tarea de clasificación, considerando los tiempos de ejecución y el porcentaje de predicción.

Prueba	Predicción Promedio	MIN	MAX	Segundos	MSE Promedio
1	86.99%	86.07%	87.58%	22.45	2.08
2	86.48%	85.79%	86.99%	12.52	2.15
3	86.93%	86.33%	87.46%	30.94	2.09
4	84.19%	83.54%	84.73%	73.95	2.50
5	86.46%	85.79%	87.02%	28.54	2.18
6	**86.96%**	**86.18%**	**87.46%**	**11.79**	**2.08**
7	86.60%	85.89%	86.94%	23.87	2.15
8	84.33%	82.85%	84.76%	96.42	2.48
9	87.04%	86.66%	87.35%	81.39	2.07
10	87.00%	86.27%	88.18%	61.94	2.10
Nota: Cada una de las pruebas referidas se realizó diez veces *(k=10-fold cross validation)*					

Tabla 5.8 Resultados del clasificador LR para diez pruebas con 10 folds cada una

Una vez completadas las pruebas con 10 folds, se llevaron a cabo las pruebas con 3 y 5 folds (veces que se llevan a cabo las pruebas y en los que se dividen los datos en entrenamiento y prueba), empleando los parámetros que se obtuvieron en la prueba 6, como se presenta en la Tabla 5.9. Los resultados demuestran que el promedio obtenido en las pruebas de 3 y 5 folds es muy cercano al que emplea 10 folds (0.35% y 0.11% de diferencia respectivamente).

# Folds	Predicción Promedio	MIN	MAX	Segundos	MSE Promedio
3	86.61%	86.42%	86.78%	2.19	2.13
5	86.85%	86.69%	87.04%	4.23	2.09

Tabla 5.9 Resultados del clasificador LR para prueba óptima con 3 y 5 folds

5.4.5. Experimentos con LSVC

Las máquinas de vectores de soporte lineales (SVM) fueron introducidas por Cortes & Vapnik (1995). Este algoritmo es ampliamente usado en problemas de clasificación y regresión, siendo una de sus aplicaciones prácticas, en problemas referentes a clasificación de textos (Ramesh & Sathiaseelan, 2015; Sebastiani, 2002; Shafiabady et al., 2016), razón por la cual fue seleccionado este clasificador. Para realizar las pruebas con el clasificador LSVC, en primera instancia se buscaron trabajos en el estado del arte que emplearan este algoritmo de ML y que hubieran estado sujetos, pero no limitados, a tareas de clasificación de textos.

A partir de esta búsqueda se encontraron los parámetros y los valores que los investigadores emplearon en su estudio para replicar dichos experimentos en el entorno tecnológico local construido para las pruebas de la primera fase. En la Tabla 5.10 se presentan los parámetros que se emplearon durante las pruebas, resaltando en negrita el conjunto de parámetros con los que se obtuvieron los mejores porcentajes de predicción; además, se presentan entre paréntesis los valores default del clasificador que no se ajustaron en algunas de las pruebas.

Para la primera prueba, se emplearon los valores establecidos por Yang & Loog (2018) los cuáles trabajaron con el parámetro C (parámetro de penalización del término de error); esta configuración corresponde a la prueba 1, la cual se presenta en la Tabla 5.10. Este clasificador se tomó también en cuenta en estas pruebas como se explica en el apartado 5.4.4, además que los autores también emplean este clasificador en sus experimentos. Posteriormente, se empleó la configuración default del clasificador, siguiendo uno de los experimentos realizados por Onan et al. (2017); esta configuración corresponde a la prueba 2, la cual se presenta en la Tabla 5.10. Este clasificador se tomó también en cuenta en estas pruebas como se explica en el apartado 5.4.4, además que los autores también emplean este clasificador en sus experimentos.

Por otra parte, se modificaron los valores de los parámetros C y multi-clase, en

concordancia con algunas de las pruebas hechas en el estudio realizado por Salles et al. (2018); esta configuración corresponde a las pruebas 3 a 8, mientras que la prueba 9 es una variación de estas pruebas modificando el valor C, estas configuraciones se presentan en la Tabla 5.10. Estos autores proponen una mejora del clasificador *random forests* para lo cual efectúan una comparativa de su método propuesto contra una serie de clasificadores entre los que se encuentra LSVC, por lo cual se tomó este trabajo como referencia para las pruebas, además de haber empleado Yelp como uno de sus conjuntos de datos de prueba.

Por último, como se indica en la prueba 10 de la Tabla 5.10, se establecieron parámetros propios para probar una configuración alterna, la descripción de estos parámetros se describe a continuación de la tabla referida. Estos parámetros se determinaron después de haber realizado una serie de pruebas alternativas para obtener los mejores resultados y de estos se tomó la mejor combinación de parámetros.

#	C	Multi class	Parámetros adicionales	Autores de Parámetros
1	100	(ovr)	No se aplicaron	(Yang & Loog, 2018)
2	(1)	(ovr)	No se aplicaron	(Onan et al., 2017)
3	0.03	(ovr)	No se aplicaron	(Salles et al., 2018)
4	0.05	(ovr)	No se aplicaron	
5	870.53	(ovr)	No se aplicaron	
6	102.65	(ovr)	No se aplicaron	
7	0.13	Crammer Singer	No se aplicaron	
8	0.03	Crammer Singer	No se aplicaron	
9	0.23	Crammer Singer	No se aplicaron	
10	**.1**	**(ovr)**	**Dual = False (default true)** **Tol = 1e-3 (default 1e-4)**	**Configuración propia**

Tabla 5.10 Valores de los parámetros en las pruebas efectuadas con el clasificador LSVC

La función de los parámetros empleados en el clasificador LSVC se explica a continuación: el parámetro de regularización C establece el margen que separa a los vectores de soporte. El parámetro multi-clase determina la estrategia multiclase si cuando se contienen más de dos clases; “ovr” entrena a n_classes de clasificadores de uno-contra-el-resto, mientras que "crammer_singer" optimiza un objetivo conjunto sobre todas las clases (Scikit-Learn, 2018), en el caso de las pruebas realizadas, se emplearon ambas estrategias para observar su comportamiento en cuanto a la predicción. Por otra parte, el parámetro dual selecciona el algoritmo para resolver el problema de optimización dual o principal. El parámetro tol establece la tolerancia para el criterio de parada del algoritmo.

Resultados

Después de haber llevado a cabo las diez pruebas que se ejecutaron diez veces por cada configuración de parámetros, se observa que el conjunto óptimo de parámetros resultó ser el correspondiente a la prueba 10, como se concentra y detalla en la Tabla 5.11. En este caso, se tienen dos pruebas que arrojan el mismo porcentaje de predicción: la 9 y 10, se está tomando la 10 al ser la que lleva menos tiempo de ambas en completar la tarea de clasificación. Por lo que, por otra parte, puede establecerse que en otro ambiente de pruebas podrían evaluarse ambos tipos de configuraciones para examinar la óptima de acuerdo a otro conjunto de datos con características similares a los que se están empleando. Este clasificador tiene tiempos mixtos en cuanto al tiempo en que termina las tareas, sin embargo, es de los más rápidos en cuanto a los algoritmos evaluados.

Prueba	Predicción Promedio	MIN	MAX	Segundos	MSE Promedio
1	81.81%	80.61%	82.49%	134.12	2.90
2	86.13%	85.73%	86.67%	11.97	2.22
3	86.32%	85.79%	86.73%	5.24	2.18
4	86.72%	86.19%	87.52%	5.09	2.12
5	81.39%	80.71%	82.41%	146.03	2.98
6	81.90%	81.25%	82.80%	131.82	2.91
7	86.94%	86.21%	87.47%	13.35	2.09
8	85.67%	85.38%	86.01%	10.91	2.29

9	87.10%	86.64%	87.77%	14.66	2.07
10	**87.10%**	**86.63%**	**87.44%**	**8.56**	**2.07**
Nota: Cada una de las pruebas referidas se realizó diez veces *(k=10-fold cross validation)*					

Tabla 5.11 Resultados del clasificador LSVC para diez pruebas con 10 folds cada una

Una vez completadas las pruebas con 10 folds, se llevaron a cabo las pruebas con 3 y 5 folds (veces que se llevan a cabo las pruebas y en los que se dividen los datos en entrenamiento y prueba), empleando los parámetros que se obtuvieron en la prueba 10, como se presenta en la Tabla 5.12. Los resultados demuestran que el promedio obtenido en las pruebas de 3 y 5 folds es muy cercano al que emplea 10 folds (0.32% y 0.12% de diferencia respectivamente). Por otra parte, es notable que el tiempo fue menor en la prueba para 5 folds que para 3.

# Folds	Predicción Promedio	MIN	MAX	Segundos	MSE Promedio
3	86.78%	86.49%	86.97%	3.98	2.12
5	86.98%	86.82%	87.13%	2.97	2.07

Tabla 5.12 Resultados del clasificador LSVC para prueba óptima con 3 y 5 folds

5.4.6. Experimentos con SVM-SGD

Como se ha comentado anteriormente, la categorización de textos es un problema común que tiene diversas aplicaciones, como el filtrado de correo basura o en análisis de sentimientos. Para llevar a cabo esta tarea, se tienen varios enfoques como los que se han presentado a lo largo de este capítulo, dentro de los cuáles Joachims (1998) realizó una investigación en la que concluyó que el clasificador SVM presenta mejor precisión que el clasificador Naïve Bayes para la clasificación de textos.

Siguiendo este enfoque Donchenko et al. (2017) presentan un estudio en el cual emplean el clasificador SVM con entrenamiento SGD (SVM-SGD) de scikit-learn para realizar tareas de clasificación de textos, razón por la cual se seleccionó este algoritmo.

Para efectuar las pruebas con el clasificador SVM-SGD, en primera instancia se buscaron trabajos en el estado del arte que emplearan este algoritmo de ML y que hubieran estado sujetos, pero no limitados, a tareas de clasificación de textos.

A partir de esta búsqueda se encontraron los parámetros y los valores que los investigadores emplearon en su estudio para replicar dichos experimentos en el entorno tecnológico local construido para las pruebas de la primera fase. En la Tabla 5.13 se presentan los parámetros que se emplearon durante las pruebas, resaltando en negrita el conjunto de parámetros con los que se obtuvieron los mejores porcentajes de predicción; además, se presentan entre paréntesis los valores default del clasificador que no se ajustaron en algunas de las pruebas.

En las pruebas llevadas con este clasificador se tomaron como referencia los valores empleados por Marafino, Boscardin, & Dudley (2015) para los siguientes parámetros: Eta0, regularización, y l1 ratio; las configuraciones que fueron tomadas de este trabajo corresponden a las pruebas 1 a 5 presentadas en la Tabla 5.13. No obstante que el dominio de este trabajo es la clasificación de textos biomédicos, se le hace alusión debido a la amplia selección de características (features) que se hace del texto a través de la aplicación de los métodos propuestos, lo cual puede servir de referencia para la investigación futura en dominios como el turismo.

Por otra parte, del trabajo realizado por Zareapoor et al. (2017), se tomaron como referencia los valores que los autores emplearon para estos parámetros: tasa de aprendizaje, Eta0, l1 ratio y el número de iteraciones; las configuraciones que fueron tomadas de este trabajo corresponden a las pruebas 6 a 10 presentadas en la Tabla 5.13. Este trabajo se empleó como referencia en las pruebas por haber empleado el clasificador SVM-SGD en la clasificación multiclase, la cual puede emplearse en la investigación futura para considerar tres o más clases de un conjunto de datos a clasificar.

#	Tasa de aprendizaje	Eta0	Regula-rización	L1 ratio	Iteraciones	Autores de Parámetros
1	(Constant)	1.4	L2	(0.15)	(5)	(Marafino et al., 2015)
2	(Constant)	1.4	L1	(0.15)	(5)	
3	(Constant)	1.4	(L2)	0	(5)	
4	(Constant)	1.4	(L2)	0.5	(5)	
5	(Constant)	1.4	(L2)	1	(5)	

6	Constant	.001	(L2)	(0.15)	500	(Zareapoor et al., 2017)
7	Constant	.01	(L2)	(0.15)	400	
8	**Constant**	**.01**	**(L2)**	**(0.15)**	**300**	
9	Constant	.1	(L2)	(0.15)	200	
10	Constant	1	(L2)	(0.15)	100	

Tabla 5.13 Valores de los parámetros en las pruebas efectuadas con el clasificador SVM-SGD

Respecto a este clasificador, como se comentó en el apartado 3.2.6, se refiere a una serie de clasificadores lineales (como SVM o regresión logística) que emplean el entrenamiento con gradiente estocástico de descenso. La función de los parámetros referidos anteriormente dentro del clasificador SVM-SGD se explica a continuación: la tasa de aprendizaje se refiere al plan o programa que se utiliza para ajustar el valor que viene establecido por el parámetro Eta0 que es el valor inicial de la tasa de aprendizaje; de tal manera que entre ambos se ajusta el valor con el cual se está ajustando el criterio de la tasa de aprendizaje, y mediante qué esquema se hará; en todas las pruebas se empleó el esquema 'constante'.

El término de regularización establece la penalidad que se emplea en el modelo, siendo 'l2' el regularizador comúnmente empleado. El parámetro l1 ratio se refiere al método de regresión regularizada 'red elástica' o *'elastic net mixing parameter';* que combina linealmente las penalizaciones de L2 (método ridge) siendo l1_ratio=0 y L1 (método lasso) siendo l1_ratio=1, de tal forma que por esta razón el parámetro varía entre los límites de 1 y 0. En cuanto a las iteraciones, éstas establecen el número de veces que pasa el algoritmo sobre los datos de entrenamiento (Scikit-Learn, 2018).

Resultados

Después de haber llevado a cabo las diez pruebas que se ejecutaron diez veces por cada configuración de parámetros, se observa que el conjunto óptimo de parámetros resultó ser el correspondiente a la prueba 8, como se concentra y detalla en la Tabla 5.14. De lo cual se establece que, con el conjunto de parámetros establecidos en esta prueba, se alcanzó el punto óptimo en cuanto a la predicción. Sin embargo, como se observa en la Tabla 5.14, existe un conjunto de parámetros que obtienen un porcentaje de predicción alto en un tiempo menor al no establecer de manera fija las iteraciones del algoritmo.

Prueba	Predicción Promedio	MIN	MAX	Segundos	MSE Promedio
1	84.58%	83.92%	85.02%	2.77	2.51
2	84.39%	83.65%	84.97%	4.59	2.50
3	84.51%	84.26%	84.94%	2.66	2.49
4	84.33%	83.71%	85.29%	2.67	2.50
5	84.17%	81.21%	85.27%	2.50	2.51
6	86.70%	86.33%	86.97%	162.77	2.13
7	86.63%	86.03%	87.38%	130.17	2.13
8	**86.74%**	**85.89%**	**87.46%**	**94.19**	**2.14**
9	86.61%	86.04%	87.17%	62.94	2.14
10	86.60%	85.95%	87.13%	32.20	2.14
Nota: Cada una de las pruebas referidas se realizó diez veces *(k=10-fold cross validation)*					

Tabla 5.14 Resultados del clasificador SGD para diez pruebas con 10 folds cada una

Una vez completadas las pruebas con 10 folds, se llevaron a cabo las pruebas con 3 y 5 folds (veces que se llevan a cabo las pruebas y en los que se dividen los datos en entrenamiento y prueba), empleando los parámetros que se obtuvieron en la prueba 10, como se presenta en la Tabla 5.15. Los resultados demuestran que el promedio obtenido en las pruebas de 3 y 5 folds es muy cercano al que emplea 10 folds (0.12% y 0.16% de diferencia respectivamente).

# Folds	Predicción Promedio	MIN	MAX	Segundos	MSE Promedio
3	86.62%	86.37%	86.86%	15.72	2.14
5	86.58%	86.23%	86.87%	32.34	2.14

Tabla 5.15 Resultados del clasificador SVM-SGD para prueba óptima con 3 y 5 folds

5.4.7. Experimentos con MNB

El clasificador Naïve Bayes (NB) es un clasificador probabilístico fundamentado en el teorema de Bayes (Russell & Norvig, 2013), que fue creado para su uso en clasificación de textos (Brink et al., 2017), razón por la cual se seleccionó este algoritmo, como también lo señalan Segura-Bedmar et al. (2018). Para realizar las pruebas con el clasificador MNB, en primera instancia se buscaron trabajos en el estado del arte que emplearan este algoritmo de ML y que hubieran estado sujetos, pero no limitados, a tareas de clasificación de textos.

A partir de esta búsqueda se encontraron los parámetros y los valores que los investigadores emplearon en su estudio para replicar dichos experimentos en el entorno tecnológico local construido para las pruebas de la primera fase. En la Tabla 5.16 se presentan los parámetros que se emplearon durante las pruebas, resaltando en negrita el conjunto de parámetros con los que se obtuvieron los mejores porcentajes de predicción; además, se presentan entre paréntesis los valores default del clasificador que no se ajustaron en algunas de las pruebas.

La implementación ofrecida por la herramienta para este clasificador sólo permite establecer el valor de tres parámetros, de los cuáles sólo se empleó uno, como se explica a continuación. El parámetro Alpha se refiere al parámetro de suavizado aditivo, esto es, existen ocasiones en las que un término no existe en un conjunto de palabras, pero sí existe en otro que, a la inversa, tiene una palabra que no existe en el conjunto anterior, para lo cual se emplean la técnica también conocida como suavizado aditivo o suavizado de Laplace. Mientras que class_prior establece las probabilidades previas de las clases. En el apartado 3.2.5 se amplió el concepto de los parámetros empleados en el clasificador MNB.

Por lo que el único parámetro modificado en este clasificador fueron las probabilidades previas de las clases. En primera instancia, se especificaron estos valores tomando como referencia los métodos desarrollados por Feng, Guo, Jing, & Sun (2015); las configuraciones que fueron tomadas de este trabajo corresponden a las pruebas 1 a 5 presentadas en la Tabla 5.16. También se utilizaron en las pruebas los valores empleados en la investigación de Lee & Isa (2010); las configuraciones que fueron tomadas de este trabajo corresponden a las pruebas 6 a 10 presentadas en la Tabla 5.16.

Cabe aclarar que se acotaron las pruebas a estos dos trabajos en primer lugar porque la búsqueda que se realizó, como se ha mencionado, considera trabajos en los que se empleara el clasificador MNB para tareas de clasificación de textos, así como el hecho de que fueran trabajos citados por otros autores. Pero fundamentalmente al hecho de que,

aunque existen una gran cantidad de trabajos con estas características, y además se establecen los valores de las probabilidades de las clases, estas son hechas través de métodos donde se explora el teorema de Bayes mediante fórmulas donde se toma en cuenta el volumen de datos que se está analizando, así como otras características de los datos.

De tal manera que este tipo de estudios no se consideraron, al no ser este un trabajo en el cual se pretenda adentrar en el teorema de Bayes y sus aplicaciones dentro del dominio particular de datos que se está analizando. Por otra parte, la implementación actual de la herramienta no permite establecer estos valores a través de tales fórmulas, por lo que se prefirió utilizar trabajos que incluyeran directamente los valores empleados o bien, un valor inicial y los valores que incrementaran los mismos. Como se ha mencionado, el clasificador MNB es ampliamente empleado en tareas de clasificación de textos, por lo que los trabajos citados pueden ser empleados en la investigación futura en dominios como el turismo para evaluar tareas de selección de subconjuntos de características (Feng et al., 2015) o para mejorar el factor de ponderación dependiente del documento calculado automáticamente (Lee & Isa, 2010).

#	alpha	class_prior	fit_prior	Autores de Parámetros
1	(1.0)	0, 0.05	(True)	(Feng et al., 2015)
2	(1.0)	0.05, 0.1	(True)	
3	(1.0)	0.1, 0.3	(True)	
4	(1.0)	0.3, 0.5	(True)	
5	(1.0)	0.5, 0.7	(True)	
6	(1.0)	0.005, 0.01	(True)	(Lee & Isa, 2010)
7	(1.0)	0.01, 0.015	(True)	
8	(1.0)	0.015, 0.02	(True)	
9	(1.0)	0.02, 0.03	(True)	
10	**(1.0)**	**0.03, 0.04**	**(True)**	

Tabla 5.16 Valores de los parámetros en las pruebas efectuadas con el clasificador MNB

Resultados

Después de haber llevado a cabo las diez pruebas que se ejecutaron diez veces por cada configuración de parámetros, se observa que el conjunto óptimo de parámetros resultó ser el correspondiente a la prueba 10, como se concentra y detalla en la Tabla 5.17. En este clasificador hay que resaltar que las probabilidades previas de las clases comúnmente se establecen en función del tipo de datos que se tienen y de los objetivos del estudio a realizar si se dese emplear el clasificador MNB. Otro aspecto a destacar del clasificador MNB es que, aunque no obtiene los mejores porcentajes de predicción, es el que más rápido concluye la tarea de clasificación de datos, lo cual confirma el porqué es comúnmente empleado en aplicaciones reales de clasificación de textos.

Prueba	Predicción Promedio	MIN	MAX	Segundos	MSE Promedio
1	40.00%	39.35%	40.58%	1.23	9.60
2	78.93%	78.32%	79.81%	1.20	3.36
3	75.06%	74.57%	75.64%	1.19	3.99
4	80.51%	80.08%	81.13%	1.18	3.11
5	81.76%	81.18%	82.43%	1.19	2.92
6	78.98%	78.35%	79.57%	1.18	3.36
7	81.31%	80.09%	82.46%	1.21	2.99
8	82.15%	81.21%	83.27%	1.24	2.86
9	81.33%	80.55%	82.13%	1.20	3.00
10	**82.18%**	**81.27%**	**83.03%**	**1.21**	**2.86**
Nota: Cada una de las pruebas referidas se realizó diez veces *(k=10-fold cross validation)*					

Tabla 5.17 Resultados del clasificador MNB para diez pruebas con 10 folds cada una

Una vez completadas las pruebas con 10 folds, se llevaron a cabo las pruebas con 3 y 5 folds (veces que se llevan a cabo las pruebas y en los que se dividen los datos en entrenamiento y prueba), empleando los parámetros que se obtuvieron en la prueba 10, como se presenta en la Tabla 5.18. Los resultados demuestran que el promedio obtenido

en las pruebas de 3 y 5 folds es muy cercano al que emplea 10 folds (0.03% y 0.05% de diferencia respectivamente, siendo la menor diferencia registrada en cualquiera de los clasificadores).

# Folds	Predicción Promedio	MIN	MAX	Segundos	MSE Promedio
3	82.15%	82.08%	82.30%	0.37	2.87
5	82.13%	81.73%	82.37%	0.49	2.86

Tabla 5.18 Resultados del clasificador MNB para prueba óptima con 3 y 5 folds

5.4.8. Comparación de los mejores resultados por clasificador. Fase 1

Después de haber realizado todas las pruebas concernientes a los seis clasificadores mediante el método de validación cruzada para $k = 3$, 5 y 10 folds, se proceden a comparar los resultados obtenidos para las pruebas con $k = 10$ folds. Este factor de comparación se establece en virtud de que con este número de particiones se obtuvieron los mejores resultados del porcentaje de predicción respecto a la clasificación binaria llevada a cabo. En la Tabla 5.19 se presentan los mejores resultados obtenidos por cada clasificador, empleando las mismas métricas presentadas en los apartados anteriores.

