Universidad de Valladolid

ESCUELA DE INGENIERÍA INFORMÁTICA (SG)

Grado en Ingeniería Informática de Servicios y Aplicaciones

Aplicabilidad del Procesamiento de Lenguaje Natural al análisis de errores en logs de aplicativos utilizados en el área de BSS (Business Support System)

Alumna: Noemi Arada Pérez

Tutores: José Ignacio Farrán Martín

Rubén Martínez Alonso

A mis cachorras,
por todos los ratos que este
documento os ha arrebatado.

Abstract

Versión en castellano:

Este estudio se ha realizado en colaboración con la empresa HPE CDS; en él se resume una investigación sobre la aplicabilidad de la tecnología de Machine Learning en la lectura de logs, y más concretamente la lectura de logs generados por las máquinas de Mediación de una empresa de telecomunicaciones. La finalidad de este trabajo es determinar la viabilidad de la implantación de la tecnología mencionada a fin de prevenir posibles abortos de las máquinas en esta área de la telefonía.

Se analiza por un lado el funcionamiento de la tecnología Machine Learning y, por otro, cómo funciona el Business Support System dentro del área de Mediación de Telefonía. Gracias a este análisis, se sugiere la fórmula para aplicar dicha tecnología a la mencionada lectura de logs, así como su viabilidad en un entorno real.

La metodología para la realización de las pruebas ha sido el análisis de los datos desde distintas perspectivas de tratamiento de los mismos, mediante el lenguaje Python y su librería scikit-learn.

Al final del estudio se exponen una serie de conclusiones derivadas de las distintas pruebas realizadas, y se abren diferentes vías de análisis, de cara a la realización de otros estudios prácticos.

English Version

This document has been created in collaboration with the company HPE CDS; it summarizes a research on the applicability of the Machine Learning technology in the reading of logs, and more precisely the reading of logs generated by the Mediation machines of a telecommunication company. The goal of this work is to determine the feasibility of using the technology mentioned above in order to prevent possible abortions of the Mediation machines.

On one hand, the way of working of the Machine Learning technology is analyzed and, on the other hand, we study how the Business Support System works within the area of Telephony Mediation. Thanks to this analysis, the strategy to apply the Machine Learning technology is studied, as well as its viability in a real environment.

The methodology to make the tests has been the analysis of the data from different perspectives, with different treatments, with the aid of the Python language and its library scikit-learn. At the end of this study, a series of conclusions are shown, with different tests and different ways of analysis carried out, in order to succeed in other practical studies.

Índice general:

Listado de Imágenes

Listado de Tablas

1. Introducción

La necesidad por parte de muchas empresas de optimizar la información que poseen, especialmente cuando se trata de grandes volúmenes de la misma, es una realidad creciente. Para el ojo humano resulta imposible analizar esta información de forma manual, se requeriría una cantidad de personal inimaginable con unos conocimientos muy específicos en cuanto a matemáticas, estadística, programación… además de que el análisis sería un trabajo tedioso y repetitivo.

Con el objetivo de analizar esta información voluminosa, en lo que a las empresas de telecomunicaciones compete, nace este trabajo. A través de la tecnología de Machine Learning que, si bien no es novedosa, sí lo es el descubrimiento de las múltiples capacidades y posibilidades de la misma; se explotarán posibilidades de análisis dentro del área de Business Support System. Concretamente se centrará en la operadora de telefonía de Vodafone, a quién desde hace años provee servicios la empresa HPE CDS, y más específicamente en el área de Mediación de Telefonía.

En la sociedad actual, prácticamente a cada paso y en cada minuto, una persona genera datos. Datos derivados de consumos móviles, gps, domótica, aplicaciones tan sencillas como aquellas que cuentan pasos, redes sociales, etc. Estos datos al final del día de muchas personas, tienen un volumen considerable, que a fin de obtener información como por ejemplo referida a hábitos de consumo, tiene un importante valor para las empresas. La tecnología de Machine Learning, proveniente de la rama de la ciencia de la Inteligencia Artificial, es utilizada para lo que se denomina ‘Data Science’, o la ciencia de los datos. Esta ciencia, está centrada en grandes volúmenes de datos. Últimamente se ha oído mucho hablar sobre ella en diferentes medios, puesto que sus posibilidades son casi infinitas: desde análisis de ecografías y otras pruebas médicas para detección de algunos cánceres, pasando por análisis de imágenes de cámaras de seguridad o traducción instantánea del habla a distintos idiomas.

Aplicando la tecnología de Machine Learning a los logs de Mediación de Telefonía, a través de un exhaustivo estudio en las trazas de los ficheros de logs de las máquinas de Mediación de Telefonía, se podrán comprobar las posibilidades de que éstas generen un error fatal que provoque la caída del sistema. Estas caídas, aunque son poco frecuentes, dada la estabilidad del sistema, se producen. La caída del sistema supone retrasos en la colección de los datos, que en ocasiones se traduce en una pérdida económica más o menos significativa y, que en el 100% de los casos, genera situaciones de estrés y nerviosismo a los trabajadores, además de horas de trabajo fuera del horario laboral, que deberán ser remuneradas. Por todas estas razones, además de una clara, que es la optimización de los recursos, nace la idea de este estudio, puesto que si se pueden prevenir esta clase de errores que generan el colapso del sistema, se tendrían no sólo trabajadores más felices, sino que además el balance de cuentas sería más beneficioso.

2. Objetivos

El título de este estudio es "Aplicabilidad del Procesamiento de Lenguaje Natural al análisis de errores en logs de aplicativos utilizados en el área de BSS (Business Support System)".

El objetivo general de este trabajo es comprender el reto que supone la comprensión del lenguaje natural para las computadoras, así como la interpretación del lenguaje máquina. Eso supone, además, explicar en qué consiste la tecnología de Machine Learning, su funcionamiento, problemática y aplicabilidad y, finalmente, estudiar la posibilidad de que esta tecnología sea aplicable a un campo como la lectura de logs generados en los sistemas de telecomunicaciones, tales como las operadoras de telefonía. El estudio culminará con una pequeña ejemplificación sobre lo que el Machine Learning puede aportar a través de la lectura e interpretación de los logs, dentro del área de Business Support System.

Para alcanzar este objetivo general, se establecen 4 objetivos específicos y concretos, y que a su vez se corresponderían con los 4 apartados del trabajo, a saber:

1) Definir, aplicar y desarrollar el funcionamiento básico de Machine Learning

En este apartado se proporcionará una idea general de lo que es: la lingüística y el lenguaje, el lenguaje natural y el lenguaje formal, el Procesamiento del Lenguaje Normal y finalmente la tecnología de Machine Learning. Este apartado es importante de cara a obtener los conocimientos necesarios para tener una idea general de lo que engloba el Machine Learning, así como la aclaración de muchos conceptos que son importantes a la hora de comprender el grueso de este documento.

2) Saber interpretar y prevenir los errores en la lectura de logs a través de Machine Learning

En el segundo de los apartados de este estudio se explicará qué son y para qué se utilizan los logs, así como el funcionamiento de Machine Learning aplicado a la lectura de logs. También se abordará el funcionamiento y peculiaridades de la herramienta eIUM, ampliamente utilizada en la mediación de telefonía y finalmente, un análisis de tecnología Machine Learning aplicado a la lectura de logs.

3) Valorar la implantación de la tecnología Machine Learning en Mediación de Telefonía

En el tercer apartado, se valorará la posibilidad de instaurar Machine Learning para la lectura de logs en el área del Business Support System, haciendo una breve introducción al mismo, el impacto que podría tener, los pros y contras que podría suponer, etc.

4) Realizar un resteo ejemplificado de Machine Learning en Mediación de Telefonía

Este último apartado tratará de ejemplificar una pequeña simulación de la utilización de la tecnología de Machine Learning en el área de Business Support System.

Por todo ello, podemos decir que el objeto de estudio de este trabajo no se corresponde con una aplicación, una web o una extensión. A lo largo de los apartados que se han detallado, se podrá tener una idea bastante aproximada de cómo de útil y eficiente, podría resultar una tecnología tan puntera como Machine Learning, al aplicarse a un ámbito hasta ahora desconocido, como pueda ser el área de Business Support System, que se encarga de la mayoría de operaciones que se realizan dentro de una operadora de telefonía.

Para finalizar, es importante tener en mente que, en el mundo de la informática por encima de cualquier otro campo de la ciencia, se debe aplicar una máxima que es: ‘renovarse o morir’.

3. Glosario de términos

3.1 Siglas:

ALA: Autómata Linealmente Acotado

ASCII: American Standard Code for Information Interchange

BD: Big Data

BBDD: Base de Datos

BSS: Business Support System

CDR: Call Data Recorder.

CEO: Chief Executive Officer, se corresponde con el máximo responsable de una empresa

COMPAS: Correctional Offender Management Profiling for Alternative Santions

DM: Data Mining, Minería de Datos

E2E: End to End

FTP: File Transfer Protocol. Es un protocolo de transferencia de ficheros poco seguro

eIUM: enhanced Interactive Unified Mediation

GSM: Global System for Mobile

GT: Global Title. Dirección única referida al destino, utilizada por el protocolo SCCP para enrutamiento de mensajes en redes.

HP: Hewlett Packard

HPE: Hewlett Packard Enterprise

IA: Inteligencia Artificial

IMSI: International Mobile Subscriber Identity. Código único integrado en la SIM de cada móvil que utilizan las operadoras de telefonía.

IMT: International Mobile Telecommunication

KDD: Knowledge Discovery in Databases

LN: Lenguaje Natural

LTE: Long Term Evolution

ML: Machine Learning

MO: Mobile Originated

MSC: Mobile Switching Central. Comienza, termina y canaliza las llamadas, como si fuera una central telefónica de red fija, aunque se funciona tanto en fijos como en móviles.

MSISDN: Mobile Station Integrated Services Digital Network. Se corresponde con un número de teléfono móvil.

MT: Mobile Terminated

NIST: National Institute of Standards and Technology

NLTK: Natural Language Tool Kit

NME: Normalized Metered Events o Eventos Medios Normalizados

NRN: Network Routing Number. Es un prefijo que permite identificar las llamadas a números portados, posee 6 dígitos, entre ellos el código de operador (Comisión Nacional de los Mercados y la Competencia, 2017).

OSS: Operational Support System

Sw: Software

PLN: Procesamiento del Lenguaje Natural

PYME(S): Pequeña(s) y mediana(s) empresa(s)

RNAs: Redes Neuronales Artificiales

SIM: Subscriber Identity Module. Tarjeta inteligente que se utiliza en teléfonos móviles, almacenan la clave del servicio del suscriptor que sirve para identificarse en la red. Esta tarjeta permite que el cliente se identifique como tal a través de diferentes dispositivos.

SSCP: System Security Certifier Partitioner.

SVM: Support Vector Machine (Máquinas de Vectores de Soporte)

UMTS: Universal Mobile Telecommunications System

VLR: Visitor Location Register

VPC: Virtual Private Cloud o Red Privada en la Nube.

3.2 Terminología:

Aborto: Se refiere a un error grave en la ejecución de uno o varios procesos, éstos generan un error en cadena que finaliza con la caída generalizada del sistema. Es un error grave o muy grave que requeriría el reinicio del sistema.
Árbol de decisión: Mapa de resultados posibles de una serie de decisiones relacionadas entre sí.
Chi-cuadrado: O chi-cuadrado de Pearson, fue desarrollada por este matemático y estadístico alrededor del año 1900. Esta prueba evalúa los datos de la distribución observada con la distribución esperada (Pértega Díaz S., Pita Fernández S.,2004).

Complementos: Indican el lugar, tiempo o circunstancia en la que sucede la acción de la oración
Corpus: Conjunto cerrado de textos o de datos destinado a la investigación científica
K-Nearest Neighbours: algoritmo que clasifica datos por similitud. Se trata de un algoritmo de aprendizaje supervisado.
Mantenimiento predictivo: El mantenimiento Predictivo son las acciones que se toman para la detección de fallos o defectos de maquinaria en los componentes de un sistema, a fin de evitar que éstos se hagan difíciles de manejar o de mayor magnitud, el objetivo del Mantenimiento Predictivo es que el nivel de servicio de los equipos sea el adecuado (Olarte W., Botero M., Cañon B., 2010).
Minería de datos: Campo que aplica la estadística y las ciencias de computación en grandes volúmenes de datos, cuyo fin es encontrar patrones dentro de esos datos
Modularidad: Capacidad de un sistema para trabajar dividido en varias partes para alcanzar un objetivo común
Objeto directo: Persona, animal o cosa que recibe la acción realizada por el sujeto
Objeto indirecto: Persona, animal o cosa que recibe indirectamente la acción realizada por el sujeto
RMON: Sus siglas provienen de Monitoreo Remoto o Remote Monitoring, se trata de una extensión hacia un espacio de direcciones tipo MIB que se creó para monitorear el mantenimiento de las redes de área local remotas, "RMON captura datagramas directamente del medio y desde allí puede analizar el datagrama completo y proporcionar un análisis detallado de la red LAN como un todo" (García A., 2000, pág. 348)
Servidor proxy: Es una "interfaz de comunicación que actúa como intermediaria entre dos sistemas informáticos, como el navegador de tu ordenador o móvil y la propia red de Internet" (Nodus Trends, 2017, p. 2).
SNMP: *Simple Network Management Protocol*, o protocolo simple de administración de red. Se utiliza para administrar los dispositivos que están conectados en una misma red (García A., 2000, pág. 343)
TimeStamp: Timestamp es un número que se refiere a la cantidad de segundos transcurridos desde las 00:00:00 UTC del 1 de enero de 1970. El estándar ISO 8601 se encarga de los formatos de las fechas, muchas veces los timestamps siguen dichos estándares.
VoIP: Voice Over Internet Protocol, o voz sobre protocolo de internet. Permite comunicarse a través de la voz utilizando internet, una de sus mayores ventajas es su precio económico.

4. Definición, aplicación y funcionamiento básico de Machine Learning

En este primer apartado, se tratará de explicar la tecnología de Machine Learning dentro del campo que la engloba, que es la Inteligencia Artificial, dando las nociones básicas para la comprensión de esta tecnología. De igual importancia es la contextualización de la empresa CDS, en torno a la cual, se ha fraguado este documento.

4.1 Contextualización: la empresa CDS y su interés en el ML en Mediación de telefonía

La necesidad histórica de comunicación por parte del ser humano lo ha empujado a la creación de sistemas y tecnologías cada vez más complejas. La velocidad a la que estas tecnologías evolucionan las hace inabarcables, incluso para aquellos se dedican profesionalmente al sector. Gracias a esta evolución se han forjado especializaciones de lo más dispares en torno a la informática.

Machine Learning es una de esas tecnologías, que, aunque diste de ser novedosa, su evolución en los últimos años ha sido notable. A lo largo de este documento, se pretenderán adquirir los conocimientos básicos necesarios para la comprensión no sólo del funcionamiento, sino también de la cantidad de campos que abarca hoy en día.

Si hace años empresas como Siemens o Motorola llevaban la voz cantante en telecomunicaciones y tecnologías punteras, hoy en día la lista de las empresas más o menos grandes que se dedican a este sector es casi infinita. Una de esas empresas es CDS. CDS es una empresa perteneciente a la compañía *Hewlett Packard Enterprise* (HPE), antes conocida como *Hewlett Packard*, (HP) fue fundada en el año 1939 por David Packard y William R. Hewlett. Aunque inicialmente la compañía lanzó un oscilador de baja frecuencia, a día de hoy se trata de una multinacional que se dedica a impresoras de pequeño y gran formato, tintas, portátiles y servicios tecnológicos de todo tipo.

ILUSTRACIÓN 2 LOGOTIPO HPE (HPE OFFICIAL WEBSITE)

ILUSTRACIÓN 3 LOGOTIPO DE HP (HP OFFICIAL WEBSITE)

ILUSTRACIÓN 1 LOGOTIPO DE CDS (CDS OFFICIAL WEBSITE)

CDS es una empresa subsidiaria [1] de HPE; que se dedica a "servidores, almacenamiento, equipamiento de red, soluciones Cloud y servicios de Tecnologías de la Información" (CDS, 2018).

[1] Una empresa es subsidiaria de otra, cuando una de ellas controla a la otra; la que es controlada se considera subsidiaria mientras que la otra, es llamada matriz.

CDS se encarga de las operaciones EMEA (Europe, the Middle East and Africa), y aunque es subsidiaria de HPE, se trata de una entidad legal independiente que proporciona capacidad de servicio de múltiples proveedores para los clientes de HPE.

La compañía CDS se formó tras la adquisición de Synstar plc por parte de HPE. La empresa Synztar pl., tenía una amplia experiencia en la prestación de servicios de múltiples proveedores en Europa. Partiendo de esa experiencia, se ha creado una empresa de soluciones de servicio tan flexibles como personalizadas; con los valores propios de la empresa HPE.

El objeto de estudio de este documento trata sobre la Mediación de Telefonía que se engloba dentro del área de Business Support System de la empresa CDS. La Mediación de Telefonía por su parte, engloba servicios externalizados por varias multinacionales dedicadas a las telecomunicaciones, como por ejemplo Orange, Vodafone o Telefónica.

Aunque esto se explicará en profundidad en apartados posteriores (*Valoración de la implantación de la tecnología Machine Learning en Mediación de Telefonía), e*n pocas palabras, la Mediación, se encarga de que toda la información que generan las centrales de telefonía, se procese, almacene y tarifique de forma correcta. Esto significa que cada vez que un usuario realiza algún tipo de consumo de fijo o móvil, éste se registra dentro de una central de telefonía. Las operadoras deben registrar este consumo, procesarlo, almacenarlo y tarificarlo. Para ello trabajan en conjunto diversos equipos: red, incidencias (referido a incidencias técnicas, sin contacto directo con el usuario final), fraude, etc. Todos los equipos de Mediación cooperan entre sí, a través de un engranaje perfecto a fin de que no se pierda información y la que proceda, sea tarificada de acuerdo con el contrato con el usuario final. De los cobros se encargan otros equipos.

Las operadoras telefónicas, por las características de sus procesos, trabajan con grandes volúmenes de datos, volúmenes de datos proporcionales al tráfico de datos que se produce entre los clientes de éstas. Este tráfico de datos comprende desde llamadas de teléfono ordinarias entre usuarios, pasando por servicios de buzón, desvíos, llamadas a números especiales: tipo 900 o cortos (como el de emergencias 112 o el teléfono de atención a la mujer maltratada 116), tráfico de datos, sms, videollamadas, etc. Todo este tráfico genera grandes cantidades de datos que las operadoras deben gestionar. De esta gestión de datos, se generan una serie de ficheros a los que se les denomina *logs*. Los logs no son otra cosa, que ficheros que poseen información histórica sobre diferentes aplicaciones, sistemas o dispositivos; por ejemplo, sobre los ficheros que se procesan dentro de una máquina, las entradas (log-in) de los usuarios a un sistema o los accesos al correo electrónico por parte de los usuarios; para simplificar la comprensión del concepto, etc. Un ejemplo de log '*a nivel de usuario*' sería el acceso al histórico de las pestañas abiertas en un navegador; se podría considerarse como un fichero de log, puesto que posee datos sobre las páginas web consultadas con las horas y el día en el que se consultaron.

Existen múltiples ficheros de logs, puesto cada mínima aplicación que está en funcionamiento, genera este tipo de ficheros. Este estudio se centrará en los logs que se generan al procesar los datos provenientes de las centrales telefónicas. Las centrales telefónicas son las encargadas de que se produzca tanto el tráfico de datos móviles como

de comunicar los dispositivos a través de llamadas de voz, sms y mms. Concretamente, serán los ficheros de log generados para una de las compañías con las que trabaja CDS: la teleco multinacional Vodafone.

Está demostrado que la aplicación de ML a grandes volúmenes de datos, supone multitud de ventajas dentro de ámbitos dispares. Puesto que Vodafone posee grandes volúmenes de datos que en determinadas circunstancias deben ser analizados; la aplicación de ML, podría agilizar esta analítica e incluso prevenir de posibles problemas y por lo tanto generar una ventaja económica para CDS, que es la empresa encargada de la Mediación de Vodafone.

El objetivo principal de este estudio es probar si es posible o no la implantación de la tecnología ML para poder prevenir posibles caídas en el sistema de colección de datos; y de implantarse qué tipo de ventajas e inconvenientes podría presentar. De probarse que ML funciona y realmente previene con un tiempo adecuado la caída del sistema o de algunos de sus procesos, se podrían realizar labores de prevención para que esto no ocurriera o minimizar el impacto de la misma.

Las ventajas de una tecnología capaz de proporcionar información con un margen de tiempo amplio sobre posibles errores que provoquen la caída total, parcial o de determinados procesos de sistema, revertiría en una serie de ventajas dentro del área de Mediación:

1. Se podrían desarrollar instrumentos o procedimientos para prevención de las caídas o minimización de su impacto
2. La caída del sistema implica que expertos en el sistema abandonen sus tareas para dedicarse a tiempo completo a la solución de la problemática, lo cual repercute económicamente en las cuentas de la compañía; si se consigue que el sistema tenga menos errores, la repercusión económica será menor a la actual.
3. En general, un sistema que funciona con unos errores mínimos, se traduce en menos atención sobre el mismo, y menos impacto económico para su mantenimiento.
4. Con la implantación de ML en la lectura de logs, tal vez se descubran otra serie de errores o mejoras, que actualmente resultan invisibles al ojo humano.
5. Este trabajo además es una buena oportunidad para observar otros campos en los que se podría aplicar la tecnología ML en un futuro dentro de la compañía.

Si bien es cierto que, para la implantación de ML dentro de un área tan extensa y crítica como la Mediación de telefonía, se requeriría la opinión de un experto, un equipo de expertos o una empresa que se dedique exclusivamente a ello; esta persona o grupo de personas deberían tener un control total o altísimo sobre los entresijos del sistema. La persona que realiza este documento, no poseía grandes conocimientos previos sobre lo que supone la implantación de ML o sobre la tecnología en cuestión; sin embargo, tras nueve años trabajando en el área de Mediación de Telefonía, realizando guardias,

proyectos de mejora y prevención con la herramienta *eIUM*[2] (enhanced Interactive Unified Mediation), que es la principal plataforma de trabajo en el área de Mediación; se podría decir que tiene los conocimientos necesarios como para saber qué puntos del sistema o de la plataforma de trabajo, son susceptibles de mejoras.

Con lo dicho hasta el momento, se pretende que el lector comprenda que, aunque la valoración del estudio sobre la implantación no arroje resultados concretos, el objetivo es estudiar cómo de compleja es la tecnología de ML, cómo de complejo puede resultar el procedimiento de implantación desde la base y si es posible realizarlo con los conocimientos que se tienen sobre el sistema, o se precisan estudios de otro tipo.

Para finalizar, sólo comentar que la implantación de ML no siempre resulta todo lo bonita que la pintan, también se comenten errores, como derivados de la implantación de la tecnología, como los que se describe en el artículo de Dave Gershgorn (Gershgorn D., 2018), en la que el *Bank of America*, denegó sistemáticamente préstamos a una serie de clientes, debido a que el algoritmo de ML no estaba correctamente implantado.

Con todo lo dicho, se espera dar una visión general sobre ML, ventajas, desventajas, funcionalidades, funcionamiento y aplicabilidad dentro de la lectura de logs de la Mediación de Telefonía de la compañía CDS, a fin de mejorar las labores de prevención de errores dentro su sistema.

4.2 Introducción

La Inteligencia Artificial (IA) se podría definir como la ciencia que estudia la simulación de la inteligencia humana, desarrollada por software (sw) informático, "adquiriendo las capacidades de razonar, aprender y autocorregirse*"* (Euroforum, 2018, p.1). En resumidas cuentas, de lo que tratará la IA, será de imitar, definir y anticiparse al lenguaje natural humano. Es de esta anticipación de lo que la tecnología de Machine Learning (ML a partir de ahora) se hace gala, puesto que pretende la predicción de datos de diferente tipología.

La capacidad de razonamiento del ser humano, se ha producido a través de un largo proceso evolutivo. Paralelamente a este proceso, se ha producido la evolución del lenguaje natural. El uso del lenguaje natural es fundamental; puesto que requiere de un complejo proceso cerebral, conexiones neuronales, además de entonación o lenguaje gestual.

Antes de explicar cualquier concepto o conocimiento sobre la tecnología Machine Learning, resulta conveniente la mención de una serie de conceptos propios de la IA, a fin de tener una base sostenible y la capacidad de comprender el funcionamiento básico de la tecnología ML. Sin más dilación, se procederá a una breve descripción de los conceptos más importantes que se manejarán a la hora de describir el funcionamiento de la tecnología objeto de este estudio, de cara a su aplicabilidad dentro del análisis de logs en los aplicativos de Business Support System (BSS).

[2] Se dedicará un apartado completo a la explicación de esta plataforma, es la herramienta principal con la que se trabaja en Mediación

4.3 La lingüística y el lenguaje

La lingüística es la ciencia que estudia el lenguaje humano y las lenguas (Diccionario de la Lengua Española, ss. Ff.). Dentro de la lingüística se puede distinguir:

- Lingüística descriptiva o sincrónica: En la que se estudian los componentes de la lengua, sus relaciones y las posibles estructuras que conforman.
- Lingüística histórica, diacrónica o evolutiva: en la que se estudian las relaciones que unen términos y la propia evolución de la lengua, estudiando la transformación de la lengua a través del tiempo.
- Lingüística aplicada: en la que se analizan las aplicaciones prácticas que se hacen de la lingüística teórica: traducción, enseñanza, etc. (Knepp D., 2010, pág. 1, p. 3-5)

De acuerdo al objeto de estudio de este documento, conviene ampliar la información referida a dos de los tipos de lingüística que se han mencionado, puesto que tienen que ver con el análisis que la IA realiza a la hora de trabajar con el lenguaje natural, que se describirá más adelante.