Clasificador	Predicción Promedio	MIN	MAX	Segundos	MSE Promedio
MLPC	87.21%	86.42%	87.83%	1279.92	2.04
SVC	86.94%	86.55%	87.40%	25832.11	2.11
LR	86.96%	86.18%	87.46%	11.79	2.08
LSVC	87.10%	86.63%	87.44%	8.56	2.07
SVM-SGD	86.74%	85.89%	87.46%	94.19	2.14
MNB	82.18%	81.27%	83.03%	1.21	2.86
Nota: Cada una de las pruebas referidas se realizó diez veces *(k=10-fold cross validation)*					

Tabla 5.19 Mejores resultados de los clasificadores evaluados para 10 folds

Como se observa en la Tabla 5.19, el clasificador que obtiene el mejor porcentaje de predicción es el clasificador MLPC con un 87.21% promedio de acierto, mientras que el clasificador MNB es el que presenta el menor porcentaje con un 82.18% promedio de acierto, no obstante, este clasificador es el que realiza la tarea de clasificación en el menor tiempo: sólo 1.21 segundos. Por otra parte, en cuanto a la relación Predicción/Tiempo el clasificador LSVC es el que presenta los mejores valores al ser el clasificador con el segundo mejor promedio con el segundo mejor tiempo.

Mientras que el clasificador SVC es el que puede considerarse el menos favorable para este tipo de tareas de clasificación al haberse tomado la mayor cantidad de tiempo para realizar la misma tarea de clasificación que el resto de los clasificadores. En la Figura 5.3 se presenta de forma gráfica esta información donde se comparan los porcentajes de acierto en la predicción obtenidos por los clasificadores.

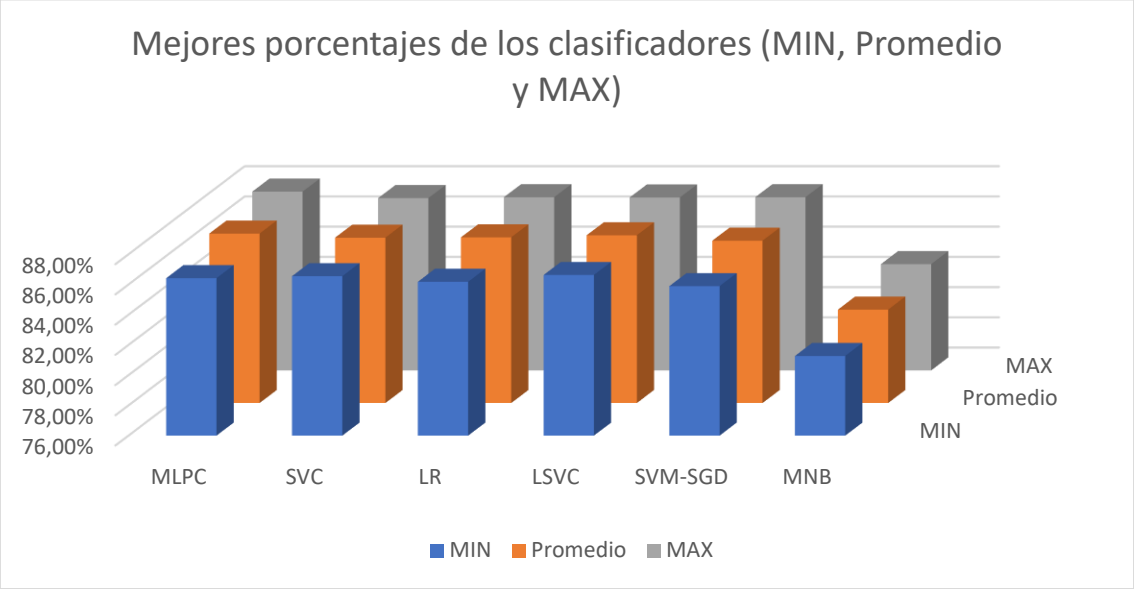

Figura 5.3 Mejores porcentajes de los clasificadores

Por lo que se concluye que para el conjunto de datos utilizado en las pruebas se puede emplear el clasificador MLPC si lo que se desea es obtener el mejor porcentaje de predicción, si se necesita realizar la tarea en el menor tiempo se puede emplear el clasificador MNB, mientras que si se persigue obtener el mejor porcentaje de predicción en el menor tiempo posible, el clasificador LSVC es la mejor alternativa de uso. En la investigación futura se podrían probar otros conjuntos de datos para este tipo de tareas (clasificación binaria de textos) en este u otros entornos no BD y entonces evaluar si se mantienen los respectivos promedios.

5.4.9. Medidas de calidad de las alternativas. Fase 1

Para comparar los procesos de clasificación sobre la información de las propuestas estudiadas se incorporan criterios estadísticos fundados en las pruebas de Kruskal–Wallis (Vasconcelos, Almeida, & Gonçalves, 2015) con un 95% de confianza. La prueba de Kruskal–Wallis (Kruskal & Wallis, 1952) es un método no paramétrico empleado para comprobar si un conjunto de datos provienen de una misma población, que se utiliza cuando se ha determinado que existe diferencia estadística significativa entre varianzas.

Este método también se emplea para comparar dos o más ejemplos independientes cuyos tamaños de la muestra pueden ser de igual o diferente tamaño. Dado que no es un método paramétrico, esta prueba no asume que los residuos tengan una distribución normal, a diferencia del método ANOVA que se explica en el Apartado 5.5.9. En la Figura 5.4 se presenta el esquema de la prueba de contraste de hipótesis realizado durante la validación de los resultados cuando la población es mayor a 2. En la Tabla 5.19 se presentaron los resultados obtenidos por cada clasificador, como se ha comentado, el clasificador MLP es el que obtuvo el mejor porcentaje de predicción durante las pruebas de la primera fase.

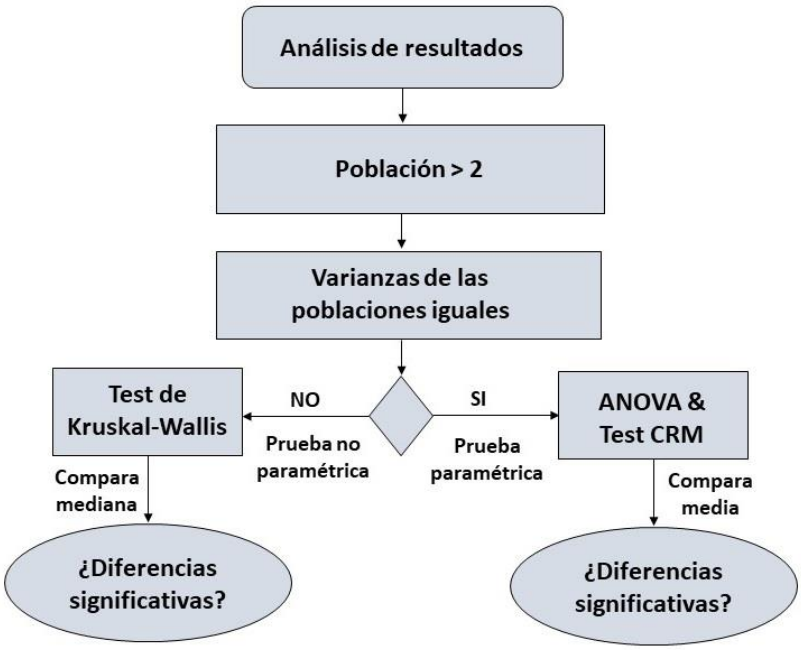

Figura 5.4 Esquema de la prueba de contraste de hipótesis realizado durante la validación de los resultados (adaptado de (González-Carrasco, 2010))

En la Figura 5.5 se presenta el gráfico de cajas y bigotes asociado a los resultados obtenidos por cada clasificador en cuanto al factor de predicción. Las diferentes cajas muestran que existen diversos valores de medición alcanzados por los clasificadores, obteniéndose en algunos casos puntos externos como en los casos del clasificador MNB, o menos notable como en el clasificador MLP. La mayoría de las cajas presentan valores en el rango del 0.8 al 0.9, con excepción del clasificador SVC el cual, como se observa, presenta valores muy dispersos en el factor de predicción. Se observa cierta simetría en el clasificador MLP mientras que las demás cajas tienen asimetría positiva o negativa.

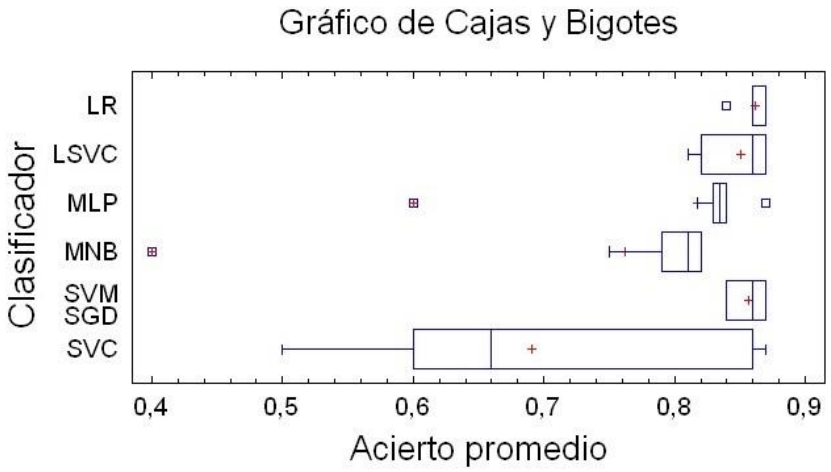

Figura 5.5 Gráfico de cajas y bigotes para el factor de predicción

En la Figura 5.6 se presenta el gráfico de análisis de media por cada clasificador así como la media global y un límite de decisión igual al 95%. Todos los clasificadores excepto SVC están dentro de los límites de decisión además de ser considerablemente distintos a la media global, aunque existe una menor diferencia con el clasificador MLP.

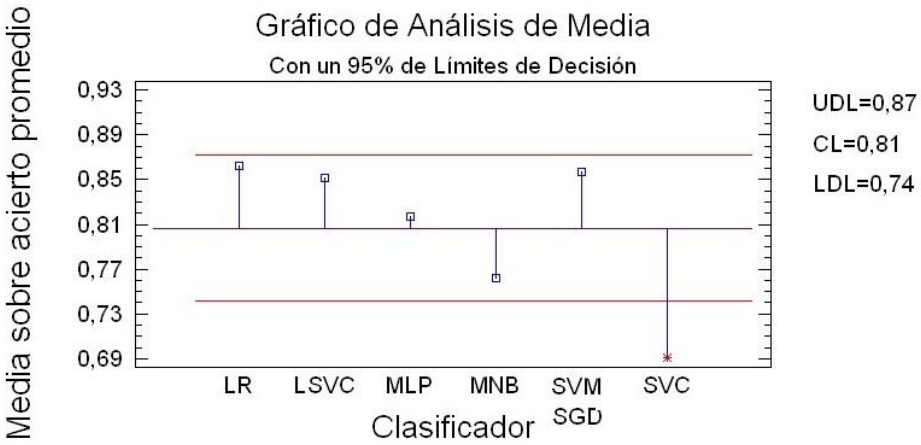

Figura 5.6 Gráfico de análisis de media para los escenarios de predicción

En la Figura 5.7 se muestra el diagrama de dispersión para observar si hubiese modelos de comportamiento, tendencias y observaciones independientes. Como se puede apreciar, se tiene una gran dispersión en el clasificador MNB, seguido de los clasificadores SVC y MLP. Sin embargo, se puede afirmar que existe menor dispersión en la mayoría de los clasificadores respecto a los valores superiores de las predicciones registradas.

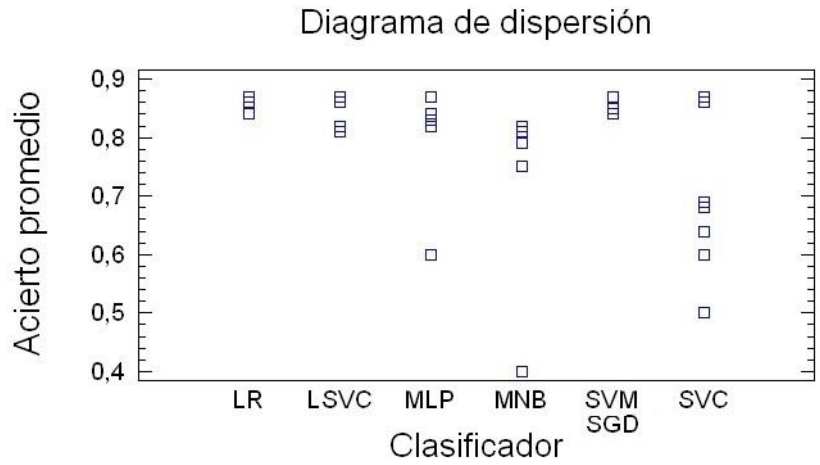

Figura 5.7 Diagrama de dispersión sobre acierto promedio

En la Figura 5.8 se tiene la gráfica de residuos de los algoritmos empleados. Los residuos son similares a los porcentajes de acierto menos el valor medio del grupo del cual se originan y exhiben que el cambio en los clasificadores LR, LSVC y SVM-SGD es aproximadamente la misma; mientras que para MLP, MNB y SVC existe mayor cambio.

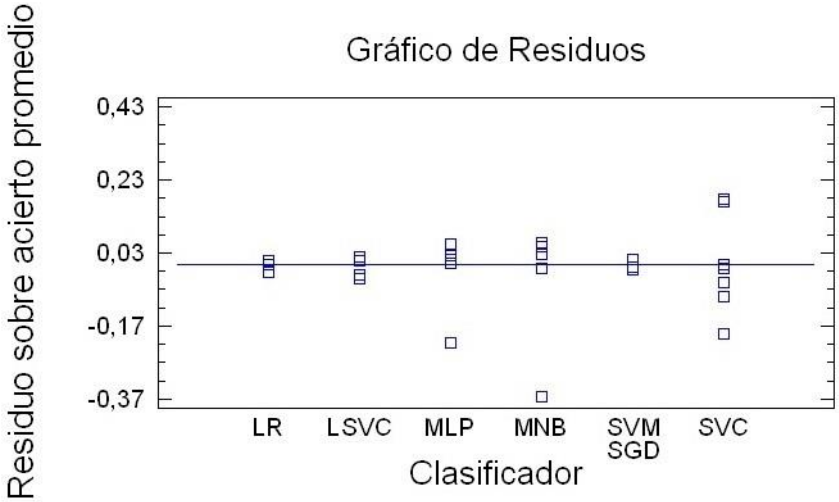

Figura 5.8 Diagrama de residuos sobre acierto promedio

En la Tabla 5.20 se refleja el análisis de contraste de la varianza que, como se observa, los p-valores computados son menores de 0.05, lo cual lleva a establecer desigualdad estadística referible en las desviaciones típicas con un 95.0 % del nivel de confianza. Esto incumple uno de los principales supuestos que se encuentran tras el análisis de varianza pudiendo anular el empleo del ANOVA. Por lo que se ha empleado la prueba no paramétrica de Kruskal-Wallis para no comparar las medias sino las medianas.

Contraste de Varianza		
	Valor	p-Valor
Contraste C de Cochran	0.42381	0.0097195
Contraste de Bartlett	3.85236	0.0
Contraste de Hartley	115.213	
Test de Levene	2.74015	0.0281006

Tabla 5.20 Análisis aplicando el contraste de la varianza

La prueba de Kruskal-Wallis que se presenta en la Tabla 5.21 comprueba la hipótesis nula de igualdad de medianas del Rango Promedio dentro de los 6 niveles del tamaño muestral. La información de los niveles se combina primero y luego se arreglan de menor a mayor. Posteriormente se computa el rango medio para la información en cada nivel. Encontrando diferencia significativa estadística en las medianas con un nivel de confianza del 95.0%, dado que el p-valor es menor de 0.05.

Clasificador	**Tamaño muestral**	**Rango Promedio**
LR	10	44.2
LSCV	10	37.3
MLP	10	28.65
MNB	10	13.6
SGD	10	41.25
SVC	10	18.0
Estadístico = 27.0503		P-valor = 0.0000557703

Tabla 5.21 Análisis aplicando el test de Kruskal-Wallis

El análisis estadístico realizado en este apartado valida los resultados presentados en la Tabla 5.19 la cual concentra todos los resultados obtenidos en las pruebas llevadas a cabo. Como se demuestra, los clasificadores MLPC y LSVC son los que mejor comportamiento presentan, considerando la media del acierto promedio y los errores MSE. Además la prueba de Kruskal-Wallis revela que hay significativas diferencias acerca de la mediana de las distribuciones en la muestra de cada red.

Los resultados presentados en la Tabla 5.19 pueden ser obtenidos debido a que el entorno no BD empleado proporciona indicadores sobre el rendimiento de los clasificadores como la precisión (accuracy) obtenida (promedio, máxima y mínima) y el valor MSE. Lo cual valida la hipótesis "1. Es posible definir un framework en el que se puedan aplicar diversos indicadores de estudio independientemente del volumen de datos a analizar". Como se ha demostrado a lo largo del Apartado 5.3.5, en todos los experimentos realizados se emplearon las métricas de estudio ya descritas, lo cual ha permitido evaluar esta fase del modelo tanto a nivel individual por cada clasificador como de forma general, con lo cual se valida la hipótesis 1.

5.5. Pruebas y resultados de la segunda fase

En este capítulo se trata la evaluación concerniente al MLM planteado en la fase dos del modelo presentado en el apartado 4.1.1. Se retoma el framework a partir de la etapa Análisis de Datos en función de que las etapas: Sistema de Archivos Distribuido, Administrador de Recursos del Clúster y Acceso a Datos, ya fueron en su momento presentadas y desarrolladas en el apartado 4.3, a las cuáles se les valida a partir de lo que estas entregan como resultado y que es, una arquitectura de BD, en este caso Spark, sobre la cual se llevan a cabo las pruebas del entorno BD.

En este entorno de BD se pone a prueba el MLM construido en la primera fase, para evaluar los clasificadores que se tienen en común para ambas fases. Las hipótesis que se quieren validar con las pruebas de esta fase son: la 1. "Es posible definir un framework en el que se puedan aplicar diversos indicadores de estudio independientemente del volumen de datos a analizar" y la 3. "Los modelos de aprendizaje automático definidos para trabajar en un entorno con un volumen reducido de información se pueden extrapolar a un entorno GVDNE para realizar el mismo tipo de análisis".

5.5.1. Preprocesamiento del texto

En el apartado 5.4.1 se dio a conocer los pasos que se siguieron para procesar el corpus de acuerdo a lo propuesto por el framework. En esta fase se siguen los mismos pasos que en la etapa anterior, con la excepción al paso 8 que menciona que se vectoriza el texto. Como se explicó en el apartado 4.3.4 se emplean dos métodos separados, uno para realizar TF y el otro para el proceso IDF, procesos que se explicaron en los apartados 3.5.2 al 3.5.4. Tras los pasos citados se obtiene la matriz conocida como BOW. Posterior a esto, se procedió a llevar a cabo las pruebas con los diversos clasificadores que, como se ha mencionado, el procedimiento para preprocesar el texto fue el mismo en todos los experimentos, de forma que no se produjera un sesgo en los resultados por esta causa.

5.5.2. Experimentos con MLP

Las generalidades del perceptrón multicapa (MLP) y el porqué de su utilización en las pruebas del MLM se ha detallado en el apartado 5.4.2, de igual manera en dicho apartado se explicaron los parámetros que se emplearon atendiendo a los autores que utilizaron este algoritmo para tareas de clasificación de texto. Por lo que en el presente apartado se explica la parte que se modificó para poder emplear este algoritmo con sus parámetros y valores para poderlo utilizar en el entorno de BD debido a que esta implementación es diferente a la empleada en el entorno no BD. A diferencia de la primera fase, se efectuaron cinco pruebas adicionales a las diez existentes para poder realizar la validación estadística necesaria.

Inicialmente se habían programado una serie de pruebas que estuvieran acordes tanto a lo hecho en la primera etapa como a las pruebas llevadas a cabo en los demás clasificadores. En una primera serie de pruebas se configuró la red neuronal para que tuviera 30000 nodos en la capa de entrada como se efectuó durante las pruebas de la primera fase. Este número de nodos se estableció de esta manera para que fueran iguales al número de features que se generan durante el proceso de vectorización del corpus. Sin embargo y a diferencia de lo que se tenía en la primera fase, donde el paquete tecnológico ajusta de forma automática los nodos de entrada a la red neuronal de acuerdo al número de features, para posteriormente realizar el posterior proceso de clasificación, en el entorno BD los pasos anteriores no suceden de esta manera para este algoritmo.

En el entorno BD el paquete empleado (Spark ML) no ajusta de forma automática el número (n) de nodos de entrada de la red neuronal de acuerdo a los features, en su lugar, estos se tienen que establecer de forma directa. De esta manera se probó inicialmente con n=30000 que eran los nodos que se habían probado en la primera fase (de acuerdo a la documentación de scikit-learn), sin embargo el hardware en el que fue probada la solución

no podía procesar tal número de nodos de entrada al producirse desbordamiento de memoria. Por lo anterior, se estuvieron probando diversos conjuntos de nodos de entrada a la red neuronal para que dejaran de producirse los errores de memoria del hardware descritos; esto también comprende el reducir el número de features del vectorizador.

El mayor número de nodos de entrada de la red neuronal que se pudieron probar sin problemas fue de n=1500, siendo este el número de features que también tuvieron que ajustarse en el vectorizador. Aunque este número difiere de lo hecho en la primera fase, este número no afectó al resto de clasificadores empleados en la segunda fase, en los cuáles si pudieron emplearse los 30000 features. Por otra parte, dado que el proceso de vectorización busca encontrar los términos menos frecuentes que a su vez aparecen en la mayor cantidad de documentos (opiniones o reviews) esto no afecta el objetivo de realizar la tarea de clasificación cuya finalidad es encontrar los términos con el mayor valor de TF-IDF.

Por otra parte, como se ha comentado anteriormente, la implementación que se tiene en el entorno BD difiere a la del entorno no BD, por lo que tuvo que reajustarse el plan de pruebas previsto en la primera fase para poder aplicarlo en la segunda; estas diferencias se explican a continuación. El parámetro *solver* o solucionador tiene menos opciones en Spark ML: l-bfgs o gd, los cuáles se corresponden con los *solver* LBFGS o SGD de Scikit-learn, no se cuenta con Adam que fue el *solver* que obtuvo los mejores resultados en la predicción, por esta razón sólo se emplearon las opciones con que se cuenta actualmente.

El parámetro tasa de aprendizaje es referido en este clasificador como *stepSize*. El parámetro Epochs es referido en este clasificador como *maxIter*. Otra de las diferencias con Scikit-learn es que en Spark no se puede establecer la función de activación de la capas, en su lugar, cada capa tiene la función de activación sigmoide (Logistic). El parámetro momentum no se tiene disponible en esta implementación. Mientras que la configuración de la red es obligatoria establecerla, como se indica en la columna con este nombre de la Tabla 5.22.

Los parámetros de la columna configuración de la red refieren lo siguiente: [1500, 100, 2] significa que se tiene una capa de entrada con 1500 nodos de acuerdo al número de features explicado anteriormente, una capa oculta de 100 nodos y una capa de salida de dos nodos la cual se establece de acuerdo al número de etiquetas, en este caso dos (positivos y negativos). Otro ejemplo de configuración de la red neuronal es [1500, 50, 50, 2] en cuyo caso se tiene una capa de entrada con 1500 nodos, dos capas oculta de 50 nodos respectivamente y una capa de salida de dos nodos. El resto de los parámetros y sus diversas configuraciones se presentan en la Tabla 5.22; la mejor combinación de parámetros se resalta en negritas.

El plan de pruebas descrito en la Tabla 5.22 se explica a continuación: inicialmente no se contempló probar una a una cada una de las pruebas empleadas en la primera fase debido a que se tuvieron que descartar todas aquellas que emplearon Adam como solver, con excepción de la prueba 9 que fue la mejor en esas pruebas. La prueba 1 busca reflejar los resultados de la prueba 4 de la fase 1 (Tabla 5.1). Las pruebas 2 y 3 buscan reflejar los resultados de las pruebas 6 y 8 respectivamente de la fase 1. Mientras que la prueba 9 aunque emplea Adam, este tiene de base un optimizador SGD por lo que se probó con el parámetro solver=”gd”.

#	solver	stepSize	maxIter	Configuración de la red	Autores de Parámetros
1	l-bfgs	0.001	100	[1500, 100, 2]	(Liu et al., 2017)
2	gd	0.0003	50	[1500, 100, 2]	(Xia et al., 2018)
3	gd	0.0001	100	[1500, 100, 2]	
4	gd	0.001	50	[1500, 100, 2]	Propia
5	l-bfgs	0.001	50	[1500, 100, 2]	(Segura-Bedmar et al., 2018)
6	l-bfgs	0.05	50	[1500, 100, 2]	
7	l-bfgs	0.05	100	[1500, 100, 2]	
8	**l-bfgs**	**0.001**	**50**	**[1500, 100, 100, 2]**	
9	l-bfgs	0.05	50	[1500, 100, 100, 2]	(Xia et al., 2018)
10	l-bfgs	0.0001	50	[1500, 100, 100, 2]	
11	l-bfgs	0.001	50	[1500, 50, 50, 2]	
12	l-bfgs	0.001	100	[1500, 100, 100, 2]	
13	l-bfgs	0.001	100	[1500, 50, 50, 2]	
14	l-bfgs	0.001	50	[1500, 50, 50, 2]	
15	l-bfgs	0.001	50	[1500, 150, 150, 2]	

Tabla 5.22 Valores de los parámetros en las pruebas efectuadas con el clasificador MLP - Fase 2

Posteriormente tras haber llevado a cabo las primeras cuatro pruebas, se observó que sólo la que empleó el *solver='l-bfgs'* era la que entregaba los mejores resultados, por lo que las posteriores pruebas descritas alteran en este valor las pruebas efectuadas por los diversos autores que se expusieron en las correspondientes pruebas de la primera fase. Sin embargo, al haber sólo dos opciones no se contaba con mayor margen de decisión, por lo que se empleó el solver *l-bfgs* en el resto de pruebas. Las pruebas 5 a 8 son una adaptación de las pruebas 1 y 3 de la Tabla 5.1 en las que se varió el resto de los parámetros disponibles para observar el porcentaje de acierto en la predicción.

En lo referente a las pruebas 9 y 10 de la Tabla 5.22, estas buscan reflejar las pruebas 5 y 8 respectivamente de la primera fase tomando como referencia la tasa de aprendizaje y la configuración de la red. El resto de pruebas (11 a la 15) se llevaron a cabo con el fin de observar si se mejoraba el porcentaje de predicción variando los Epochs o la configuración de la red, no derivando lo anterior en una mejora en los resultados obtenidos en el plan inicial de diez pruebas

Resultados

Después de haber llevado a cabo las quince pruebas que se ejecutaron diez veces por cada configuración de parámetros, se observa que el conjunto óptimo de parámetros resultó ser el correspondiente a la prueba 8, como se concentra y detalla en la Tabla 5.23. De lo cual se deduce que, para el conjunto de datos de prueba disponible, la combinación de 50 iteraciones (*epochs*) y una red neuronal conteniendo dos capas ocultas de 100 nodos cada una, resultó ser la óptima ya que obtuvo el mejor factor de predicción.

#	Predicción Promedio	Segundos	MSE
1	84.16%	1773	0.02
2	53.03%	784	0.45
3	53.04%	1498	0.46
4	53.52%	808	0.44
5	86.89%	1005	0.13
6	87.27%	998	0.14
7	84.32%	1855	0.005

8	**88.36%**	**1409**	**0.17**
9	88.28%	1390	0.17
10	88.25%	1426	0.17
11	88.14%	783	0.16
12	85.70%	2650	0.10
13	84.23%	1609	0.06
14	86.49%	736	0.13
15	88.34%	2100	0.17
Nota: Cada una de las pruebas referidas se realizó diez veces *(k=10-fold cross validation)*			

Tabla 5.23 Resultados del clasificador MLP para quince pruebas con 10 folds - Fase 2

Como se observa en la Tabla 5.23, los porcentajes de predicción en la clasificación son altos habiéndose entrenado la red incluso con 1500 nodos de entrada, quedando para el trabajo futuro probar este clasificador con un número mayor de nodos de entrada en este entorno BD pero con una base tecnológica de mayores prestaciones.

Una vez completadas las pruebas con 10 folds, se llevaron a cabo las pruebas con 3 y 5 folds (veces que se llevan a cabo las pruebas y en los que se dividen los datos en entrenamiento y prueba), empleando los parámetros que se obtuvieron en la prueba 8, como se presenta en la Tabla 5.24. Los resultados demuestran que el promedio obtenido en las pruebas de 3 y 5 folds es muy cercano al que emplea 10 folds (0.58% y 0.25% de diferencia respectivamente).

Folds	Predicción Promedio	Segundos	MSE
3	87.78%	439	0.17
5	88.11%	736	0.17

Tabla 5.24 Resultados del clasificador MLP para prueba óptima con 3 y 5 folds - Fase 2

5.5.3. Experimentos con SVC

En la versión más reciente del paquete Spark ML no se cuenta con la implementación del Clasificador de vectores de soporte – C (SVC) por lo que las pruebas con este algoritmo no pudieron ser llevadas a cabo en la segunda fase.

5.5.4. Experimentos con LR

Las generalidades del clasificador Regresión Logística (LR) y el porqué de su utilización en las pruebas del MLM se ha detallado en el apartado 5.4.4, de igual manera en dicho apartado se explicaron los parámetros que se emplearon atendiendo a los autores que utilizaron este algoritmo para tareas de clasificación de texto. Por lo que en el presente apartado se explica la parte que se modificó para poder emplear este algoritmo con sus parámetros y valores para poderlo utilizar en el entorno de BD debido a que esta implementación es diferente a la empleada en el entorno no BD. A diferencia de la primera fase, se realizaron cinco pruebas adicionales a las diez existentes para poder hacer la validación estadística necesaria.

En la librería empleada en la primera fase se tiene el parámetro Penalty que en este clasificador especifica la norma empleada en la penalización para indicar si se trata del tipo L1 o L2; en el caso de Spark ML esto se indica modificando el parámetro elasticNetParam haciéndolo igual a 0 (L2) o 1 (L1). El parámetro que especifica el valor máximo de iteraciones es el mismo en ambos entornos. Mientras que el parámetro C que en Scikit-learn indica el inverso de la fuerza de regularización, en Spark ML se corresponde con el parámetro regParam. En la Tabla 5.25 se muestran los valores de los parámetros que se emplearon en las pruebas efectuadas con el clasificador LR; la mejor combinación de parámetros se resalta en negritas.

En este entorno no se cuenta con el parámetro Solver para este clasificador, el cual fue empleado en las pruebas de la fase uno, por lo que las pruebas que se llevaron a cabo en esta fase y con este clasificador emulan a aquellas de la primera, no haciendo referencia a ningún autor en particular debido a esta causa, pero se tomaron en cuenta algunos de los valores usados en las pruebas desarrolladas en la primera fase. A diferencia del clasificador MLP, con este algoritmo sí pudieron emplearse los 30000 features generados durante el proceso de vectorización.

#	elasticNetParam – l1(1) o l2 (0)	maxIter	regParam
1	elasticNetParam=0	100	100.0
2	elasticNetParam=0	100	10.0
3	elasticNetParam=1	100	1.0
4	elasticNetParam=0	50	1.0
5	elasticNetParam=0	100	1.0
6	elasticNetParam=0	200	1.0
7	**elasticNetParam=0**	**250**	**1.0**
8	elasticNetParam=0	300	1.0
9	elasticNetParam=0	100	0.1
10	elasticNetParam=0	200	0.1
11	elasticNetParam=0	220	1.0
12	elasticNetParam=0	250	5.0
13	elasticNetParam=0	250	10.0
14	elasticNetParam=0	250	25.0
15	elasticNetParam=0	250	50.0

Tabla 5.25 Valores de los parámetros en las pruebas efectuadas con el clasificador LR - Fase 2

El plan de pruebas descrito en la Tabla 5.25 se explica a continuación: se exploraron primero ciertas pruebas aleatorias con los parámetros maxIter, regParam y elasticNetParam, observando que este último establecido a 1 entregaba el peor promedio de predicción. Posteriormente, se observó que el parámetro RegParam establecido en 1.0 ofrecía los mejores resultados en la predicción promedio, por lo que se realizaron una serie de pruebas variando el número máximo de iteraciones para encontrar el punto donde se alcanzara el valor máximo de predicción, siendo 250 el valor óptimo de este parámetro. A partir de este valor se efectuaron cinco pruebas adicionales con el fin de obtener valores que se emplean más adelante en la validación estadística.

Resultados

Después de haber llevado a cabo las quince pruebas que se ejecutaron diez veces por cada configuración de parámetros, se observa que el conjunto óptimo de parámetros resultó ser el correspondiente a la prueba 7, como se concentra y detalla en la Tabla 5.26. De lo cual se deduce que, para el conjunto de datos de prueba disponible, la combinación de 250 iteraciones máximas, regularización L2 y regParam=1.0, resultó ser la combinación óptima de parámetros ya que obtuvo el mejor factor de predicción.

#	Predicción Promedio	Segundos	MSE
1	83.48%	65	0.40
2	88.48%	64	0.38
3	49.99%	57	0.40
4	91.33%	85	0.13
5	91.31%	67	0.13
6	91.34%	76	0.13
7	**91.34%**	**74**	**0.13**
8	91.30%	75	0.13
9	90.82%	84	0.06
10	90.78%	83	0.06
11	91.30%	76	0.13
12	89.81%	65	0.30
13	88.54%	65	0.38
14	86.16%	63	0.40
15	84.58%	64	0.40
Nota: Cada una de las pruebas referidas se realizó diez veces *(k=10-fold cross validation)*			

Tabla 5.26 Resultados del clasificador LR para quince pruebas con 10 folds - Fase 2

Una vez completadas las pruebas con 10 folds, se llevaron a cabo las pruebas con 3 y 5 folds (veces que se llevan a cabo las pruebas y en los que se dividen los datos en entrenamiento y prueba), empleando los parámetros que se obtuvieron en la prueba 7, como se presenta en la Tabla 5.27. Los resultados demuestran que el promedio obtenido en las pruebas de 3 y 5 folds es muy cercano al que emplea 10 folds (0.37% y 0.18% de diferencia respectivamente).

Folds	Predicción Promedio	Segundos	MSE
3	90.97%	26	0.13
5	91.16%	40	0.13

Tabla 5.27 Resultados del clasificador LR para prueba óptima con 3 y 5 folds - Fase 2

5.5.5. Experimentos con LSVC

Las generalidades del clasificador LSVC y el porqué de su utilización en las pruebas del MLM se ha detallado en el apartado 5.4.5, de igual manera en dicho apartado se explicaron los parámetros que se emplearon atendiendo a los autores que utilizaron este algoritmo para tareas de clasificación de texto. Por lo que en el presente apartado se explica la parte que se modificó para poder emplear este algoritmo con sus parámetros y valores para poderlo utilizar en el entorno de BD debido a que esta implementación es diferente a la empleada en el entorno no BD. A diferencia de la primera fase, se realizaron cinco pruebas adicionales a las diez existentes para poder hacer la validación estadística necesaria.

Una de las diferencias que se tienen en los parámetros del clasificador LSVC empleado en la segunda fase es que en esta no se cuenta con el parámetro Multiclass, del cual no se tiene información sobre cómo está implementado internamente, por lo que no se pueden aplicar las pruebas descritas en la Tabla 5.10 de la fase 1, no obstante, se variaron otros parámetros que se tenían disponibles. El parámetro C que establece el valor del término de error se encuentra en Spark ML como regParam, para lo cual se tomaron algunos de los valores empleados en las pruebas correspondientes de la fase 1. Otro parámetro que se empleó fue el que establece la tolerancia (tol) el cual es similar en ambos entornos. En la Tabla 5.28 se muestran los valores de los parámetros que se emplearon en las pruebas efectuadas con el clasificador LSVC; la mejor combinación de parámetros se resalta en negritas.

#	regParam	tol	maxIter
1	100	0.001	5
2	100	0.0001	5
3	1	0.001	5
4	1	0.0001	5
5	0.1	0.001	5
6	0.1	0.0001	5
7	0.03	0.0001	5
8	0.05	0.0001	5
9	870.53	0.0001	5
10	102.65	0.0001	5
11	0.05	0.0001	10
12	0.5	0.0001	5
13	0.05	0.01	5
14	0.05	0.0001	4
15	**0.05**	**0.0001**	**8**

Tabla 5.28 Valores de los parámetros en las pruebas efectuadas con el clasificador LSVC - Fase 2

El plan de pruebas descrito en la Tabla 5.28 se explica a continuación: las primeras pruebas fueron hechas tomando como base el parámetro regParam de acuerdo al correspondiente parámetro C de la primera fase, en la tabla referida se encuentran estos valores distribuidos entre la prueba 1 a la 10. Posteriormente se ajustó el valor de tolerancia en diversos valores observando que con 0.0001 se obtenía un mejor resultado en la predicción. Finalmente, se ajustó el máximo de iteraciones que, aunque mayoritariamente se probó con un valor de 5, al realizar las pruebas extras necesarias para la posterior validación estadística se encontró que con un valor de 8 se obtenía el mejor resultado.

Resultados

Después de haber llevado a cabo las quince pruebas que se ejecutaron diez veces por cada configuración de parámetros, se observa que el conjunto óptimo de parámetros resultó ser el correspondiente a la prueba 15, como se concentra y detalla en la Tabla 5.29. De los resultados obtenidos se deduce que, para el conjunto de datos de prueba disponible, la combinación de regParam=1.0, tolerancia=0.0001 y 8 iteraciones máximas, resultó ser la combinación óptima de parámetros ya que obtuvo el mejor factor de predicción.

#	Predicción Promedio	Segundos	MSE
1	70.03%	318	0.40
2	69.97%	310	0.40
3	91.25%	297	0.11
4	91.30%	293	0.11
5	91.38%	293	0.11
6	91.39%	289	0.11
7	91.36%	301	0.11
8	91.43%	292	0.11
9	69.28%	307	0.40
10	69.95%	310	0.40
11	90.99%	507	0.07
12	91.33%	292	0.11
13	91.42%	290	0.11
14	90.09%	254	0.14
15	**91.52%**	**438**	**0.09**
Nota: Cada una de las pruebas referidas se realizó diez veces *(k=10-fold cross validation)*			

Tabla 5.29 Resultados del clasificador LSVC para quince pruebas con 10 folds - Fase 2

Una vez completadas las pruebas con 10 folds, se llevaron a cabo las pruebas con 3 y 5 folds (veces que se llevan a cabo las pruebas y en los que se dividen los datos en entrenamiento y prueba), empleando los parámetros que se obtuvieron en la prueba 15, como se presenta en la Tabla 5.30. Los resultados demuestran que el promedio obtenido en las pruebas de 3 y 5 folds es muy cercano al que emplea 10 folds (0.47% y 0.17% de diferencia respectivamente).

Folds	Predicción Promedio	Segundos	MSE
3	91.05%	131	0.09
5	91.35%	214	0.09

Tabla 5.30 Resultados del clasificador LSVC para prueba óptima con 3 y 5 folds - Fase 2

5.5.6. Experimentos con SVM-SGD

En Spark existen dos paquetes tecnológicos para realizar tareas de ML: Spark ML y Spark MLlib. Como se ha comentado durante las pruebas de la segunda fase, las pruebas llevadas a cabo con los diversos clasificadores han sido llevadas a la práctica empleando la librería Spark ML, en la cual no existe aún este algoritmo de ML. No obstante, este algoritmo existe en la librería Spark MLlib, la cual no se empleó para esta fase de pruebas debido a que no permite reproducir el proceso de vectorización del corpus requerido para su uso por los clasificadores.

El haber empleado este clasificador habría implicado un sesgo en los resultados finales debido a que no se hubiera llevado a cabo el mismo procesamiento previo al empleo de este algoritmo, por lo que las pruebas para este clasificador no se llevaron a cabo debido a los motivos expuestos. Por otra parte, como se afirma por Deshpande & Kumar (2018), la interfaz de Spark ML es más versátil y flexible, además de tener la ventaja de poder emplear DataFrames, mientras que Spark MLlib (basado en RDDs) se espera sea removido en el futuro; por lo que también se espera que este algoritmo sea incluido en una futura versión de Spark ML.

5.5.7. Experimentos con MNB

Las generalidades del clasificador MNB y el porqué de su utilización en las pruebas

del MLM se ha detallado en el apartado 5.4.7, de igual manera en dicho apartado se explicaron los parámetros que se emplearon atendiendo a los autores que utilizaron este algoritmo para tareas de clasificación de texto. Por lo que en el presente apartado se explica la parte que se modificó para poder emplear este algoritmo con sus parámetros y valores para poderlo utilizar en el entorno de BD debido a que esta implementación es diferente a la empleada en el entorno no BD. A diferencia de la primera fase, se realizaron cinco pruebas adicionales a las diez existentes para poder hacer la validación estadística necesaria.

El clasificador MNB es, de los cuatro clasificadores empleados, el más parecido en cuanto a su implementación al empleado en la primera fase, los parámetros utilizados se aprecian en la Tabla 5.31; la mejor combinación de parámetros se resalta en negritas. Como se observa en la tabla referida, el parámetro smoothing se corresponde con el parámetro Alpha del clasificador empleado en la primera fase. Mientras que existe una clara correspondencia entre los parámetros *class prior* empleados para establecer los valores de las prioridades de las clases.

#	smoothing	class_prior
1	1.0	0, 0.05
2	1.0	0.05, 0.1
3	1.0	0.1, 0.3
4	1.0	0.3, 0.5
5	1.0	0.5, 0.7
6	1.0	0.005, 0.01
7	1.0	0.01, 0.015
8	1.0	0.015, 0.02
9	1.0	0.02, 0.03
10	1.0	0.03, 0.04
11	10.0	0.1, 0.3
12	20.0	0.1, 0.3
13	40.0	0.1, 0.3

14	**70.0**	**0.1, 0.3**
15	100.0	0.1, 0.3

Tabla 5.31 Valores de los parámetros en las pruebas efectuadas con el clasificador MNB - Fase 2

El plan de pruebas descrito en la Tabla 5.31 se explica a continuación: en el caso de este clasificador fue posible emplear los valores de los parámetros empleados por trabajos hechos por otros autores como se representa en la Tabla 5.16. Debido a esto, se siguieron los valores de los incrementos establecidos en las prioridades de las clases para las pruebas de la primera fase para obtener el mejor valor de predicción, el cual fue con la sucesión: 0.1 y 0.3. Originalmente al realizar las primeras diez pruebas, esta combinación entregaba los mejores resultados para un valor de smoothing=1.0. Sin embargo al llevar a cabo el plan extendido de cinco pruebas adicionales, se tuvo que variar el parámetro smoothing encontrando un mejor ajuste cuando este valor se establece en 70.0.

Resultados

Después de haber llevado a cabo las quince pruebas que se ejecutaron diez veces por cada configuración de parámetros, se observa que el conjunto óptimo de parámetros resultó ser el correspondiente a la prueba 14, como se concentra y detalla en la Tabla 5.32. De lo cual se deduce que, para el conjunto de datos de prueba disponible, la combinación de smoothing=70.0 con prioridades previas de las clases de 0.1 y 0.3 resultó ser la combinación óptima de parámetros ya que obtuvo el mejor factor de predicción.

#	Predicción Promedio	Segundos	MSE
1	44.99%	60	0.40
2	80.81%	60	0.16
3	80.91%	61	0.15
4	80.79%	61	0.16
5	80.80%	62	0.16
6	80.85%	60	0.16

7	80.78%	61	0.16
8	80.64%	60	0.16
9	80.74%	61	0.16
10	80.55%	61	0.16
11	81.07%	64	0.16
12	81.19%	61	0.16
13	81.50%	60	0.16
14	**81.60%**	**60**	**0.16**
15	81.33%	60	0.16
Nota: Cada una de las pruebas referidas se realizó diez veces *(k=10-fold cross validation)*			

Tabla 5.32 Resultados del clasificador MNB para quince pruebas con 10 folds - Fase 2

Como se observa en la tabla anterior, los valores de predicción obtenidos (con excepción del primero) se encuentran distanciados por un margen de separación estrecho, además de que se completa la tarea de clasificación de casi todas las pruebas en el mismo tiempo. Una vez completadas las pruebas con 10 folds, se llevaron a cabo las pruebas con 3 y 5 folds (veces que se llevan a cabo las pruebas y en los que se dividen los datos en entrenamiento y prueba), empleando los parámetros que se obtuvieron en la prueba 14, como se presenta en la Tabla 5.33. Los resultados demuestran que el promedio obtenido en las pruebas de 3 y 5 folds es muy cercano al que emplea 10 folds (0.32% y 0.08% de diferencia respectivamente).

Folds	**Predicción Promedio**	**Segundos**	**MSE**
3	81.28%	24	0.16
5	81.52%	34	0.16

Tabla 5.33 Resultados del clasificador MNB para prueba óptima con 3 y 5 folds - Fase 2

5.5.8. Comparación de los mejores resultados por clasificador. Fase 2

Después de haber llevado a cabo todas las pruebas concernientes a los seis clasificadores mediante el método de validación cruzada para k = 3, 5 y 10 folds, se proceden a comparar los resultados obtenidos para las pruebas con k = 10 folds. Este factor de comparación se establece en virtud de que con este número de particiones se obtuvieron los mejores resultados del porcentaje de predicción respecto a la clasificación binaria realizada sobre los datos. En la Tabla 5.34 se presentan los mejores resultados obtenidos por cada clasificador, empleando las métricas disponibles en la fase 2 que son actualmente soportadas por el entorno BD.