Por un lado, está la lingüística descriptiva, que "analiza los elementos del lenguaje verbal y describe las clases de palabras, la sintaxis, las reglas gramaticales y los fonemas" (Knepp D., 2010, pág. 1, p. 7) y al mismo tiempo es la que proporciona la gramática de la lengua. Resultaría interesante, por tanto, poder definir e identificar alguna de la terminología que se ha utilizado, a fin de aclarar el significado de la lingüística descriptiva. La gramática se podría definir como la parte de la lingüística que "estudia el conjunto de normas y principios que rige una lengua" (Diccionario de Términos, ss. Ff.).

Por otro lado, está la lingüística aplicada, que tiene un carácter multidisciplinario, estudiando desde la enseñanza-aprendizaje de lenguas, patologías en el lenguaje, traducción, la aplicación de las nuevas tecnologías o la planificación lingüística. Cabe destacar en este punto, la lingüística matemática, que estudia las propiedades matemáticas de la lengua, y que a su vez comprende dos ramas: lingüística estadística y computacional. Para la lingüística estadística, se mide la gramática formal y la computacional se centra en los fenómenos lingüísticos: la relación entre la informática y la lingüística. Uno de los objetivos que tiene la ciencia de la computación a día de hoy, es que los ordenadores puedan recrear la capacidad del lenguaje.

Ahora que ya se tiene una idea más clara de lo que la lingüística en general puede llegar a abarcar, se pasará a lo concreto: el lenguaje. Existen una gran variedad de definiciones de la palabra *lenguaje*. De acuerdo con el Diccionario de la Lengua Española, el lenguaje es la "facultad del ser humano de expresarse y comunicarse con los demás a través del sonido articulado o de otros sistemas de signos ". (Diccionario de la Lengua Española, ss. Ff.). Se debe tener en consideración, sin embargo, que a día de hoy se utiliza la palabra *lenguaje* para describir la forma en la que se comunican animales como los delfines o los primates. A modo de aproximación al objeto de este estudio, se podría definir de una forma más técnica, como la que expone en la Universidad Politécnica de Madrid, como "un sistema más o menos complejo, que

asocia contenidos de pensamiento y significación a manifestaciones simbólicas tanto orales como escritas" (Montero J.M., ss. Ff., pág. 11).

Para que el lenguaje exista como tal, será necesario un conjunto de símbolos que permitan la construcción de mensajes. Será el intercambio de estos mensajes, su envío y recepción lo que permita que se produzca la comunicación (Carlos Torres L., ss. Ff., pág. 5)

Existen multitud de tipos de lenguaje que se dividen a partir de la forma o finalidad de los mismos, como su nivel de naturalidad, de acuerdo a su elemento comunicativo empleado, etc. Por las características propias de este estudio, se focalizará entre dos tipos de lenguaje de acuerdo con su nivel de naturalidad: natural en el que se engloban los idiomas como inglés, francés, alemán, etc., y artificial, teniendo en cuenta que este último a su vez engloba otros como el literario, científico y técnico, y formal (Corbin, 2018, p. 4).

4.3.1 El lenguaje natural y el lenguaje formal.

4.3.1.1 Lenguaje natural: definición

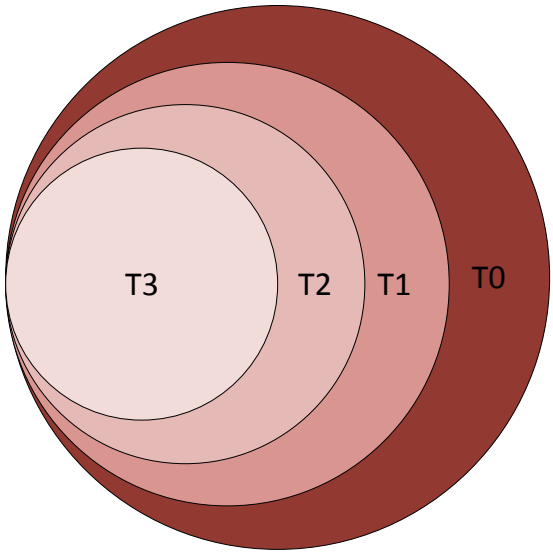
El lenguaje natural (LN) es la forma de comunicación que se utiliza entre humanos. El LN está en constante evolución, no se trata de un simple medio de comunicación; sirve para expresar emociones y situaciones complejas: como el amor, las leyes, los sentimientos, etc. Además de ser una potente herramienta para la comprensión del razonamiento humano (Cortés Vázquez, Vega Huerta, Pariona Quispe, 2009, pág. 46, p.6-7).

Dentro del lenguaje natural se pueden destacar las siguientes características:

- Se ha ido desarrollando y organizando a través de un proceso evolutivo
- La poli semántica o la capacidad de una palabra para adquirir varios significados
- Dificultad o imposibilidad de una formalización completa, entendiendo como formalización el proceso en el que un conjunto de palabras adquiere un sentido concreto (López Takeyas B., 2005, pág. 1, p. 8)

4.3.1.2 El lenguaje formal

El término formal hace referencia al hecho de poder expresar a través de símbolos matemáticos, ecuaciones u otra simbología matemática las relaciones, los componentes o los comportamientos que se detectan en el LN (Lahoz-Beltrá R., 2004, pág. 17). Se podría definir el lenguaje formal como "establecimiento de una serie de propiedades o fórmulas, que definan unívocamente las oraciones correctas que componen un lenguaje natural*"* (Montero J. M., ss. Ff., pág. 15, p.3). Los lenguajes formales son creados por el hombre y tienen fines científicos tales como la formalización o la representación de una información de forma simbólica, de forma que sea comprensible para un ordenador. No se trata de lenguajes que surgen de un proceso evolutivo natural como en el caso del LN.



T0: Gramáticas formales
T1: Gramáticas sensibles al contexto
T2: Gramáticas libres de contexto
T3: Gramáticas regulares

ILUSTRACIÓN 4 JERARQUÍA NOAM CHOMSKY (ELABORACIÓN PROPIA, JERARQUÍA NOAM CHOMSKY)

A partir de 1950, los lenguajes formales fueron tomados en consideración por la comunidad científica y la lingüística teórica. Uno de los principales precursores de los lenguajes formales fue Noam Chomsky, que a través de lo que él denominó como *Jerarquías* (Ilustración 4), consiguió demostrar que se pueden construir modelos matemáticos que reflejen el grado de complejidad estructural de los lenguajes naturales (Balari S., 2014, pág. 7-15). Cada una de las gramáticas descritas en la imagen, define el tipo de lenguaje que engloba, yendo de mayor (T0) a menor (T3) complejidad computacional:

- Gramáticas formales = lenguajes formales. Lo que se traduce en que genera lenguajes formales, estudia el lenguaje a través de modelos formales y matemáticos. Una gramática formal, representa como una cuaterna G = (V, Σ, Q0, P), donde:

"V es un conjunto finito llamado alfabeto de símbolos […].

Σ es otro conjunto finito, que verifica $V \cap \Sigma = \emptyset$ y se suele denominar alfabeto de símbolos terminales.

$Q0 \in V$ es una "variable" distinguida que se denomina símbolo inicial.

$P \subseteq (V \cup \Sigma)* \times (V \cup \Sigma)*$ es un conjunto finito llamado conjunto de producciones […]". (Pardo Vasallo L. M. y Gómez Pérez D.,2010, Pág. 8)

- Gramáticas sensibles al contexto o dependientes del contexto = lenguajes sensibles al contexto. Lo que se traduce en que genera lenguajes sensibles al contexto. Este lenguaje podría ser interpretado por un Autómata Linealmente Acotado (ALA), que es similar a la máquina de Turing, aunque más limitado.
- Gramáticas libres de contexto = lenguajes libres de contexto. Lo que se traduce en que generan lenguajes libres de contexto. Es un tipo de lenguaje formal, que tiene aplicación por ejemplo en la creación de compiladores de lenguajes de programación.
- Gramáticas regulares = lenguajes regulares. Lo que se traduce en que generan lenguajes regulares. En este caso, las cadenas se forman por la concatenación de símbolos y no hay relación entre las distintas partes de la cadena. El autómata capaz de interpretar estos lenguajes se denomina finito.

Dentro de la IA, se utiliza la Teoría de Lenguajes Formales; que estudia los lenguajes centrándose en sus propiedades estructurales, definiendo clases de complejidad estructural y estableciendo las relaciones existentes entre ellas (Balari S., 2014, pág. 7-15).

Dentro de los lenguajes formales, se encuentran los lenguajes de programación.

4.3.2 PLN: Procesamiento del Lenguaje Natural

La comunicación es la base y el corazón del ser humano. El lenguaje es la vía que abre la comunicación entre humanos, que ayuda a comprender conceptos complejos como comunidad, empatía, amor, justicia o pertenencia. Uno de los grandes hitos de la IA es llegar a comprender el lenguaje humano en toda su extensión.

Cuando se habla de Procesamiento de lenguaje natural (PLN), se refiere a la forma que la IA tiene para comprender e imitar el lenguaje natural que utilizan los seres humanos (Euroforum, 2018, p. 5), para ello será necesaria la utilización de las ciencias de la computación y la lingüística.

Hasta hace unos años, tener el poder de comunicarse hablando directamente con un ordenador o un teléfono móvil era cosa de la ficción, pero gracias a los avances en el Procesamiento del Lenguaje Natural, hoy día es una realidad. Ahora mismo es posible avisar al coche a través de la voz de que se ha sufrido un accidente y que el ordenador de abordo avise a los servicios pertinentes para poder ser atendidos, se puede pedir al móvil que envíe un mensaje de texto a una persona concreta, dictándole qué es lo que se quiere enviar o algo tan sencillo como apretar el botón de 'traducir' de Facebook para comprender lo que otros han escrito; los avances en este campo son innumerables.

De forma general, los usos más comunes del PLN, descritos por Scott Sims, *CEO* de la empresa *Buzzlogix,* son los siguientes:

- **Análisis:** consiste en la extracción de la información útil de documentos o conjuntos de documentos de extensiones variables y provenientes de múltiples fuentes. Puede tratarse de búsquedas muy complejas a través de datos estructurados o semi-estructurados. Se utiliza en sectores tan variados como la medicina, la ciencia en general, farmacia, biología, datos financieros, estadísticas, minería de datos…
- **Traducción automática**: se trata del procesamiento del lenguaje natural de forma que se pueda traducir automáticamente de un lenguaje a otro.
- **Extracción de entidades nominales o Named Entity Extraction**: la definición de entidad con nombre es una palabra o una frase que identifica un elemento o un conjunto de elementos con atributos similares. Esto significa, que, si un usuario introduce una palabra o una frase, se contempla la posibilidad de que haya errado en su ortografía y/o gramática, de forma que se realiza la búsqueda en base a ello, para una mejor optimización. Este tipo de búsquedas incluyen una infinidad de datos: nombres, teléfonos, ubicaciones geográficas, correos electrónicos, empresas, procedimientos de todo tipo, etc. Facilita en gran medida la minería de textos. Un ejemplo para que se pueda entender este punto sería cómo al lanzar la búsqueda sobre 'la victoria de Franco', podrían obtenerse los mismos resultados que al lanzar la búsqueda 'la derrota de la Segunda República'. Las dos búsquedas no tienen palabras en común, sin embargo, se refieren al mismo acontecimiento histórico.

- **Recuperación de la información**: o búsqueda de información relevante. En este punto se refiere a que en cualquier país del primer mundo existen toneladas de documentos escritos a mano, sobre la historia del propio país. Cuando se trata de que un humano digitalice esa información, no existe gran problema, puesto que, si faltan palabras o letras en el documento, la mente humana es capaz de rellenarlas. No ocurre lo mismo con los ordenadores, para los que existe una problemática a la hora de rellenar los espacios en blanco.
- **Resolución Co-referenciada:** aplicada al principio de polisemántica, en un texto buscaría las palabras que se utilizan para referirse a un mismo objeto.
- **Resumen automático:** su finalidad principal es extraer la información útil de documentos, a través de resúmenes o buscando palabras clave. Que la red está sobrecargada de información es un hecho, de forma que el poder mejorar las búsquedas semánticas a través de las palabras más relevantes, ayuda en gran medida a la forma en la que los motores realizan las búsquedas, optimizando las mismas. (Sims, 2015, p. 5-11)

La forma en la que trabaja el PLN, se basa en la definición del LN a través de una serie de niveles: fonológico, morfológico, sintáctico, semántico y pragmático.

4.3.2.1 Niveles de arquitectura PLN

Nivel fonológico

Estudia la relación de las palabras con los sonidos que las representan (ITlligent, 2017).

Nivel morfológico

> En primer lugar, es conveniente definir el significado de léxico. "léxico es el conjunto de información sobre cada palabra que el sistema utiliza para su procesamiento. Las palabras que forman parte del diccionario están representadas por una entrada léxica, y en caso de que ésta tenga más de un significado o diferentes categorías gramaticales, tendrá asignada diferentes entradas.
> En el léxico se incluye la información morfológica, la categoría gramatical, irregularidades sintácticas y representación del significado". (Sosa E., 1997, p. 23-28)

Una vez definido lo que es el léxico por su estrecha relación con la funcionalidad de este nivel, su misión es encontrar la relación entre las unidades mínimas que forman una palabra: sufijos y prefijos y realizar una clasificación de las mismas a nivel morfológico. Un ejemplo para ello podría ser en análisis de la palabra gatos: de la que se podría decir sustantivo+masculino+plural.

El léxico de forma regular contiene únicamente la raíz de las palabras con formas regulares, siendo el analizador morfológico el encargado de determinar si el resto de los atributos de la palabra (como género o número) poseen los atributos adecuados.

Nivel sintáctico

Para comprender este nivel, es necesario definir qué es la estructura sintáctica de una oración, que no es más que el orden de las palabras que la componen. A su vez, comprende tres niveles, que son: nivel de palabras, nivel de oración y nivel de párrafos.

- **Nivel de palabras**: el sustantivo y el verbo, que es el núcleo del predicado de la oración. Alrededor de estos dos elementos se ubican artículos, adjetivos, adverbios, conjunciones, preposiciones, pronombres o interjecciones.
- **Nivel de oración:** se refiere al orden en el que el sujeto y el predicado se sitúan, además de los objetos directos e indirectos y los complementos.
- **Nivel de párrafos**: los párrafos sirven para estructurar la información y ordenar las ideas de forma lógica y jerarquizada. Esta forma de trabajo, requiere que se identifiquen las ideas principales, secundarias y complementarias, y una vez identificadas se escriban por párrafos. En el párrafo se expresarán las ideas separadas por coma, punto, puntos suspensivos o punto y coma, entre otros.

La función principal del nivel sintáctico es determinar cómo las palabras se unen para formar oraciones gramaticalmente correctas. El resultado de este análisis, será generar el esqueleto de las categorías sintácticas constituidas por cada una de las unidades léxicas de la oración (Sosa E., 1997, p. 23-28).

Nivel semántico

El nivel semántico se ocupa del significado de las palabras y el sentido que éstas le dan a la oración. Tal y como se ha explicado con anterioridad, las palabras pueden tener varios significados, de forma que complican en gran medida la comprensión de las oraciones por parte de las computadoras. En este nivel, por lo tanto, el PLN sigue presentando problemas, en cuanto a que las técnicas de representación del significado no han obtenido los resultados deseados, aunque se han realizado avances importantes en este sentido.

La propiedad polisemántica de las palabras puede dar lugar a diversas interpretaciones sobre el significado de una oración. La mejor forma de abordar el problema es con la modularidad, a través de la cual es posible distinguir entre el significado independiente de la palabra y aquel que depende del contexto (Sosa E., 1997, p. 23-28).

El nivel semántico se ocupará por tanto del significado que las palabras tienen por sí mismas y sin tener en cuenta el significado general de la oración, su contexto o la intención que el hablante pretende darle.

Nivel pragmático

Trata sobre cómo el significado de las palabras en distintas situaciones, puede afectar al significado de las oraciones, es decir que el significado de la palabra varía en función del contexto en el que se encuentre. Por ejemplo, cuando se trata de expresiones hechas como ‘a pies juntillas’ o ‘ir al grano’. Es una de las partes más complejas del análisis, dada la ambigüedad lingüística. El significado de la oración dependerá en gran medida de la intención del interlocutor y también de las oraciones inmediatamente anteriores.

4.3.3 Funcionamiento general del PLN

Tal y como se ha explicado anteriormente, la comprensión del lenguaje para los computadores es un proceso complejo. El LN posee propiedades que merman la efectividad de los sistemas de análisis textual.

La forma de realizar el PLN es a través del *lenguaje Python*[3] (Seif G.,2018, p.12), concretamente a través del *Natural Language Tool Kit* (NLTK).

A la hora de procesar el LN, las computadoras no comprenden las palabras, por lo tanto, los programas preparados a tal menester, deberán convertir esas palabras en números. Para optimizar el trabajo del algoritmo, el texto a analizar deberá estar "*limpio*", es decir, lo más libre de errores gramaticales y ortográficos posible y preferiblemente con pocos signos de puntuación. Para la realización de esta tarea hay muchos procedimientos marcados para hacerlo manualmente, librerías propias en Python y otros lenguajes y programas que se dedican exclusivamente a ello. En resumidas cuentas, la cuestión es seleccionar la metodología que se considere más óptima, más rápida y mejor (Bownlee J., 2017, p.11-20). Pero ¿En qué consiste esta limpieza? A continuación, se ejemplificará brevemente la metodología utilizando NLKT, estos serían algunos de los puntos de partida, aunque existen más variantes a analizar dentro del texto:

1. Se seleccionará un texto libre de marcas, es decir, un texto plano sin formato (tipo .txt); en caso contrario, se deberá limpiar de marcas el texto antes de su procesamiento.
2. Relectura del texto: se debe conocer aquello que se va a analizar. Se debe comprobar que no hay faltas de ortografía, tipografía, o de puntuación y que el texto está escrito de manera comprensible.
3. Tener en cuenta signos de puntuación: comas, puntos, apóstrofes, signos de interrogación, admiración, etc.
4. ¿Hay o no hay guiones?
5. Existen en el texto nombres propios (que comienzan por mayúscula)
6. Se encuentran en el texto caracteres numéricos
7. Aparecen o no delimitadores de secciones o puntos como a), b), c) o Sección I, Sección II, etc.

Ejemplo: Se parte de un texto muy sencillo

No te quieres enteraaaar, ¡**Ye! ¡Ye!**

Que te quiero de verdaaaad, ¡**Ye! ¡Ye! ¡Ye! ¡Ye!**

1. Se pasa el texto a texto plano y sin formatos, eliminando la **negrita**:

> No te quieres enteraaaar, ¡Ye! ¡Ye!
>
> Que te quiero de verdaaaad, ¡Ye! ¡Ye! ¡Ye! ¡Ye!

Aunque el alargamiento de la letra 'a' enfatiza la letra de la famosa canción, en este caso no tiene sentido, por lo tanto, habría que eliminarlo:

[3] Lenguaje de alto nivel, de manejo relativamente sencillo, muy utilizado a día de hoy, especialmente para tratamiento de grandes volúmenes de texto. Se explicará más profundamente en apartados futuros.

> No te quieres enterar, ¡Ye! ¡Ye!
>
> Que te quiero de verdad, ¡Ye! ¡Ye! ¡Ye! ¡Ye!

2. Existen diversos signos de puntuación a tener en cuenta
3. No hay guiones en el texto (importantes a tener en cuenta sobretodo en caso de palabras que se completan de una línea a otra)
4. No existen nombres propios, aunque si mayúsculas por inicio de frase y expresión
5. No se encuentran caracteres numéricos

Una vez realizado el análisis, se deberá dividir el texto en oraciones de más o menos 70 caracteres y tokenizar, a través de las herramientas propias del NLKT. La Tokenización consiste básicamente en separar la frase en palabras:

Frase ejemplo: *No te quieres enterar ¡Ye! ¡Ye! Que te quiero de verdad ¡Ye! ¡Ye! ¡Ye! ¡Ye!*.

Tokenización ➔ "No" "te" "quieres" "enterar" "Ye" "Ye" "que" "te" "quiero" "de" "verdad" "Ye" "Ye" "Ye" "Ye"

Una vez el texto está tokenizado se procede a lo que se denomina como 'Bolsa de palabras' o 'Bag words model', que es la forma de representar los datos del texto de forma que el computador pueda comprenderlo. Se construye una matriz *nxt* donde *n* es el número de documentos y *t* es el número de términos únicos (Paruchuri V.,2013, p.35-45).

TABLA 1 MATRIZ PLN

	No	Te	quieres	Enterar	Ye	Que	quiero	De	Verdad
1	1	0	1	1	0	1	1	1	1
2	0	1	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0
6	0	0	0	0	1	0	0	0	0

Dentro de la bolsa de palabras, el orden no es relevante, de lo que se trata es de extraer la información importante en el texto. El propio programa está preparado

para discernir palabras que son iguales, pero están incorrectamente escritas o se diferencia entre mayúsculas y minúsculas, como, por ejemplo:

"*No te quieres enterar"*

"no te qiueres enetrar"

De esta forma, aunque se optimiza bastante el algoritmo, al igualar mayúsculas y minúsculas se pierde información, del mismo modo que se podrían perder algunas palabras, por ejemplo, en el caso de que se iguale: la palabra "*enterar*" por "*entrar*". Dentro de NLKT existen diversidad de funciones dedicadas a esto, funciones que guardan la información sobre mayúsculas y minúsculas y todas las modificaciones que se realizan sobre los caracteres del texto, pero explotar este punto haría que este texto se extendiera innecesariamente; por lo tanto, con la simple mención realizada, será más que suficiente para poder vislumbrar la amplitud de características y opciones que posee el NLKT.

Una vez realizados los pasos anteriormente mencionados; es el momento de que se distingan qué características del texto son importantes y cuáles no. Para eso existen dos formas: la primera, creando un modelo de *Machine Learning* (ML); que se estudiará más adelante, y también de modelo de error; y la segunda haciendo un test de *chi-cuadrado* (Paruchuri V., 2013).

Básicamente, en resumen, el algoritmo PLN de Python está preparado para seguir a grandes rasgos, los siguientes pasos:

1. A partir del texto, genera un árbol sintáctico sobre el que se realiza un análisis semántico; que genera una representación semántica.
2. A partir de la representación semántica, se asocian una serie de valores numéricos al texto.
3. Extrae los valores numéricos que ha asociado. Para que se comprenda mejor, asociará un número concreto, por ejemplo, a la palabra que aparece en el texto '*avión*' y extraerá el número de veces que esa palabra aparece en el texto; y así con todas las demás.
4. Los valores se derivan al texto y se agregan a una serie de vectores
5. Los vectores se colocan en una matriz, donde cada raíz representará un pedazo del texto.
6. El algoritmo descubrirá cual de esos valores es relevante y cuál no, aplicando metodologías matemáticas.
7. El valor de relevancia se determina a través de las librerías, que detectan qué palabras son relevantes y cuáles no.
8. El algoritmo se alimenta de las palabras más específicas y relevantes dentro del texto. (Paruchuri V.,2013)

Con lo dicho hasta el momento, ya se tiene una visión general del funcionamiento del PLN, a partir de este momento se analizarán los puntos que resultan más relevantes dentro de este estudio, como el *Machine Learning*. En apartados posteriores, se ejemplificará de forma gráfica este procedimiento.

4.4 Machine Learning

Machine Learning es una tecnología puntera que nació de la IA y que hoy en día es utilizada por tecnologías tan potentes como Amazon o Google.

Antes de proseguir, es interesante diferenciar la tecnología de ML, del PLN; puesto que en ocasiones son terminologías que van muy de la mano y pueden llegar a confundirse. La diferencia más obvia entre ambos términos es su finalidad:

- ML: su objetivo principal es el de predecir
- PLN: su objetivo principal es comprender el lenguaje natural

4.4.1 Breve historia y definición de Machine Learning

La historia del Machine Learning (ML) se remonta a la propia historia de la IA, con Alan Turing a la cabeza, quien en 1950 creó el '*Test Turing*', en el que una máquina era capaz de hacer creer a un humano, de que era un humano más. En los años posteriores se crearían juegos de damas capaces de mejorar destrezas y aprender de cada partida; y años más tarde, alrededor de 1967, se crearía el "*Nearest Neighbour*", considerado el padre en el reconocimiento de patrones (Rodríguez Rama J. M., 2018, pág. 9, p. 1-3).

Se podría decir que el ML nació en torno a los años 90, gracias al aumento en la capacidad de procesamiento de los ordenadores; y pasó de ser una subdisciplina de la IA, a ser una disciplina en sí misma. Arthur Samuel (1901-1990), científico matemático, pionero en IA y en el desarrollo de los juegos de ordenador; definió el ML como "campo de estudio que da a los ordenadores la capacidad de aprender sin ser explícitamente programados" (Francois Pujet J., 2016, p. 4).