Clasificador	Predicción Promedio	Segundos	MSE
MLP	88.36%	1409	0.17
LR	91.34%	74	0.13
LSVC	91.52%	438	0.09
MNB	81.60%	60	0.16
Nota: Cada una de las pruebas referidas se realizó diez veces *(k=10-fold cross validation)*			

Tabla 5.34 Mejores resultados de los clasificadores evaluados para 10 folds - Fase 2

Como se observa en la Tabla 5.34, el clasificador que obtiene el mejor porcentaje de predicción es el clasificador LSVC con un 91.52% promedio de acierto, mientras que el clasificador MNB es el que presenta el menor porcentaje con un 81.60% promedio de acierto, no obstante, este clasificador es el que realiza la tarea de clasificación en el menor tiempo: sólo 60 segundos. Por otra parte, en cuanto a la relación Predicción/Tiempo el clasificador LR es el que presenta los mejores valores al ser el clasificador con el segundo mejor promedio con el segundo mejor tiempo. Mientras que el clasificador MLP es el que puede considerarse el menos favorable para este tipo de tareas de clasificación al haberse tomado la mayor cantidad de tiempo para efectuar la misma tarea de clasificación que el resto de los clasificadores habiendo obtenido el tercer mejor promedio de predicción.

En la Figura 5.9 se presentan de forma gráfica todos los resultados obtenidos por los cuatro clasificadores habiéndose llevado a cabo las 15 pruebas establecidas. En dicha figura se resaltan además los mejores resultados que se obtuvieron por clasificador. Como

se observa en la Figura 5.9, los mejores resultados se comienzan a obtener a partir de la prueba 7, después de haber estado experimentando con algunos parámetros en las pruebas previas.

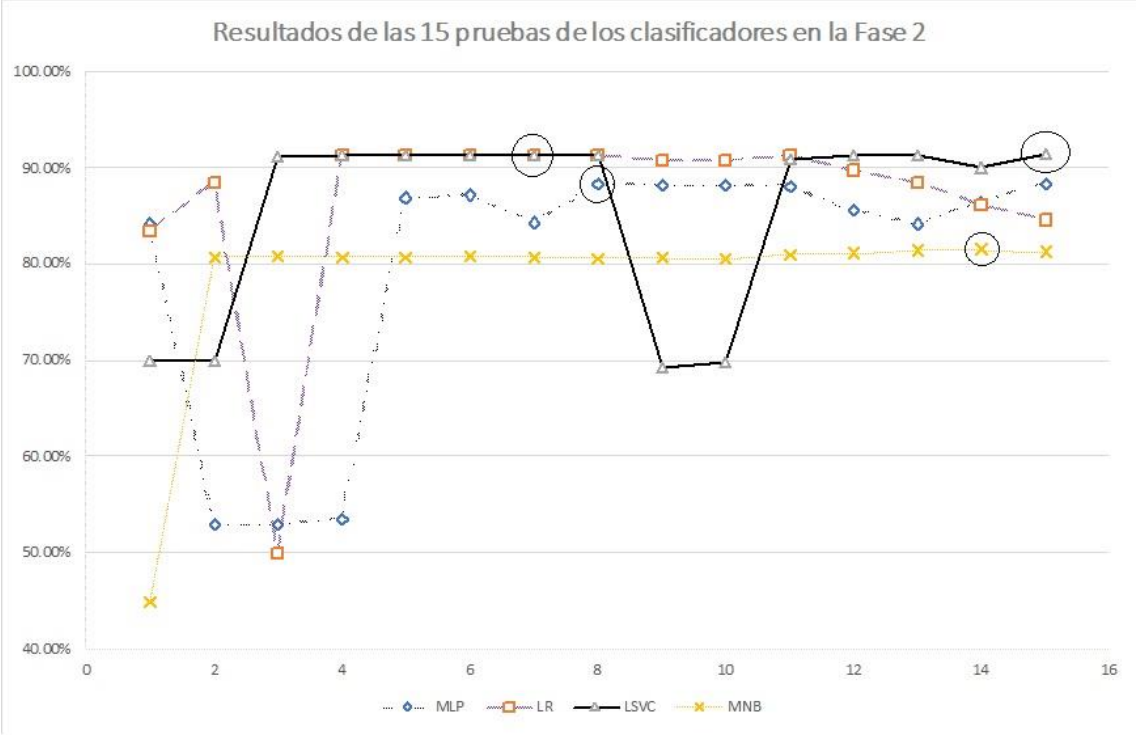

Figura 5.9 Desviación de los diversos clasificadores para el porcentaje de acierto y resultados conseguidos en las mejores pruebas

Por lo que se concluye que para el conjunto de datos utilizado en las pruebas se puede emplear el clasificador LSVC si lo que se desea es obtener el mejor porcentaje de predicción, si se necesita realizar la tarea en el menor tiempo se puede emplear el clasificador MNB, mientras que si se persigue obtener el mejor porcentaje de predicción en el menor tiempo posible, el clasificador LR es la mejor alternativa de uso. En la investigación futura se podrían probar otros conjuntos de datos para este tipo de tareas (clasificación binaria de textos) en este u otros entornos de BD y entonces evaluar si se mantienen los respectivos promedios.

5.5.9. Medidas de calidad de las alternativas. Fase 2

Para contrastar los procesos de clasificación sobre la información de las propuestas estudiadas se incorporan criterios estadísticos fundados en las pruebas de Anova & Test CRM, también conocida como la prueba de rangos múltiples de Duncan (Onezi, Khalifa, El-Metwally, & Househ, 2018) con un 95% de confianza. En la Figura 5.4 se presentó el

esquema de la prueba de contraste de hipótesis realizado durante la validación de los resultados cuando la población es mayor a 2.

El análisis de la varianza (ANOVA, ANalysis Of VAriance por su terminología en inglés) (Fisher, 1950) se origina a partir de los conceptos de regresión lineal. El ANOVA permite determinar si diversos procesos presentan diferencias significativas o si puede suponerse que las medias poblacionales no difieren. Por lo que el método ANOVA admite la contrastación de la hipótesis nula H_0 en la cual las medias de n poblaciones (siendo n > 2) son similares, respecto a la hipótesis alternativa en la que como mínimo, una población es diferente al resto respecto al valor que se espera.

En lo que respecta a la prueba CRM también conocida como Contraste de Rangos Múltiples (Duncan, 1955), es un método que pertenece a la clase general de procedimientos de comparación múltiple que emplea el rango estadístico estudiado q_r para comparar conjuntos de medias. El resultado de la prueba indica que las medias son significativamente distintas entre sí. La prueba CRM también se caracteriza por realizar todas las comparaciones por parejas.

En la Tabla 5.34 se presentaron los resultados obtenidos por cada clasificador, como se ha comentado, el clasificador LSVC es el que obtuvo el mejor porcentaje de predicción durante las pruebas de la segunda fase. En la Figura 5.10 se presenta el gráfico de cajas y bigotes asociado a los resultados obtenidos por cada clasificador en cuanto al factor de predicción. Las diferentes cajas muestran que existen diversos valores de medición alcanzados por los clasificadores, obteniéndose en algunos casos puntos externos con excepción del clasificador LSVC. La mayoría de las cajas presentan valores en el rango del 0.8 al 0.9, además de que las cajas presentan asimetría positiva o negativa.

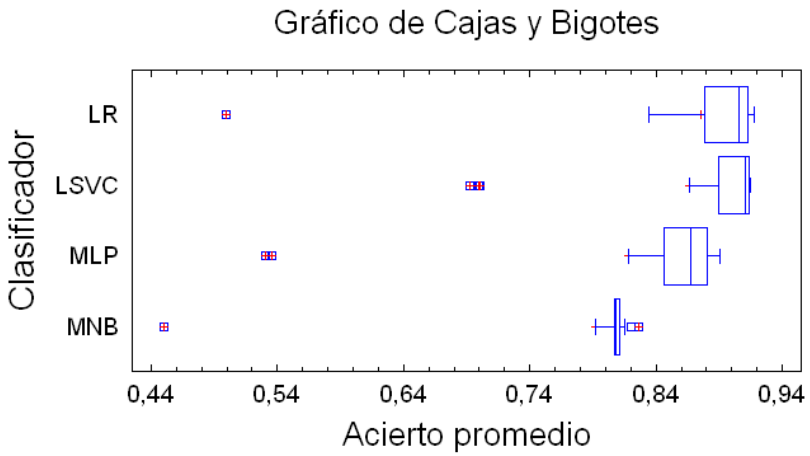

Figura 5.10 Gráfico de cajas y bigotes para el factor de predicción - Fase 2

En la Figura 5.11 se presenta el gráfico de análisis de medias por cada clasificador así como la media global y un límite de decisión igual al 95%. Todos los clasificadores están dentro de los límites de decisión además de ser distintos a la media global.

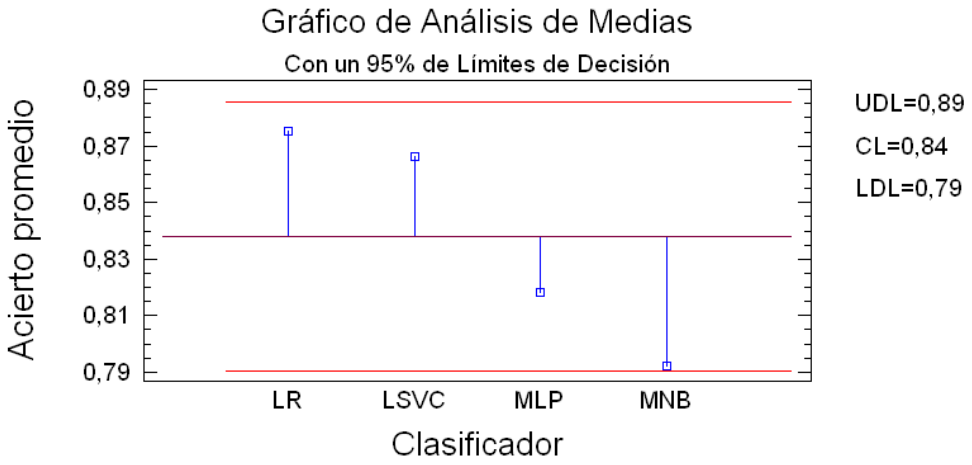

Figura 5.11 Gráfico de análisis de media para los escenarios de predicción - Fase 2

En la Figura 5.12 se tiene el diagrama de dispersión para observar si hubiese modelos de comportamiento, tendencias y observaciones independientes. Como se puede apreciar, se tiene una gran dispersión en todos los clasificadores con excepción de LSVC. Sin embargo, se puede afirmar que existe menor dispersión en la mayoría de los clasificadores respecto a los valores superiores de las predicciones registradas.

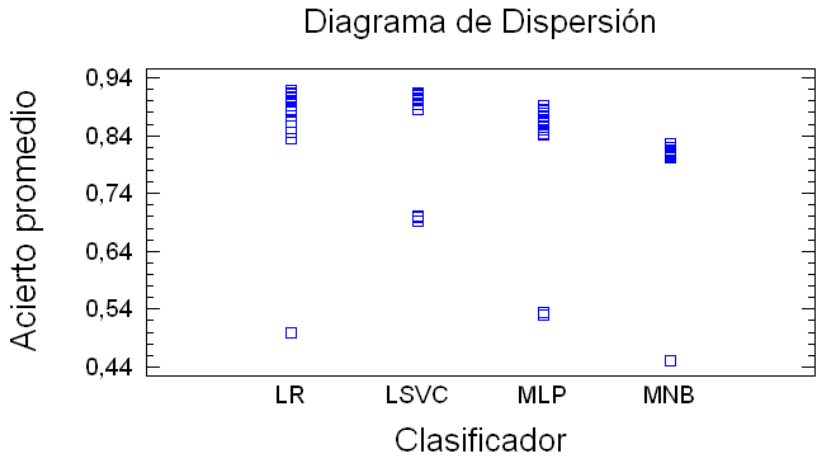

Figura 5.12 Diagrama de dispersión sobre acierto promedio - Fase 2

En la Figura 5.13 se tiene la gráfica de residuos de los algoritmos empleados. Los residuos son similares a los porcentajes de acierto menos el valor medio del grupo del cual se originan y exhiben que el cambio en los clasificadores es casi idéntico; siendo LSVC el que registra el menor cambio.

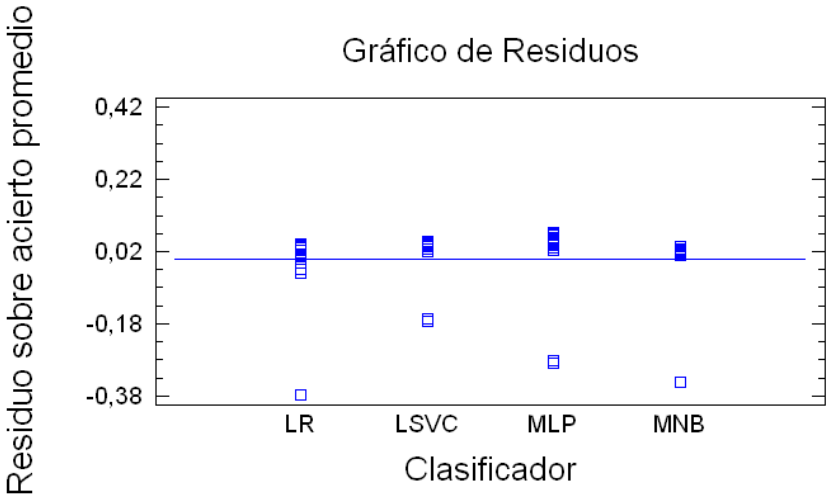

Figura 5.13 Diagrama de residuos sobre acierto promedio - Fase 2

El cuarto estadístico presentado en la Tabla 5.35, demuestra la hipótesis nula en la cual la desviación típica del p-Valor en los 4 niveles del Valor, es idéntica. También es interesante ver los p-valores, pueso que el menor de los tres es mayor o igual a 0,05, no se encuentra diferencia estadísticamente significativa en las desviaciones típicas cuando el nivel de confianza es igual a 95,0%.

Contraste de Varianza		
	Valor	p-Valor
Contraste C de Cochran	0.40775	0.073561
Contraste de Bartlett	1.06052	0.224219
Contraste de Hartley	2.36642	
Test de Levene	0.62585	0.600493

Tabla 5.35 Análisis aplicando el contraste de la varianza - Fase 2

La Tabla 5.36 descompone la varianza de las Sumas de cuadrados en dos elementos: un elemento entre grupos y un elemento dentro de esos grupos. El Cociente-F con valor de 3.30204, es el cociente de la estimación entre grupos y la estimación dentro de los grupos. Dado que en el test F el p-valor es menor de 0.05, existe desigualdad

estadísticamente significativa entre la media del porcentaje de acierto de cada algoritmo con un 95.0 % del nivel de confianza.

Fuente	Sumas de cuadrados	Grados Libertad	Cuadrado Medio	Cociente-F	P-Valor
Entre grupos (SCE)	0.093673	3	0.031224	3.30	0.0247
Intra grupos (SCR)	0.718663	76	0.009456		
Total (SCT)	0.812336	79			

Tabla 5.36 Análisis aplicando el Método ANOVA

Para establecer la diferencia entre las medias, se llevó a cabo el test CRM presentado en la Tabla 5.37. En la tabla se emplea un método de comparación múltiple para establecer la diferencia entre las medias. La parte inferior de la salida presenta la desigualdad considerada entre pares de medias. Junto a los 2 pares hay un asterisco indicando que éstos presentan diferencias estadísticamente significativas con un 95,0% del nivel de confianza.

Contraste	Diferencias	+/-Límites
LR - LSVC	0.00902	0.06125
LR - MLP	0.05693	0.06125
LR - MNB	*0.08321	0.06125
LSVC - MLP	0.04791	0.06125
LSVC - MNB	*0.07419	0.06125
MLP - MNB	0.02628	0.06125
* indica una diferencia significativa		

Tabla 5.37 Análisis mediante el Test CRM

En la Tabla 5.38, ordenados por la media, se reconocen dos grupos semejantes según

la alineación del símbolo X en la columna. Los niveles que tienen X en las columnas crean un grupo de medias en las que no existen diferencias estadísticamente significativas. El procedimiento empleado para distinguir entre las medias es el de las menores diferencias significativas de Fisher. Con dicho procedimiento, existe un riesgo de 5.0% de tomar a cuenta los pares de medias como distintos cuando la diferencia real es 0.

Clasificador	Media (%)	Grupos homogéneos
MNB	79.206	X
MLP	81.833	XX
LSVC	86.624	X
LR	87.526	X

Tabla 5.38 Análisis aplicando Grupos homogéneos

El análisis estadístico realizado en este apartado valida los resultados presentados en la Tabla 5.34 la cual concentra todos los resultados conseguidos en las pruebas efectuadas. Como lo han probado los resultados del clasificador LR, no sólo son mayores en media, sino que son distintos y presenta igualdad con los clasificadores MLP y LSVC. Si se considera el estimador de referencia, la media del porcentaje de acierto, el rendimiento del clasificador LR es el más preciso.

Los resultados presentados en la Tabla 5.34 pueden ser obtenidos debido a que el entorno BD empleado proporciona indicadores sobre el rendimiento de los clasificadores como el promedio de la precisión (accuracy) obtenida y el valor MSE. Lo cual valida la hipótesis "1. Es posible definir un framework en el que se puedan aplicar diversos indicadores de estudio independientemente del volumen de datos a analizar". Como se ha demostrado a lo largo del Apartado 5.5, en todos los experimentos realizados se tuvieron que emplear las métricas de estudio descritas anteriormente, lo cual ha permitido evaluar esta fase del modelo tanto a nivel individual por cada clasificador como de forma general, validando la hipótesis 1.

De igual manera, con estos experimentos ha quedado validada la hipótesis 3 que establece: "Los modelos de aprendizaje automático definidos para trabajar en un entorno con un volumen reducido de información se pueden extrapolar a un entorno GVDNE para realizar el mismo tipo de análisis". Es decir, que el MLM creado durante la primera fase pudo ser puesto a prueba en la segunda, para efectuar el mismo análisis que, en ambos

casos, fue el de analizar la información disponible para llevar a cabo tareas de clasificación binaria.

5.6. Comparación de ambas fases

Una vez que se han completado las tareas de análisis y clasificación de la información para las dos fases del framework, es posible realizar una comparación del rendimiento máximo alcanzado por los algoritmos evaluados en ambas fases. En la Tabla 5.39 se presenta la relación de los niveles máximos de predicción obtenidos por los clasificadores: MLP, LR, LSVC y MNB. Cabe recordar que aunque por un lado se tiene la cuestión que algunos parámetros de ajuste de los algoritmos no están presentes en ambas plataformas, por el otro, se está evaluando el mismo algoritmo base para la clasificación.

Clasificador	Fase	Predicción Promedio	Segundos	MSE Promedio / MSE
MLP	**1**	**87.21%**	1280	2.04
MLP	2	88.36%	1409	0.17
LR	1	86.96%	12	2.08
LR	2	91.34%	74	0.13
LSVC	1	87.10%	9	2.07
LSVC	**2**	**91.52%**	438	0.09
MNB	**1**	82.18%	**1.2**	2.86
MNB	**2**	81.60%	**60**	0.16

Tabla 5.39 Comparación de los mejores resultados obtenidos en las dos fases

Como se puede apreciar en la Tabla 5.39, en la Fase 1 el clasificador con el mejor promedio es MLP, mientras que MNB es el que registró el menor tiempo de clasificación para ambas fases. Por su parte, en la Fase 2 se tiene que el clasificador LSVC es el que registra el mejor promedio de predicción de las dos fases, mientras que MNB es el que efectúa esta tarea en el menor tiempo. En ambas fases el clasificador MLP es el que registra el mayor tiempo de ejecución para completar la tarea predictiva, mientras que MNB es el que lo realiza en el menor tiempo, tomando como base de comparación el rendimiento dentro de su propio entorno.

De forma general se podría afirmar que las tareas de clasificación no requieren tanto tiempo de cómputo, quedando para la investigación futura el repetir estos experimentos con un volumen mayor de información y observar su comportamiento y tiempos en la predicción. En la Figura 5.14 se ilustra de forma gráfica la comparativa de estos clasificadores donde se puede observar a priori que los clasificadores de la segunda fase obtuvieron mayores porcentajes de acierto que aquellos correspondientes a la primera fase. En la investigación futura se tendrán que evaluar nuevamente estos resultados cuando las implementaciones de los algoritmos sean más similares y entonces poder establecer si las diferencias permanecen o han aumentado para alguna de las fases o en alguno de los clasificadores.

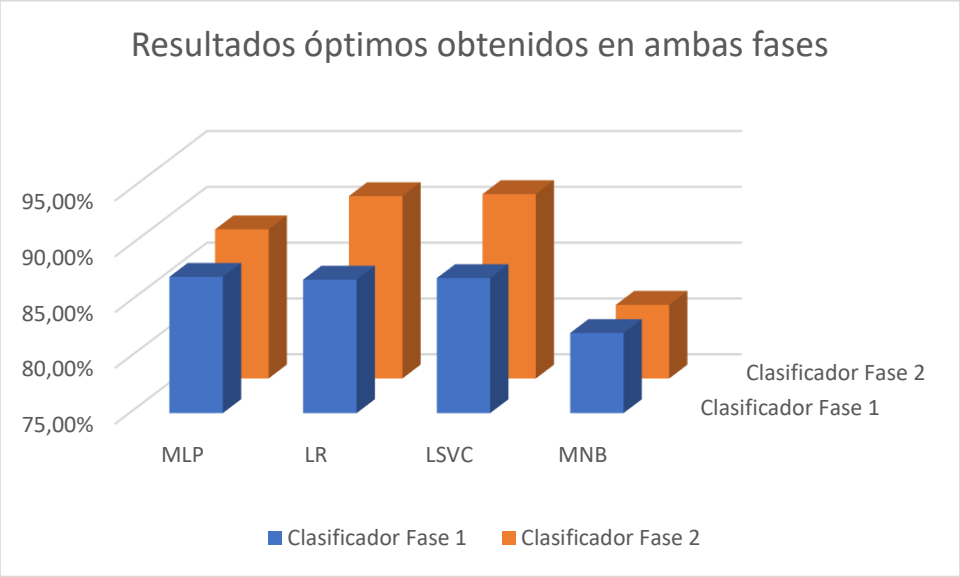

Figura 5.14 Resultados óptimos obtenidos por los clasificadores en común de las dos fases del framework

En un entorno real de predicción ya sea en una u otra fase, se tendrá que evaluar si se privilegia emplear uno u otro algoritmo en función del porcentaje de predicción alcanzado o del tiempo de procesamiento requerido para terminar dicha tarea. Por otra parte, si se compara el rendimiento del clasificador LSVC en ambas fases contra el resto de clasificadores, se observa que de manera general este es el mejor clasificador que se evaluó para las tareas de clasificación en los conjuntos de prueba disponibles.

Los resultados presentados en la Tabla 5.39 pueden ser obtenidos debido a que tanto el entorno no BD como el entorno BD empleados proporcionan indicadores sobre el rendimiento de los clasificadores como el promedio de la precisión (accuracy) obtenida y el valor MSE. Lo cual valida la hipótesis “1. Es posible definir un framework en el que se puedan aplicar diversos indicadores de estudio independientemente del volumen de datos a analizar”. En este apartado se concentran y reflejan los resultados obtenidos en

los experimentos realizados en ambas fases para lo cual se tuvieron que emplear las métricas de estudio mencionadas. Lo anterior ha permitido evaluar los clasificadores en común para ambas fases del modelo, dando lugar a la validación de la hipótesis 1.

Tras haber analizado los resultados obtenidos a partir de los experimentos llevados a cabo en las dos fases se valida la hipótesis 2 que indica lo siguiente: "Las técnicas de análisis de aprendizaje automático propuestas por el framework detectan patrones que permiten establecer predicciones sobre los datos de cualquier volumen". Las técnicas de análisis a las que se refiere esta hipótesis son precisamente las que emplean clasificadores de ML. Mientras que los patrones a los que se hace mención están referidos a descubrir en las clases que se generaron (positivos y negativos), aquellos términos (features) que permiten diferenciar una clase de otra. De tal manera que las predicciones referidas puedan ser aplicadas al dominio en estudio, en este caso el de la hotelería.

Por otra parte, la hipótesis 3 establece lo siguiente: "Los modelos de aprendizaje automático definidos para trabajar en un entorno con un volumen reducido de información se pueden extrapolar a un entorno GVDNE para realizar el mismo tipo de análisis", tras lo demostrado en este Apartado, se ha establecido la comprobación de esta hipótesis. La anterior afirmación se basa en el hecho de que en la Tabla 5.39 y la Figura 5.14 se aporta evidencia de que el MLM construido en la primera fase pudo ser empleado en la segunda, para realizar el mismo tipo de análisis mediante técnicas de GVDNE.

5.7. Visualización de datos

Esta es la última etapa del framework, sobre la cual se ha establecido anteriormente su importancia en los apartados 3.4.16, 3.4.17 y 4.3.5 tanto desde una perspectiva teórica como dentro de su pertinencia dentro del modelo propuesto. Aunque existen diversas herramientas de inteligencia de negocios (Business Intelligence) en el mercado como Tableau, Qlik o SAS, que ciertamente ofrecen mayores funcionalidades de visualización, se prefirió emplear librerías de Python debido a que éstas emplean el procesamiento y la transformación que se ha realizado con los datos para poder presentar los resultados del análisis de una forma intuitiva de manera que ésta pueda ser captada por gente no experta en el área.

Para determinar qué tipo de visualizaciones se deseaban obtener, uno de los primeros objetivos fue establecer qué información podría ser vital o importante de conocer en un entorno real. De esta manera se planteó dar respuesta a las siguientes preguntas: ¿cuál es la polaridad del sentimiento que tienen realmente los clientes dentro de las cinco categorías originales de evaluación? y ¿cuál es la polaridad del sentimiento que tienen los clientes cuando esta es agrupada para realizar la clasificación binaria? Con esta finalidad se trabajó con dos fuentes de información: para las cinco clases, aquella donde se tenían todas las clases con un mismo número de instancias; y para la clasificación binaria se

emplea la misma fuente con que se llevó a cabo la clasificación de las dos fases expuesta anteriormente en este capítulo.

El diagrama de caja en la Figura 5.15 muestra la distribución del puntaje de sentimiento para cinco clases, donde 1 es absoluto positivo y -1 es absoluto negativo. Como se puede ver en dicha figura, las revisiones calificadas con una estrella tienen un puntaje de sentimiento neutral ligeramente inclinado hacia negativo. Para las estrellas subsecuentes se observa que a medida que aumenta la calificación también lo hace el sentimiento. Esta figura también permite analizar la información para los aspectos cualitativos, ya que podría interpretarse como: los usuarios que otorgaron la calificación de una estrella tienen un sentimiento bastante neutral hacia la unidad de negocios

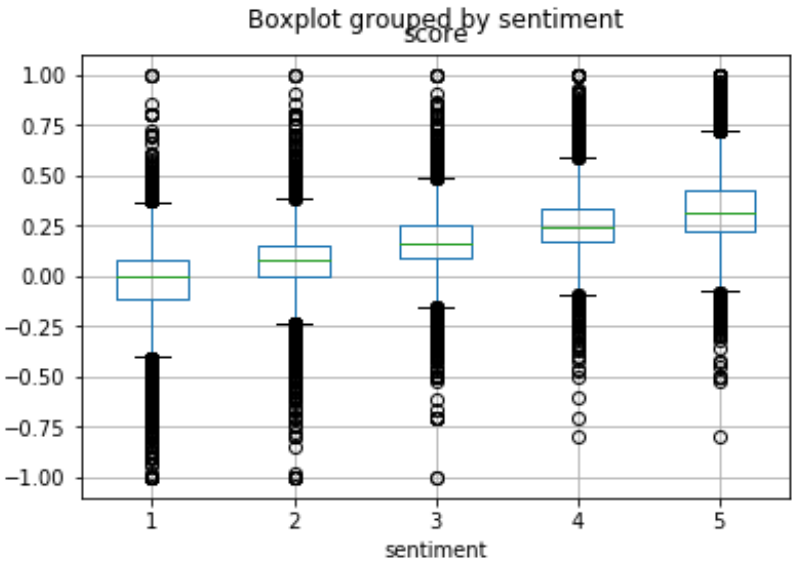

Figura 5.15 Boxplot que indica la polaridad real del sentimiento para cinco clases o estrellas[15]

Por otro lado, la polaridad del sentimiento observada para una estrella no es tan negativa, lo que indica que existen áreas de oportunidad en las cuáles es posible mejorar el servicio al cliente. Por otro lado, las reviews que tienen una calificación de 5 estrellas tienen un puntaje de sentimiento en el área positiva que apenas alcanza la mitad del área positiva absoluta, lo que también implica que muchos servicios podrían mejorarse en el área de atención al cliente aún para aquellos negocios que hayan recibido críticas favorables.

Con respecto a la clasificación binaria, la Figura 5.16 muestra la distribución de la

[15] Esta imagen fue tomada del artículo presentado por (Jimenez-Marquez, Gonzalez-Carrasco, Lopez-Cuadrado, & Ruiz-Mezcua, 2019)

puntuación de sentimiento para categorías negativas (1 estrella) y positivas (5 estrellas). Esta figura también permite procesar datos para aspectos cuantitativos: la representación de la clase 1 (una y dos estrellas) es muy similar a la representación de la clase 1 en la Figura 5.15 que indica, según el peso TFIDF de la palabra, qué tan similares pueden ser las reviews de una y dos estrellas. Para la clase 5 (tres, cuatro y cinco estrellas), la Figura 5.16 muestra que a pesar de que la agrupación de estas revisiones tiene un puntaje de sentimiento positivo, el sentimiento general no alcanza la mitad del área positiva absoluta.

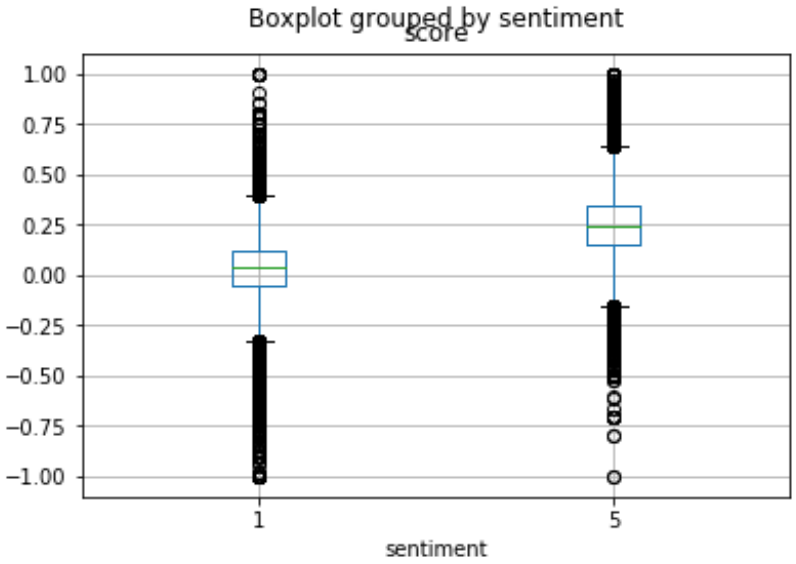

Figura 5.16 Boxplot que indica la polaridad real del sentimiento para dos clases o estrellas[16]

Para determinar la siguiente visualización se planteó la siguiente pregunta: ¿cuáles son las principales palabras que los clientes expresan en sus comentarios? Para poder dar respuesta a esta interrogante se exploraron los coeficientes TFIDF de los features utilizando el clasificador LR, que se presenta en la Figura 5.17. Estos resultados muestran las palabras principales que los usuarios expresan con respecto a los servicios. En esta figura, 0 representa neutralidad, mientras que los valores negativos representan las palabras más negativas mencionadas en las opiniones; el lado derecho de la figura muestra las palabras más positivas incluidas en las opiniones. En otras palabras, se podría decir que esta figura presenta el análisis de la información tanto cualitativa como cuantitativamente. Las palabras debajo de las barras son las palabras derivadas en su forma de raíz (*stem*).

[16] Esta imagen fue tomada del artículo presentado por (Jimenez-Marquez, Gonzalez-Carrasco, Lopez-Cuadrado, & Ruiz-Mezcua, 2019)

La Figura 5.17 es la que presenta más información sobre los aspectos cualitativos de los datos. Como se ve en esta figura, ciertos términos como 'belleza', 'grande' o 'amor' reflejan una imagen positiva del negocio como la expresan los usuarios, mientras que términos como 'peor', 'horrible' o 'terrible' expresan decepción cuando se alojan en un hotel o utilizan ciertos servicios de este. También en esta figura se muestra la parte superior de los mejores y peores términos encontrados en el corpus desde un aspecto cualitativo, sin embargo, durante la experimentación de datos, otros términos relacionados con circunstancias específicas como "aire acondicionado" o "estacionamiento" se encontraron con coeficientes de TFIDF más bajos (Jimenez-Marquez et al., 2019).

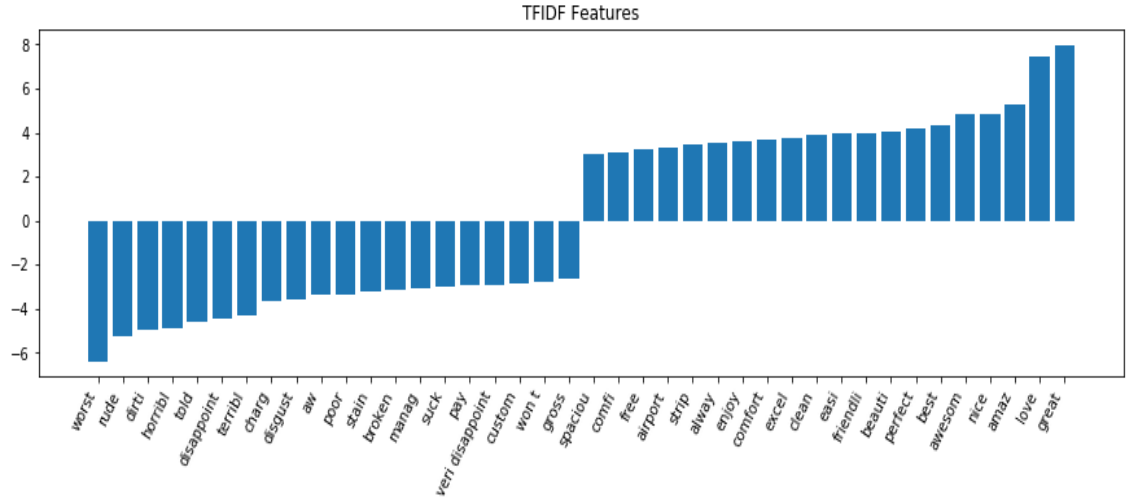

Figura 5.17 Representación de los términos con peor y mejor referencia por los usuarios[17]

Por último, en la Figura 5.18 se tiene un gráfico que representa los términos empleados con mayor frecuencia en las críticas vertidas por los usuarios. Esta es una manera muy descriptiva de presentar las palabras más empleadas en las críticas puesto que los términos más frecuentes se presentan con el tamaño de fuente más alto, y van disminuyendo hasta los menos frecuentes mientras que el tamaño del gráfico permita su representación. La ventaja que presenta este tipo de representaciones es que se puede apreciar de forma instantánea cuáles son los términos empleados con mayor frecuencia, además de que pueden ser obtenidos a partir de diversas manipulaciones de los datos, como mostrar sólo los términos más positivos o más negativos.

[17] Esta imagen fue tomada del artículo presentado por (Jimenez-Marquez, Gonzalez-Carrasco, Lopez-Cuadrado, & Ruiz-Mezcua, 2019)

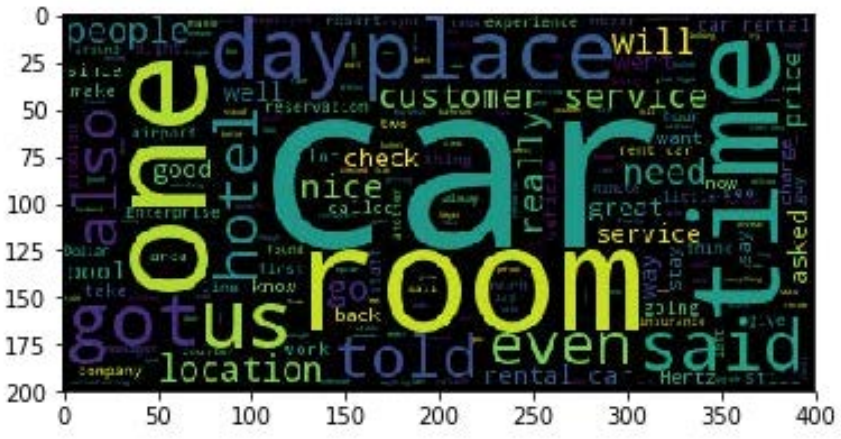

Figura 5.18 Representación de nube de palabras para los términos empleados con más frecuencia

El haber obtenido las visualizaciones presentadas anteriormente también valida la hipótesis 2 definida como: “Las técnicas de análisis de aprendizaje automático propuestas por el framework detectan patrones que permiten establecer predicciones sobre los datos de cualquier volumen”. Esto en función de que los gráficos presentados en la Figura 5.15 o la Figura 5.17 fueron obtenidos a partir del análisis previo que se llevó a cabo con los clasificadores de ML, los cuáles permitieron encontrar patrones en los datos, independientemente del volumen de estos.

5.8. Sumario

Como lo establece el framework, se siguió una serie de etapas divididas en dos fases para el análisis de BSD. El MLM construido en la primera fase mediante el empleo de diversos clasificadores de ML, tiene la intención de servir como base para el análisis que se lleva a cabo en el entorno BD de la segunda fase. Esta suposición se fundamenta en el hecho de que las dos fases comparten el mismo propósito: analizar un conjunto de datos no estructurados en entornos tecnológicos distintos. El framework no está vinculado a un proveedor, metodología o algoritmo de ML específicos, lo cual es su característica más distintiva con respecto a otros modelos existentes de BSD.

Por otra parte, se observó que no todos los métodos aplicados en la fase del entorno no BD pueden ser directamente escalables a las técnicas de análisis del entorno BD, dado que no existe una correspondencia total de los algoritmos de ML, así como de sus

implementaciones internas. Debido a lo anterior, en la investigación futura se deberá poner a prueba nuevamente el modelo propuesto, cuando los entornos no BD y BD tengan implementaciones semejantes de los algoritmos de ML empleados y evaluar sus resultados.

En este capítulo se ha descrito la validación de las tres hipótesis que se plantearon al inicio de la investigación. La primera de las hipótesis planteadas afirmaba que en el framework propuesto se podrían aplicar diversos indicadores de estudio independientemente del volumen de datos a analizar. Esta hipótesis se ha validado al presentar los resultados del porcentaje de acierto de predicción obtenidos por los diversos clasificadores, pues fueron obtenidos empleando diversas métricas.

La segunda hipótesis planteaba que las técnicas de análisis de aprendizaje automático propuestas por el framework detectan patrones que permiten establecer predicciones sobre los datos de cualquier volumen. Esta hipótesis se ha validado presentando los resultados de la clasificación binaria, donde las técnicas de análisis permitieron obtener los patrones que identifican a una crítica como positiva o negativa, empleando técnicas que pueden ser empleadas tanto en volúmenes reducidos como grandes de datos.

La tercera hipótesis suponía que los MLM definidos en un entorno no BD pueden aplicarse en un entorno GVDNE para realizar el mismo análisis sobre los datos. Esta hipótesis se ha validado al presentar un conjunto de algoritmos de ML que en un principio fueron aplicados en la primera fase, y posteriormente pudieron ser aplicados en la segunda para analizar la información.

Con estas hipótesis se ha comprobado la capacidad del framework planteado al principio de la investigación para poder establecer predicciones basadas en el análisis y la comparación de datos cualitativos contra cuantitativos en dos entornos tecnológicos que se diferencian por el volumen (grande o reducido) de datos que se puede procesar en cada uno.

Capítulo 6. Conclusiones y Trabajo Futuro

Este capítulo expone las conclusiones de la investigación llevada a cabo durante la realización de esta tesis doctoral. Además de hacer una revisión de los aportes primordiales de la tesis, se plantean una serie de futuras líneas de investigación que pueden servir como continuidad a dicha tesis. Por último, se listan las publicaciones que se llevaron a cabo con motivo de esta investigación.

6.1. Conclusiones

Al comienzo de la presente tesis se destacó el aumento en la cantidad de datos que se están generando cada segundo. Ante este escenario, los proveedores de soluciones tecnológicas tienen que innovar cada día para poder ofrecer productos que den soporte a la gestión y análisis de datos a gran escala. Debido a esta vertiginosidad, es complejo desde una perspectiva científica el poder utilizar o evaluar cada herramienta disponible en la oferta tecnológica para solucionar algún problema particular. Por lo que se detectó la necesidad de aportar a la ciencia modelos que reúnan las características de las mejores prácticas empleadas en el manejo de bases de datos tradicionales, combinándolas con técnicas modernas de análisis de datos, como el ML, para establecer un marco en el cual puedan ser integradas las herramientas tecnológicas que sean requeridas.

Por otra parte, se destacó el hecho de que los datos que se están generando actualmente provienen de diversas fuentes, entre estos los que constituyen el foco de interés provienen de las redes sociales, conteniendo texto libre sobre diversas temáticas y en contextos dispares. Ante esto, se requiere una categoría particular de modelos que puedan realizar el análisis de este tipo de información, ya que el texto y los demás atributos o variables que le acompañan pueden proporcionar detalles sobre la información que normalmente no se obtiene mediante el empleo de técnicas convencionales. Por lo que se planteó que, mediante técnicas de ML, PLN y BD se pueden conseguir los objetivos citados a través de la definición de un framework para el análisis predictivo de datos no estructurados.

En el estado del arte se trataron los trabajos más destacados que competen a las áreas transversales que conciernen a esta investigación, primero comparándolas una contra otra y al final las tres áreas, resaltando que en la intersección de estas se encuentra el framework planteado en la tesis. Posteriormente, en el estado de la técnica se presentan los diversos elementos que componen el framework, como se pudo apreciar en este

capítulo, cada etapa tiene un sustento teórico por detrás en las que diversas disciplinas pueden entrar en acción dependiendo de la tarea a realizar. Por otra parte, se observa que a pesar de que cada etapa es una actividad individual *per se* con un objetivo claro, estas se encuentran integradas para dar soporte a la etapa siguiente; y así la primera fase también da soporte a la segunda. Por lo que se establece que las diversas teorías que componen el framework han sido cuidadosamente seleccionadas para poder resolver el problema establecido.

Con respecto a la solución propuesta, se presentó el framework desde un punto de vista conceptual en el cual se detallan las etapas y fases que lo integran. Como se pudo apreciar, ambas fases coexisten de forma independiente en lo que respecta al entorno tecnológico, pero a su vez la fase uno proporciona un MLM a la segunda fase en el que se integra una propuesta tanto para el preprocesamiento de los datos empleando PLN, como una serie de algoritmos de ML para la clasificación del texto.

En lo que respecta a la evaluación y validación del modelo, se desarrollaron una serie de pruebas tanto en la fase no BD como en la fase BD para comparar el rendimiento del MLM que se construyó en la primera fase y se empleó en la segunda. Para esto se configuraron los clasificadores de acuerdo a lo recomendado en el estado del arte en base a autores que hubiesen comprobado los algoritmos evaluados. Esta serie de pruebas permitieron la posterior validación de las hipótesis planteadas al principio de la investigación, como se desarrolló a lo largo del Capítulo 5.

Los objetivos planteados al inicio de la investigación se cumplieron satisfactoriamente como se explica a continuación: en primera se instancia se especificó que el framework sería construido con el objetivo de poder establecer predicciones basadas en el análisis y la comparación de datos cualitativos contra cuantitativos, empleando principalmente técnicas de ML, BD y PLN, lo cual quedó demostrado tanto en la solución propuesta como en la validación del modelo. Posteriormente, se obtuvo la información a utilizar en la investigación y a partir de esta se dedujo el corpus que, como se ha explicado, se escogió el de hotelería por su amplia aplicabilidad y adopción en otros ámbitos de investigación.

Continuando con los objetivos, se aplicaron las técnicas que se establecieron al inicio de la investigación para poder resolver el problema. Por último, como se exploró en el capítulo anterior, se compararon los resultados obtenidos en ambas fases para comprobar la eficacia del framework al ser puesto a prueba bajo dos ambientes tecnológicos diferentes empleando herramientas y técnicas distintas. De tal manera que los objetivos planteados además de haber sido cubiertos satisfactoriamente durante la investigación contribuyeron a resolver el problema planteado al inicio de la tesis doctoral.

Los principales resultados y contribuciones de la tesis se enlistan a continuación:

1. Se cuenta con un marco de referencia, y a la vez un framework teórico que permite organizar un conjunto de etapas que tienen como fin común el análisis de información no estructurada, primordialmente textual o de lenguaje natural.
2. Aunque en el estado del arte existen modelos que realizan funciones parecidas a las presentadas en esta investigación, estos por lo general están ligados a ciertas técnicas para resolver un problema específico. Por lo que se puede afirmar que el framework es novedoso en función de que las técnicas a utilizar para resolver los problemas de análisis de la información quedan abiertas en el sentido de que las técnicas propuestas pueden ser sustituidas con el fin de mejorar los resultados.
3. En el framework se propone efectuar el análisis de los datos en dos fases, de tal manera que al realizar un proyecto de investigación que involucre decidir el establecer o contratar una arquitectura BD, se pueda llevar a cabo dicho análisis previamente en un entorno no BD. Una vez que se hayan obtenido resultados intermedios, se puede llevar el MLM al entorno BD. Esto además representa la ventaja de que se podrá comprobar si el análisis de los datos requiere técnicas propias de BD en función del volumen de estos y las capacidades de hardware que sean requeridas.
4. El framework es libre de dominio en el sentido de que el análisis que se realizó en el MLM puede ser aplicado a dominios que compartan las características de la información analizada (cualitativa y cuantitativa), variando algunos elementos durante el preprocesamiento del texto y ajustando el entrenamiento de los algoritmos de ML.