La traducción literal de *Machine Learning* es aprendizaje de máquinas. Machine Learning "es una disciplina científica del ámbito de la Inteligencia Artificial que crea sistemas que aprenden automáticamente" (González A., 2014, p.1). Con *aprender*, se refiere a que el algoritmo sea capaz de localizar patrones complejos dentro del volumen de datos de trabajo. Para funcionar utilizará métodos probabilísticos, estadísticos y matemáticos (Estévez M., 2018, p.4), que funcionarán combinados a través tanto de *algoritmos de aprendizaje no supervisado*, que son algoritmos donde solo se conocen las variables de entrada, que son agrupadas en función de sus similitudes; como de *algoritmos de aprendizaje supervisados*, que encuentran patrones ocultos en los datos de entrada (Estévez M., 2017, p. 10-12).

4.4.2 Usos más comunes

Puesto que el ML es una disciplina muy versátil, el número de campos al que se puede aplicar es bastante extenso, siendo la tendencia a ocupar cada vez más ámbitos. Algunos

de los usos más comunes son los siguientes:

- **Predicciones:** Aplicado a *Business Inteligence* (BI), se trata de convertir los datos en información y ésta a su vez, transformarla en fuente de conocimiento; para poder ser aplicado en la toma de decisiones de los negocios. Basándose en ingentes cantidades de datos, ayuda a los negocios a saber qué se va a vender más o cuál será el producto que obtendrá las mayores pérdidas (Cohen G., 2012, p.3).
- **Detección de intrusos:** Puesto que dentro de la red la información fluye de manera masiva y ésta se utiliza para hacer todo tipo de transacciones, cada vez es más necesaria la presencia de expertos en seguridad informática. Con el fin de disminuir la dependencia existente de estos expertos, se han desarrollado programas ML que detecten posibles intrusos; concretamente sistemas de detección de intrusos a través de la minería de datos (Peluffo I., Capobianco M., Echaiz J.,2014).
- **Antivirus:** Las grandes compañías invierten millones de dólares en mejorar sus sistemas de seguridad. A fin de disminuir de forma palpable el número de ciberataques que se producen, se combaten con antivirus basados en ML, capaces de aprender de sus propias vulnerabilidades y mejorar por sí mismos la protección que se brinda al sistema, demostrando que la aplicación de esta tecnología mejora sustancialmente el porcentaje de malware detectado (Raghunarayan R., 2018).
- **Clasificación de texto:** se podría definir como "la acción ejecutada por un sistema artificial sobre un conjunto de elementos para ordenarlos en clases o categorías" (Cárdenas Juan P., Olivares G., Alfaro R., 2014). Debido a los grandes volúmenes de texto que se almacenan en dispositivos electrónicos, la clasificación de textos a través de ML ha cobrado gran relevancia, debido a las mejoras tanto en la facilidad de la implementación como en los resultados obtenidos.
- **Productividad:** La aplicación de ML sobre los datos de la empresa, permite localizar mejoras en costos, tiempo y procesos de fabricación (DigitalHouse, 2018)
- **Recomendación de productos**: Aplicando ML a las preferencias de los usuarios para así realizar recomendaciones de compra de servicios u objetos de forma más certera y ajustada a sus gustos y preferencias (Walid Ghobar E., 2017, pág. 11).
- **Bots de soporte:** "Un bot es un programa informático que imita el comportamiento humano simple y realiza una tarea automatizada como puede ser programar una alarma o mostrar el tiempo cuando se solicita el parte meteorológico" (Cossío A., 2018, pág. 20). Un chatbot es un bot conversacional, es decir, un bot capaz de mantener una conversación con un humano. Como por ejemplo Alexa o Siri.

- **Customer Churn:** o pérdida de clientes. Gracias a la aplicación de ML sobre los datos que se tienen del cliente, se puede averiguar y prevenir porqué un cliente quiere dejar de utilizar los servicios de la compañía, del mismo modo que se pueden perfilar qué características poseen los clientes que más posibilidades tienen de abandonar los servicios que ofrece la empresa (Schoenbaum D., 2018, p.9-11).
- **Lingüística computacional**: se ha abordado este punto en apartados anteriores (*1.2 Procesamiento del Lenguaje Natural*), en el que se mencionaban los siguientes usos: análisis, traducción automática, extracción de entidades nominales o *named entity extraction*, recuperación de la información, resolución co-referenciada o resumen automático.
- **Deep Learning:** esta es sin duda una de las ramas del ML que más ha crecido. Deep Learning es un tipo particular de ML, base de algunos productos de Google. Presenta la particularidad de necesitar una red neuronal artificial jerárquica; similar a la que se presenta en los mamíferos; en la que el primer nivel aprende algo sencillo y envía la información al siguiente, que la procesa, añade información adicional, se la envía al siguiente y así sucesivamente (Ramírez V., 2018, p. 4-12). Representa un acercamiento al sistema nervioso humano, imitando su arquitectura para detectar características ocultas en los datos de entrada (Arrabales R., 2016, p. 12-17).
- **Biométrica:** Consiste en aplicar la tecnología de ML a la biometría; siendo la biometría *"*el reconocimiento automático de los individuos en función de sus características biológicas y de comportamiento" (Universidad Internacional de Valencia, 2016).

4.4.3 Funcionamiento básico de ML

ML funciona básicamente encontrando una función o una relación entre los datos de entrada y los datos de salida, para que se comprenda mejor, conviene imaginar por ejemplo un juego de ajedrez: a la hora de programar se podría abordar la problemática desde dos frentes:

1. El primero de los frentes se ocuparía de estudiar todas las jugadas posibles dentro del juego de ajedrez, teniendo en cuenta la posición de las piezas del oponente y las propias y los movimientos que se permiten en las piezas. Es particularmente complejo, puesto que es prácticamente imposible que el programador controle todas las jugadas posibles de antemano.
2. El segundo de los frentes, o la segunda gran opción; sería poder introducir las reglas del juego en el programa y que éste decida cuáles son los mejores movimientos, en base a que *'ganar'* es el objetivo y '*perder*' es la penalización. El programa será capaz de aprender cuáles son los movimientos que tienen *'premio'* y evitará en la medida de lo posible perder; de esta forma *aprenderá* a jugar (Maverick L., 2017, p. 1-9)

El término de aprendizaje, posee varios significados; por ejemplo, de acuerdo con el economista y politólogo Herbert Alexander Simon, el aprendizaje son "cambios adaptivos en el sistema para hacer la misma tarea de la misma población de una manera

más eficiente y efectiva la próxima vez*"* (Cyert R. M., Simon H. A. y Trow D. B.,1956, v. 29, pág. 237-248) o con el científico computacional y profesor en la Universidad de Carnegie Mellon (CMU) en Estados Unidos, Tom Mitchell, que lo define como "un programa de computadora se dice que aprende de experiencia E con respecto a una clase de tareas T y medida de desempeño D, si su desempeño en las tareas en T, medidas con D, mejoran con experiencia E*"*(Mitchell M. T., 1997, pág. 2-7). La búsqueda en general de ML, es que los programas mejoren sus algoritmos a partir de su experiencia. Tal y como se pinceló con anterioridad, para llevar a cabo su trabajo, ML utilizará dos tipos de aprendizaje: supervisado y no supervisado, que pese a lo que pueda parecer el significado de supervisado y no supervisado, poco o nada tienen que ver con la intervención humana dentro de los algoritmos de aprendizaje. A continuación, se especificará en qué consiste el aprendizaje supervisado y no supervisado.

4.4.3.1 Aprendizaje supervisado:

A grandes rasgos, con el aprendizaje supervisado se prepara al algoritmo concediéndole una serie de entradas, con sus salidas. La finalidad que tiene el proceso es que el algoritmo sea capaz de combinarlas y con la información resultante, hacer predicciones. Para ello se dispondrá de un conjunto de datos, formado a su vez por un conjunto de entrenamiento y otro de prueba (García Cambronero C., 2012, pág. 1). Al disponer de estos datos, el algoritmo se '*entrena*' con el histórico de datos y '*aprende*' a asignar el valor esperado de salida, de forma que puede predecir qué valores van a aparecer en las salidas a partir de las entradas.

Un ejemplo de aprendizaje supervisado, puede darse dentro del correo electrónico. Cuando un usuario recibe un correo, el propio buzón determina si es o no '*spam*'. Antes de enviarlo a la carpeta de 's*pam'*, la información del correo recibido ha sido procesada a través del algoritmo de aprendizaje supervisado, en el que, a partir de datos como el remitente, las etiquetas del histórico de correos electrónicos, la IP, el texto o las imágenes recibidas, el propio buzón del correo puede discernir si se trata o no de un correo no deseado, enviándolo directamente y sin previo paso por el *buzón de entrada*, hacia el directorio de *spam*.

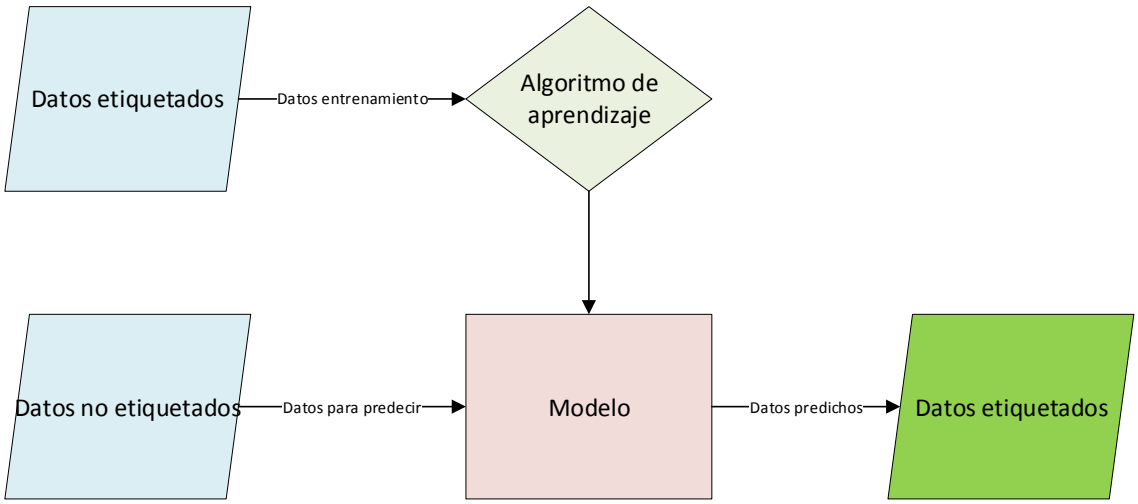


ILUSTRACIÓN 5 APRENDIZAJE SUPERVISADO (ELABORACIÓN PROPIA)

Los dos tipos de aprendizaje supervisado que existen son a través de regresión y a través de la clasificación, siendo a su vez:

- Regresión: predice un número, como por ejemplo el precio de un artículo, el número de entradas de un festival, etc.
- Clasificación: predice una categoría, como en el ejemplo anterior del *spam*.

Los ejemplos de uso más común de algoritmos de aprendizaje supervisado son de lo más variopintos, desde las redes neuronales artificiales, árboles de decisión o máquinas de vectores de soporte, pasando por la programación de vehículos autónomos.

4.4.3.2 Aprendizaje no supervisado

Aunque el funcionamiento entre los algoritmos de aprendizaje supervisado y no supervisado es muy similar, poseen ciertas diferencias. La diferencia fundamental es que los algoritmos de aprendizaje no supervisado o *clustering*, no poseen información sobre las salidas que se esperan a unas determinadas entradas, además de que no etiquetan o clasifican la información; buscan similitudes dentro de los datos de entrada, aunque nada garantiza que esas similitudes vayan a proporcionar información de utilidad.

El aprendizaje no supervisado, toma su nombre por lo subjetivo de su algoritmo, ya que a priori, no tiene respuestas correctas o incorrectas, sino que sirve para descubrir y presentar estructuras de datos que tienen cierto interés (Gonzalo de Alba A., 2018, p. 12-15).

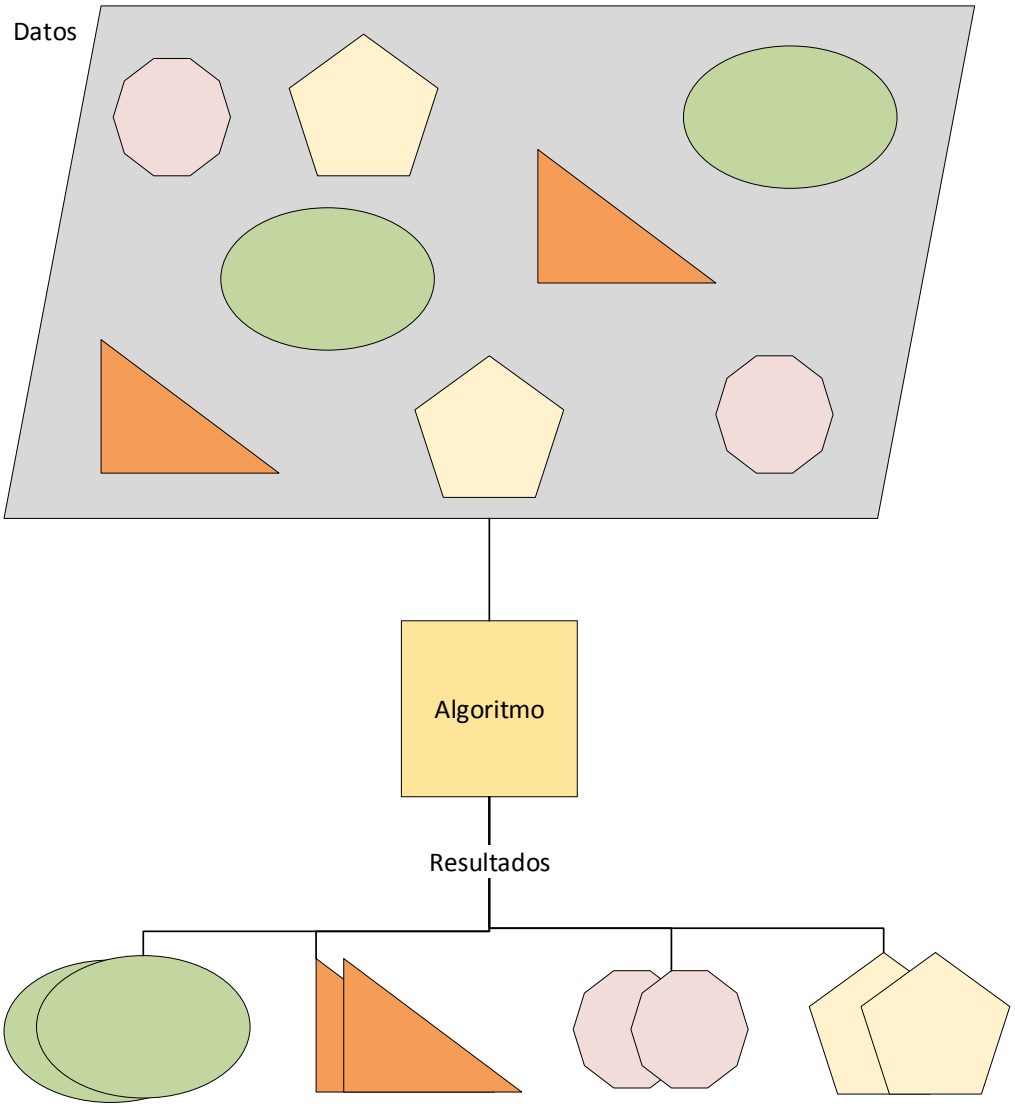


ILUSTRACIÓN 6 APRENDIZAJE NO SUPERVISADO (ELABORACIÓN PROPIA)

Existen dos tipos de algoritmo no supervisado:

- **Hebbiano**: pretende medir la familiaridad o extraer características de los datos de entrada.
- **Competitivo y cooperativo:** trabaja a través de redes neuronales artificiales, las neuronas individuales aprenden a especializarse sobre conjuntos de patrones similares y a detectar características comunes de los patrones de entrada (Muñoz Pérez J., ss. Ff., pág. 1-3).

Las aplicaciones principales para las que se utilizan los algoritmos de aprendizaje no supervisado son para la compresión sin pérdida (lossless compression) y el tratamiento de imágenes.

No se podría especificar cuál de los dos algoritmos resulta mejor, más rápido o más óptimo. La diferencia entre la utilización del algoritmo de lenguaje supervisado, no supervisado, o una mezcla de los dos tipos, dependerá del tipo de problema que haya que resolver y de los datos que se posean.

5. Interpretación y prevención de errores en la lectura de logs a través de Machine Learning, la Mediación de telefonía y el Business Support System

Tal y como reza el enunciado, en este punto se abordará la interpretación y prevención de errores en la lectura de los logs, a través de ML; para ello se explicarán qué son y para qué sirven los logs, generalidades y particularidades de los mismos, la problemática y el reto que suponen para el mundo de la informática y su arquitectura. Además, dentro de este apartado, se hablará del Business Support System, qué es, en qué consiste, y qué parte del mismo engloba a la Mediación de Telefonía.

Para finalizar, se hablará de la herramienta eIUM de HP, que resulta básica en la Mediación; se explicará en qué consiste y el tipo de logs que genera, para finalmente explicar la aplicabilidad de la tecnología de ML dentro de la lectura de los logs.

5.1 La minería de datos

Aunque la minería de datos puede sonar como un término novedoso, apareció por primera vez en los años sesenta, cuando las capacidades del mundo de la informática aún estaban *en pañales*. También conocida como *Data Mining*, se podría definir la minería de datos como "el conjunto de técnicas y tecnologías que permiten explorar grandes bases de datos con el objetivo de encontrar patrones que nos puedan aportar información valiosa en la toma de futuras decisiones" (Brunet Robert, 2015, p.3).

La minería de datos surge como respuesta a una necesidad, la necesidad de interpretar una enorme cantidad de datos, a fin de poder utilizarlos para obtener información útil que ayudara a las empresas a comprender, por ejemplo, los datos de consumo de sus clientes o las preferencias de los mismos, a fin de conseguir sus objetivos.

Los pasos que los mineros de datos deben llevar a cabo para poder extraer esta información son los siguientes (Brunet Robert, 2015, p.4):

- Determinación de los objetivos: se deben definir los objetivos que el cliente pretende conseguir a partir de la explotación del data mining.

- Procesamiento de los datos: este paso requiere seleccionar, cribar y transformar la base de datos del cliente, para extraer la información útil.

- Determinación del modelo: llegados a este punto, se debe elegir el algoritmo que regirá las búsquedas a través de la IA, es un paso delicado puesto que hay multitud de algoritmos (árbol de decisión, regresión lineal, red neuronal…), el objetivo será la obtención de los mejores resultados de acuerdo con los objetivos que se pretenden conseguir

- Análisis de los resultados

La interpretación de logs forma parte de la minería de datos, aunque la pretensión en los análisis de logs suele ser la detección de problemáticas o posibles

mejoras dentro de los sistemas, es importante saber que lo que se manejan son grandes cantidades de datos y el objetivo final es la búsqueda de información útil para la mejora en el sistema.

5.2 El Business Support System

Hablar del Business Support System (a partir de ahora BSS), va a requerir una descripción de cómo funcionan los servidores de telecomunicaciones. Los servidores de telecomunicaciones, poseen dos grandes áreas: BSS y OSS (Operational Support System). Esta división podría realizarse entre más áreas, dependiendo de las necesidades de las compañías.

Una traducción literal del BSS, podría ser *Sistema de Apoyo Empresarial*, aunque sería más acertado definirlo como un conjunto de servicios que las empresas de telecomunicaciones ofrecen o poseen para poder realizar sus operaciones comerciales. Aun explicándolo así, no queda completamente claro qué es el BSS.

Si bien a continuación se explicará en qué consiste el BSS y el OSS y cuáles son sus funciones principales, se debe tener en consideración la cercanía de los dos términos y que, aunque se haga el esfuerzo de definir y delimitar las labores de cada una de las áreas, dependiendo de la configuración de las organizaciones, estas definiciones podrían variar significativamente.

- **BSS**: se asocia con la capa de gestión empresarial de los procesos internos (Rodríguez Olim D. J., 2017, pág 26), el elemento complementario al OSS, que se encarga de la administración de los elementos del negocio, incluye herramientas para la atención al cliente, cobro, facturación, etc. (Fernando Negrete J., 2014, p.5).
- **OSS**: se asocia con la entrega de servicios a los clientes (Rodríguez Olim D. J., 2017, pág 26), engloba los sistemas de red que están vinculados con las operaciones; por ejemplo, detección de errores, requisitos de la red, mantenimiento, etc. Es lo que permite a los operadores de telecomunicaciones mantener sus redes en funcionamiento (Fernando Negrete J., 2014, p.4).

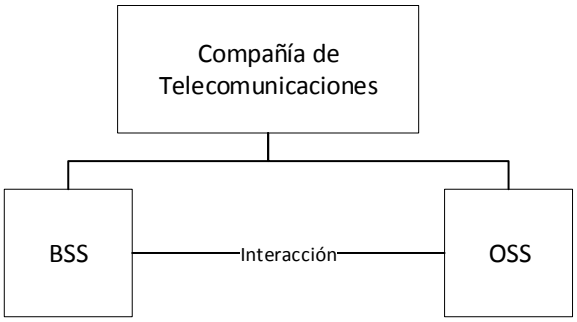


ILUSTRACIÓN 7 COMBINACIÓN BSS-OSS (ELABORACIÓN PROPIA)

Tal y como se observa en la imagen superior (Ilustración 7), la combinación de BSS unido a OSS, conforma todo el sistema de ingeniería de telecomunicaciones, son áreas que están en constante crecimiento y evolución, puesto que deben realizar importantes modificaciones acordes a la evolución de las telecomunicaciones y de la sociedad, así como labores de optimización de los recursos.

Lo cierto es que los límites de OSS y los de BSS están tan estrechamente unidos, que en la actualidad se refiere a estas herramientas de soporte de negocio como si fueran una única: el BSS/OSS o OSS/BSS.

Se podría considerar que el soporte de negocio BSS/OSS posee un enfoque E2E, es decir de extremo a extremo, que incluye desde que se detecta una necesidad en la organización hasta que está necesidad resulta satisfecha, pasando por todos los procesos intermedios.

La necesidad de comprender a grandes rasgos el funcionamiento de las centrales telefónicas, radica en la información que éstas generan y cómo las operadoras de telefonía procesan esta información. Para su comprensión se hará un breve repaso a la historia de la tecnología móvil, que ha impulsado la evolución de las centrales desde su creación hasta el día de hoy.

5.2.1 Breve historia del móvil

Para explicar este apartado, se podrían mencionar a las grandes mentes gracias a las cuales se ha fraguado el imponente sistema de comunicaciones que la humanidad posee en pleno siglo XXI; personalidades tan importantes como Alexander Graham Bell creador del primer teléfono patentado de la historia (1876), pasando por Guglielmo Marconi, Nikola Tesla o la empresa AT&T; sin embargo, el objeto de este apartado se centrará en el funcionamiento de las centrales telefónicas a día de hoy, la recepción de la información por parte de las operadoras y el procesamiento de la misma.

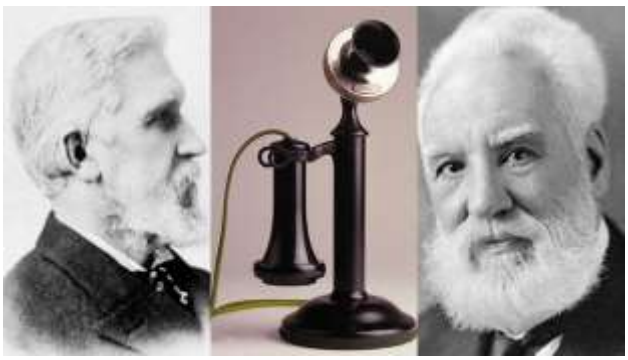


ILUSTRACIÓN 8 ALEXANDER GRAHAM BELL Y EL PRIMER TELÉFONO DE LA HISTORIA (TECHNOISTORIA WEBSITE)

La idea de 'central telefónica' surge por primera vez hacia el año 1949, época en la que los radio-teléfonos dominaban las comunicaciones y en cada ciudad importante se erguía tan sólo una central telefónica. La llegada en 1973 del primer teléfono móvil marcó un antes y un después en los sistemas de telecomunicaciones, gracias al doctor Martin Cooper, considerado el padre de la telefonía celular. Tan sólo 10 años después la AMPS (Sistema Avanzado de Telefonía Móvil) lanza un ancho de banda desde 800 MHz, con un sistema completamente automatizado y crea el que sería el primer estándar en telefonía móvil del mundo

ILUSTRACIÓN 9 MARTIN COOPER REALIZANDO LA PRIMERA LLAMADA DESDE TELÉFONO MÓVIL (XATAKAMOVIL)

(Inzaurralde M., Isi J., Garderes J., 2016, pág 6-8).

La segunda generación de móviles nace al comienzo de los años 90 de la mano de un nuevo estándar y de la telefonía digital, el conocido como GSM o *Global System for Mobile Communications*. El ancho de banda sigue aumentando y las prestaciones del teléfono móvil van en aumento, ofreciendo la posibilidad de enviar mensajes de texto además de la posibilidad de hablar y escuchar. Un dato relevante es que, en 1997, la telefonía móvil ya contaba con más de 50 millones de usuarios en todo el mundo.