En un principio se planteó que el MLM construido durante la primera fase se empleara en la segunda (con sus respectivos detalles técnicos). El estado actual de la técnica contempla que se tengan realizar ajustes a ambos modelos tanto en la primera como en la segunda fase al momento de la implementación. No obstante, las plataformas tecnológicas no BD y BD están en constante evolución debido al impacto actual que se ha generado por la demanda de más recursos para el análisis de datos. Por lo que es posible que en el futuro cercano puedan llegar a converger en más áreas y aspectos técnicos que los que se tienen actualmente.

El análisis sobre cómo los usuarios califican los servicios o sus experiencias de cliente tanto cuantitativa como cualitativamente es un tema recurrente en ciencias de la computación. Los aspectos complejos que existen en la subjetividad de las revisiones de texto y las diversas características existentes en los conjuntos de datos (precio, ubicación, horarios de apertura, influenciadores, etc.) hacen que el análisis de datos en la era del BD sea una tarea permanente y en evolución. Sólo las compañías que integren estas metodologías en sus estrategias comerciales pueden evolucionar y mantenerse por delante de los competidores en la era basada en datos, también conocida como la industria 4.0.

6.2. Futuras líneas de investigación

La investigación futura se deberá enfocar en aplicar el MLM resultante de la primera etapa en la segunda mediante técnicas y algoritmos de ML específicos de BD. Las investigaciones futuras también deberían poder explorar cómo ampliar el framework a través de la integración de métodos avanzados de ML, entre estos el aprendizaje profundo (deep learning). Además, se propone ampliar la investigación a través de un estudio que explore las diferencias entre los dos paradigmas de computación expuestos (no BD y BD) para especificar por qué estas diferencias impiden en ocasiones que se empleen los mismos métodos.

La investigación futura también deberá incluir variaciones en los algoritmos aplicados para considerar más categorías de sentimiento. Es decir, que se debe ampliar la cantidad de categorías para considerar todas aquellas que se encuentran relacionadas al texto mediante una etiqueta o clasificación categórica. Lo anterior abre interesantes líneas de investigación en el área de PLN y ML ya que, como se ha mencionado recurrentemente a lo largo de este trabajo, la subjetividad que se encuentra naturalmente ligada al texto (entendido este como: expresiones, opiniones, comentarios, etc.) es un área muy compleja de analizar, un ser humano tendría incluso resultados más bajos que un algoritmo de ML clasificando textos y relacionándolos con una etiqueta dada. Si a lo anterior se le agrega que este tipo de información se encuentra en muchas ocasiones relacionada con otros atributos, las posibilidades de hacer investigación desde distintos enfoques para extraer resultados diversos son casi infinitas.

Como se pudo ver en el capítulo anterior, las tareas de PLN de la primera fase, no se adaptaron de forma plena en la fase de BD. Por lo que la investigación actual podría ampliarse mediante la repetición de los experimentos de la primera fase y aplicar el MLM resultante en una arquitectura diferente de BD que cuente con una interfaz similar a la del entorno no BD empleado. Una extensión mayor de este trabajo sería comparar diversos entornos no BD contra diversos entornos BD.

Finalmente, una línea de investigación que se exploró durante esta tesis es: realizar pronósticos a partir de datos no estructurados con contenido textual. Los modelos existentes que generan pronósticos suelen emplear técnicas como series de tiempo o análisis de regresión, y comúnmente los datos empleados son de tipo numérico, perteneciendo estos a diversas categorías y orígenes. Sin embargo, emplear datos textuales dentro de este tipo de modelos podría ser de gran interés tanto para investigadores como para interesados en el área. Schneider & Gupta (2016) expresan lo siguiente: “los datos textuales a menudo se omiten de los modelos predictivos”. El saber lo que podría acontecer dentro de un periodo de tiempo corto o mediano a partir del análisis y el pronóstico de un conjunto de datos textuales sería de gran utilidad y valor en diversos ámbitos de la ciencia, la ingeniería y la tecnología.

6.3. Publicaciones realizadas

Derivado de los resultados obtenidos en esta investigación y el alcance planteado para la misma, se realizaron las siguientes publicaciones:

- Publicaciones en revista con factor de impacto.

  Title: Towards a Big Data Framework for Analyzing Social Media Content
  Journal: International Journal of Information Management
  Authors: Jose Luis Jimenez-Marquez, Israel Gonzalez-Carrasco, José Luis Lopez-Cuadrado, Belén Ruiz-Mezcua.
  Impact Factor JCR (2017): 4.516. Posición 3/88 Q1.
  Area: Information Science & Library Science.

  Este artículo presenta el framework en su conjunto desde un punto de vista teórico y de implementación. También presenta los resultados experimentales obtenidos después de haber completado la primera fase.

  Title: Challenges and Opportunities in Analytic-Predictive Environments of Big Data and Natural Language Processing for Social Network Rating Systems.
  Journal: IEEE Latin America Transactions.
  Authors: Jose Luis Jimenez-Marquez, Israel Gonzalez-Carrasco, José Luis Lopez-Cuadrado.
  Impact Factor JCR (2017): 0.502. Q4.
  Area: Computer Science, Information Systems.

  En este artículo se presenta el estado del arte sobre los trabajos elaborados hasta ese momento que conjuntan diversas áreas y técnicas sobre las que se fundamenta el framework.

- Publicaciones en congresos internacionales.
  - Getting the value out of data using high-level analysis techniques in the tourism sector. 2nd Global Conference on Applied Physics, Mathematics and Computing (2018), Madrid, Spain.

  En este congreso se dio a conocer el proyecto de investigación que se estaba elaborando, así como los resultados obtenidos hasta ese momento.

  - Towards a common architecture for social data analysis. 2nd Global Conference on Applied Computing in Science & Engineering (2017), Islas Canarias, Spain.

  En este congreso se presentó el proyecto en una etapa inicial de la investigación, en la cual además de plantear una primera aproximación al framework, se dieron a conocer los aspectos teóricos relevantes del mismo.

Bibliografía

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., … Zheng, X. (2016). TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems. *CoRR*, *abs/1603.0*. Retrieved from http://arxiv.org/abs/1603.04467

Agerri, R., Artola, X., Beloki, Z., Rigau, G., & Soroa, A. (2015). Big data for Natural Language Processing: A streaming approach. *Knowledge-Based Systems*, *79*, 36–42. https://doi.org/https://doi.org/10.1016/j.knosys.2014.11.007

Ahmad, M., Aftab, S., Muhammad, S., & Waheed, U. (2017). Tools and Techniques for Lexicon Driven Sentiment Analysis: A Review. *International Journal of Multidisciplinary Sciences and Engineering*, *8*, 17–23.

Ain, Q. T., Ali, M., Riaz, A., Noureen, A., Kamran, M., Hayat, B., & Rehman, A. (2017). Sentiment Analysis Using Deep Learning Techniques: A Review. *International Journal of Advanced Computer Science and Applications*, *8*(6), 424–433.

Akram, S. K. W., Raheman, M., Jagadeesh, N., Teja, P. V., & Krishna, R. S. (2018). Prediction of Service Rating by Exploring Behavior of User's From Social Websites. *Iconic Research And Engineering Journals*, *1*(10), 37–41.

Al-Smadi, M., Qawasmeh, O., Al-Ayyoub, M., Jararweh, Y., & Gupta, B. (2018). Deep Recurrent neural network vs. support vector machine for aspect-based sentiment analysis of Arabic hotels' reviews. *Journal of Computational Science*, *27*, 386–393. https://doi.org/https://doi.org/10.1016/j.jocs.2017.11.006

Alahmadi, D. H., & Zeng, X.-J. (2015). ISTS: Implicit social trust and sentiment based approach to recommender systems. *Expert Systems with Applications*, *42*(22), 8840–8849. https://doi.org/10.1016/j.eswa.2015.07.036

Alapati, S. R. (2018). *Expert Apache Cassandra Administration*. Apress. https://doi.org/10.1007/978-1-4842-3126-5

Allahbakhsh, M., Ignjatovic, A., Benatallah, B., Beheshti, S.-M.-R., Foo, N., & Bertino, E. (2014). Representation and querying of unfair evaluations in social rating systems. *Computers & Security*, *41*, 68–88. https://doi.org/https://doi.org/10.1016/j.cose.2013.09.008

Amine, A., Elberrichi, Z., & Simonet, M. (2010). Evaluation of Text Clustering Methods Using WordNet. *International Arab Journal of Information Technology*, *7*(4), 349–357. Retrieved from http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.178.4261&rep=rep1&type=pdf

Amith, M., He, Z., Bian, J., Lossio-Ventura, J. A., & Tao, C. (2018). Assessing the practice of biomedical ontology evaluation: Gaps and opportunities. *Journal of Biomedical Informatics*, *80*, 1–13. https://doi.org/https://doi.org/10.1016/j.jbi.2018.02.010

Andersson, B.; Beals, B. (2018). Big Data in Oil and Gas. Retrieved January 1, 2017, from https://www.slideshare.net/bjorna/big-data-in-oil-and-gas

Andrade, D., Tamura, A., & Tsuchida, M. (2018). Exploiting covariate embeddings for classification using Gaussian processes. *Pattern Recognition Letters*, *104*, 8–14. https://doi.org/https://doi.org/10.1016/j.patrec.2018.01.011

Araque, O., Corcuera-Platas, I., Sánchez-Rada, J. F., & Iglesias, C. A. (2017). Enhancing deep learning sentiment analysis with ensemble techniques in social applications. *Expert Systems with Applications*, *77*, 236–246. https://doi.org/10.1016/j.eswa.2017.02.002

Arbib, M. A. (1995). Brain theory and neural networks. *MIT Press, Cambridge, MA*.

Armbrust, M., Ghodsi, A., Zaharia, M., Xin, R. S., Lian, C., Huai, Y., … Franklin, M. J. (2015). Spark SQL: Relational Data Processing in Spark. *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data - SIGMOD '15*, 1383–1394. https://doi.org/10.1145/2723372.2742797

Artola, X., Beloki, Z., & Soroa, A. A stream computing approach towards scalable NLP (2014).

Atzeni, P., Bugiotti, F., Cabibbo, L., & Torlone, R. (2016). Data modeling in the NoSQL world. *Computer Standards & Interfaces*, 11. https://doi.org/https://doi.org/10.1016/j.csi.2016.10.003

Bagchi, S. (2015). Performance and Quality Assessment of Similarity Measures in Collaborative Filtering Using Mahout. *Procedia Computer Science*, *50*, 229–234. https://doi.org/https://doi.org/10.1016/j.procs.2015.04.055

Bansal, H., Chauhan, S., & Mehrotra, S. (2016). *Apache Hive Cookbook*. Packt Publishing Ltd.

Barber, D. (2017). *Bayesian Reasoning and Machine Learning*.

Barker, R. (1990). *Case*Method: Entity Relationship Modelling*. Addison-Wesley.

Baruh, L., & Popescu, M. (2017). Big data analytics and the limits of privacy self-management. *New Media & Society*, *19*(4), 579–596. https://doi.org/10.1177/1461444815614001

Bello-Orgaz, G., Jung, J. J., & Camacho, D. (2016). Social big data: Recent achievements and new challenges. *Information Fusion*, *28*, 45–59. https://doi.org/https://doi.org/10.1016/j.inffus.2015.08.005

Bengfort, B., & Kim, J. (2016). *Data Analytics with Hadoop*. O'Reilly Media.

Bilal, M., Oyedele, L. O., Qadir, J., Munir, K., Ajayi, S. O., Akinade, O. O., … Pasha, M. (2016). Big Data in the construction industry: A review of present status, opportunities, and future trends. *Advanced Engineering Informatics*, *30*(3), 500–521. https://doi.org/https://doi.org/10.1016/j.aei.2016.07.001

Bird, S., Klein, E., & Loper, E. (2009). *Natural Language Processing with Python*. O' Reilly.

Brink, H., Richards, J. W., & Fetherolf, M. (2017). *Real World Machine Learning*. Manning Publications Co.

Byrd, R. H., Lu, P., Nocedal, J., & Zhu, C. (1995). A Limited Memory Algorithm for Bound Constrained Optimization. *SIAM Journal on Scientific Computing*, *16*(5), 1190–1208. https://doi.org/10.1137/0916069

Calì, D., Condorelli, A., Papa, S., Rata, M., & Zagarella, L. (2011). Improving intelligence through use of Natural Language Processing. A comparison between NLP interfaces and traditional visual GIS interfaces. *Procedia Computer Science*, *5*, 920–925. https://doi.org/10.1016/j.procs.2011.07.128

Cankurt, S., & Subasi, A. (2015). Developing tourism demand forecasting models using machine learning techniques with trend , seasonal , and cyclic components. *Balkan Journal of Electrical & Computer Engineering*, *3*(1), 42–49.

Cao, L., Sun, Y., Wang, S., & Li, M. (2016). Detecting malicious behavior and collusion for online rating system. In *2016 IEEE Trustcom/BigDataSE/ISPA* (pp. 1046–1053). https://doi.org/10.1109/TrustCom.2016.0174

Carcillo, F., Dal Pozzolo, A., Le Borgne, Y. A., Caelen, O., Mazzer, Y., & Bontempi, G. (2018). SCARFF: A scalable framework for streaming credit card fraud detection with spark. *Information Fusion*, *41*, 182–194. https://doi.org/10.1016/j.inffus.2017.09.005

Carneiro, N., Figueira, G., & Costa, M. (2017). A data mining based system for credit-card fraud detection in e-tail. *Decision Support Systems*, *95*, 91–101. https://doi.org/https://doi.org/10.1016/j.dss.2017.01.002

Carpenter, J., & Hewitt, E. (2016). *Cassandra: The Definitive Guide*. O'Reilly Media.

Chang, C.-C., & Lin, C.-J. (2011). LIBSVM: A Library for Support Vector Machines. *ACM Trans. Intell. Syst. Technol.*, *2*(3), 27:1--27:27. https://doi.org/10.1145/1961189.1961199

Chang, Y., Ku, C., & Chen, C. (2017). Social media analytics: Extracting and visualizing Hilton hotel ratings and reviews from TripAdvisor. *International Journal of Information Management*, (April), 1–17. https://doi.org/10.1016/j.ijinfomgt.2017.11.001

Chen, L., Yan, D., & Wang, F. (2017). User perception of sentiment-integrated critiquing in recommender systems. *International Journal of Human Computer Studies*, *000*(September), 1–17. https://doi.org/10.1016/j.ijhcs.2017.09.005

Chen, P. P.-S. (1976). The Entity-relationship Model - Toward a Unified View of Data. *ACM Trans. Database Syst.*, *1*(1), 9–36. https://doi.org/10.1145/320434.320440

Chen, Y., Fu, X., Yue, K., Liu, L., & Liu, L. (2016). Ranking Online Services by Aggregating Ordinal Preferences. In S. Song & Y. Tong (Eds.), *Web-Age Information Management* (pp. 41–53). Cham: Springer International Publishing.

Cho, J. Y., Jin, H. W., Lee, M., & Schwan, K. (2014). Dynamic core affinity for high-performance file upload on Hadoop Distributed File System. *Parallel Computing*, *40*(10), 722–737. https://doi.org/10.1016/j.parco.2014.07.005

Claveria, O., & Torra, S. (2014). Forecasting tourism demand to Catalonia: Neural networks vs. time series models. *Economic Modelling*, *36*(January 2014), 220–228. https://doi.org/10.1016/j.econmod.2013.09.024

Coelho, L. P., & Richert, W. (2015). *Building Machine Learning Systems with Python*. Packt Publishing Ltd. https://doi.org/10.1007/s13398-014-0173-7.2

Colace, F., De Santo, M., Greco, L., Moscato, V., & Picariello, A. (2015). A collaborative user-centered framework for recommending items in Online Social Networks. *Computers in Human Behavior*, *51*, 694–704. https://doi.org/10.1016/j.chb.2014.12.011

Cortes, C., & Vapnik, V. (1995). Support-Vector Networks. *Machine Learning*, *20*(3), 273–297. https://doi.org/10.1023/A:1022627411411

Cunha, J., Silva, C., & Antunes, M. (2015). Health Twitter Big Bata Management with Hadoop Framework. *Procedia Computer Science*, *64*, 425–431. https://doi.org/https://doi.org/10.1016/j.procs.2015.08.536

Cybulski, J. L., Keller, S., Nguyen, L., & Saundage, D. (2015). Creative problem solving in digital space using visual analytics. *Computers in Human Behavior*, *42*, 20–35. https://doi.org/https://doi.org/10.1016/j.chb.2013.10.061

Dali, H., & Yutaka, M. (2015). Predicting tourism trends through the data of online communications. In *The 29th Annual Conference of the Japanese Society for Artificial Intelligence* (pp. 1–4).

Dar, S. (2016). *A simulator for Spark scheduler*. Eindhoven.

Das, S., Behera, R. K., kumar, M., & Rath, S. K. (2018). Real-Time Sentiment Analysis of Twitter Streaming data for Stock Prediction. *Procedia Computer Science*, *132*, 956–964. https://doi.org/https://doi.org/10.1016/j.procs.2018.05.111

Demir-Kavuk, O., Kamada, M., Akutsu, T., & Knapp, E. W. (2011). Prediction using

step-wise L1, L2 regularization and feature selection for small data sets with large number of features. *BMC Bioinformatics*, *12*(1), 1–10. https://doi.org/10.1186/1471-2105-12-412

Deshpande, A., & Kumar, M. (2018). *Artificial Intelligence for Big Data* (First Edit). Birmingham, UK: Packt Publishing Ltd.

Dessì, D., Fenu, G., Marras, M., & Recupero, D. R. (2018). Bridging learning analytics and Cognitive Computing for Big Data classification in micro-learning video collections. *Computers in Human Behavior*. https://doi.org/https://doi.org/10.1016/j.chb.2018.03.004

Detours, V., Dumont, J. E., Bersini, H., & Maenhaut, C. (2003). Integration and cross-validation of high-throughput gene expression data: comparing heterogeneous data sets. *FEBS Letters*, *546*(1), 98–102. https://doi.org/https://doi.org/10.1016/S0014-5793(03)00522-2

Dhanalakshmi, V., Bino, D., & Saravanan, A. M. (2016). Opinion mining from student feedback data using supervised learning algorithms. In *2016 3rd MEC International Conference on Big Data and Smart City (ICBDSC)* (pp. 1–5). https://doi.org/10.1109/ICBDSC.2016.7460390

Do, T., & Tran-Nguyen, M.-T. (2016). Incremental Parallel Support Vector Machines for Classifying Large-Scale Multi-class Image Datasets. In *Future Data and Security Engineering. FDSE 2016. Lecture Notes in Computer Science* (Vol. 10018, pp. 20–39). https://doi.org/10.1007/978-3-319-48057-2

Dodd, M., Grant, A., & Seruwagi, L. (2011). *Artificial Intelligence Through the Eyes of the Public*.

Donchenko, D., Ovchar, N., Sadovnikova, N., Parygin, D., Shabalina, O., & Ather, D. (2017). Analysis of Comments of Users of Social Networks to Assess the Level of Social Tension. *Procedia Computer Science*, *119*, 359–367. https://doi.org/https://doi.org/10.1016/j.procs.2017.11.195

Donnelley, J. E. (1995). WWW media distribution via Hopwise Reliable Multicast. *Computer Networks and ISDN Systems*, *27*(6), 781–788. https://doi.org/10.1016/0169-7552(95)00048-C

Drabas, T., & Lee, D. (2017). Learning PySpark., 273.

Duncan, D. B. (1955). Multiple Range and Multiple F Tests. *Biometrics*, *11*(1), 1–42. Retrieved from http://www.jstor.org/stable/3001478

Duric, A., & Song, F. (2012). Feature selection for sentiment analysis based on content and syntax models. *Decision Support Systems*, *53*(4), 704–711. https://doi.org/10.1016/j.dss.2012.05.023

Duvvuri, S., & Singhal, B. (2016). *Spark for Data Science* (First Edit). Birmingham, UK:

Packt Publishing Ltd.

Ekmekci, M. (2011). Sustainable reputations with rating systems. *Journal of Economic Theory*, *146*(2), 479–503. https://doi.org/https://doi.org/10.1016/j.jet.2010.02.015

Elgendy, N., & Elragal, A. (2016). Big Data Analytics in Support of the Decision Making Process. *Procedia Computer Science*, *100*, 1071–1084. https://doi.org/https://doi.org/10.1016/j.procs.2016.09.251

Elshawi, R., Sakr, S., Talia, D., & Trunfio, P. (2018). Big Data Systems Meet Machine Learning Challenges: Towards Big Data Science as a Service. *Big Data Research*. https://doi.org/https://doi.org/10.1016/j.bdr.2018.04.004

Erevelles, S., Fukawa, N., & Swayne, L. (2016). Big Data consumer analytics and the transformation of marketing. *Journal of Business Research*, *69*(2), 897–904. https://doi.org/https://doi.org/10.1016/j.jbusres.2015.07.001

Ertemel, A. V. (2015). Consumer Insight as Competitive Advantage Using Big Data and Analytics. *International Journal of Commerce and Finance*, *1*(1), 45–51.

Evelson, B., & Yuhanna, N. (2012). *The Forrester Wave™: Advanced Data Visualization (ADV) Platforms, Q3 2012*. Retrieved from https://www.forrester.com/report/The+Forrester+Wave+Advanced+Data+Visualization+ADV+Platforms+Q3+2012/-/E-RES71903

Fan, R.-E., Chang, K.-W., Hsieh, C.-J., Wang, X.-R., & Lin, C.-J. (2008). LIBLINEAR: A Library for Large Linear Classification. *Journal of Machine Learning Research*, *9*(2008), 1871–1874. https://doi.org/10.1038/oby.2011.351

Fasale, A., & Kumar, N. (2015). *YARN Essentials*. PACKT Publishing.

Fayyad, U., Piatetsky-Shapiro, G., & Smyth, P. (1996). From data mining to knowledge discovery in databases. *AI Magazine*, 37–54. https://doi.org/10.1145/240455.240463

Felbermayr, A., & Nanopoulos, A. (2016). The Role of Emotions for the Perceived Usefulness in Online Customer Reviews. *Journal of Interactive Marketing*, *36*, 60–76. https://doi.org/https://doi.org/10.1016/j.intmar.2016.05.004

Feng, G., Guo, J., Jing, B.-Y., & Sun, T. (2015). Feature subset selection using naive Bayes for text classification. *Pattern Recognition Letters*, *65*, 109–115. https://doi.org/https://doi.org/10.1016/j.patrec.2015.07.028

Finlay, S. (2014). *Predictive Analytics, Data Mining and Big Data. Myths, Misconceptions and Methods.* Palgrave Macmillan.

Fisher, R. A. (1950). *Statistical methods for research workers* (11th ed.(r). Edinburgh: Oliver & Boyd.

Flath, C. M., & Stein, N. (2018). Towards a data science toolbox for industrial analytics

applications. *Computers in Industry*, *94*, 16–25. https://doi.org/10.1016/j.compind.2017.09.003

Forbes. (2017). Poor-Quality Data Imposes Costs and Risks on Businesses, Says New Forbes Insights Report. Retrieved August 29, 2018, from https://www.forbes.com/sites/forbespr/2017/05/31/poor-quality-data-imposes-costs-and-risks-on-businesses-says-new-forbes-insights-report/#5641d1a8452b

Freund, Y., & Schapire, R. E. (1999). Large Margin Classification Using the Perceptron Algorithm. *Machine Learning*, *37*(3), 277–296.

Gantz, J., & Reinsel, D. (2011). *Extracting Value from Chaos*.

García-Gil, D., Ramírez-Gallego, S., García, S., & Herrera, F. (2017). A comparison on scalability for batch big data processing on Apache Spark and Apache Flink. *Big Data Analytics*, *2*(1), 11. https://doi.org/10.1186/s41044-016-0020-2

García-Pablos, A., Cuadros, M., & Rigau, G. (2018). W2VLDA: Almost unsupervised system for Aspect Based Sentiment Analysis. *Expert Systems with Applications*, *91*, 127–137. https://doi.org/https://doi.org/10.1016/j.eswa.2017.08.049

García-Pedrajas, N., & Ortiz-Boyer, D. (2011). An empirical study of binary classifier fusion methods for multiclass classification. *Information Fusion*, *12*(2), 111–130. https://doi.org/10.1016/j.inffus.2010.06.010

Gates, A. (2011). *Programming Pig*. O'Reilly Media.

Gates, A., Natkovich, O., Chopra, S., Kamath, P., Narayanamurthy, S. M., Olston, C., … Srivastava, U. (2009). Building a High-level Dataflow System on Top of Map-Reduce: The Pig Experience. *Proceedings of the VLDB Endowment*, *2*(2), 1414–1425. https://doi.org/10.14778/1687553.1687568

Geetha, M., Singha, P., & Sinha, S. (2017). Relationship between customer sentiment and online customer ratings for hotels - An empirical analysis. *Tourism Management*, *61*, 43–54. https://doi.org/10.1016/j.tourman.2016.12.022

Ghaddar, B., & Naoum-Sawaya, J. (2018). High dimensional data classification and feature selection using support vector machines. *European Journal of Operational Research*, *265*(3), 993–1004. https://doi.org/10.1016/j.ejor.2017.08.040

Ghiassi, M., Skinner, J., & Zimbra, D. (2013). Twitter brand sentiment analysis : A hybrid system using n -gram analysis and dynamic artificial neural network. *Expert Systems With Applications*, *40*(16), 6266–6282. https://doi.org/10.1016/j.eswa.2013.05.057

Giatsoglou, M., Vozalis, M. G., Diamantaras, K., Vakali, A., Sarigiannidis, G., & Chatzisavvas, K. C. (2017). Sentiment analysis leveraging emotions and word embeddings. *Expert Systems with Applications*, *69*, 214–224. https://doi.org/https://doi.org/10.1016/j.eswa.2016.10.043

Glenski, M., Stoddard, G., Resnick, P., & Weninger, T. (2018). GuessTheKarma : A Game to Assess Social Rating Systems. In *Proceedings of the ACM on Human-Computer Interaction,* (Vol. 2).

González-Carrasco, I. (2010). *Análisis, Optimización y Evaluación de Modelos de Redes de Neuronas Artificiales para la Clasificación y Predicción de Impactos de Alta Velocidad sobre Distintos Materiales*. Universidad Carlos III de Madrid, Madrid.

Gordon, C. (2016). Decoding Buzzwords: Big Data, Predictive Analytics, Business Intelligence. In *The Big Analytics* (pp. 137–140).

Gross, A., & Murthy, D. (2014). Modeling virtual organizations with Latent Dirichlet Allocation: A case for natural language processing. *Neural Networks*, *58*, 38–49. https://doi.org/https://doi.org/10.1016/j.neunet.2014.05.008

Gudivada, V. N., Rao, D., & Raghavan, V. V. (2015). Big Data Driven Natural Language Processing Research and Applications. In *Handbook of Statistics* (Vol. 33, pp. 203–238). Elsevier Inc. https://doi.org/10.1016/B978-0-444-63492-4.00009-5

Guerrero, J. I., García, A., Personal, E., Luque, J., & León, C. (2017). Heterogeneous data source integration for smart grid ecosystems based on metadata mining. *Expert Systems with Applications*, *79*, 254–268. https://doi.org/https://doi.org/10.1016/j.eswa.2017.03.007

Guo, Y., Barnes, S. J., & Jia, Q. (2017). Mining meaning from online ratings and reviews: Tourist satisfaction analysis using latent dirichlet allocation. *Tourism Management*, *59*, 467–483. https://doi.org/10.1016/j.tourman.2016.09.009

Habib, M., Chang, V., Batool, A., & Ying, T. (2016). Big data reduction framework for value creation in sustainable enterprises. *International Journal of Information Management*, *36*(6), 917–928. https://doi.org/10.1016/j.ijinfomgt.2016.05.013

Halpin, T., & Morgan, T. (2008). 8 - Entity Relationship Modeling. In T. Halpin & T. Morgan (Eds.), *Information Modeling and Relational Databases (Second Edition)* (Second Edi, pp. 305–343). San Francisco: Morgan Kaufmann. https://doi.org/https://doi.org/10.1016/B978-012373568-3.50012-6

Hosmer, D. W., & Lemeshow, S. (2000). *Applied Logistic Regression*. *Wiley Series in Probability and Sattistics*. https://doi.org/10.2307/2074954

Hossin, M., & Sulaiman, M. N. (2015). A Review on Evaluation Metrics for Data Classification Evaluations. *International Journal of Data Mining & Knowledge Management Process (IJDKP)*, *5*(2), 1–11. https://doi.org/10.5121/ijdkp.2015.5201

Hossin, M., Sulaiman, M. N., Mustapha, A., Mustapha, N., & Rahmat, R. W. (2011). A hybrid evaluation metric for optimizing classifier. In *2011 3rd Conference on Data Mining and Optimization (DMO)* (pp. 165–170). https://doi.org/10.1109/DMO.2011.5976522

Hu, H., Wen, Y., Chua, T., & Li, X. (2014). Toward Scalable Systems for Big Data Analytics: A Technology Tutorial. *IEEE Access*, *2*, 652–687. https://doi.org/10.1109/ACCESS.2014.2332453

Iqbal, R., Doctor, F., More, B., Mahmud, S., & Yousuf, U. (2016). Big Data analytics: Computational intelligence techniques and application areas. *International Journal of Information Management*. https://doi.org/10.1016/j.ijinfomgt.2016.05.020

Ishwarappa, & Anuradha, J. (2015). A brief introduction on Big Data 5Vs characteristics and Hadoop technology. In *International Conference on Intelligent Computing, Communication & Convergence* (Vol. 48, pp. 319–324). https://doi.org/10.1016/j.procs.2015.04.188

Jiang, D., Luo, X., Xuan, J., & Xu, Z. (2017). Sentiment Computing for the News Event Based on the Social Media Big Data. *IEEE Access*, *5*, 2373–2382. https://doi.org/10.1109/ACCESS.2016.2607218

Jiang, L., Li, C., Wang, S., & Zhang, L. (2016). Deep feature weighting for naive Bayes and its application to text classification. *Engineering Applications of Artificial Intelligence*, *52*, 26–39. https://doi.org/https://doi.org/10.1016/j.engappai.2016.02.002

Jimenez-Marquez, J. L., Gonzalez-Carrasco, I., & Lopez-Cuadrado, J. L. (2018). Challenges and Opportunities in Analytic-Predictive Environments of Big Data and Natural Language Processing for Social Network Rating Systems. *IEEE Latin America Transactions*, *16*(2), 592–597. https://doi.org/10.1109/TLA.2018.8327417

Jimenez-Marquez, J. L., Gonzalez-Carrasco, I., Lopez-Cuadrado, J. L., & Ruiz-Mezcua, B. (2019). Towards a big data framework for analyzing social media content. *International Journal of Information Management*, *44*, 1–12. https://doi.org/https://doi.org/10.1016/j.ijinfomgt.2018.09.003

Joachims, T. (1998). Text categorization with Support Vector Machines: Learning with many relevant features. In C. Nédellec & C. Rouveirol (Eds.), *Machine Learning: ECML-98* (pp. 137–142). Berlin, Heidelberg: Springer Berlin Heidelberg.

Kakadia, D. (2015). *Apache Mesos Essentials*. Packt Publishing Ltd.

Kang, M., Ahn, J., & Lee, K. (2018). Opinion mining using ensemble text hidden Markov models for text classification. *Expert Systems with Applications*, *94*, 218–227. https://doi.org/https://doi.org/10.1016/j.eswa.2017.07.019

Karau, H., Konwinski, A., Wendell, P., & Zaharia, M. (2015). *Learning Spark*. O'Reilly Media, Inc.

Kim, J., Han, M., Lee, Y., & Park, Y. (2016). Futuristic data-driven scenario building: Incorporating text mining and fuzzy association rule mining into fuzzy cognitive map. *Expert Systems with Applications*, *57*, 311–323.

https://doi.org/10.1016/j.eswa.2016.03.043

Kim, W., Choi, B.-J., Hong, E.-K., Kim, S.-K., & Lee, D. (2003). A Taxonomy of Dirty Data. *Data Mining and Knowledge Discovery*, *7*(1), 81–99. https://doi.org/10.1023/A:1021564703268

Kim, Y. (2014). Convolutional Neural Networks for Sentence Classification. *CoRR*, *abs/1408.5*. Retrieved from http://arxiv.org/abs/1408.5882

Kingma, D. P., & Ba, J. (2015). Adam: A Method for Stochastic Optimization. In *International Conference on Learning Representations* (pp. 1–15). https://doi.org/http://doi.acm.org.ezproxy.lib.ucf.edu/10.1145/1830483.1830503

Kreps, J., Narkhede, N., & Rao, J. (2011). Kafka: a Distributed Messaging System for Log Processing. In *NetDB workshop 2011* (p. 7).

Krishnan, S. P. T., & Ugia, J. (2015). *Building Your Next Big Thing with Google Cloud Platform*. Apress.

Kruskal, W. H., & Wallis, W. A. (1952). Use of Ranks in One-Criterion Variance Analysis. *Journal of the American Statistical Association*, *47*(260), 583–621. Retrieved from http://www.jstor.org/stable/2280779

Kwon, H., Park, Y., & Geum, Y. (2018). Toward data-driven idea generation: Application of Wikipedia to morphological analysis. *Technological Forecasting and Social Change*, *132*, 56–80. https://doi.org/https://doi.org/10.1016/j.techfore.2018.01.009

Lakshman, A., & Malik, P. (2010). Cassandra — A Decentralized Structured Storage System. *Operating Systems Review*, *44*, 35–40.

Landset, S., Khoshgoftaar, T. M., Richter, A. N., & Hasanin, T. (2015). A survey of open source tools for machine learning with big data in the Hadoop ecosystem. *Journal of Big Data*, *2*(1), 24. https://doi.org/10.1186/s40537-015-0032-1

Lara, J. A., Lizcano, D., Martínez, M., & Pazos, J. (2014). Data preparation for KDD through automatic reasoning based on description logic. *Information Systems*, *44*, 54–72. https://doi.org/10.1016/j.is.2014.03.002

Leavitt, N. (2010). Will NoSQL Databases Live Up to Their Promise? *Computer*, *43*(2), 12–14. https://doi.org/10.1109/MC.2010.58

Lee, L. H., & Isa, D. (2010). Automatically computed document dependent weighting factor facility for Naïve Bayes classification. *Expert Systems with Applications*, *37*(12), 8471–8478. https://doi.org/https://doi.org/10.1016/j.eswa.2010.05.030

Lee, S., & Choeh, J. Y. (2014). Predicting the helpfulness of online reviews using multilayer perceptron neural networks. *Expert Systems with Applications*, *41*(6), 3041–3046. https://doi.org/https://doi.org/10.1016/j.eswa.2013.10.034

Leong, L., & Chamberlin, T. (2010). *Magic Quadrant for Cloud Infrastructure as a*

Service and Web Hosting.

Li, F., Gui, Z., Wu, H., Gong, J., Wang, Y., Tian, S., & Zhang, J. (2018). Big enterprise registration data imputation: Supporting spatiotemporal analysis of industries in China. *Computers, Environment and Urban Systems*, *70*, 9–23. https://doi.org/https://doi.org/10.1016/j.compenvurbsys.2018.01.010

Li, J., Ott, M., & Varadarajan, B. (2013). Identifying manipulated offerings on review portals. In *2013 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Li, X., Pan, B., Law, R., & Huang, X. (2017). Forecasting tourism demand with composite search index. *Tourism Management*, *59*, 57–66. https://doi.org/10.1016/j.tourman.2016.07.005

Li, Y., Zhang, Z., Peng, Y., Yin, H., & Xu, Q. (2018). Matching user accounts based on user generated content across social networks. *Future Generation Computer Systems*, *83*, 104–115. https://doi.org/https://doi.org/10.1016/j.future.2018.01.041

Lin, T. Y. (T. Y. . (2002). Attribute transformations for data mining I: Theoretical explorations. *International Journal of Intelligent Systems*, *17*(2), 213–222. https://doi.org/10.1002/int.10017

Liu, H., Hussain, F., Tan, C. L., & Dash, M. (2002). Discretization: An Enabling Technique. *Data Mining and Knowledge Discovery*, *6*(4), 393–423. https://doi.org/10.1023/A:1016304305535

Liu, H., & Motoda, H. (1998). *Feature Selection for Knowledge Discovery and Data Mining*. Norwell, MA, USA: Kluwer Academic Publishers.

Liu, J., Liu, C., & Huang, Y. (2017). Multi-granularity sequence labeling model for acronym expansion identification. *Information Sciences*, *378*, 462–474. https://doi.org/https://doi.org/10.1016/j.ins.2016.06.045

Liu, K., & Dong, L. (2012). Research on Cloud Data Storage Technology and Its Architecture Implementation. *Procedia Engineering*, *29*, 133–137. https://doi.org/10.1016/j.proeng.2011.12.682

Liu, X.-L., Guo, Q., Hou, L., Cheng, C., & Liu, J.-G. (2015). Ranking online quality and reputation via the user activity. *Physica A: Statistical Mechanics and Its Applications*, *436*, 629–636. https://doi.org/https://doi.org/10.1016/j.physa.2015.05.043

Liu, Y., Teichert, T., Rossi, M., Li, H., & Hu, F. (2017). Big data for big insights: Investigating language-specific drivers of hotel satisfaction with 412,784 user-generated reviews. *Tourism Management*, *59*, 554–563. https://doi.org/10.1016/j.tourman.2016.08.012

Machanick, P. (2005). A distributed systems approach to secure Internet mail. *Computers*

& Security, *24*(6), 492–499. https://doi.org/10.1016/j.cose.2005.03.007

Maletti, A. (2016). Survey: Finite-state technology in natural language processing. *Theoretical Computer Science*, *679*, 1–16. https://doi.org/10.1016/j.tcs.2016.05.030

Manning, C. D., Bauer, J., Finkel, J., & Bethard, S. J. (2014). The Stanford CoreNLP Natural Language Processing Toolkit. In *52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations* (pp. 55–60).

Manning, C. D., Raghavan, P., & Schutze, H. (2008). *An Introduction to Information Retrieval*. https://doi.org/10.1109/LPT.2009.2020494

Manochandar, S., & Punniyamoorthy, M. (2018). Scaling feature selection method for enhancing the classification performance of Support Vector Machines in text mining. *Computers & Industrial Engineering*, *124*, 139–156. https://doi.org/https://doi.org/10.1016/j.cie.2018.07.008

Mäntylä, M. V., Graziotin, D., & Kuutila, M. (2018). The evolution of sentiment analysis—A review of research topics, venues, and top cited papers. *Computer Science Review*, *27*, 16–32. https://doi.org/10.1016/j.cosrev.2017.10.002

Marafino, B. J., Boscardin, W. J., & Dudley, R. A. (2015). Efficient and sparse feature selection for biomedical text classification via the elastic net: Application to ICU risk stratification from nursing notes. *Journal of Biomedical Informatics*, *54*, 114–120. https://doi.org/https://doi.org/10.1016/j.jbi.2015.02.003

Marine-Roig, E. (2017). Measuring destination image through travel reviews in search engines. *Sustainability*, *9*(8), 1–18. https://doi.org/10.3390/su9081425

Marine-Roig, E., & Anton-Clavé, S. (2015). Tourism analytics with massive user-generated content: A case study of Barcelona. *Journal of Destination Marketing & Management*, *4*(3), 162–172. https://doi.org/https://doi.org/10.1016/j.jdmm.2015.06.004

Mauro, A. De, Greco, M., & Grimaldi, M. (2015). What is big data ? A consensual definition and a review of key research topics. In *AIP Conference Proceedings* (Vol. 1644, pp. 97–104). https://doi.org/10.1063/1.4907823

Meng, X., Bradley, J., Yavuz, B., Sparks, E., Venkataraman, S., Liu, D., … Talwalkar, A. (2015). MLlib: Machine Learning in Apache Spark. *Journal of Machine Learning Research*, *17*, 1–7. https://doi.org/10.1145/2882903.2912565

Midgett, C., Bendickson, J. S., Muldoon, J., & Solomon, S. J. (2017). The Sharing Economy and Sustainability: A Case for Airbnb. *Small Business Institute Journal*, *13*(2), 51–71. Retrieved from https://sbij.org/index.php/SBIJ/article/view/265/222

Moraes, R., Valiati, J. F., & Gaviao Neto, W. P. (2013). Document-level sentiment classification: An empirical comparison between SVM and ANN. *Expert Systems with Applications*, *40*(2), 621–633. https://doi.org/10.1016/j.eswa.2012.07.059

Mueller, A. C., & Guido, S. (2016). *Introduction to Machine Learning with Python* (First Edit). O'Reilly Media, Inc.

Mueller, J. P., & Massaron, L. (2016). *Machine Learning For Dummies*. John Wiley & Sons, Inc.

Müller, O., Junglas, I., vom Brocke, J., & Debortoli, S. (2016). Utilizing big data analytics for information systems research: challenges, promises and guidelines. *European Journal of Information Systems*, *25*(4), 289–302. https://doi.org/10.1057/ejis.2016.2

Nadkarni, P. M., Ohno-Machado, L., & Chapman, W. W. (2011). Natural language processing: An introduction. *Journal of the American Medical Informatics Association*, *18*(5), 544–551. https://doi.org/10.1136/amiajnl-2011-000464

Nagalavi, D., & Hanumanthappa, M. (2016). N-gram Word Prediction Language Models to Identify the Sequence of Article Blocks in English E-Newspapers. In *2016 International Conference on Computational Systems and Information Systems for Sustainable Solutions* (pp. 307–311).

Nakov, P., Ritter, A., Rosenthal, S., Sebastiani, F., & Stoyanov, V. (2016). SemEval-2016 Task 4: Sentiment Analysis in Twitter. In *Proceedings of SemEval-2016* (Vol. 50, pp. 1–18). https://doi.org/10.1109/IISA.2016.7785373

Nema, W., & Tang, Y. (2017). Consensus-based ranking of wikipedia topics. In *Proceedings of the International Conference on Web Intelligence - WI '17* (pp. 114–124). https://doi.org/10.1145/3106426.3106529

Nesi, P., Pantaleo, G., & Sanesi, G. (2015). A hadoop based platform for natural language processing of web pages and documents. *Journal of Visual Languages & Computing*, *31*, 130–138. https://doi.org/https://doi.org/10.1016/j.jvlc.2015.10.017

Ngo-Ye, T. L., & Sinha, A. P. (2014). The influence of reviewer engagement characteristics on online review helpfulness: A text regression model. *Decision Support Systems*, *61*, 47–58. https://doi.org/https://doi.org/10.1016/j.dss.2014.01.011

Ojeda, T., Bilbro, R., & Bengfort, B. (2018). *Applied Text Analysis with Python*. O'Reilly Media.

Olmedilla, M., Martínez-Torres, M. R., & Toral, S. L. (2016). Harvesting Big Data in social science: A methodological approach for collecting online user-generated content. *Computer Standards and Interfaces*, *46*, 79–87. https://doi.org/10.1016/j.csi.2016.02.003

Olston, C., Reed, B., Srivastava, U., Kumar, R., & Tomkins, A. (2008). Pig Latin: A Not-so-foreign Language for Data Processing. In *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data* (pp. 1099–1110). New York, NY, USA: ACM. https://doi.org/10.1145/1376616.1376726

Onan, A., Korukoğlu, S., & Bulut, H. (2017). A hybrid ensemble pruning approach based on consensus clustering and multi-objective evolutionary algorithm for sentiment classification. *Information Processing & Management*, *53*(4), 814–833. https://doi.org/https://doi.org/10.1016/j.ipm.2017.02.008

Onezi, H. Al, Khalifa, M., El-Metwally, A., & Househ, M. (2018). The impact of social media-based support groups on smoking relapse prevention in Saudi Arabia. *Computer Methods and Programs in Biomedicine*, *159*, 135–143. https://doi.org/https://doi.org/10.1016/j.cmpb.2018.03.005

Ou, W., Huynh, V.-N., & Sriboonchitta, S. (2018). Training attractive attribute classifiers based on opinion features extracted from review data. *Electronic Commerce Research and Applications*, *32*, 13–22. https://doi.org/https://doi.org/10.1016/j.elerap.2018.10.003

Oussous, A., Benjelloun, F.-Z., Lahcen, A. A., & Belfkih, S. (2017). Big Data technologies: A survey. *Journal of King Saud University - Computer and Information Sciences*. https://doi.org/https://doi.org/10.1016/j.jksuci.2017.06.001

Owen, S., Anil, R., Dunning, T., & Friedman, E. (2011). *Mahout in Action*. Greenwich, CT, USA: Manning Publications Co.

Özköse, H., Sertac, E., & Gencer, C. (2015). Yesterday, Today and Tomorrow of Big Data. *Procedia - Social and Behavioral Sciences*, *195*, 1042 – 1050.

Pääkkönen, P., & Pakkala, D. (2015). Reference Architecture and Classification of Technologies, Products and Services for Big Data Systems. *Big Data Research*, *2*(4), 166–186. https://doi.org/https://doi.org/10.1016/j.bdr.2015.01.001

Pan, B., & Yang, Y. (2017). Forecasting Destination Weekly Hotel Occupancy with Big Data. *Journal of Travel Research*, *56*(7), 957–970. https://doi.org/10.1177/0047287516669050

Pang, B., & Lee, L. (2004). A Sentimental Education: Sentiment Analysis Using Subjectivity Summarization Based on Minimum Cuts. In *Proceedings of the Association for Computational Linguistics* (pp. 271–278). https://doi.org/10.3115/1218955.1218990

Pang, B., Lee, L., & Vaithyanathan, S. (2002). Thumbs up? Sentiment Classification using Machine Learning Techniques. *Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing - EMNLP '02*, *10*(July), 79–86. https://doi.org/10.3115/1118693.1118704

Pantano, E., Priporas, C. V., & Stylos, N. (2017). 'You will like it!' using open data to predict tourists' response to a tourist attraction. *Tourism Management*, *60*, 430–438. https://doi.org/10.1016/j.tourman.2016.12.020

Perrons, R. K., & McAuley, D. (2015). The case for "n«all": Why the Big Data revolution

will probably happen differently in the mining sector. *Resources Policy*, *46*, 234–238. https://doi.org/https://doi.org/10.1016/j.resourpol.2015.10.007

Piazza, A., & Davcheva, P. (2015). Sentiment Classification and Visualization of Product Review Data. In M. Hofmann & A. Chisholm (Eds.), *Text Mining, Web Mining, and Visualization Use Cases Using Open Source Tools* (pp. 133–152). Taylor & Francis Group.