La tercera generación de móviles viene de la mano de la convergencia de la voz y datos inalámbricos con Internet, con las aplicaciones multimedia y las transmisiones de grandes cantidades de datos, además de la llegada de los vídeos a las aplicaciones y las videollamadas (Rodríguez Gámez O., Hernández Perdomo R., Torno Hidalgo L., García Escalona L. y Rodríguez Romero R., 2005, pág 5-6).

En estos momentos estamos inmersos en la cuarta generación de móviles, caracterizada por su gran velocidad, capacidad de transmisión de datos y la facilidad de conectarse prácticamente desde cualquier lugar.

ILUSTRACIÓN 10 EVOLUCIÓN DEL MÓVIL (CULTURACION WEBSITE)

El avance en el mundo del móvil es imparable. Está a punto de llegarla quinta generación. Ya existen operadoras que se han lanzado a la conquista de esta tecnología, como *Huawei* o *Vodafone* (Álvarez E., 2018). Se especula con la posibilidad de aplicar el 'internet de las cosas' de manera masiva, las ciudades inteligentes, la realidad aumentada, vídeos de calidades inimaginables… Lo cierto es que en cuanto a tecnología se refiere, se están dando pasos agigantados en un corto espacio de tiempo, así que cualquier escenario es posible.

5.2.2 Las centrales de telefonía

Aunque la tecnología que llevan las centrales de telefonía es muy compleja, en este apartado se explicará a grandes rasgos su funcionamiento, a fin de poder esbozar una idea general de cómo una llamada desde un móvil, termina reflejándose dentro de la factura mensual.

Tal y como se ha explicado en el apartado anterior, la 3G trajo consigo el sistema *Universal Mobile Telecommunications System*, a partir de ahora UMTS, conocido también como Telecomunicación Internacional Móvil 2000 o IMT 2000.

Una de las grandes ventajas del sistema UMTS, es que proporciona una velocidad alta en las comunicaciones móviles, con un coste relativamente bajo. Aunque su nacimiento tiene raíces en la base de la tercera generación, sigue vigente y en uso a

día de hoy. Pero cada vez está siendo más desplazado por el uso de la LTE o *Long Term Evolution*, de la 4G, caracterizada por su alta velocidad, que responde mejor a las crecientes necesidades de los usuarios.

En las décadas de los 70-80, se creó un sistema de *células*. Las células no son otra cosa que áreas. Por ejemplo, una ciudad se divide en X células de forma hexagonal a imitación de los panales de abejas y de tamaño variable, dependiendo de la densidad de población de la misma. Cada célula posee una central o estación base.

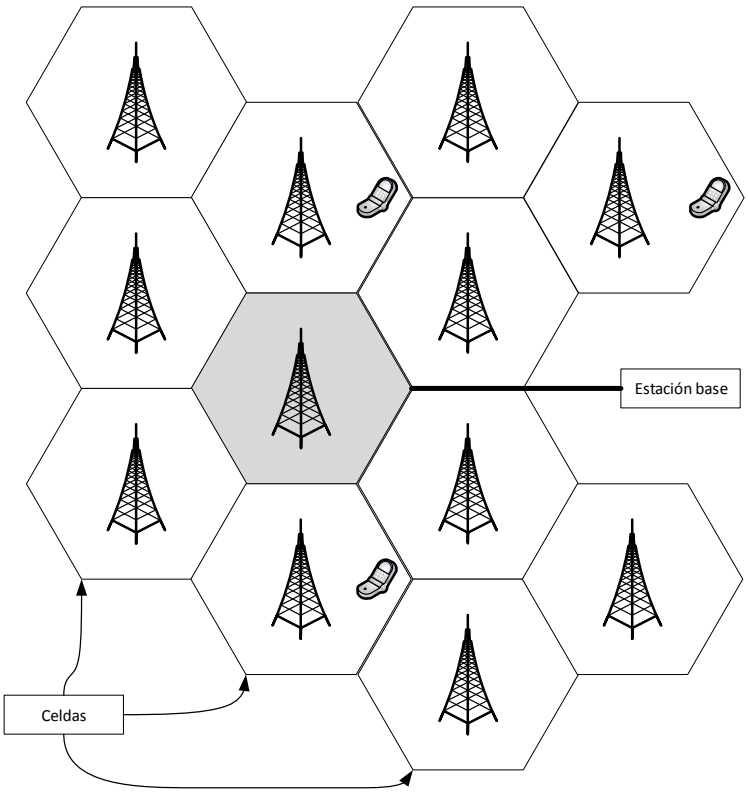


ILUSTRACIÓN 11 DISEÑO DE LAS REDES DE TELEFONÍA MÓVIL (ELABORACIÓN PROPIA)

Las centrales sirven para conectar radios de baja potencia, como son los dispositivos móviles. Las centrales o estaciones base, administran las llamadas de los móviles de una determinada región geográfica o área de cobertura. Sus cometidos son la "asignación de frecuencias, el manejo de una base de datos en la que se registran todas las llamadas que se realizan, el registro de todos los usuarios potenciales de la región y la transmisión de la señal a otras estaciones base" (Lara Velázquez P., Gallardo López L., Gutiérrez Miguel Á., 2000). Se componen de *antena o antenas emisoras y* receptoras de señales de radio, equipos electrónicos y eléctricos, baterías y sistemas de refrigeración. En cada célula se pueden producir simultáneamente cientos de llamadas. El sistema de centrales permite la movilidad a los usuarios, ya que en el transcurso de una llamada se puede pasar de una célula a otra, sin que el usuario sea consciente de ello. Cuando el usuario cambia de célula, la célula abandonada libera el subcanal utilizado y lo deja libre para que otro usuario lo utilice.

Debido al creciente tráfico de datos en los móviles, también suministran cobertura para ellos; pero no se detecta la necesidad de ahondar más en el funcionamiento de esta tecnología, puesto que el objeto de estudio de este documento se centra más en los resultados de las comunicaciones, que en las comunicaciones en sí mismas.

Gracias a los datos que proporcionan las centrales, trabajan las operadoras telefónicas. Con ellos se puede saber la duración de la comunicación, la razón por la que se cortó (pérdida de cobertura, interrupción por parte de un usuario, etc.), si se llamaba a

un teléfono desviado, a un buzón de voz, con llamada oculta, desde un fijo, desde un móvil y una larga lista de etcéteras en forma de características propias de la llamada.

A partir de estos datos, las operadoras deberán discernir entre los que se envían a facturación, a retención de datos, a detección de fraude o a otros sistemas, como pueden ser otras operadoras. Las operadoras tienen acuerdos entre

La evolución del número de las estaciones base en España, para poder dar cabida a la amplia demanda de los dispositivos móviles, ha sido la siguiente, de acuerdo con el portal de estadísticas *Stadista*, es la siguiente (Ilustración 12 Evolución de las estaciones base en España: 2005 al 2016).

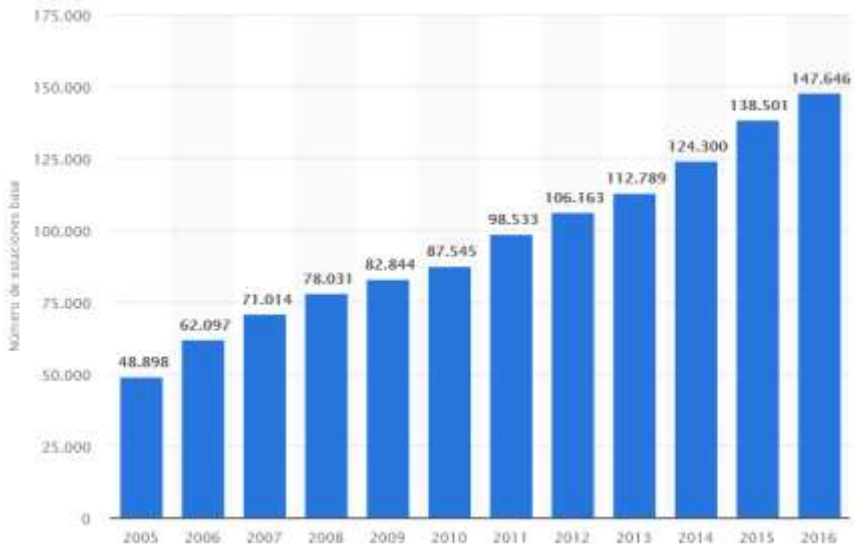


ILUSTRACIÓN 12 EVOLUCIÓN DE LAS ESTACIONES BASE EN ESPAÑA: 2005 AL 2016 (STADISTA PORTAL DE ESTADÍSTICAS)

La compra de las estaciones base por parte de las operadoras de telefonía ha sido paulatina, dado que hasta el año 1994, en España tan sólo existía una operadora: Telefónica, a partir de ese año el monopolio se liberó, comenzando tímidamente a llegar otras operadoras, como *Airtel*, a día de hoy llamada Vodafone (Herrero J., 2013). La compra de centrales ha tenido una evolución relativamente reciente, inicialmente todas eran posesión de Telefónica y poco a poco, la mayoría de las operadoras poseen en su haber centrales; aunque existen otras, conocidas como *operadoras virtuales*, cuyas políticas les impiden la posesión de las mismas y 'alquilan' las redes al resto de operadoras; pero este dato se añade como algo anecdótico puesto que el funcionamiento interno y los acuerdos entre operadoras pueden hacer prolongarse en demasía este apartado y no forman parte del objeto de estudio de este documento.

5.2.3 El BSS aplicado a la Mediación de telefonía

En este punto del estudio, donde ya se tiene una idea más o menos clara del funcionamiento, tanto del BSS/OSS, como de las centrales o estaciones base de telefonía, se procederá a describir los límites y el rango de aplicación del BSS/OSS en el área de la mediación de telefonía. Antes de continuar, se deberá tener en cuenta que

puede haber variaciones más o menos grandes dentro del ámbito que abarca el BSS/OSS, puesto que forma parte de las divisiones departamentales y de elementos de negocio que se realizan en las empresas, con el fin de obtener el mayor aprovechamiento de sus recursos.

Se podría decir que el BSS/OSS, dentro de las operadoras de telefonía puede abarcar desde el momento en el que el usuario realiza un consumo, pasando por la creación del fichero con los datos del consumo dentro de la estación base, hasta que el dato de este consumo llega a los sistemas finales y por lo tanto abarca las centrales, los distintos sistemas de procesamiento de datos y envíos a otros sistemas.

Aunque como ya se ha comentado, el concepto de BSS/OSS es lo bastante amplio como para que no se pueda especificar de qué departamentos se compone o como se divide, puesto que depende en gran medida de cómo se divide y organiza la empresa a la que pertenece, una división posible dentro de la mediación de telefonía podría ser la siguiente:

- *Product management*: o gestión de productos, apoya el desarrollo de productos, las ventas y la gestión de productos, ofertas y paquetes para empresas y clientes del mercado masivo. Incluye ofertas y descuentos de productos cruzados, precios adecuados y la gestión de cómo los productos se relacionan entre sí.
- *Customer Management*: o gestión de clientes. Para los proveedores de servicios se requiere una visión única del cliente, que obtienen gracias a las aplicaciones orientadas al cliente (gestión de relaciones con el cliente). La gestión de clientes puede considerarse como un sistema completo de gestión de las relaciones con los clientes implementado para ayudar a los agentes de atención al cliente a manejar a los clientes de una manera mejor y más informada.
- *Revenue management:* o gestión de ingresos. Se centra en la facturación, el cobro y la liquidación.
- *Order Management:* o gestión de pedidos. El BSS es a menudo el motor comercial para la gestión del cumplimiento y el aprovisionamiento de pedidos.

ILUSTRACIÓN 13 BSS ESQUEMA (INTPLE WEBSITE)

Ahora que ya se ha explicado hasta dónde puede llegar el BSS/OSS en una operadora de telefonía, se analizará los servicios de los que se compone la Mediación de telefonía. La Mediación de Telefonía, posee varios departamentos o varios grupos de trabajo, como puedan ser: fraude, desarrollos e incidencias, *Data Retention And*

Guardian Online (más conocido como *dragón*) o *billing*, cada uno de ellos con una serie de funciones bien definidas:

- Fraude: se ocupa entre otras muchas funciones, del análisis y la detección de casos potenciales de fraude, incluye la detección de desviaciones en el perfil de uso de los clientes, puede manejar grandes volúmenes de datos con tiempos de respuesta cortos y posee gran flexibilidad: capaz de interconectar, recoger y *correlacionar*[4] datos de distintas fuentes en tiempo real, además posee alarmas parametrizables para la detección de colisiones (dobles usos) y velocidad, comprueba destinos y mantiene el control de los nuevos clientes, así como las 'listas negras'.

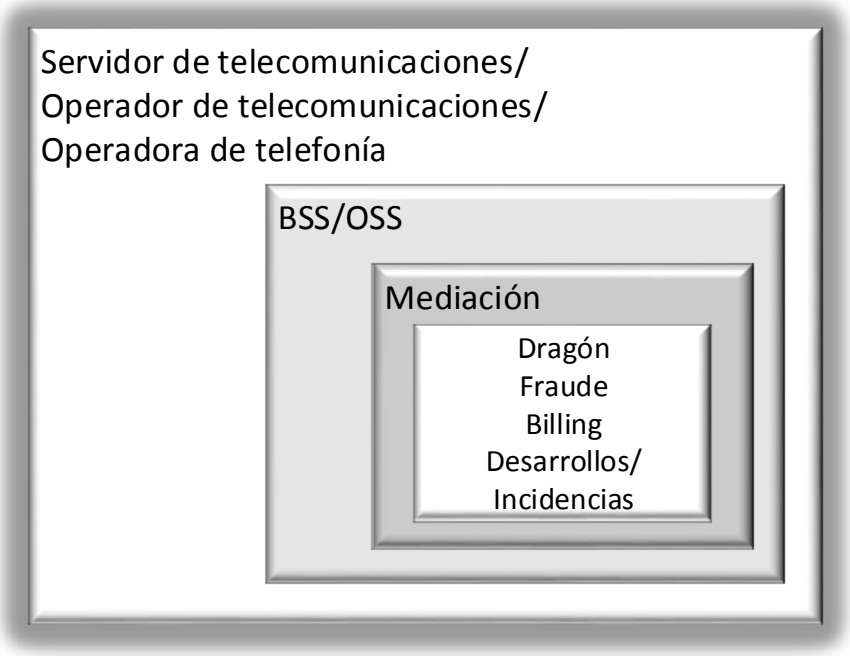


ILUSTRACIÓN 14 ESQUEMA MEDIACIÓN (ELABORACIÓN PROPIA)

- DRAGON: tiene labores muy importantes de cara a la legalidad, puesto que, entre otras funciones, se encarga de la retención de datos de movimientos de teléfonos móviles o fijos, y su archivado, estos datos se guardarán durante un tiempo determinado de acorde a la legalidad vigente; gestión de garantías y solicitudes, como las que puedan venir de mano de jueces que soliciten los movimientos telefónicos de uno o varios números; o el rastreo de números.
- Billing: se encarga de la facturación y envío de facturas a clientes finales.
- Desarrollos e incidencias: es el equipo que custodia, actualiza y modifica el sistema, de acuerdo con nuevas especificaciones o solución de problemas detectados.

A la hora de analizar los logs y de aplicar el Machine Learning o cualquier otra tecnología al sistema, es importante tener en consideración que cualquiera de estas nuevas tecnologías debería tener acceso a la información que se deriva, tanto de desarrollos como de incidencias dentro del sistema. Este sería un punto relevante puesto

[4] La correlación es una técnica estadística que se usa para determinar la relación entre dos o más variables (Ramón Gustavo S., ss. Ff., pág 1)

que, en base a las necesidades del sistema, se modifican especificaciones, flags, diferencias en la codificación de los ficheros, campos de entrada o de salida, etc. Toda esta información en caso de no estar accesible, derivará con total seguridad en un resultado final catastrófico en el que el aprendizaje automático no podrá realizarse o su ejecución será insatisfactoria o ineficiente.

5.3 Los logs

Cada vez que se produce un error dentro de un sistema, el administrador se dirige al directorio en el que se encuentran los logs principales para empezar a investigar qué es lo que ha sucedido, en base a esto, en este apartado se analizarán qué son los logs, cómo se generan y para qué se utilizan.

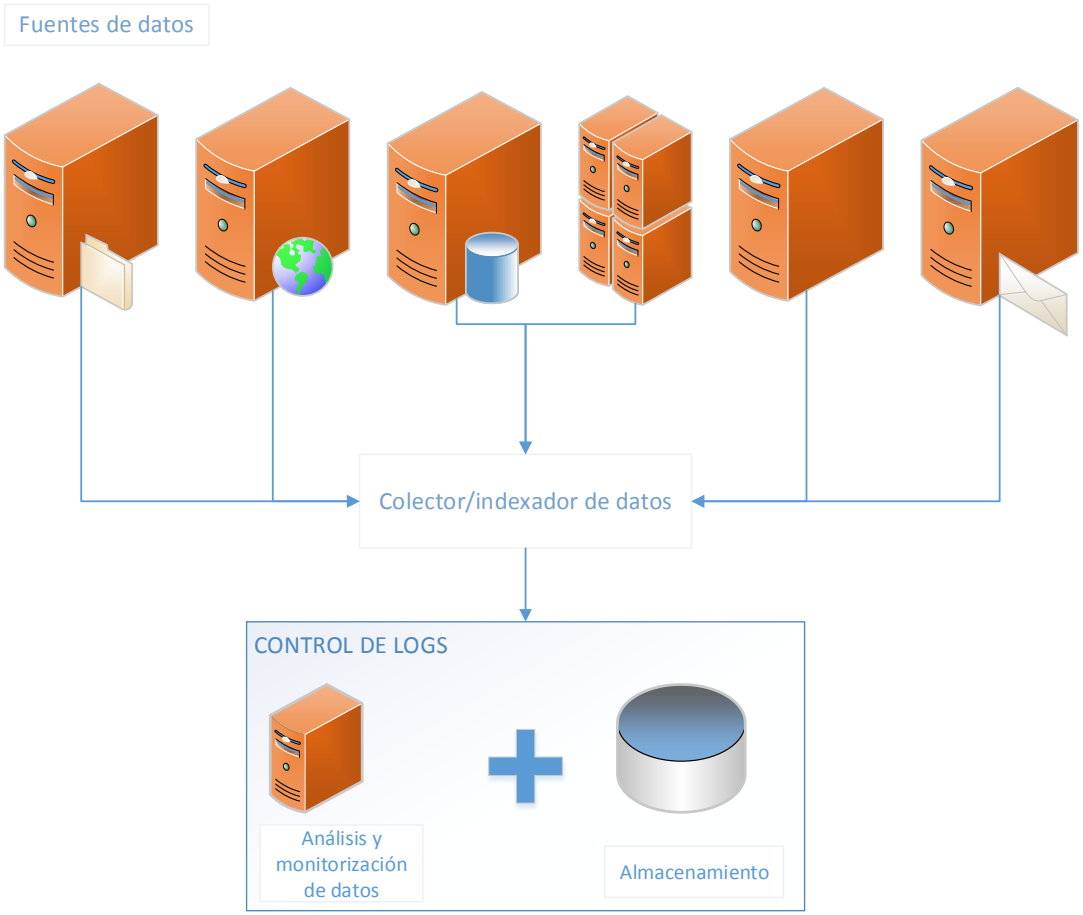


ILUSTRACIÓN 15 FUENTES DE DATOS DE LOS LOGS (ELABORACIÓN PROPIA)

5.3.1 Descripción y usos

De una forma literal, traducida del inglés, la palabra log equivale a *registrar, anotar, tomar nota, anotarse*, que tiene mucho que ver con el significado que se le da en el ámbito informático. De acuerdo con la definición encontrada en el artículo de Jose Luis Ortega Priego, miembro del Consejo Superior de Investigaciones Científicas; de la revista española de documentación científica (Ortega Priego J. L., 2004. Pág. 457, p.1),

se define a los log como: "archivos generados por un sistema operativo o por un determinado programa donde se reflejan sucesos ocurridos dentro de una aplicación".

Los logs registran toda clase de eventos que ocurren dentro de un sistema, una red, un dispositivo o una aplicación. Cada entrada que se registra dentro de ellos, se corresponde con un suceso acontecido dentro del sistema al que hacen referencia. A modo de aclaración, los logs servirían, por ejemplo, para saber qué pasos se han seguido a la hora de realizar una determinada acción que ha tenido como resultado: la generación de un documento, la ejecución de un aplicativo, el asalto a un servidor, etc. Es información que no está disponible para los usuarios de *a pie*, sino para administradores o usuarios con un perfil alto en el sistema.

Los logs son una herramienta realmente útil a la hora de mantener un sistema, puesto que generan información; y es bien sabido que, especialmente en la era de la digitalización, la información es poder. Originalmente se crearon para proporcionar información acerca de los errores que ocurrían en el sistema, a fin de analizarlos y depurarlos. Hoy en día y gracias a la información que se almacena en los ficheros de log, se puede obtener información de todo tipo sobre lo que ocurre en dispositivos, sistemas o servidores: conocer quién ha modificado determinados archivos, desde dónde se ha accedido a los servidores o qué acciones o eventos se produjeron en la última entrada en el sistema; todo dependiendo del propósito para el que los logs fueron creados. Los logs son generados de forma automática por las aplicaciones, dispositivos o los sistemas, de forma que el lenguaje que se utiliza en ellos es de tipo *formal*; no da lugar a segundas interpretaciones ni a dobles sentidos: proporciona información para aquello para lo que fue programado, ni más, ni menos.

La mayoría de los logs se generan en texto plano, lo cual resulta más sencillo a la hora de abrirlos desde cualquier editor de texto simple, como por ejemplo el *vi* de UNIX o el *wordpad* de Windows; y tanto su elaboración, conservación y revisión periódica es responsabilidad de las compañías que gestionan los sistemas en los que se producen, que suelen darle la importancia que merecen. A nivel de usuario, también se producen logs, pero no suele ser necesario su análisis.

La razón por la que la información de los logs es tan valiosa es porque permite detectar problemas dentro del sistema a prácticamente cualquier nivel: desde bases de datos corruptas, fallos en el servidor de correo o accesos no permitidos, pasando por la detección y registro de accesos, conexiones fallidas, reinicios y apagados, fallos en los log-in, caídas de servidores, modificación de ficheros (Incibe, ss. Ff., pág. 3), análisis forense, detección de ataques/intrusos (Vicente C., 2008, pág 2)... La monitorización de los logs es vital dentro de cualquier sistema; a tal punto llega su importancia que existen multitud de empresas y aplicaciones que se dedican a ello, como, por ejemplo: Splunk, Sumo Logic o GrayLog entre otras.

A la hora de analizar la información que se registra dentro de los logs, es de lo más variada, como se ha comentado anteriormente, por lo tanto, la información que se muestre dependerá de la finalidad para la que se generó el log. De acuerdo con esto, se debe realizar un estudio previo antes de generar los logs, respondiendo a las siguientes cuestiones:

- Qué actividad debe ser registrada, en base a la información que resulta relevante
- En qué formato se mostrará
- Sincronización del sistema: toda la información que se muestre debe estar regida bajo el mismo horario y formato del mismo (Incibe, Ss. Ff., pág 7)

Dentro de la tipología de logs, se podrían clasificar en dos grandes muestras: por un lado, los logs de Sistemas Operativos y elementos de Networking, que incluye Host, Linux/Unix, Windows o elementos de Networking; mientras que, por otro, la categoría sería de servicios o aplicaciones e incluiría aplicaciones de seguridad, servidores de correo, servidores de aplicaciones, monitorización o virtualización (Alonso Alegre Díez M. B., 2016, pág 10).

5.3.2 Generalidades

Este apartado se centrará en una serie de reglas que no son universales, pero que se suelen utilizar dentro de la monitorización y/o análisis de logs. Pese a la problemática que se describirá en el siguiente apartado; existen una serie de rasgos o características que se repiten dentro de los logs y que ayudan en gran medida a su monitorización y análisis.

A la hora de generalizar o protocolarizar un sistema de logs, existen varios ejemplos a seguir, como puede ser el Exchange de Windows, Syslog o el Log4J de Java, a continuación, se mostrarán sus características principales, puesto que cada uno de los sistemas de gestión de logs que se mencionarán a continuación, tienen sus propios protocolos, su propia forma de gestionar y estructurar la información que poseen.

Syslog

Syslog se utiliza para describir un protocolo como un estándar de mensajes de notificación de eventos, o lo que es lo mismo, un protocolo de generación de mensajes dentro de los ficheros de log, es especialmente popular para sistemas de red con UNIX y LINUX (Gerhards R., 2009, pág 2).

El protocolo Syslog se desarrolló alrededor de los años 80 en Estados Unidos y aunque con modificaciones constantes por las nuevas necesidades que surgen en el ámbito informático, su vigencia, uso y validez perdura hasta el día de hoy. Se puede afirmar que, a día de hoy, existen multitud de sistemas de red que aceptan Syslog: servidores, switches, firewalls y otros dispositivos (Keneth E. S., 2003, pág 2-5).

El funcionamiento básico de Syslog se muestra en la siguiente imagen (Ilustración 16):

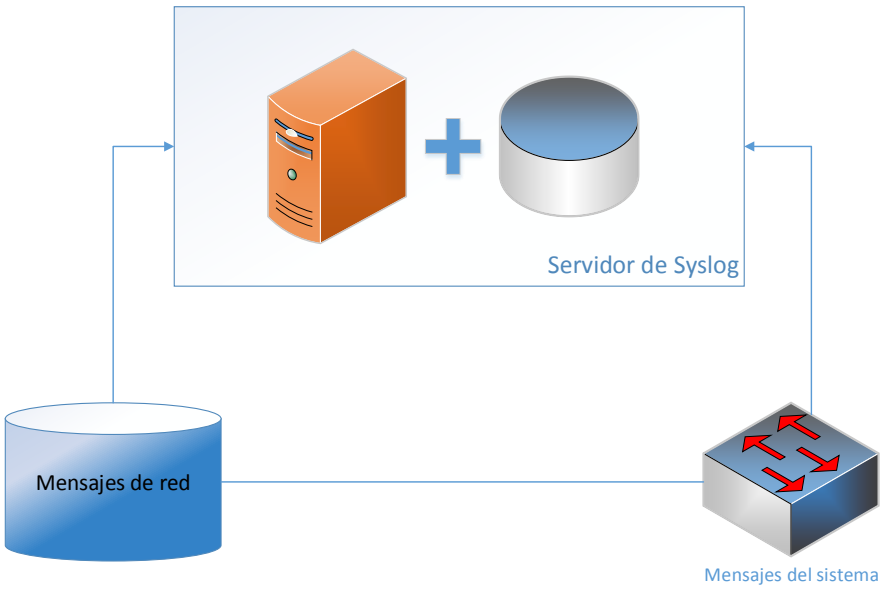


ILUSTRACIÓN 16 FUNCIONAMIENTO SYSLOG (ELABORACIÓN PROPIA)

Los mensajes de red y los mensajes del sistema se envían unificados a un mismo servidor de syslog (Ilustración 16 Funcionamiento SyslogFuncionamiento Syslog), donde debe estar configurada la recepción de los mismos. Dependiendo de cómo se configure pueden enviarse a un servidor interno, externo o a una nube. Una vez llegan y se ubican según sus especificaciones propias, será labor del administrador la interpretación de los mismos.

De acuerdo con el protocolo Syslog, cuanto menor es la nomenclatura del mensaje que se indica, mayor es la gravedad del problema que se describe, de forma que (Longvic C., 2003, pág 8-10):

TABLA 2 DESCRIPCIÓN MENSAJES SYSLOG

Nombre de la gravedad	Nivel de gravedad	Explicación
Emergencia	0	El sistema no se puede usar
Alerta	1	Se necesita una acción inmediata
Crítico	2	Condición crítica
Error	3	Condición de error
Advertencia	4	Condición de advertencia
Notificación	5	Condición normal pero importante
Informativo	6	Mensaje informativo
Depuración	7	Mensaje de depuración

Log4Java

El Log4Java es una librería de código abierto perteneciente a Apache, utilizada de forma casi universal como gestor de logs. Al igual que el Syslog, desde su creación ha sufrido infinidad de modificaciones, debido a la vertiginosa velocidad en la que cambian las necesidades en el mundo de la informática (Gómez Rodríguez V., 2011, pág 94).

Una de sus principales ventajas es que es configurable en tiempo de ejecución, además de realizarse cualquier modificación de forma relativamente sencilla, a través de uno de sus ficheros de configuración (Boyán Ivanov B., 2004).

Los mensajes de error con los que trabaja Log4Java se categorizan de la siguiente manera (Boyán Ivanov B., 2004):

TABLA 3 DESCRIPCIÓN MENSAJES LOG4JAVA

NIVEL	SIGNIFICADO
DEBUG	Muy útil mientras se está desarrollando la aplicación, sirve para ver cómo se comporta el código ante las circunstancias que se le están dando. Una vez depurado, este tipo de mensajes suelen ser eliminados, a no ser que resulten ser necesarios, en cuyo caso se modificaría su nivel.
INFO	Muestra información del programa durante la ejecución: inicio o fin de proceso lanzado, fichero procesado, etc.
WARN	Lanza una alerta sobre una situación anómala que puede suponer un riesgo aunque de momento no afecte al correcto funcionamiento del programa
ERROR	Guarda constancia de errores del programa, aunque ése siga funcionando pese a ellos. Parámetros incorrectos, ficheros corrompidos, etc. Son de gran utilidad en el análisis de las circunstancias que han rodeado un fallo en el sistema, en el procesamiento o cualquier problema que haya surgido y haya provocado una caída o un fallo de considerables proporciones.
FATAL	Mensajes muy críticos, abocados al aborto de la ejecución o abrupto final de la ejecución del programa.

Por otro lado, existen una serie de protocolos de generación de logs, como pueden ser: *W3C Extended Log file Format*, *Microsoft IIS Log File* o el *NCSA Common Log file Format (*Mohd Helmy A. W., Mohd Norzali H. M., Hafizul Fahri H. y Mohamad Farhan M. M., 2008, pág 4):

- W3C Extended Log file Format: se trata de un fichero de log escrito en formato ASCII que se puede personalizar. El formato fue creado por el *World Wide Web Consortium* (W3C), organización que promueve estándares para la evolución de la Web. Los campos se separan con espacios y el tiempo se registra en formato UTC (*Universal Time Coordinated*, referido a la hora del meridiano de Greenwich). El formato del archivo generado, es legible por herramientas de análisis genéricas. Su comienzo viene marcado con el carácter '#', que se utilizan como identificadores de campo, tal como se observa en la Ilustración 17: (Hallam-Baker P. M., Behlendorf B., 1996):

```
#Version: 1.0
#Date: 12-Jan-1996 00:00:00
#Fields: time cs-method cs-uri
00:34:23 GET /foo/bar.html
12:21:16 GET /foo/bar.html
12:45:52 GET /foo/bar.html
12:57:34 GET /foo/bar.html
```

ILUSTRACIÓN 17 EJEMPLO W3C (HALLAM-BAKER P. M., 1996)

A continuación, su formato sería el siguiente:

- Versión del log que se está utilizando (en el caso que se muestra, se corresponde con la *1.0*): *Numero_entero*
- Fecha (*Date*): es la fecha en la que la entrada del log fue añadida. Su formato es *"<date> <time>"*
- Campos (*Fields*): Especifica los campos dentro del log, separados por espacios. Su formato es el *string*.
- Fecha Inicio (Start-Date): Identifica la fecha del software en la que se generó el log. Su formato es el mismo que el de la Fecha: *"<date> <time>".*
- Fecha-Fin (End-Date): Fecha en la que la escritura del log finalizó. Su formato es el mismo que el de las demás fechas: *"<date> <time>".*
- Remarcado: Información comentada que no es objeto de análisis. Su formato es de texto plano.

Aunque como ya se ha mencionado, es el formato utilizado por el eIUM, es importante tener en cuenta que presenta desventajas como que la separación habitual de campos es con espacios en blanco, lo cual no es necesariamente malo, salvo que haya campos que posean separaciones en blanco. La ventaja que presenta este formato, es que, aunque tenga su estandarización, siempre existe la opción de modificarlo.

- Microsoft IIS Log File: en este caso, también se trabaja en formato ASCII, aunque no se puede personalizar. El formato Microsoft IIS incluye: elementos como la dirección IP del usuario, nombre de usuario, fecha de solicitud y hora, código de estado del servicio y número de bytes recibidos; en este caso los campos se separan por comas, tal como se observa en la siguiente imagen (Ilustración 18 Microsoft IIS Log File):

```
192.168.114.201, -, 03/20/98, 7:55:20,
W3SVC2, SALES1, 192.168.114.201, 4502,
163, 3223, 200, 0, GET, /DeptLogo.gif, -,
172.16.255.255, anonymous, 03/20/98,
23:58:11, MSFTPSVC, SALES1,
192.168.114.201, 60, 275, 0, 0, 0, PASS,
/intro.htm, -,
```

ILUSTRACIÓN 18 MICROSOFT IIS LOG FILE (CITESEERX WEBSITE)

- NCSA Common Log file Format: es otro fichero no configurable, nuevamente en formato ASCII. Sirve para informar sitios web pero no *FTP*s. La

información que maneja es la siguiente: información básica sobre las solicitudes de los usuarios, tales como nombre de host, nombre de usuario, fecha, hora, tipo de solicitud, estado HTTP, código, y el número de bytes enviados por el servidor. Tal y como se observa en: (**¡Error! No se encuentra el origen de la referencia.**)

```
172.21.13.45        -        REDMOND\fred
[08/Apr/1997:17:39:04     -0800]     "GET
/scripts/iisadmin/ism.dll?http/serv
HTTP/1.0" 200 3401
```

ILUSTRACIÓN 19 EJEMPLO NSCA COMMON LOG FILE FORMAT (CITESEERX WEBSITE)

5.3.3 Problemática en el análisis de logs

Aunque la monitorización de logs es una actividad muy común dentro de la monitorización de sistemas, existen ciertos problemas derivados de la falta de estandarización. Se han mencionado algunos protocolos, también existen reglas de estilo al respecto, sin embargo, no hay nada que indique de forma exacta, cómo formar los ficheros de logs. Esto genera una serie de problemáticas que se describirán a continuación, aunque existan más, se plasmarán algunas de las más comunes o relevantes para el objeto de estudio:

- la información que se proporciona es demasiada o escasa: a la hora de generar un log hay que saber qué información es necesaria y cuál resulta excesiva; puesto que exceso de información también es desinformación.
- la información que se proporciona es de poca utilidad: es importante que se acoten las palabras específicas que aportan información real al mensaje que se transmite.
- la información que se proporciona está mezclada, resultando imposible su comprensión: en ocasiones las trazas que muestran los logs resultan inconsistentes y de difícil análisis, debido a que no poseen una estructura rigurosamente definida. Sería interesante la generación de un mayor número de logs en los que la información tuviera cierto orden y/o propósitos similares.
- no existe una ubicación específica para los logs, de forma que cada aplicación o sistema, los guarda en la ubicación que se consideró ideal cuando se programó. A la hora de poner en manos de una empresa o una aplicación la monitorización de los logs, se debe afinar mucho en las búsquedas para dar con todos los ficheros de log relevantes para el sistema que se pretende monitorizar.
- no existe una extensión universal para los logs, de forma que pueden poseer extensión de tipo .lg, .tx, .log, .xml, etc. Al igual en el punto anterior, existe cierta problemática con la falta de protocolarización de los logs, puesto que hay mucha variedad en tamaños, extensiones y ubicaciones a la hora de generar los mismos.
- el tamaño de los logs no es un factor determinante en las búsquedas de los mismos, puesto que no hay un tamaño específico. Algunas aplicaciones de servidores o de dispositivos generan un fichero comprimido cada cierto tiempo

en el que guardan la información de los logs; de esta forma se evita tener un fichero demasiado grande como para trabajar con él. A priori esta acción resulta muy útil, sin embargo, dota de más complejidad la labor de monitorización, puesto que derivará en una búsqueda mayor para dar con este tipo de ficheros.

- la información que se genera es variopinta, en el sentido de que hay que contrastar la proveniente de varios sistemas para investigar dónde se produjo un error común; el resultado es un proceso complejo. Este punto se refiere a que cada sistema genera sus propios logs, dar con errores, por ejemplo, en envíos de paquetes entre servidores, es tedioso, por la falta de uniformidad en los mensajes que trasmiten, que en muchas ocasiones no basta con el acceso a uno de los servidores del envío, sino que se requiere el acceso y posterior análisis a los logs de los servidores implicados en la pérdida de información.
- el volumen de logs que hay que gestionar cada vez es mayor: puesto que cada vez hay más necesidad de digitalización, el volumen de datos que se guarda es mayor y en cuanto a los logs, no podría ser de otra forma. Debido a la utilidad de los mismos y a la instalación de distintas funcionalidades, el volumen de logs con el que se trabaja cada vez es mayor y más variado, de forma que se complica en gran medida la monitorización de la información que proporcionan (NIST, 2006, pág. 2-8)
- Inconsistencia de *TimeStamps*: la falta de uniformidad en los formatos de fecha produce esta problemática.
- Protección de información sensible: los logs guardan información sensible en muchas ocasiones: accesos al sistema desde determinadas IPs, consultas que realizan los usuarios, etc. Este tipo de información no se encuentra a la vista de cualquier usuario, se intenta esconder; lo cual nuevamente no ayuda a la hora de monitorizar la información que se genera en los logs.

Aunque la estandarización en la generación de logs no exista y genere la problemática que se ha mencionado en las líneas anteriores, con el fin de lograr un manejo de logs eficiente y efectivo, el *National Institute of Standards and Tecnology*[5] (NIST) propone una serie de pautas a seguir para cada uno de los tipos de logs que se pueden encontrar dentro de una compañía (NIST, 2006, pág 10-15):

1. Establecer políticas y procedimientos para la administración de logs
2. Priorizar la administración de logs dentro de la empresa
3. Crear y mantener una infraestructura para la administración de logs
4. Establecer procesos operativos estandarizados

5.3.4 Arquitectura en un sistema de logs

La infraestructura para la administración y gestión de logs incluye todo lo relacionado con el hardware, software, redes, y medios usados para generar, transmitir, guardar, analizar o disponer de los datos de log (NIST, 2006, pág 27). La mayoría de las grandes compañías poseen varias infraestructuras o departamentos que se dedican a la

[5] Perteneciente al Departamento de Comercio del Gobierno de los Estados Unidos, fue fundado en 1901. Es un laboratorio de ciencias físicas que se dedica a labores de medición y estandarización.

administración de logs. En esta sección del documento, se describirán los puntos más importantes de la arquitectura que se sigue habitualmente y la forma en la que los distintos niveles se relacionan entre sí.

La arquitectura típica de un sistema de gestión de logs posee los siguientes niveles (Cal González A., 2015, pág 12-17):

1. *Generación de logs:* este nivel no sólo incluye la generación de logs propiamente dicha, sino un cierto orden en la localización de los mismos, a fin de facilitar la labor del resto de niveles.
2. *Análisis y almacenamiento:* las aplicaciones o servidores encargados de recibir y analizar automáticamente todos los logs susceptibles de ello dentro de un mismo sistema.
3. *Monitorización:* esta última capa se encarga del acceso a los análisis realizados en la anterior. Se comprueban los resultados obtenidos y se realizan informes sobre los mismos.

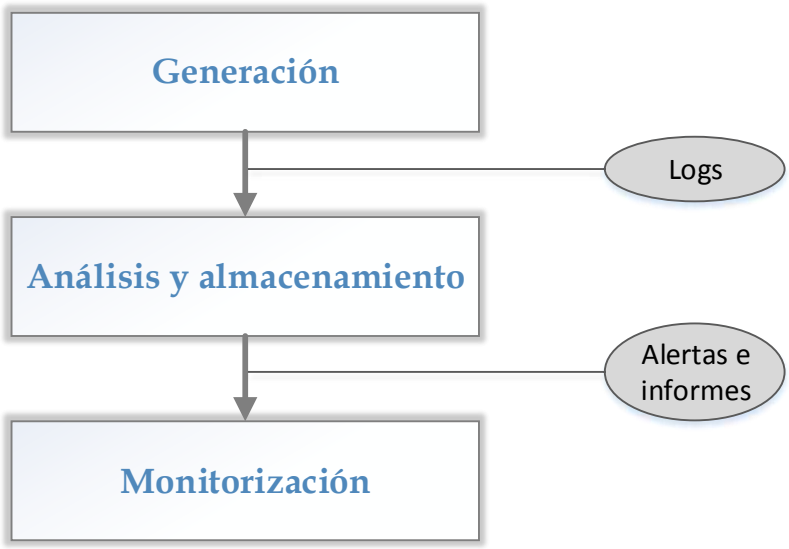


ILUSTRACIÓN 20 ARQUITECTURA DE LOS LOGS (ELABORACIÓN PROPIA)

En las siguientes líneas serán analizados un poco más en detalle cada uno de los puntos de esta arquitectura.

Generación de logs

En este nivel se incluyen todas las acciones relacionadas con:

- Parseo de logs: *o log parsing*, extrae datos de determinados logs para copiarlos dentro de otros.
- Filtro de eventos: *o event filtering*, suprime determinadas líneas que resultan superfluas para el análisis, puesto que se trata de información que no tiene interés, como por ejemplo entradas duplicadas que se producen en diferentes ficheros.
- Agregación de eventos: relaciona eventos que están vinculados de alguna manera. En ocasiones se producen muchas entradas de corte similar, lo que se

hace en este apartado es unirlas o correlacionarlas para que el análisis no resulte tan pesado.

Análisis y almacenamiento

- Rotación de logs: se llama así al proceso de cierre de un fichero log, y la apertura de otro. Este hecho se produce por ejemplo cuando el fichero de log ha excedido del tiempo o el volumen para el que el sistema ha sido programado.
- Archivo de logs: los ficheros de logs se guardan por requisitos legales o sencillamente para que la compañía disponga de determinada información que podría ser necesaria. Existen dos formas de archivado: retención y preservación. La retención archiva los registros de forma regular y la preservación conserva los registros susceptibles de contener información de utilidad durante un tiempo indeterminado.
- Compresión de logs: consiste en comprimir los ficheros de logs para que ocupen menos espacio, sin renunciar a la cantidad de información que poseen.
- Reducción de logs: se trata del borrado de logs. Cuando ha pasado un tiempo considerable, se eliminan; o también se eliminan en un espacio de tiempo inferior, aquellos cuya información no se considera relevante.
- Normalización de logs: se trata de estandarizar la información que se posee para que sea más sencillo su análisis. Un ejemplo de normalización sería en cuanto a los formatos de fecha que se producen en los logs; hhmmss, hh:mm:ss, hh mm ss, etc. La normalización consiste en este caso en una modificación, para que todos los logs monitorizados, posean el mismo formato de fecha y hora.

Monitorización

- Correlación de eventos: consiste en encontrar la relación entre dos o varias entradas de log. Se pueden utilizar métodos estadísticos o herramientas de visualización para conseguir la correlación de eventos.
- Vista de logs: consiste en abrir los ficheros de log con un editor o visor de texto para su siguiente lectura.
- Reporte de logs: de manera periódica es recomendable el resumiendo o agrupando determinada información relacionada con un evento en particular o una serie de eventos.

5.4 El eIUM y la mediación de telefonía

En este apartado se abordará la herramienta principal con la que trabajan muchas empresas de telefonía: el eIUM, una solución de la compañía HP.

Puesto que el objeto de estudio de este trabajo es la implantación de la tecnología Machine Learning dentro de un sistema de mediación de telefonía; es interesante saber cómo las operadoras pueden gestionar la información proporcionada por las centrales; de forma que conocer el funcionamiento a grandes rasgos de la herramienta eIUM y con él, cómo se estructura su sistema de logs, resulta imprescindible.

5.4.1 La solución eIUM

Existen multitud de tutoriales y guías en las que se explica con más o menos detalle el funcionamiento del eIUM, en este punto se ha seleccionado la información que resulta más relevante para el objeto de estudio de este documento, puesto que la cantidad de información que hay en la red es inmensa y tal y como se verá, es una herramienta muy completa, por lo tanto, resulta difícil abarcar o mencionar todas las funcionalidades y opciones que posee. Es por estas razones, por las que se mencionarán las partes y funcionalidades del eIUM que puedan resultar interesantes para el objeto de estudio de este documento.

5.4.1.1 Descripción

Tal como se expresa la guía oficial del HP, el eIUM es "el software HP enhanced Interative Unified (eIUM) que permite a los proveedores de servicios analizar el uso de su infraestructura y facturar a sus clientes en consecuencia" (Hewlett-Packard D.C., 2004, pág 8, p.2). El eIUM es, por lo tanto, una de las principales plataformas a la hora de gestionar la mediación, ya que se utiliza con:

- Llamadas de voz
- *IMS*: se corresponde con las siglas *IP Multimedia Subsystem*, soporta la telefonía a través de redes IP
- *DS*L: Las siglas DSL significan *Digital Subscriber Name*, que en español son las Líneas de Suscriptor General, proporciona acceso a internet, principalmente en redes ADSL (Asymmetric Digital Subscriber Line, o más conocida como línea digital de banda ancha)
- *VoIP*: Voz sobre IP, una tecnología que permite que la señal telefónica viaje a través de la red internet empleando un protocolo IP
- Cable
- 3G y 4G móvil
- Además de otros tipos de redes y tecnologías

El eIUM trabaja también en tiempo real y gestiona las redes de facturación, calidad del servicio, capacidad, etc.; en definitiva, es una solución muy completa que abarca todos los procesos que se suceden dentro de la gestión de la mediación telefónica.

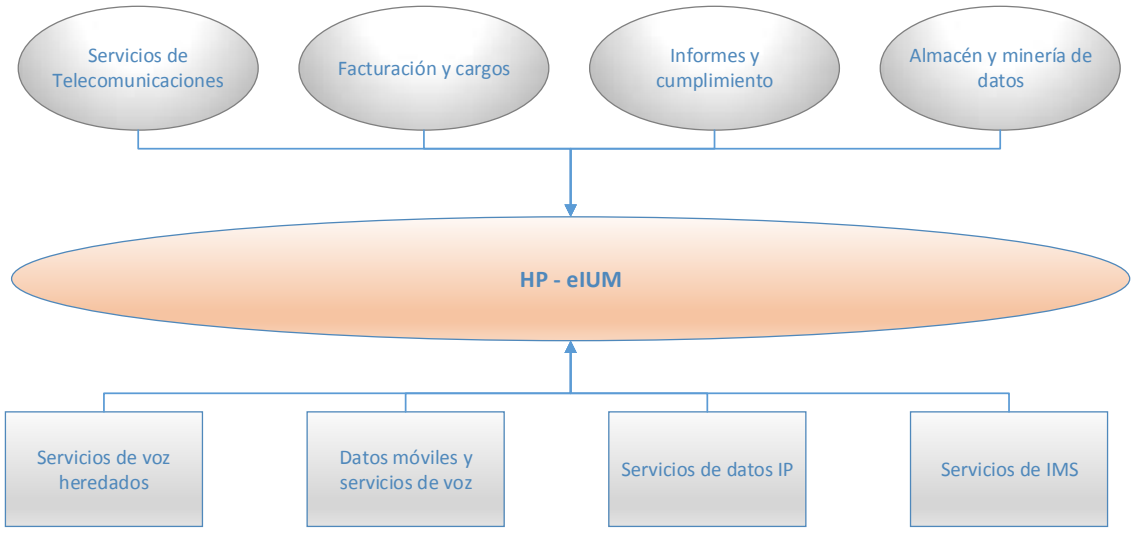


ILUSTRACIÓN 21 FUNCIONES Y SERVICIOS EIUM (ELABORACIÓN PROPIA)

La arquitectura dl eIUM es escalable y distribuida para facilitar la recopilación, agregación y correlación de datos; además de poseer un sistema de soporte empresarial que facilita la facturación o análisis de los mismos. Otra de sus ventajas, es que se puede configurar para gestionar repositorios, usuarios, servidores, entre otros, y todo ello en tiempo real. Así mismo es una herramienta capaz de procesar información de más de 50 fuentes diferentes, característica que la hace muy versátil.

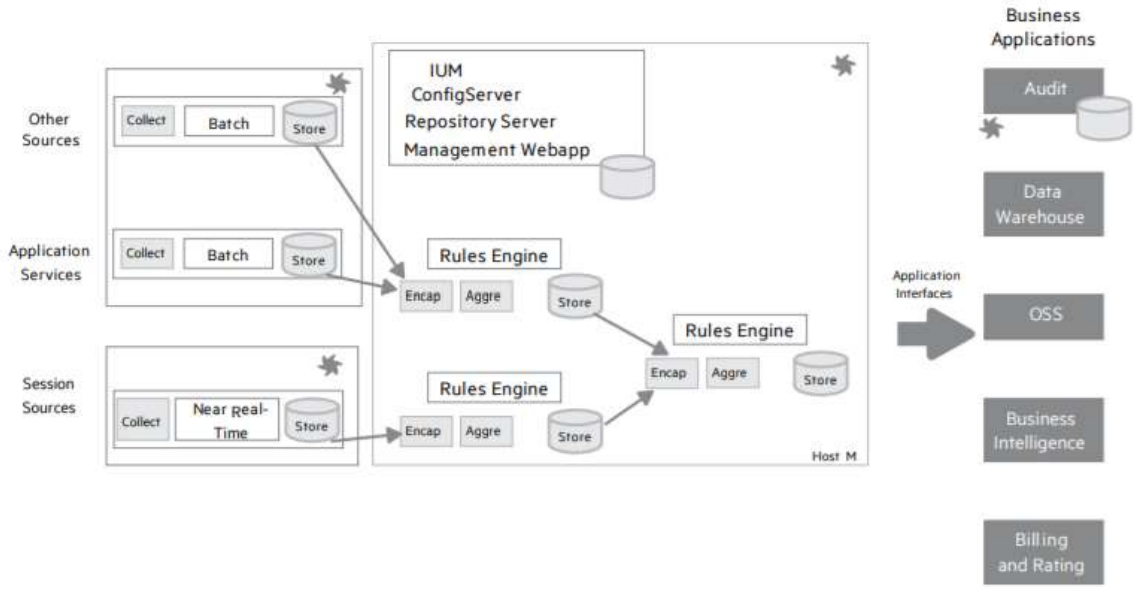


ILUSTRACIÓN 22 ARQUITECTURA HP EIUM (HP EIUM OVERVIEW GUIDE)

Gracias a todo lo mencionado el eIUM se convierte en una plataforma flexible, modular, basada en estándares, que es utilizada hoy en día por múltiples compañías líderes en el sector de la tecnología, tales como: Telco 2.0, Cloud, Smart Meeting, WiMax, WIFI y HP Networking.