Piryani, R., Madhavi, D., & Singh, V. K. (2017). Analytical mapping of opinion mining and sentiment analysis research during 2000–2015. *Information Processing and Management*, *53*(1), 122–150. https://doi.org/10.1016/j.ipm.2016.07.001

Porter, M. F. (1980). An algorithm for suffix stripping. *Program*, *14*(3), 130–137. https://doi.org/10.1108/eb046814

Prado-Daud, S., & Costa-Ribeiro, C. H. (2010). Natural language processing applied to the analysis of requirements. In *9th Brazilian Conference on Dynamics Control and their Applications* (pp. 1091–1098).

Pranckevicius, T., & Marcinkevičius, V. (2017). Comparison of Naive Bayes, Random Forest, Decision Tree, Support Vector Machines, and Logistic Regression Classifiers for Text Reviews Classification. *Baltic Journal of Modern Computing*, *5*. https://doi.org/10.22364/bjmc.2017.5.2.05

Pyle, D. (1999). *Data Preparation for Data Mining*. Morgan Kaufmann.

Qi, J., Qian, L., & Luo, Z. (2009). Distributed Structured Database System HugeTable. In M. G. Jaatun, G. Zhao, & C. Rong (Eds.), *Cloud Computing* (pp. 338–346). Berlin, Heidelberg: Springer Berlin Heidelberg.

Qi, J., Zhang, Z., Jeon, S., & Zhou, Y. (2016). Mining customer requirements from online reviews: A product improvement perspective. *Information & Management*, *53*(8), 951–963. https://doi.org/https://doi.org/10.1016/j.im.2016.06.002

Qiu, J., Liu, C., Li, Y., & Lin, Z. (2018). Leveraging sentiment analysis at the aspects level to predict ratings of reviews. *Information Sciences*, *451–452*, 295–309. https://doi.org/10.1016/j.ins.2018.04.009

Qiu, J., Wu, Q., Ding, G., Xu, Y., & Feng, S. (2016). A survey of machine learning for big data processing. *EURASIP Journal on Advances in Signal Processing*, *2016*(1), 67. https://doi.org/10.1186/s13634-016-0355-x

Rahman, M. N., Esmailpour, A., & Zhao, J. (2016). Machine Learning with Big Data An Efficient Electricity Generation Forecasting System. *Big Data Research*, *5*, 9–15. https://doi.org/https://doi.org/10.1016/j.bdr.2016.02.002

Ramesh, B., & Sathiaseelan, J. G. R. (2015). An Advanced Multi Class Instance Selection based Support Vector Machine for Text Classification. *Procedia Computer Science*, *57*, 1124–1130. https://doi.org/https://doi.org/10.1016/j.procs.2015.07.400

Ramon y Cajal, S. (1894). The Croonian lecture.—La fine structure des centres nerveux. *Proceedings of the Royal Society of London*, *55*(331–335), 444–468.

Ravi, K., & Ravi, V. (2015). A survey on opinion mining and sentiment analysis : Tasks , approaches and applications. *Knowledge-Based Systems*, *89*, 14–46. https://doi.org/10.1016/j.knosys.2015.06.015

Refaeilzadeh, P., Tang, L., & Liu, H. (2009). Cross-Validation. In L. LIU & M. T. ÖZSU (Eds.), *Encyclopedia of Database Systems* (pp. 532–538). Boston, MA: Springer US. https://doi.org/10.1007/978-0-387-39940-9_565

Reinartz, T. (2002). A Unifying View on Instance Selection. *Data Mining and Knowledge Discovery*, *6*(2), 191–210. https://doi.org/10.1023/A:1014047731786

Reuther, A., Byun, C., Arcand, W., Bestor, D., Bergeron, B., Hubbell, M., … Kepner, J. (2018). Scalable system scheduling for HPC and big data. *J. Parallel Distrib. Comput.*, *111*, 76–92. https://doi.org/10.1016/j.jpdc.2017.06.009

Rocha, L., Mourão, F., Mota, H., Salles, T., Gonçalves, M. A., & Jr., W. M. (2013). Temporal contexts: Effective text classification in evolving document collections. *Information Systems*, *38*(3), 388–409. https://doi.org/https://doi.org/10.1016/j.is.2012.11.001

Rosenthal, S., Farra, N., & Nakov, P. (2017). SemEval-2017 Task 4 : Sentiment Analysis in Twitter. In *Proceedings of SemEval-2017* (pp. 502–518).

Russell, S., & Norvig, P. (2013). *Artificial Intelligence A Modern Approach*. *Prentice Hall Series in Artificial Intelligence*. https://doi.org/10.1017/S0269888900007724

Ryza, S., Laserson, U., Owen, S., & Wills, J. (2015). *Advanced Analytics with Spark*. (A. Spencer, Ed.). O'Reilly Media.

Sadeghi, M., & Vegas, J. (2014). Automatic identification of light stop words for Persian information retrieval systems. *Journal of Information Science*, *40*(4), 1–12. https://doi.org/10.1177/0165551514530655

Saif, S., & Wazir, S. (2018). Performance Analysis of Big Data and Cloud Computing Techniques: A Survey. *Procedia Computer Science*, *132*, 118–127. https://doi.org/https://doi.org/10.1016/j.procs.2018.05.172

Salehan, M., & Kim, D. J. (2016). Predicting the performance of online consumer reviews: A sentiment mining approach to big data analytics. *Decision Support Systems*, *81*, 30–40. https://doi.org/https://doi.org/10.1016/j.dss.2015.10.006

Salles, T., Gonçalves, M., Rodrigues, V., & Rocha, L. (2018). Improving random forests by neighborhood projection for effective text classification. *Information Systems*, *77*, 1–21. https://doi.org/https://doi.org/10.1016/j.is.2018.05.006

Sangeetha, R., & Kalpana, B. (2011). Identifying Efficient Kernel Function in Multiclass

Support Vector Machines. *International Journal of Computer Applications*, *28*(8), 18–23.

Saravana-Kumar, N. M., Eswari, T., Sampath, P., & Lavanya, S. (2015). Predictive Methodology for Diabetic Data Analysis in Big Data. *Procedia Computer Science*, *50*, 203–208. https://doi.org/https://doi.org/10.1016/j.procs.2015.04.069

Sarkar, S., Vinay, S., Raj, R., Maiti, J., & Mitra, P. (2018). Application of optimized machine learning techniques for prediction of occupational accidents. *Computers and Operations Research*, *0*, 1–15. https://doi.org/10.1016/j.cor.2018.02.021

Saumya, S., Singh, J. P., Baabdullah, A. M., Rana, N. P., & Dwivedi, Y. K. (2018). Ranking online consumer reviews. *Electronic Commerce Research and Applications*, *29*(January), 78–89. https://doi.org/10.1016/j.elerap.2018.03.008

Schneider, M. J., & Gupta, S. (2016). Forecasting sales of new and existing products using consumer reviews: A random projections approach. *International Journal of Forecasting*, *32*(2), 243–256. https://doi.org/10.1016/j.ijforecast.2015.08.005

Schouten, K., & Frasincar, F. (2016). Survey on Aspect-Level Sentiment Analysis. *IEEE Transactions on Knowledge and Data Engineering*, *28*(3), 813–830. https://doi.org/10.1109/TKDE.2015.2485209

Scikit-Learn. (2018). Scikit-Learn User Guide, 2342.

Scott, J. A. (2015). *Getting Started with Apache Spark*. MapR Technologies, Inc.

Sebastiani, F. (2002). Machine Learning in Automated Text Categorization. *ACM Comput. Surv.*, *34*(1), 1–47. https://doi.org/10.1145/505282.505283

Segura-Bedmar, I., Colón-Ruíz, C., Tejedor-Alonso, M. Á., & Moro-Moro, M. (2018). Predicting of anaphylaxis in big data EMR by exploring machine learning approaches. *Journal of Biomedical Informatics*, *87*, 50–59. https://doi.org/https://doi.org/10.1016/j.jbi.2018.09.012

Serrano-Guerrero, J., Olivas, J. A., Romero, F. P., & Herrera-Viedma, E. (2015). Sentiment analysis : A review and comparative analysis of web services. *Information Sciences*, *311*, 18–38. https://doi.org/10.1016/j.ins.2015.03.040

Shadroo, S., & Rahmani, A. M. (2018). Systematic survey of big data and data mining in internet of things. *Computer Networks*, *139*, 19–47. https://doi.org/https://doi.org/10.1016/j.comnet.2018.04.001

Shafiabady, N., Lee, L. H., Rajkumar, R., Kallimani, V. P., Akram, N. A., & Isa, D. (2016). Using unsupervised clustering approach to train the Support Vector Machine for text classification. *Neurocomputing*, *211*, 4–10. https://doi.org/https://doi.org/10.1016/j.neucom.2015.10.137

Shah, N., Irani, Z., & Sharif, A. M. (2017). Big data in an HR context: Exploring

organizational change readiness, employee attitudes and behaviors. *Journal of Business Research*, *70*, 366–378. https://doi.org/https://doi.org/10.1016/j.jbusres.2016.08.010

Shen, X., & Choudhary, A. (2004). A high-performance distributed parallel file system for data-intensive computations. *Journal of Parallel and Distributed Computing*, *64*(10), 1157–1167. https://doi.org/10.1016/j.jpdc.2004.07.001

Shirdastian, H., Laroche, M., & Richard, M. O. (2017). Using big data analytics to study brand authenticity sentiments: The case of Starbucks on Twitter. *International Journal of Information Management*, (August), 0–1. https://doi.org/10.1016/j.ijinfomgt.2017.09.007

Shvachko, K., Kuang, H., Radia, S., & Chansler, R. (2010). The Hadoop Distributed File System. In *2010 IEEE 26th Symposium on Mass Storage Systems and Technologies (MSST)* (pp. 1–10). https://doi.org/10.1109/MSST.2010.5496972

Siegel, E. (2013). *Predictive Analytics: The Power to Predict Who Will Click, Buy, Lie, or Die*. Wiley.

Silberschatz, A., Korth, H., & Sudarshan, S. (2006). *Fundamentos de bases de datos*. McGraw-Hill.

Silva, R. M., Alberto, T. C., Almeida, T. A., & Yamakami, A. (2017). Towards filtering undesired short text messages using an online learning approach with semantic indexing. *Expert Systems with Applications*, *83*, 314–325. https://doi.org/https://doi.org/10.1016/j.eswa.2017.04.055

Silva, R. M., Almeida, T. A., & Yamakami, A. (2017). MDLText: An efficient and lightweight text classifier. *Knowledge-Based Systems*, *118*, 152–164. https://doi.org/https://doi.org/10.1016/j.knosys.2016.11.018

Singh, J. P., Irani, S., Rana, N. P., Dwivedi, Y. K., Saumya, S., & Roy, P. K. (2017). Predicting the "helpfulness" of online consumer reviews. *Journal of Business Research*, *70*, 346–355. https://doi.org/https://doi.org/10.1016/j.jbusres.2016.08.008

Smits, G. F., & Jordaan, E. M. (2002). Improved SVM Regression using Mixtures of Kernels. In *Proceedings of the 2002 International Joint Conference on Neural Networks* (pp. 2785–2790).

Sohangir, S., Wang, D., Pomeranets, A., & Khoshgoftaar, T. M. (2018). Big Data: Deep Learning for financial sentiment analysis. *Journal of Big Data*, *5*(1), 3. https://doi.org/10.1186/s40537-017-0111-6

Soleymani, M., Garcia, D., Jou, B., Schuller, B., Chang, S. F., & Pantic, M. (2017). A survey of multimodal sentiment analysis. *Image and Vision Computing*, *65*, 3–14. https://doi.org/10.1016/j.imavis.2017.08.003

Song, H., & Liu, H. (2017). Predicting Tourist Demand Using Big Data. In *Analytics in Smart Tourism Design* (pp. 13–30). https://doi.org/10.1007/978-3-319-44263-1

Spector, L. (2006). Evolution of artificial intelligence. *Artificial Intelligence*, *170*, 1251–1253. https://doi.org/10.1016/0020-0255(70)90034-4

Subramanian, J., & Simon, R. (2013). Overfitting in prediction models - Is it a problem only in high dimensions? *Contemporary Clinical Trials*, *36*(2), 636–641. https://doi.org/10.1016/j.cct.2013.06.011

Sun, S., Luo, C., & Chen, J. (2016). A review of natural language processing techniques for opinion mining systems. *Information Fusion*, *36*, 10–25. https://doi.org/10.1016/j.inffus.2016.10.004

Swarup, P. (2012). Artificial intelligence. *International Journal of Computing and Corporate Research*, *2*(4), 1–16.

Talha, A., & Kara, R. (2016). A performance evaluation of in-memory databases. *Journal of King Saud University - Computer and Information Sciences*, 6–11. https://doi.org/10.1016/j.jksuci.2016.06.007

Tan, W., Blake, M. B., Saleh, I., & Dustdar, S. (2013). Social-network-sourced big data analytics. *IEEE Internet Computing*, *17*(5), 62–69. https://doi.org/10.1109/MIC.2013.100

Tanana, M., Hallgren, K. A., Imel, Z. E., Atkins, D. C., & Srikumar, V. (2016). A Comparison of Natural Language Processing Methods for Automated Coding of Motivational Interviewing. *Journal of Substance Abuse Treatment*, *65*, 43–50. https://doi.org/10.1016/j.jsat.2016.01.006

Team, A. (2016). AzureML: Anatomy of a machine learning service. In L. Dorard, M. D. Reid, & F. J. Martin (Eds.), *Proceedings of The 2nd International Conference on Predictive APIs and Apps* (Vol. 50, pp. 1–13). Sydney, Australia: PMLR. Retrieved from http://proceedings.mlr.press/v50/azureml15.html

Tessari, M. (2018). *4 formas de solucionar los problemas más comunes a la hora de preparar datos*. Londres: Tableau Software.

Thorsby, J., Stowers, G. N. L., Wolslegel, K., & Tumbuan, E. (2017). Understanding the content and features of open data portals in American cities. *Government Information Quarterly*, *34*(1), 53–61. https://doi.org/10.1016/j.giq.2016.07.001

Thottuvaikkatumana, R. (2016). *Apache Spark 2 for Beginners*. Birmingham, UK: Packt Publishing Ltd.

Thusoo, A., Sen Sarma, J., Jain, N., Shao, Z., Chakka, P., Zhang, N., … Murthy, R. (2010). Hive - A Petabyte Scale Data Warehouse Using Hadoop. *Proceedings - International Conference on Data Engineering*.

Turner, V., Gantz, J., Reinsel, D., & Minton, S. (2014). *The Digital Universe of Opportunities: Rich Data and the Increasing Value of the Internet of Things*.

Tutkan, M., Can, M., & Akyokus, S. (2016). Helmholtz principle based supervised and unsupervised feature selection methods for text mining. *Information Processing and Management*, *52*, 885–910. https://doi.org/10.1016/j.ipm.2016.03.007

Vaddeman, B. (2016). *Beginning Apache Pig*. Apress.

Vajirakachorn, T., & Chongwatpol, J. (2017). Application of business intelligence in the tourism industry: A case study of a local food festival in Thailand. *Tourism Management Perspectives*, *23*, 75–86. https://doi.org/10.1016/j.tmp.2017.05.003

Valdivia, A., Luzón, M. V., & Herrera, F. (2017). Sentiment Analysis in TripAdvisor. *IEEE Intelligent Systems*, *32*(4), 72–77. https://doi.org/10.1109/MIS.2017.3121555

Vasconcelos, M., Almeida, J. M., & Gonçalves, M. A. (2015). Predicting the popularity of micro-reviews: A Foursquare case study. *Information Sciences*, *325*, 355–374. https://doi.org/https://doi.org/10.1016/j.ins.2015.07.001

Wamba, S. F., Akter, S., Edwards, A., Chopin, G., & Gnanzou, D. (2015). How 'big data' can make big impact: Findings from a systematic review and a longitudinal case study. *International Journal of Production Economics*, *165*, 234–246. https://doi.org/https://doi.org/10.1016/j.ijpe.2014.12.031

Wang, G., Gunasekaran, A., Ngai, E. W. T., & Papadopoulos, T. (2016). Big data analytics in logistics and supply chain management: Certain investigations for research and applications. *International Journal of Production Economics*, *176*, 98–110. https://doi.org/https://doi.org/10.1016/j.ijpe.2016.03.014

Watt, J., Borhani, R., & Katsaggelos, A. K. (2016). *Machine Learning Refined*. Cambridge University Press.

Willett, P. (2006). The Porter stemming algorithm: then and now. *Program*, *40*(3), 219–223. https://doi.org/10.1108/00330330610681295

Wong, D. F., Lu, Y., & Chao, L. S. (2016). Bilingual recursive neural network based data selection for statistical machine translation. *Knowledge-Based Systems*, *108*, 15–24. https://doi.org/https://doi.org/10.1016/j.knosys.2016.05.003

Xia, W., Zhu, W., Liao, B., Chen, M., Cai, L., & Huang, L. (2018). Novel architecture for long short-term memory used in question classification. *Neurocomputing*, *299*, 20–31. https://doi.org/https://doi.org/10.1016/j.neucom.2018.03.020

Xiang, Z., Du, Q., Ma, Y., & Fan, W. (2017). A comparative analysis of major online review platforms : Implications for social media analytics in hospitality and tourism. *Tourism Management*, *58*, 51–65. https://doi.org/10.1016/j.tourman.2016.10.001

Xiang, Z., Schwartz, Z., Gerdes, J. H., & Uysal, M. (2015). What can big data and text

analytics tell us about hotel guest experience and satisfaction? *International Journal of Hospitality Management*, *44*, 120–130. https://doi.org/https://doi.org/10.1016/j.ijhm.2014.10.013

Xie, K. L., Zhang, Z., & Zhang, Z. (2014). The business value of online consumer reviews and management response to hotel performance. *International Journal of Hospitality Management*, *43*(October 2014), 1–42. https://doi.org/10.1016/j.ijhm.2014.07.007

Xu, Z., Jiang, H., Kong, X., Kang, J., Wang, W., & Xia, F. (2016). Cross-Domain Item Recommendation Based on User Similarity. *Computer Science and Information Systems*, *13*(2), 359–373. https://doi.org/10.2298/CSIS150730007Z

Yadav, R. (2017). *Apache Spark 2.x Cookbook*. Packt Publishing.

Yadollahi, A., Shahraki, A. G., & Zaiane, O. R. (2017). Current State of Text Sentiment Analysis from Opinion to Emotion Mining. *ACM Comput. Surv.*, *50*(2), 25:1--25:33. https://doi.org/10.1145/3057270

Yala, A., Barzilay, R., Salama, L., Griffin, M., Sollender, G., Bardia, A., … Hughes, K. S. (2017). Using machine learning to parse breast pathology reports. *Breast Cancer Research and Treatment*, *161*(2), 203–211. https://doi.org/10.1007/s10549-016-4035-1

Yang, Y., & Loog, M. (2018). A variance maximization criterion for active learning. *Pattern Recognition*, *78*, 358–370. https://doi.org/https://doi.org/10.1016/j.patcog.2018.01.017

Yoo, S., Song, J., & Jeong, O. (2018). Social media contents based sentiment analysis and prediction system. *Expert Systems with Applications*, *105*, 102–111. https://doi.org/https://doi.org/10.1016/j.eswa.2018.03.055

Yuan, H., Xu, H., Qian, Y., & Li, Y. (2016). Make your travel smarter: Summarizing urban tourism information from massive blog data. *International Journal of Information Management*, *36*(6), 1306–1319. https://doi.org/10.1016/j.ijinfomgt.2016.02.009

Yue, X., Di, G., Yu, Y., Wang, W., & Shi, H. (2012). Analysis of the combination of natural language processing and search engine technology. *Procedia Engineering*, *29*, 1636–1639. https://doi.org/10.1016/j.proeng.2012.01.186

Yüksel, S. (2007). An integrated forecasting approach to hotel demand. *Mathematical and Computer Modelling*, *46*(7–8), 1063–1070. https://doi.org/10.1016/j.mcm.2007.03.008

Zareapoor, M., Shamsolmoali, P., Jain, D. K., Wang, H., & Yang, J. (2017). Kernelized support vector machine with deep learning: An efficient approach for extreme multiclass dataset. *Pattern Recognition Letters*, 1–10. https://doi.org/https://doi.org/10.1016/j.patrec.2017.09.018

Zhang, L., Stoffel, A., Behrisch, M., Mittelstadt, S., Schreck, T., Pompl, R., … Keim, D. (2012). Visual analytics for the big data era. A comparative review of state-of-the-art commercial systems. In *2012 IEEE Conference on Visual Analytics Science and Technology (VAST)* (pp. 173–182). https://doi.org/10.1109/VAST.2012.6400554

Zhang, L., Wang, S., & Liu, B. (2018). Deep Learning for Sentiment Analysis: A Survey. *CoRR*, *abs/1801.0*. Retrieved from http://arxiv.org/abs/1801.07883

Zhang, L., Wu, Z., Bu, Z., Jiang, Y., & Cao, J. (2018). A pattern-based topic detection and analysis system on Chinese tweets. *Journal of Computational Science*, *28*, 369–381. https://doi.org/https://doi.org/10.1016/j.jocs.2017.08.016

Zhang, W., Du, Y., Yoshida, T., & Wang, Q. (2018). DRI-RCNN: An approach to deceptive review identification using recurrent convolutional neural network. *Information Processing & Management*, *54*(4), 576–592. https://doi.org/https://doi.org/10.1016/j.ipm.2018.03.007

Zhang, W., Yoshida, T., & Tang, X. (2008). Text classification based on multi-word with support vector machine. *Knowledge-Based Systems*, *21*(8), 879–886. https://doi.org/https://doi.org/10.1016/j.knosys.2008.03.044

Zhang, W., Yoshida, T., & Tang, X. (2011). A comparative study of TF*IDF, LSI and multi-words for text classification. *Expert Systems with Applications*, *38*(3), 2758–2765. https://doi.org/https://doi.org/10.1016/j.eswa.2010.08.066

Zheng, W., Liu, Q., Zhang, M., Wan, K., Hu, F., & Yu, K. (2018). J-TEXT distributed data storage and management system. *Fusion Engineering and Design*, *129*, 207–213. https://doi.org/https://doi.org/10.1016/j.fusengdes.2018.02.058

Zhou, K., Zeng, J., Liu, Y., & Zou, F. (2018). Deep sentiment hashing for text retrieval in social CIoT. *Future Generation Computer Systems*, *86*, 362–371. https://doi.org/https://doi.org/10.1016/j.future.2018.03.047

Zhou, L., Pan, S., Wang, J., & Vasilakos, A. V. (2017). Machine learning on big data: Opportunities and challenges. *Neurocomputing*, *237*, 350–361. https://doi.org/https://doi.org/10.1016/j.neucom.2017.01.026

Zhu, X., & Goldberg, A. B. (2009). *Introduction to Semi-Supervised Learning*. Morgan & Claypool. https://doi.org/https://doi.org/10.2200/S00196ED1V01Y200906AIM006