Aunque este estudio se centrará en el procesamiento de la información y la información que se genera a partir de éste, es interesante saber que con la herramienta eIUM además se podría:

- Poseer una plataforma de que admite servicios prepagos, facturación en caliente y cobro en tiempo real
- Proporcionar a un suscriptor información sobre los cargos aplicables antes de que se utilice un servicio, para que de esta forma el suscriptor deba aceptar los cargos para utilizar el servicio en cuestión
- Proporcionar herramientas a los clientes para que tengan acceso a un servicio de control de consumo
- Mediación y gestión de los servicios domésticos como alarmas, medidores, etc.
- Proporcionar información analítica "Big Data"

5.4.1.2 Componentes y funcionamiento

La arquitectura del eIUM se divide en varios niveles, preparados para aprovechar al máximo los recursos hardware y software y minimizar la *latencia* (tiempo de respuesta).

A continuación, se explicarán algunos de sus componentes principales.

- **Los NME:**

Sus siglas se corresponden con *Normalized Metered Events*, o lo que es lo mismo los Eventos Medios Normalizados, son la estructura de datos interna que utilizan los componentes de eIUM. De esta forma todos los datos que se leen y procesan a través del eIUM, se trasforman en NMEs. Un NME es un evento de formato variable (ASCII o binario) que contiene registrada información a través de los campos que lo componen. Cada campo se corresponde con un atributo, que posee un nombre y un tipo; éstos dependerán del tipo de datos que se estén procesando. Un ejemplo de NME sería el siguiente, en el que se separa cada campo por comas (extraído de un fichero ASCII de la compañía Euskaltel y modificados algunos campos con "#" por razones de confidencialidad):

```
20180923;003522;5;1;0225###-
180614######@#####_GI;346########;35-38-37-64-38-35-62-31-
2D-32-34-36-30-2D-31-37-65-32-2D-38-65-62-64-2D-30-31-30-
36-30-##-##-##-##-##-##-##;0;S;
Telecable;010117####;0;346########;4;MT;AllDays;3;300;0.000
000;0;300;0.000000;Cg:346########,Cd:346########,NrnB:
900555,LocB:34609###### ,Route:34609######;
```

Que se corresponde con:

TABLA 4 INFORMACIÓN EN UN NME

Campo (*Ejemplo del NME*)	Formato	Descripción
Date (*20180923*)	Formato yyyymmdd	Fecha de generación del registro en formato yyyymmdd hh:mm:ss.xxx
Time *(003522)*	Formato hhmmss	Hora de generación del registro en formato hhmmss
File Version *(5)*	1 dígitos	Versión del fichero. Toma el valor 5 para la sesión actual.

Campo (*Ejemplo del NME*)	Formato	Descripción
Service Type *(1)*	Numérico (1 dígitos)	Tipo de servicio: 1:Voz 2:Video
Global ORE Id *(0225###-180614######@#####_GI)*	Alfanumérico	Identificador de la sesión de cobro. Está formado por los siguientes elementos: • Identificador de la tarea (8 dígitos en base 30) • Fecha • Nombre del nodo que está cursando el mensaje.
Service Node Address Reference *(346########)*	Alfanumérico	Identificación del nodo de servicio. En el caso de voz se identifica mediante el Global Title (GT).
Service Node Session Id (*35-38-37-64-38-35-62-31-2D-32-34-36-30-2D-31-37-65-32-2D-38-65-62-64-2D-30-31-30-36-30-##-##-##-##-##-##-##*)	Alfanumérico	Identificador de la llamada asociada
Session Sequence Number (*0*)	Numérico, (max. 4 dígitos)	Número de secuencia del cdr en la sesión.
Session Start/Continue/End mark (*S*)	Un carácter (S/I/E)	Indicador de fin de sesión. Valor *E* para el último cdr de una sesión
Brand BSS Reference (*Telecable*)	Alfanumérico	
Charged Subscriber Account (*010117####*)	Alfanumérico	
Charged Subscriber Reported Identity Type (*0*)	Numérico	Tipo de elemento: 0: MSISDN 1: IMSI
Rated Subscriber Reported Identity (*346########*)	Alfanumérico	Información de la entidad (IMSI, MSISDN u otro) tal y como se informa por el nodo de servicio
Service Originator Pattern (*4*)	Alfanumérico	

Campo (*Ejemplo del NME*)	Formato	Descripción
Direction (*MT*)	Alfanumérico	Dirección del CDR MO: Entrante MT: Saliente
Price BssReference (*AllDays*)	Alfanumérico	
Traffic Units Type (*3*)	Entero (máx. 12 dígitos)	Unidades de tráfico: 1: evento 2: datos 3: tiempo
Traffic Units (*300*)	Decimal.	Valores negativos indican cantidades devueltas al subscriber para el evento especificado.
Amount Charged (*0.000000*)	Decimal (0 a 100)	Valor > 0 => Crédito Valor < 0 => Cargo
Accumulated Traffic Units per Session (*0*)	Decimal	Tráfico acumulado en la sesión, incluyendo todas las subsesiones
Accumulated Rated Traffic Units per SubSession (*300*)	Decimal	
Additional Service Info		Información adicional de la llamada separados por ,
Cg (*346########*)		Número llamante
Cd (*346########*)		Número llamado
LocB (*34609######*)		Localización del VLR o el MSC del número llamado
NrnB (*900555)*		NRN del número B
Route (*34609######*)		Prefijo de ruta de salida para la llamada.

La información de los NMEs se extrae y procesa a través de los colectores. Es posible visualizar la información de los ficheros que contienen los NMEs, bien a través de visores de texto plano, para el caso de los ficheros ASCII; o bien a través de herramientas propias de visualización de ficheros binarios. Por lo general no resulta necesaria la visualización de ficheros, puesto que a la hora de realizar pruebas se facilita la información necesaria en la documentación de las mismas; sin embargo, en ocasiones es interesante tener acceso a la información de los eventos, de cara a la resolución de posibles incidencias.

- **Los Colectores:**

Son los procesos fundamentales dentro del eIUM. Su misión es procesar la información que contienen los ficheros, divididos en NMEs. Cabe destacar que los NME son los únicos registros que utiliza el eIUM.

Una vez los colectores recogen los ficheros, bien sea conectándose a otras máquinas o sistemas o bien éstas, los ubican en la que se encuentra el colector; éstos procesan los datos a través de su lógica y envían los datos procesados a los sistemas a los que pertenezcan: facturación, fraude, etc.

Los datos que soportan y que por lo tanto se procesan dentro de estos colectores del eIUM, pueden ser de los siguientes tipos:

- Conmutadores de voz, por ejemplo, de Alcatel, Ericsson, Lucent, Nokia, Nortel y Siemens
- Dispositivos de *VoIP*
- Dispositivos de punto de acceso *WLAN* (*Wireless Local Area Network* o red de área local inalámbrica. Es una red de tipo local que no precisa de cables para su utilización) y puntos de control de servicio
- Servidores AAA que utilizan protocolos, por ejemplo, Diámetro y *RADIUS* (*Remote Access Dial In User Service,* es un protocolo seguro y flexible, se utiliza por ejemplo en redes de establecimientos como restaurantes u hoteles)
- Enrutadores *IP* (Internet Protocol, o protocolo de internet. Es un número único que identifica un dispositivo en la red), por ejemplo, Cisco Netflow
- Archivos de registro del *servidor web* (Dispositivo de almacenamiento de páginas web, encargado de almacenar contenidos y difundirlos a través de la red)
- Archivos de registro del *servidor proxy*
- *MIB* (*Management Information Base*, es un tipo de base de datos que contiene información estructurada de forma jerárquica proveniente de todos los dispositivos que conforman una red) o Base de Información para la Gestión de *SNMP* y *RMON*
- Tráfico de datos y voz móvil desde los conmutadores GSM / *GPRS (*General Packet Radio Service o Servicio General de Paquetes de Radio. Es un sistema GSM de transmisión de voz) o *CDMA* (*Code Division Multiple Access,* o multiplexación por división de código. El término multiplexación se refiere a la unión de varias señales en una sola, a través de un único medio de transmisión, permitiendo transmisiones simultáneas. El CDMA además utiliza el espectro expandido, de forma que permitiría el envío simultáneo de señales a través de una banda ancha de frecuencias)

La forma en la que trabajan los colectores es la siguiente: un fichero comienza a ser procesado por el colector; en primer lugar, los datos del fichero entran en el *encapsulador*, que es el encargado de su lectura y validación, una vez validado, pasa al *agregador*, que posee una serie de reglas con las que se modifica y trasforma la información obtenida. A partir de esas reglas, el *DataStore* se encarga del envío de esa información en forma de NMEs procesados a los distintos sistemas de destino.

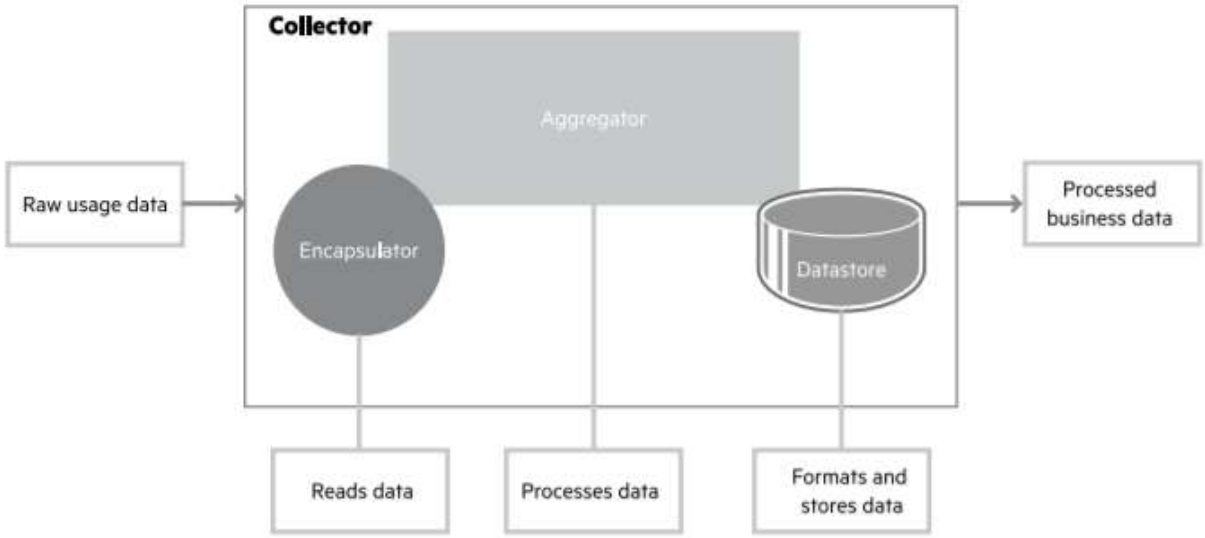


ILUSTRACIÓN 23 FUNCIONES DEL COLECTOR (HP EIUM OVERVIEW GUIDE P. 2., THIRD EDITION)

Existen varios tipos de colectores dependiendo las funciones a las que estén destinados, del mismo modo que existen diferentes niveles de colección dentro de la lógica de mediación.

Un ejemplo del funcionamiento de los colectores a distintos niveles sería el siguiente:

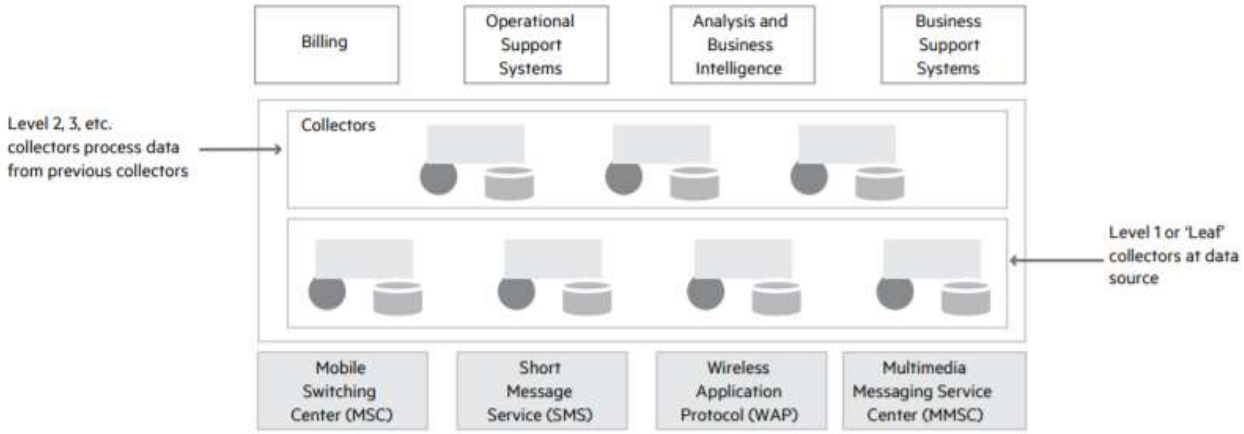


ILUSTRACIÓN 24 NIVEL 2 Y POSTERIORES DE COLECCIÓN (HP EIUM OVERVIEW GUIDE P. 2., THIRD EDITION, PÁG. 28)

Una parte importante del eIUM es que es capaz de detectar los posibles duplicados que se producen dentro de las centrales, para descartarlos y que no aparezcan en último lugar, que son las facturas de los clientes; también se ocupa de los registros erróneos o incoherentes, y de su reprocesamiento.

De esta forma se quedarían completamente cubiertas las necesidades de procesamiento de cualquier operadora de telefonía.

- **El almacenamiento:**

Una vez procesada la información, los ficheros que se generan intermedios, auditorías y demás, se guardan temporalmente en las máquinas de eIUM. La ubicación de las máquinas es segura y el espacio disponible, variable, de acuerdo a las necesidades existentes.

- **Otros componentes:**

El hecho de que este punto no se extienda más y sólo se mencionen de pasada el resto de componentes, no significa que sean menos importantes; sino que no son relevantes para el objeto de estudio. Estos componentes son:

- *Sesion Server*: es el componente principal para el procesamiento de datos en tiempo real, gracias a su velocidad, se utiliza para la gestión de datos de prepago.
- *Configuration Server*: mantiene y almacena la configuración de cada colector
- *Administration Agent*: es el primer proceso que se inicializa, se encarga de inicializar los demás.
- *Database Engine*: permite la configuración, modificación y consulta de una o más bases de datos que trabajan con MySQL u Oracle.
- *Launch Pad*: se encarga de las labores de implementación además de labores de monitoreo.
- *Operations Console*: la consola de operaciones es una aplicación que monitorea y administra el eIUM a diario, mostrando todos los procesos que se ejecutan en el mismo.
- *Correlator*: correlaciona datos para informar sobre el uso de red
- *Report Collector*: genera informes sobre los datos generados
- *Web aplication Server*: proceso de Java que admite las operaciones de eIUM consola, gestor de datos de referencia, informes eIUM e informes de auditoría.
- *Schedule Server*: permite la realización y ejecución de operaciones programadas
- *Management Server*: servidor central en tiempo real. Se utiliza para diversas labores, la más importante es como contenedor para la gestión de servidores y el servicio de sondeo.
- *Repository Server*: se utiliza como un repositorio de contenido donde se pueden cargar, registrar y extraer archivos para que otros usuarios de eIUM puedan utilizarlos.
- *Studio Server*: es un componente del motor en tiempo real capaz de soportar un gran número de procesos.

5.4.2 Los logs en eIUM

Como ya se ha explicado con anterioridad, la lectura de logs es importante en cualquier sistema, para la solución eIUM no podría ser de otra forma.

La única forma que posee la herramienta eIUM de comprobar que el procesamiento y envío ha sido correcto es a través de los logs de sus máquinas y sus sistemas de auditorías, en ellos se muestra información del fichero que se está procesando con la fecha y la hora del mismo, qué ficheros genera para envíos a otros sistemas también. con fecha y hora y registra posibles errores dentro de esos envíos. Los errores que puede registrar son de tipología varia: pérdida de la conectividad, máquinas llenas e incapaces de aceptar más ficheros o envíos duplicados entre otras.

A fin de facilitar la lectura e interpretación de los logs, la herramienta eIUM posee un mismo formato para todos los logs que genera, el formato seleccionado es el

W3C, de esta forma se posee una estructura definida y un conjunto de campos sabidos. En anteriores versiones de eIUM, no existía esta política de formato común, de forma que las auditorías se complicaban debido a que era muy difícil correlacionar eventos de distintos registros y conectarlos a una sola acción del usuario.

Tal y como es de esperar, todo el sistema eIUM posee diversos sistemas de alarmado. Esto significa que existen dispositivos que envían avisos en caso de que se produzcan situaciones indeseadas y, por lo tanto, requiere la atención y supervisión de estas alarmas en horario 24x7; es decir que, en todo momento, a diferentes niveles y en distintos ámbitos, el sistema está bajo la atenta mirada de sus responsables. De cara a preservar la estabilidad del sistema, además del chequeo permanente de logs a través de diversos sistemas de alarmado, éstos (los ficheros de log) se almacenan durante un determinado periodo de tiempo. La razón de conservar los ficheros de log durante un tiempo, es para poder analizar posibles problemas que se presenten, puesto que las trazas de los logs resultan de gran utilidad en cualquier análisis de situación.

Lo cierto es que la mayoría de las trazas de los logs muestran información de poca utilidad, en el sentido en el que la lectura de los mismos es línea a línea y de forma manual, es decir a través de unos ojos humanos, no entraña demasiada utilidad; a no ser casos muy concretos, como la detección de ciertas problemáticas. Sin embargo, estas problemáticas se podrían detectar a través de la tecnología ML entre otras, con el aprendizaje automático.

5.5 Aplicabilidad de ML en la lectura e interpretación de logs

La aplicabilidad de ML en la lectura de logs no es algo nuevo, puesto que, a grandes rasgos, se trataría de automatizar procesos a partir de los eventos que se producen dentro del sistema y se registran dentro de los logs del mismo.

Con la aplicación de ML en la lectura de logs se podrían, entre otras opciones, clasificar los eventos de forma que se permita ‘desatender’ determinadas operaciones, especialmente las repetitivas, además de estructurar la información de los eventos del sistema, de forma que resulte más comprensible al ojo humano (Durant Kathleen T. y Smith Michael D., 2006, pág. 2-3) y sea más sencilla la detección de situaciones anómalas. La ventaja que supone ML sobre otras tecnologías o sobre otros métodos de *monitoreo* de logs, radica en la anticipación: al analizar los datos a través de ML se puede predecir el comportamiento a medio/largo plazo de los logs, en base a su comportamiento actual.

La falta de estructuración y uniformidad dentro de los logs, como se ha comentado previamente, hace que se almacene información de forma *libre*, que la hace de difícil comprensión. Por suerte, esto no ocurre con la información que se almacena en los logs de eIUM, lo cual supone una enorme ventaja a la hora de aplicar ML o cualquier otra tecnología.

La finalidad de la aplicación de ML dentro del sistema de BSS en mediación de telefonía, será encontrar un modelo o un procedimiento que, haciendo uso de los datos históricos del sistema, sus entradas y salidas, consiga predecir el comportamiento de los procesos del sistema en el futuro.

Para poder aplicar ML dentro de la lectura e interpretación de logs, se deberá definir un espacio n-dimensional, donde n es el número de atributos. Cada atributo

poseerá una posición única dentro del espacio, que dependerá de los valores de sus atributos. Para poner cada evento del log en contexto, se deberá calcular en qué medida los elementos se relacionan entre sí (Girardin L. y Brodbeck D., 1998, p. 299-300).

5.5.1 Estado de la cuestión

Como se comentó anteriormente, el análisis de logs utilizando la tecnología de ML, no es una novedad. En la siguiente lista, se mostrarán estudios que avalan la efectividad de la aplicación de ML en la lectura de logs aplicado a diferentes campos, aunque existen infinidad de estudios de este tipo, se han seleccionado aquellos que se han encontrado que podrían resultar de mayor interés dentro del campo que se está investigando:

1. Año 2011: Tomoya Mizumoto, Mamoru Komachi, Masaaki Nagata, Yuji Matsumoto (Mizumoto T., Komachi M., Nagata M., Matsumoto Y., 2011) escriben un artículo sobre ML aplicada a la lectura de logs, en este caso, a partir de una amplia muestra de estudiantes japoneses, se intenta extraer el *corpus* en el estudio de idiomas, concretamente japonés-inglés, basándose en que el acceso a esta información puede ser una fuente de conocimiento interesante tanto para otros alumnos como para profesores. Pese a la rigurosidad del estudio, se concluye que el idioma japonés posee demasiados caracteres (el modelo generado de caracteres es superior al de las palabras que se pueden formar con ellos) como para poder hacer una predicción sobre la siguiente palabra que se introducirá.
2. Año 2012: Umit Ozertem, Olivier Chapelle, Emre Velipasaog y Pinar Donmez (Ozertem Umit, Chapelle O., Velipasaog E. y Donmez P., 2012), procedentes de distintas empresas y universidades de los Estados Unidos, publicaron un estudio sobre ML aplicado a la sugerencia de respuestas en las búsquedas que se realizan en los distintos buscadores. Para la realización del estudio, se apoyaron en los logs de búsquedas que se generaban en cada una de ellas. El propósito del estudio demostrar que, a través de la aplicación de ML, las probabilidades de que el usuario encontrara una información útil y relevante, eran más altas que con las tecnologías que se utilizaban en el momento de realización del estudio.
3. Año 2014: en la conferencia del conocimiento de descubrimiento y data mining, Ruben Siops, Dmitry Fradklin Fabian Moerchen y Zhuang Wang escribieron un interesante artículo sobre *el Mantenimiento Predictivo*, en el que se aplicaba ML a la lectura de logs de equipos médicos. Las conclusiones fueron varias, entre ellas una mejora en la calidad de los datos obtenidos, aunque para muestras grandes la aplicación de ML no fue más rápida que los métodos tradicionales; sin embargo, en muestra pequeñas los beneficios superaron sus expectativas (Fradkin D., Mörchen F., Wang Z., 2015).
4. Año 2015: en la universidad de Hong Kong, un grupo de estudiantes crea una pequeña aplicación que es capaz de analizar los logs a partir de Machine Learning y técnicas de manejo de ruido; y además proponer sugerencias a los desarrolladores para las líneas de log futuras. La aplicación en cuestión se

llama *LogAdvisor* y es actualmente utilizada por Microsoft y dos programas más de código abierto. Una de las conclusiones a las que se llegó parte de la complejidad que requiere el proceso de la documentación de los registros, es decir, la complejidad de hacer comprender a la máquina el porqué de las líneas de log; partiendo de ese punto, el resto del estudio culmina en una propuesta: registrar dentro de la propia aplicación un repositorio en el que la máquina comprenda o vincule mejor las modificaciones dentro del código con los logs de salida. Pese a que la propuesta no llegó a hacerse efectiva, el resultado final, no obstante, fue muy satisfactorio, puesto que según se afirma en las conclusiones queda demostrada la viabilidad y eficacia del LogAdvisor (Jieming Z., Pinjia H., Qiang F., Hongyu Z., Michael R. y Dongmei Z., 2015)

5. Año 2016: Sung-Bae Cho realiza un estudio en el que aplica ML a la geolocalización de smatphones a través de lectura de logs. Se aplica un modelo híbrido de *kNN* y árbol de decisión para el reconocimiento de predicción y localización. En sus conclusiones se demuestra que se consigue una predicción del movimiento superior al 90% en los experimientos realizados (Cho Sung-Bae, 2016)
6. Año 2016: en la Segunda Conferencia de Seguridad en Big Data en la Nube, celebrada en Nueva York, se presentó un análisis que demostraba la mejora en la seguridad en la red, mejorando la detección de vulnerabilidades. La metodología de trabajo que se utilizó, trabajaba cuatro frentes, por un lado, una plataforma de análisis de Big Data, por otro un conjunto de detección de datos atípicos, por otro un mecanismo para obtener retroalimentación de los sistemas de seguridad preinstalados y, por último, un módulo de aprendizaje supervisado a través de ML. El estudio analiza un número total de 3,6 millones de líneas de log, los resultados demuestran una mejor de más del 3% en la detección de ataques al sistema y una reducción del 5% en la detección de falsos ataques (Veeramachaneni Kalyan, Arnaldo Ignacio, Vamsi Korrapati, 2016).

Con los estudios mencionados, se puede afirmar que, aunque hay trabajos previos que relacionan la lectura de logs con ML, sin embargo, no se aplica ninguno de ellos a los logs de telefonía. Aunque tienen ciertas similitudes con este estudio, ninguno de los mencionados se acerca al ámbito de aplicación de éste y por esta razón resulta tan interesante.

La sociedad ha cambiado vertiginosamente en los últimos años, la razón principal ha venido de la mano de una revolución informática a todos los niveles posibles. Ya no es sólo la forma en la que el ser humano se comunica y relaciona, es también la forma en la que la informática y los datos están presentes en el día a día de cualquier persona. Paralelamente a esta revolución informática y social, se encuentra la aplicación de ML, cuyo crecimiento ha sido exponencial en los últimos años. La presencia de esta tecnología en tantos ámbitos la hace polivalente, sin embargo, hay muchos campos que aún están sin explotar; uno de ellos es la lectura de los logs generados dentro de la Mediación de telefonía. Tal como se ha descrito en los apartados previos, se trata de un campo muy específico dentro de la informática.

El interés de este estudio reside en primer lugar, en su originalidad, dado que no se han encontrado estudios previos sobre la aplicación de ML a la lectura de logs de Mediación de Telefonía y, en segundo lugar, en que para cualquier Operadora Telefónica mostrar una nueva forma de analizar o explotar los datos de sus logs, siempre es una puerta que se abre a la automatización, a un control más riguroso de los datos, a sistemas más estables y una larga lista de etcéteras; la realidad es que las posibilidades de aplicación de ML en cualquier campo son, a día de hoy, prácticamente infinitas.

6. Valoración de la implantación de la tecnología Machine Learning en Mediación de Telefonía

Habiendo descrito previamente lo que es la tecnología de ML y dónde se puede aplicar y teniendo una idea general de lo que es la Mediación de Telefonía; en este punto se estudiará cómo se podría realizar la implantación de ML, a priori, compleja; sin embargo, no hace falta una larga explicación para comprender que, de implantarse de forma correcta, ML será capaz de encontrar patrones de comportamiento con los datos obtenidos, que permitirán la prevención y mejora del sistema que monitorice.

Puesto que en apartados anteriores se ha expuesto a grandes rasgos cómo funciona ML, lo que se pretende en este apartado es la realización de un pequeño estudio para ver cómo se podría integrar esta tecnología dentro del BSS, aplicada a la Mediación de Telefonía y más concretamente, aplicado a la lectura de logs.

6.1 Estudiando la implantación de ML

Antes de continuar, hay que destacar algo muy importante en este estudio: para poder instalar la tecnología de ML dentro de un entorno tan amplio como el de la Mediación, se necesitaría un experto en la materia con una dilatada experiencia, y no un investigador estudiante. Partiendo de esa base, lo que se pretende con este punto es mostrar cómo se podría abordar el problema de la instalación de la tecnología ML a pequeña escala.

Hasta ahora se ha hablado de Data Mining, Machine Learning y se han dado pequeñas pinceladas de estadística. Llegando a este punto, se va hacer una reflexión sobre las diferencias entre estos tres campos:

- Machine Learning: es una tecnología de la computación, derivada de la IA, que crea algoritmos a partir de una serie de patrones dentro de los datos. Ayuda a las compañías en la toma de decisiones en base a los datos analizados, su objetivo es predictivo y posee la capacidad de aprender de los datos que procesa, de forma que el factor humano deja de ser necesario.
- Estadística: es un subcampo de las matemáticas, busca las relaciones entre variables para poder predecir resultados.
- Data Mining: busca el conocimiento dentro de las vastas cantidades de datos que se generan hoy en día, para ello utiliza los mismos algoritmos o muy similares a machine learning. Sin el factor humano, no tendría sentido.

TABLA 5 ML, ESTADÍSTICA Y DM

Características	Machine Learning	Estadística	Data Mining
Finalidad: predicción	X	X	
Finalidad: conocimiento			X
Es autónoma	X		
Trabaja con grandes cantidades de datos	X		X
Precisa del factor humano		X	X
Funciona mejor con pequeñas muestras		X	

Tanto Machine Learning como Data Mining utilizan modelos estadísticos, la relación entre ambas podría ser la siguiente: DM se sirve de ML para alcanzar el conocimiento, que es su objetivo.

Una vez aclarado este punto, para poder estudiar la implantación de ML dentro del sistema, se van a explotar una serie de pasos a seguir:

1. Definición de los objetivos que se pretenden conseguir
2. Selección de los datos aplicables
3. Elección de indicadores de éxito
4. Creación del data mining
5. Estudio de algoritmos y modelos
6. Implantación

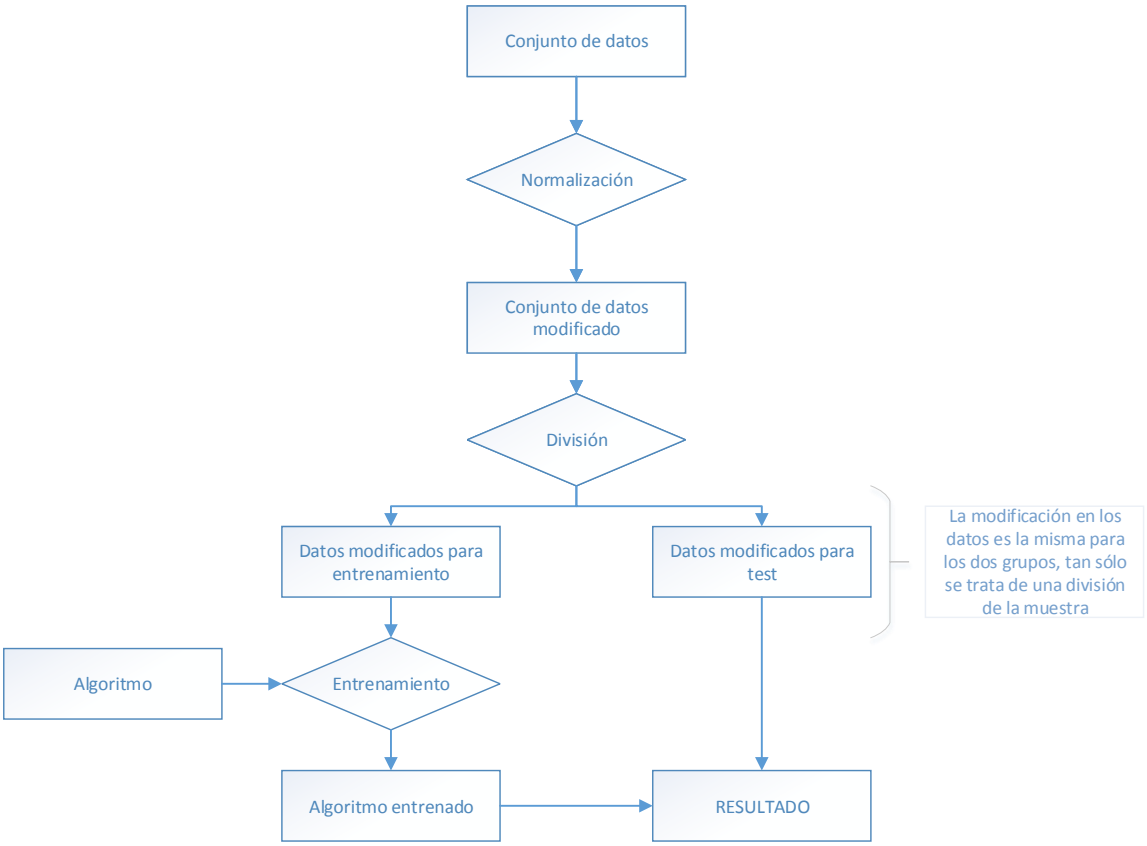


ILUSTRACIÓN 25 ALGORITMO ML (ELABORACIÓN PROPIA)

En el esquema de la imagen (Ilustración 25 Algoritmo ML) se menciona el entrenamiento, sin embargo, y pese a su importancia, no se ha hablado sobre qué es el entrenamiento. El entrenamiento de los datos consiste en proporcionar datos para que el algoritmo de ML aprenda de ellos, por lo tanto, se entiende que estos datos deberán contener la 'respuesta correcta' o lo que es lo mismo: *destino o atributo de destino*. Esta parte resulta particularmente interesante, puesto que gracias a estos entrenamientos se podrá seleccionar el algoritmo que mejores resultados obtenga según los datos que se le introducen (Román V., 2019).

Asimismo, hay una serie de términos que se precisan conocer:

- Algoritmo: conjunto de pasos a seguir u operaciones a realizar específicas para resolver un problema o ejecutar una tarea. Dentro del campo de ML, el algoritmo transformará y analizará los datos.
- Modelo: representación matemática de las relaciones dentro de un conjunto de datos, o lo que es lo mismo, forma simplificada de realizar las predicciones a partir de la aproximación a la realidad.
- Características o variables: elementos que forman el conjunto de datos
- Dataset, dataframe: conjunto de datos cargados en la plataforma de trabajo de ML

6.1.1 Objetivos que se pretenden abarcar con la implantación de ML

Dentro de un entorno de la magnitud de la Mediación de Telefonía, las opciones de aplicabilidad de ML dentro de los distintos sistemas y departamentos, podrían derivar en un documento excesivamente extenso. Es por esta razón por la que el estudio se centrará en una parte, aunque esto no implica que se mencionen y esbocen las posibles aplicaciones de ML dentro de la Mediación de Telefonía.

Cuando se trabaja con ML, se trabaja con algoritmos y estos algoritmos se entrenan con datos. Para saber qué algoritmo consigue unos resultados más óptimos se deberá en primer lugar, acotar el área en la que se pretende implantar ML y definir cuál es la problemática que se va a solucionar o la mejora que se va a proporcionar. Será necesario por las razones expuestas, explicar qué problemáticas puede resolver ML y analizar cuáles pueden ser los algoritmos que consigan el resultado más óptimo.

Como se ha mencionado, la aplicación de ML puede tener varios cometidos, como pueden ser las que se explicarán a continuación: clasificación, regresión, identificación de similitudes, clustering, agrupación por ocurrencias, profiling, predicción de vínculos, modelado casual y reducción de datos.

1. **Clasificación:** es el proceso de predecir la clase de datos dados. Se trata de aproximar una función de mapeo (f) de las variables de entrada (x) a las variables de salida (y). Las variables de salida serán usualmente denominadas como *etiquetas* o *categorías*. Como ejemplo se podría poner la clasificación por parte del correo electrónico entre *spam* o no-*spam*.
2. **Regresión:** e diferencia de la clasificación, la regresión tendrá como objetivo predecir valores continuos, es decir, a partir de unos datos, determinar un valor. Un ejemplo claro de un caso de regresión sería que basándose en unos valores como puedan ser ubicación, número de baños, número de habitaciones y metros cuadrados, se obtenga el valor de venta en euros de una casa.
3. **Identificación de similitudes:** se trataría de la identificación de patrones o tendencias en datos aparentemente inconexos.
4. **Clustering:** también denominada agrupación por ocurrencias o co-ocurrencias, es una forma de segmentar la información en grupos iguales o similares. Se utiliza en fases de explotación preliminar de datos y también en medicina, tanto para el diagnóstico de enfermedades a través de imágenes (escáneres, ecografías, etc.), como para monitorear redes sociales o mercados financieros.
5. **Profiling:** ampliamente utilizado en la detección de anomalías. Se analizan una serie de datos y a partir de éstos, se construyen perfiles (por ejemplo, perfiles de

usuarios). Gracias a estos perfiles se puede perfilar la tendencia por ejemplo en datos de consumo de móvil en base a la edad del usuario, zona geográfica, sexo, ect.

6. **Predicción de vínculos:** busca la conexión entre elementos. Se utiliza en redes sociales, si hay dos personas que tienen muchos amigos en común, tal vez esas dos personas son amigas, así que se sugiere al usuario que agregue a esa persona dentro de su red.
7. **Modelado causal:** trata de encontrar las causas que provocan que se suceda un evento. Se utilizan sobre todo para comprender el comportamiento del consumidor. A partir de una serie de datos de entrada y objetivos a conseguir dados, se debe buscar las razones o variables que determinen porqué un determinado usuario, por ejemplo, realiza la compra de un determinado producto o también, para la detección dentro de una cadena de montaje del elemento o conjunto de elementos que han provocado la reducción o parada del proceso productivo.
8. **Reducción de datos:** a partir de grandes cantidades de datos, se determina cuáles de ellos son de utilidad y cuáles no, desechando los que no aportan al contenido general, sin perder información. Esta técnica se aplica por ejemplo en diagnósticos para determinados tipos de cáncer. La pretensión es reducir la dimensionalidad de los datos manteniendo la máxima cantidad posible de información (Galarza Hernández J., 2017, pág 6).

A tenor de lo expuesto, la variedad de tareas en las que se puede aplicar ML dentro de un conjunto de datos, es bastante amplia. Es por esta razón que el objeto de estudio deberá focalizarse hacia una única dirección, a fin de no eternizar el proceso de investigación ni la extensión de este documento. El foco de atención del estudio, serán por lo tanto los logs generados por la herramienta eIUM donde se centrará la implantación de ML, su misión será prevenir cualquier tipo de problemática asociada a la información que contienen; o lo que es lo mismo, ver si es posible predecir con cierto margen de tiempo la caída de los logs.

6.1.2 Selección de los datos aplicables

Una vez se tienen las ‘variables a analizar’ que, para el caso de este estudio, son los ficheros de log, es importante responder a una serie de preguntas: ¿Se trata de variables discretas o variables continuas? ¿Aplica el aprendizaje supervisado o es más interesante el no-supervisado? Cuando se hayan respondido a estas preguntas, será el momento de seleccionar de entre los posibles, el algoritmo que mejor se adapte a la cobertura de las necesidades o mejoras que se pretenden conseguir.

De acuerdo con expertos como Alberto Coronado (Coronado A., 2017), la selección del tipo de algoritmo se puede realizar a través de la siguiente gráfica, siempre teniendo en cuenta el tipo de datos con los que se está trabajando:

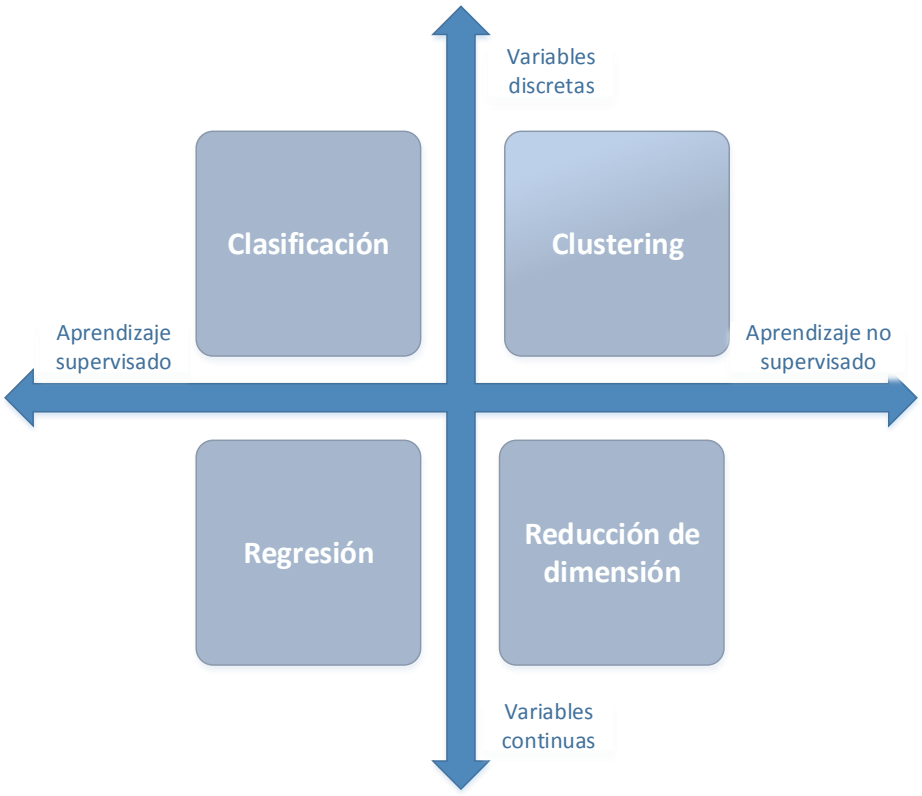


ILUSTRACIÓN 26 TIPOS DE ALGORITMOS ML (ELABORACIÓN PROPIA)

6.1.2.1 Variables discretas vs variables continuas

En primer lugar, se debería definir qué es una variable continua, qué es una variable discreta y una vez se tengan claros esos conceptos; habrá que identificar los distintos tipos de variables dentro de un log para poder englobarlos dentro del conjunto de variables continuas o discretas. De esta forma quedarían descartados al menos dos de los tipos de algoritmo que se pueden diferenciar en el gráfico.

Variable discreta:

La probabilidad nació a raíz de los juegos de azar, e inicialmente sus cálculos se realizaban a partir del conteo. La probabilidad discreta se utiliza, por ejemplo, para calcular la probabilidad que existe de que, en un dado caiga un número u otro. Es este tipo de probabilidad el que define las variables discretas, es decir, variables que toman una serie de valores dados en un intervalo y que no pueden tomar otros, sólo los que pertenecen a un conjunto, por ejemplo: un dado de seis caras puede tomar únicamente seis valores (1,2,3,4,5,6).

Si todos los resultados posibles poseen la misma probabilidad de ocurrir dentro de un espacio finito, la probabilidad de un evento se define como el número total de resultados en el evento, dividido entre el número de resultados posibles en el espacio muestra.

La probabilidad *P(E)* de un evento *E* del espacio muestra finito *S* es:

$$P(E)=|E| / |S|$$

Las variables discretas se utilizan por ejemplo para determinar el tiempo de ejecución de los algoritmos dentro de la computación (Johnsonbaugh R., 2005, pág. 76)

Variable continua:

Si la variable discreta sólo puede tomar una serie de valores dentro de un rango, la variable continua, sin embargo, podría tomar cualquier valor dentro de un rango acotado; por ejemplo, entre 1 y 2, la variable continua podría tomar un 1.1, 1.2, 1.4…

Las variables continuas no se pueden medir con total exactitud, su valor está directamente relacionado con la exactitud del instrumento de medición al que se refieren. Esto significa que una persona cuyo peso es de 65 kg, significa que dependiendo de la exactitud de la balanza ese mismo peso podría oscilar entre 65.21kg, 65.2kg o 65.18kg.

6.1.2.2 Aprendizaje supervisado vs aprendizaje no supervisado

Como se vio en el apartado (4.4.3 Funcionamiento básico de Machine Learning) la diferencia básica entre el aprendizaje supervisado y no supervisado, es la disposición o no, de una serie de datos de salida a partir de los datos de entrada disponibles. Gracias a estos datos de salida, se podría realizar un *entrenamiento* del algoritmo que permita predecir las salidas que pueden darse a partir de las entradas disponibles.

Para el análisis de logs que se pretende realizar, el objetivo es que la capacidad del modelo para hacer nuevas predicciones sea la máxima posible, esto se conoce como el proceso de *generalización*.

Aunque no se haya mencionado, además del aprendizaje supervisado y no supervisado, existe un tercero, que es el *aprendizaje reforzado*. La razón por la que no se ha mencionado, es porque se utiliza para algoritmos de juego de partidas, se basa en la recompensa obtenida en cada interacción que realiza. Se podría decir que la recompensa mide el grado de corrección de la acción a la hora de conseguir un objetivo determinado.

Puesto que en el caso de los logs se dispone de un histórico más o menos amplio de datos de entrada, con sus respectivas salidas; se podría decir que se trataría de un aprendizaje supervisado.

6.1.3 Elección de indicadores de éxito

Los indicadores de éxito podrán ser calculados una vez se pone en marcha el algoritmo de predicción, para ello se habrá tenido que seleccionar un algoritmo y realizar el entrenamiento pertinente con los datos de prueba.

Dentro de cada tipo de aprendizaje, se da una forma de medición diferente, de forma que habría:

- Métodos de clasificación ➔ Matriz de confusión
- Métodos de regresión ➔ Error Absoluto Medio (EMA), Error Cuadrático Medio (ECM), Raíz del Error Cuadrático Medio (RECM), etc.
- Métodos de Clustering ➔ Distancia Euclídea, Distancia Negativa, etc.
- Métodos de reducción de la dimensión ➔ Aproximaciones basadas en vecinos cercanos, aproximaciones paramétricas, etc.

Para los métodos de aprendizaje no supervisado (Clustering y reducción de la dimensión) las métricas son menos precisas y más subjetivas, por lo tanto, el estudio no se centrará en ellas.

6.1.3.1 Indicadores de éxito en Métodos de Clasificación

ILUSTRACIÓN 27 MATRIZ DE CONFUSIÓN (ELABORACIÓN PROPIA)

En la siguiente imagen (Ilustración 27 Matriz de confusión), se observará la teoría en la que radica este apartado: la matriz de confusión, matriz de error o tabla de contingencia. Se utiliza para evaluar la precisión de un clasificador, es decir, de un algoritmo de clasificación (Sánchez Muñoz J. M., 2016, pág. 11). Aceptada por la ISO 19157, se utiliza para medir la exactitud de una serie de valores. Se trata de discriminar los distintos tipos de errores que pueden darse a la hora de aplicar un algoritmo de clasificación para ML.

Donde:

- TP: Verdaderos positivos o *true positives*. Se define como el "número de elementos pertenecientes a la clase positiva que el clasificador ha predicho como positivos" (Gámez Granados J. C., 2017, pág. 27)
- FP: Falso Positivo o *false positive*. Se define como el "número de elementos pertenecientes a la clase negativa que han sido clasificados como positivos" (Gámez Granados J. C., 2017, pág. 27)
- TN: Verdaderos negativos o *true negatives*. Se define como el "número de elementos de la clase negativa que han sido clasificados como positivos" (Gámez Granados J. C., 2017, pág. 27)
- FN: Falso Negativo o *false negative*. Se define como el "número de elementos de la clase positiva que han sido clasificados como negativos" (Gámez Granados J. C., 2017, pág. 27)

Los verdaderos negativos, verdaderos positivos, falsos negativos o falsos positivos, se calcularán de la siguiente forma: una vez elegido el modelo, el algoritmo, obtenidos los datos de entrenamiento, etc.; se comprobará si la predicción que realiza el algoritmo es acertada o no; de forma que se marcarán los resultados como la matriz indica, dada la muestra de datos disponible, tal como se indica en el vídeo de la Universidad Politécnica de Valencia (Despujol Zabala I., 2018).

La representación matemática sería la siguiente:

$$M(g) = \left\{ n_{i,j}; \sum_{i,j=1}^{Q} n_{i,j} = N \right\}$$

Donde (Gámez Granados J. C., 2017, pág. 27):

- $n_{i,j}$ representa el número de patrones predichos por el clasificador *g* en la clase *j* cuando en realidad pertenecen a la clase *i*.
- n_i se define como el número de patrones que pertenecen a la clase *n* e *i.*
- n_j es el número de patrones clasificados en la clase *j*

Dentro de las métricas de clasificación, se podrían mencionar las siguientes, aunque hay muchas más:

- Tasa de error o *Missclasification rate*

$$\text{Tasa de error} = \frac{FP + FN}{Total}$$

- Exactitud o *accurancy*

$$\text{Exactitud} = \frac{VP + VN}{Total}$$

- Sensibilidad, exhaustividad y tasa de verdaderos positivos o *recall*, *sensibility y true positive rate*. Es el porcentaje de patrones de la clase positiva que se clasificaron correctamente respecto al total de patrones de la clase positiva.

$$\text{Sensibilidad} = \frac{VP}{Total\ Positivos}$$

- Precisión, mide el porcentaje de patrones que se han clasificado de forma correcta:

$$\text{Precisión} = \frac{VP}{Total\ clasificados\ positivos}$$

- Valor de predicción negativo

$$\text{VPN} = \frac{VN}{Total\ clasificados\ negativos}$$

6.1.3.2 Indicadores de éxito en Métodos de Regresión

La pretensión en los problemas de regresión es intentar predecir un valor real asociado a una entrada determinada. Un ejemplo podría ser a partir de las precipitaciones de un año, calcular que volumen de energía puede generar una central eléctrica.

Son múltiples las métricas de medida del éxito en los Métodos de Regresión. En este apartado se mencionarán algunas, aunque hay muchas más. Una notación común que merece ser nombrada es que en las métricas de regresión la variable *Y* hace referencia a los valores reales de salida, mientras que la

1. Error cuadrático medio (o *Mean Square Error*): media de los errores cuadráticos cometidos en la predicción

$$\mathrm{MSE} = \frac{1}{N}\sum_{i=0}^{N}\left(\hat{Y}_i - Y_i\right)^2$$

2. Raíz del Error cuadrático medio (o *Root Mean Square Error*): raíz cuadrada de la media de los errores cuadráticos cometidos en la predicción.

$$RMSE = \sqrt{MSE} = \sqrt{\frac{1}{N}\sum_{i=0}^{N}\left(\hat{Y}_i - Y_i\right)^2}$$

3. Error Medio Absoluto (o Mean Absolute Error), también conocido como MAE: media de los errores en valor absoluto cometidos en la predicción.

$$\mathrm{MAE} = \frac{1}{N}\sum_{i=0}^{N}\left|\hat{Y}_i - Y_i\right|^2$$

6.1.4 Creación del data mining

Como se ha comentado con anterioridad, el data mining nace de la necesidad de gestionar las grandes cantidades de datos que se poseen, de forma que resulten útiles. Estas grandes cantidades de datos, exceden la capacidad del ser humano para ser procesadas o analizadas, de forma que se requiere de otras metodologías más eficientes.

La diferencia entre el Data Mining (DM a partir de ahora) y el ML, es básicamente que el DM necesita de la intervención humana y busca el conocimiento, mientras que el ML no precisa de la misma a excepción del momento en el que se definen los datos (estudio-implantación). Para que la aplicación de ML sea completa, la máquina debe aprender automáticamente a partir de los modelos de datos, a través de un algoritmo de autoaprendizaje, de forma que se mejora su rendimiento.

Hablar de DM es imposible sin mencionar el KDD o *Knowledge Discovery in Databases*, que es el "es el proceso completo de extracción de información, que se encarga además de la preparación de los datos y de la interpretación de los resultados obtenidos" (Molina López J. M. y García Herrero J., 2006, pág 1-2). La minería de datos es un paso dentro del KDD, se corresponde con la etapa analítica del KDD, puesto que se ocupa del análisis de grandes cantidades de datos, buscando la relación entre los mismos. La forma en la que se busca tal relación es a través de patrones y modelos dentro de los datos recopilados. Mientras que el KDD busca una analítica completa de los datos dentro de una serie de pasos, el DM, es sólo uno de esos pasos (dependiendo del autor se divide en más o menos pasos):

1. Selección de datos
2. Preprocesamiento
3. Transformación de datos
4. Data Mining
5. Conocimiento

Se han encontrado autores que definen el KDD y el DM como si del mismo proceso se tratara, sin embargo, no se va a ahondar en estas definiciones, puesto que no aportan información valiosa a este documento.

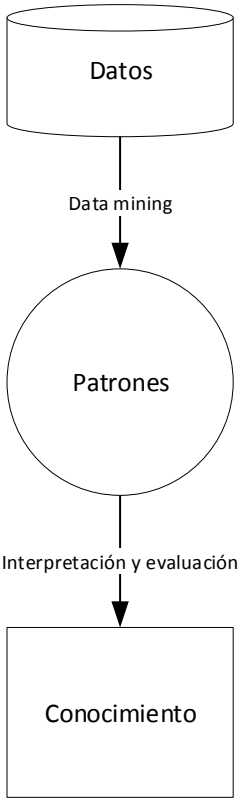


ILUSTRACIÓN 28 DATA MINING (ELABORACIÓN PROPIA)

Los datos que se recogen en los logs, encierran una serie de hechos o sucesos, mientras que los patrones definirán un modelo aplicable al subconjunto de datos. KDD implica un proceso iterativo e interactivo para buscar modelos, patrones o parámetros, que resulten útiles para el sistema (Molina López J. M. y García Herrero J., 2006, pág 1-2).

6.1.4.1 Limpieza de datos

La limpieza de datos es uno de los pasos más importantes con los que se encuentra el desarrollador de ML. Su importancia radica en que más información no significa mejor información.

Para la reducción de datos existe una técnica llamada 'reducción de la dimensionalidad' que se encuentra plasmada en la imagen (Ilustración 25 Algoritmo MLAlgoritmo ML). Los algoritmos de reducción de la dimensionalidad tienen como función la reducción del número de variables en una colección de datos (Lafuente A., 2018).

Los beneficios que tiene la reducción de datos (Lafuente A., 2018):

- Identificación y eliminación de variables o información irrelevante
- Ahorro de coste y tiempo
- Reducción de la complejidad
- Simplificación de los resultados

Existen dos grandes métodos para la reducción de los datos: el primero es la selección de variables, que básicamente consiste en seleccionar el conjunto de variables óptimo en base a los conocimientos que se tienen de los datos y la segunda es el análisis de componentes principales, que radica en profundos conocimientos matemáticos.

6.1.4.2 Selección de variables

La limpieza de datos que se va a aplicar implica el *tratamiento* de los mismos. En primer lugar, se van a detectar trazas que no son muy relevantes dentro de la detección de problemas.

El conjunto de variables que se define como óptimo dentro de un conjunto de datos, será aquel que contenga las variables más significativas del conjunto de datos original. Este conjunto óptimo se puede calcular en base al conocimiento en el campo o por lógica, aunque también existen otros métodos más matemáticos, aunque no necesariamente mejores que el primero:

- En base a la *correlación de las variables*: por ejemplo, dentro del campo de la medicina, se podría medir la asociación entre tener los ojos azules y desarrollar un cáncer de hígado, pero probablemente esta no sería la relación más significativa de los datos.
- En base a la *consistencia*: se trata de eliminar variables redundantes. Por ejemplo, dentro del ejemplo de cáncer de hígado, y teniendo la información sobre el estado general del hígado no sea relevante si esa persona ingería alcohol habitualmente o no (porque ese dato probablemente se vea a través de otras variables).

Tanto la correlación de variables como la consistencia de las mismas, se puede calcular mediante programas matemáticos como WEKA, que además es *open source*[6], o MATLAB, que es de pago, entre otros.

6.1.4.3 Análisis de componentes principales

También llamado PCA o Principal Component Analysis. Es un algoritmo que sirve para extraer las Características, donde se combinan las entradas de una forma determinada y permite la eliminación de algunas variables que se consideran 'poco relevantes'.

Cuando se selecciona una muestra de datos, la tendencia es pensar que cuantas más variables se tomen, mejor es esa muestra. Sin embargo, como ya se ha comentado, si se toman demasiadas variables, se tienden a considerar demasiados escenarios y en muchos casos, muchas de esas variables miden cosas similares.

Aunque este apartado se trata de reducir datos, es importante resaltar que, en términos matemáticos, a mayor información, mayor variabilidad o varianza. A mayor variabilidad de los datos (varianza) "se considera que existe mayor información, lo cual está relacionado con el concepto de entropía" (Marín Diazaraque J. M., 2014).

[6] Open Source o código abierto, basado en la colaboración de los usuarios, permite el acceso al código fuente y su modificación.

Debido a la complejidad de esta metodología, no se especificará más que lo que ya se ha hecho; puesto que requiere de una serie de conocimientos matemáticos de muy alto nivel.

6.1.4.4 Clasificación de los datos

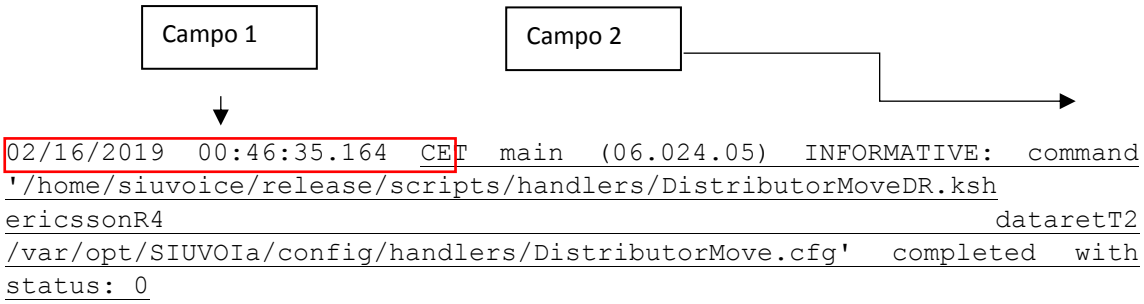
Además de realizar una reducción más o menos significativa dentro de los datos, se deberá proceder a su reestructuración.

La reestructuración de los datos resulta necesaria, de cara a que el programa en el que se van a tratar, pueda comprender los datos- Tal y como se presentan ahora mismo, no son legibles por ningún programa de tratamiento de datos. Lo que se pretende es la generación de una matriz de datos, para tener un formato legible por los programas de tratamiento de datos. Para ello será necesario:

1. División las líneas de código en campos
2. Modificación del formato de fecha, de forma que sea fácilmente legible
3. Volcado de los datos obtenidos en una matriz

1.- División de las líneas de código en campos

La división dentro de las líneas de log en campos es la base para el procesamiento de los ficheros de log a través de ML. Aunque existan múltiples formas a la hora de dividir una línea de log en diferentes campos, ésta se ha considerado la más adecuada:



Por un lado, el formato fecha y por otro, el resto de la línea de log. El resto de la línea de log deberá ser transformado, de forma que posea un valor determinado, a fin de poder trabajar con ésta información más cómodamente. Asimismo, la mayoría de las trazas del log serán desechadas, por no poseer información útil de cara a saber si el log abortará en un futuro inmediato, o no.

2.- Modificación del formato fecha

Como se comentó en los apartados iniciales (*Machine Learning*), el formato fecha es uno de los grandes problemas a la hora de analizar logs. Esta modificación, se realiza para que resulte más fácil el tratamiento de datos. Se podría hacer de varias formas: a través de un script y luego pasar los datos resultantes al programa que los

analizará o se podría transformar la información directamente en el programa si éste lo permite.

Se propone la modificación del formato fecha, ya que aparece en los logs y es importante de cara a la secuenciación de las trazas, sin embargo, dependiendo del tipo de datos con el que se trabaje, esta modificación será necesaria o no.

3.- Volcado de datos en una matriz

En este caso también tendríamos al menos dos opciones: por un lado, una en la que una vez se tienen los datos transformados, bastaría con meterlos en un fichero tipo Excel o de texto plano separado por campos, según las especificaciones del programa a través del cual vayan a ser tratados; y la segunda es realizar todo el trabajo dentro del propio programa.

Para tratar el texto desde un entorno ajeno a los propios matemáticos se podrían utilizar lenguajes como Python, Shell de Unix, Perl, Awk, etc.

Una de las formas de trabajo que se ha observado en diversos tutoriales (https://www.lynda.com/), es el volcado de los datos de trabajo en el programa Excel del paquete de Windows Office, tal y como se puede observar en la ilustración 29.

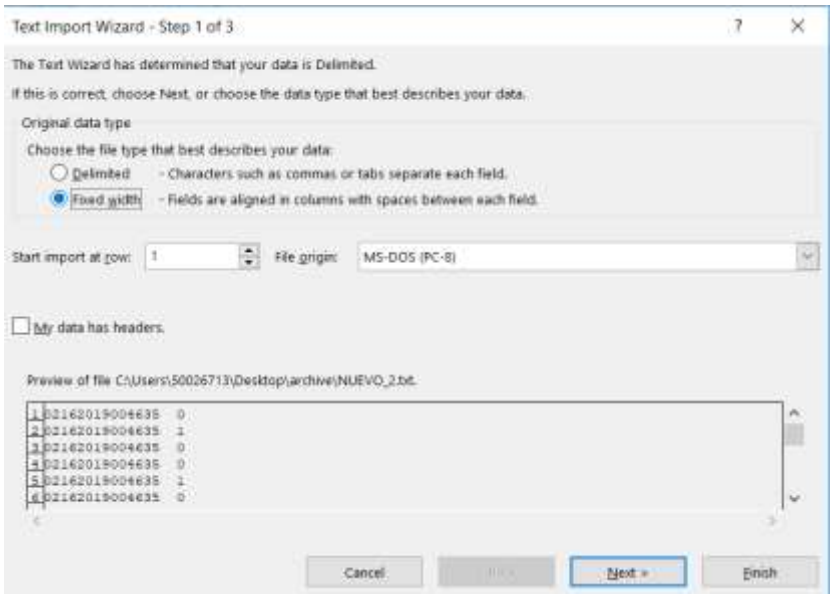


ILUSTRACIÓN 29 NORMALIZACIÓN EN EXCEL (ELABORACIÓN PROPIA)

Habría que cerciorarse de que el fichero Excel tiene el formato deseado (Normalización en Excel II):

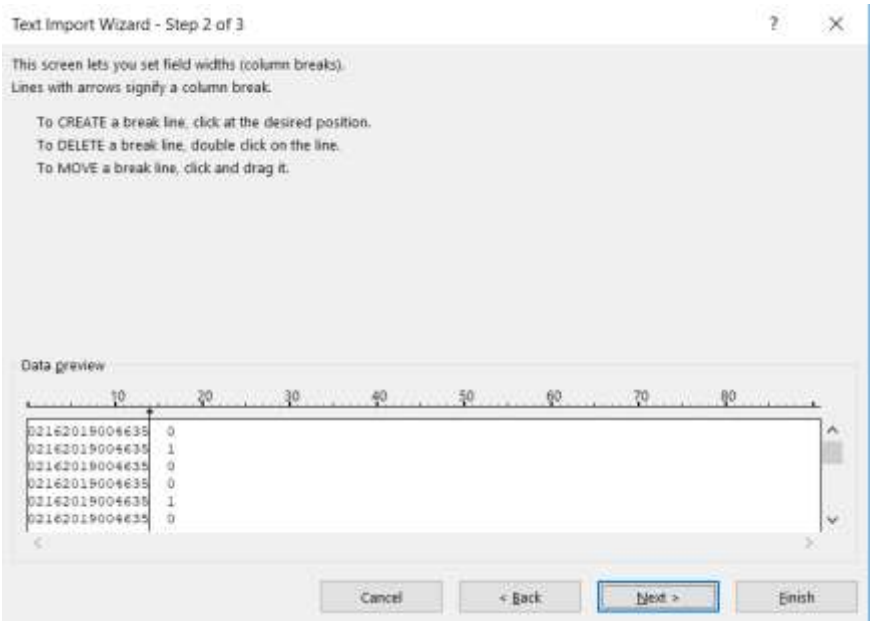


ILUSTRACIÓN 30 NORMALIZACIÓN EN EXCEL II (ELABORACIÓN PROPIA)

Y una vez comprobado, se deberá guardar con formato CSV, para que sea legible con cualquier programa de procesamiento matemático.

	A	B
1	2162019004635	0
2	2162019004635	1
3	2162019004635	0
4	2162019004635	0
5	2162019004635	1
6	2162019004635	0
7	2162019004635	0
8	2162019004635	1
9	2162019004635	1
10	2162019004635	1
11	2162019004635	0
12	2162019004635	0
13	2162019004635	0
14	2162019004635	0
15	2162019004635	0

ILUSTRACIÓN 31 MUESTRA DE DATOS NORMALIZADOS (ELABORACIÓN PROPIA)

El formato .csv es el más deseable, puesto que se podría trabajar con la matriz de datos en cualquier programa de tratamiento matemático, aunque otra opción bastante extendida es guardarlo directamente en un fichero de texto plano y extensión *.txt*, tipo wordpad o vi.

Existen multitud de programas de tratamiento matemático con los que se pueden tratar los datos, de manera previa o en su totalidad. Durante la Ejemplificación, se verá que pese a haber muchos programas en el mercado como el Matlab, Excel o Weka; a la hora de trabajar con Machine Learning, lo más extendido y habitual es programar directamente con Python. El lenguaje Python permite la realización de gráficas además

de poseer librerías propias para trabajar ML, por esta razón es el más extendido en Big Data.

6.1.5 Estudio de algoritmos y modelos

El modelado estadístico es un campo dentro de las matemáticas dedicado a encontrar relaciones entre las variables para poder predecir resultados. La forma de proceder es formalizando las relaciones entre variables en los datos en forma de ecuaciones matemáticas. De esta forma se podría trabajar con modelos geométricos, probabilísticos o lógicos.

Un algoritmo no es otra cosa que una serie de pasos estructurados y ordenados que se deben realizar para llevar a cabo una tarea. Según los objetivos que se pretendan conseguir, se trabajará con un algoritmo u otro.

La relación entre modelos y algoritmos es muy importante en ML, puesto que los modelos que se obtendrán dependerán del tipo de algoritmo que se escoja.

Para saber cuál/cuáles algoritmo(s) son más óptimos para los objetivos marcados en la implantación de ML en Mediación de telefonía, se deberá investigar cuáles son los algoritmos existentes, cuáles son sus usos, ventajas e inconvenientes; todo dentro de los algoritmos de clasificación, que son de entre todos los existentes, aquellos que resultarán más interesantes de cara a los objetivos que se pretenden conseguir. En este apartado se analizarán los más utilizados, aunque existen muchos más.

Pese a que se explicó en el apartado introductorio *(Funcionamiento básico de ML*) cómo funcionaba a grandes rasgos, a fin de refrescar la memoria, este tipo de algoritmos tienen por objeto que, a partir de los datos de entrenamiento, el algoritmo sea capaz de realizar predicciones.

Dentro de algunos de los programas de tratamiento matemático que se han mencionado, se pueden simular algunos de estos algoritmos, a fin de simplificar la tarea, puesto que la implementación de los mismos es tediosa y compleja, y se necesitará probar más de uno para poder decidir cuál de ellos, a partir de los datos proporcionados, obtiene los mejores resultados.

Dentro del grupo de los modelos de clasificación, a continuación, se mostrarán algunos de los más populares, explicándolos brevemente, puesto que la literatura que hay al respecto es tan amplia como variada, y no procede extenderse en gran medida.

6.1.5.1 Algoritmo Bayesiano:

El algoritmo bayesiano es muy importante puesto que es la base para otros algoritmos de aprendizaje como: Naivie Bayess, Gaussian Naive Bayes, multinomial Naive Bayes y Bayesian Network.

Se utilizan en problemas de clasificación y regresión.

El razonamiento bayesiano está basado en el teorema de Bayess de probabilidad. Parte del supuesto de que las cantidades de interés se rigen por las distribuciones de

probabilidad y que además se pueden tomar decisiones óptimas al razonar estas probabilidades junto con los datos observados. Gracias a ello proporciona un enfoque cuantitativo para sopesar la evidencia que apoya hipótesis alternativas.

Características propias del algoritmo bayesiano (Mitchell M. T., 1997, pág. 154-160):

1. El conocimiento previo se puede combinar con los datos observados para determinar el resultado final
2. Pueden contener hipótesis que hacen predicciones probabilísticas (por ejemplo: este perro tiene un 95% de padecer artrosis a partir de los 5 años de edad)
3. Se pueden clasificar nuevas instancias a través de la combinación de las predicciones de múltiples hipótesis y sus probabilidades
4. Hasta en los casos en los que los métodos bayesianos son computacionalmente intratables, podrían proporcionar un estándar de toma de decisiones óptimas

Puesto que el teorema de Bayes proporciona una forma para calcular la probabilidad de cada hipótesis dados los datos de entrenamiento, se podría usar como base para un algoritmo de aprendizaje directo que calcula la probabilidad para cada posible hipótesis y que luego obtenga la salida más probable (Mitchell M. T., 1997, pág. 158-160).

6.1.5.2 Árboles de decisión

De forma tanto gráfica como analítica se podrían definir a los árboles de decisión como la representación de los sucesos que pueden surgir a partir de una decisión que se asumió en un cierto momento. Podría representarse a través de reglas if-then y su finalidad es la toma de decisiones en base a la probabilidad, seleccionando la mejor opción del abanico (Berlanga Silvente V., Rubio Hurtado M. J. y Vila Baños R., 2013, pág. 1).

Dentro de los árboles de decisión, existen varios algoritmos, como puedan ser los CHAID, CHAID exhaustivo, CRT y QUEST, se seleccionará siempre el mejor, según el tipo de ajuste a nuestros datos (Berlanga Silvente V., Rubio Hurtado M. J. y Vila Baños R., 2013, pág. 2).

Los árboles de decisión se localizan dentro de la minería de datos y al igual que los demás algoritmos, poseen un importante peso en estadística e IA.

Las ventajas de un árbol de decisión son (Pérez, 2007):

- *"*Facilita la interpretación de la decisión adoptada.
- Facilita la comprensión del conocimiento utilizado en la toma de decisiones.
- Explica el comportamiento respecto a una determinada decisión.
- Reduce el número de variables independientes*."*

Se utilizan, por ejemplo, para evaluar el riesgo de crédito de los solicitantes de préstamos o para el diagnóstico de casos médicos.

6.1.5.3 K vecinos más próximos, K-nearest neighbors o Knn

Se basa en la teoría de que los prototipos más cercanos tienen una probabilidad similar. Para que se entienda mejor: "en la fase de entrenamiento, se almacenarán los vectores característicos y las etiquetas de las clases de los ejemplos de entrenamiento. En la siguiente fase, la de clasificación se evaluará el ejemplo (del que no se conoce su clase) representándolo con un vector. Se calcula la distancia entre los vectores almacenados y el nuevo vector y se seleccionan los K elementos más cercanos" (Arriagada Rodríguez M., 2015, pág. 7).

El algoritmo se utiliza para clasificar nuevas muestras dentro de valores discretos, aunque también puede usarse como método de predicción para valores continuos, dentro de los algoritmos de regresión. Esto se traduce en que sirve para realizar recomendaciones, búsquedas semánticas o detección de anomalías.

A partir de este algoritmo surgen otros relacionados, como el KNN con rechazo, en el que se rechazan algunos datos de la muestra; el KNN con distancia media, en el que la selección del vecino más próximo sufre algunas variaciones; el KNN con distancia mínima, en el que se comienza seleccionando el caso más cercano al baricentro de la clase, etc. (Moujahid A., Inza I. y Larrañaga P., ss. Ff., pág. 3-5).

Pese a ser un algoritmo muy utilizado debido a la sencillez del mismo, tiene desventajas importantes, tales como:

- Es altamente sensible al ruido
- Depende de que la función de distancia sea la adecuada.
- Se da a entender que los vecinos más cercanos proporcionan la mejor clasificación, a través de todos los parámetros (atributos) de la muestra. Sin embargo, muchos atributos son irrelevantes y en ocasiones dominan sobre la clasificación
- Está comprobado que para que este algoritmo obtenga los mejores resultados, la muestra de datos de entrenamiento deberá ser relativamente grande. Al realizarse con muestras grandes, el consumo de recursos es mayor y la velocidad inferior.

6.1.5.4 Regresión logística (Logistic regression)

El modelo de Regresión Logística parte de los algoritmos de clasificación dentro del aprendizaje supervisado. Parte de la Regresión Logística que se aplica en la estadística de análisis de datos. Se utiliza cuando se quiere relacionar una variable dependiente cualitativa con una o más variables independientes (Salas Velasco M., 1996, pág. 193). En este modelo, la variable dependiente es binaria, es decir, que tiene dos posibles resultados: 0-1, abierto-cerrado, blanco-negro, etc., aunque también podría ser de un conjunto acotado de resultados, como, por ejemplo, en cuanto a colores: blanco, negro, rosa, azul o en cuanto a razas de perro: pastor alemán, bulldog o caniche.

Generalmente se utiliza para poder estimar la probabilidad de una respuesta binaria a partir de una o más variables predictoras o independientes (González L. ,2018), aunque sus objetivos pueden ser los siguientes (Salas Velasco M., 1996, pág. 195):

- Determinar la existencia o ausencia de relación entre una o más variables independientes en apariencia
- Medir la relación en caso de que exista
- Predecir la probabilidad de que se produzca un acontecimiento en función de las variables independientes.

Algunos ejemplos del uso de modelos de regresión logística son los siguientes (Bagnato J. I., 2017, p.3):

- Clasificación del correo en Spam/NoSpam
- Dados los resultados de un tumor: Benigno/Maligno
- Dentro de un libro: novela/ciencia-ficción/romance/tutorial
- A partir de un historial bancario concesión-no concesión de un crédito

6.1.5.5 Redes neuronales artificiales (Artificial neural networks)

Las Redes Neuronales Artificiales (a partir de ahora, RNAs) son redes en las que existen elementos que procesan información de cuyas interacciones locales depende el comportamiento del conjunto del sistema.

"Las RNAs tratan de emular el comportamiento del cerebro humano, caracterizado por el aprendizaje a través de la experiencia, y la extracción de conocimiento genérico a partir de un conjunto de datos" (Levín Mangin J. P., Flórez López R. y Fernández Fernández J. M., 2008, pág. 10). La forma en la que estas estructuras imitan el funcionamiento del cerebro es mediante la construcción de sistemas cuya arquitectura se asemeja a la cerebral.

Se ha creado la tabla (Tabla 6 Cerebro-vs computadorCerebro-vs computador) para ver de una forma más clara las diferencias y similitudes entre el cerebro humano y las RNAs (Levín Mangin J. P., Flórez López R.y Fernández Fernández J. M., 2008, pág. 12):

TABLA 6 CEREBRO-VS COMPUTADOR

Características	***Cerebro Humano***	***Computador***
Velocidad de proceso	*Entre* 10^{-3} *y* 10^{2} *seg.*	*Entre* 10^{-8} *y* 10^{-9} *seg.*
Estilo de procesamiento	*Paralelo*	*Secuencial (en serie)*
Número de procesadores	*Entre* 10^{11} *y* 10^{14} *seg.*	*Pocos*
Conexiones	*10.000 por procesador*	*Pocas*
Almacenamiento del conocimiento	*Distribuido*	*En direcciones fijas (posiciones precisas)*
Tolerancia a fallos	*Amplia*	*Poca o nula*
Tipo de control del proceso	*Autoorganizado (democrático)*	*Centralizado (Dictatorial)*
Consumo de energía para ejecutar una operación en segundos	10^{-16} *Julios*	10^{-16} *Julios*

Las RNAs, se forman de manera similar a como funciona un cerebro: si éste se basa en pequeñas células llamadas neuronas, las RNAs, se basará en unidades que realizan funciones semejantes llamadas *elemento procesador*. Las funciones de los elementos procesadores serán: aprender de la experiencia, generalizar desde ejemplos

previos hasta ejemplos nuevos y abstraer las características principales a partir de una serie de datos.

En líneas generales, se podría decir que las RNAs, son un modelo para encontrar una combinación de parámetros que determinen una solución concreta, por ejemplo, dados los píxeles de una fotografía, el modelo deberá confirmar en número que hay dentro de esa fotografía, si es que lo hay.

Un elemento procesador, posee una serie de entradas que son combinadas, generalmente a través de una suma básica. La suma de las entradas, se modifica a través de determinados procesos y se obtiene un valor de salida. Esta salida puede conectarse con otras entradas de otras neuronas artificiales (elemento procesador) y así sucesivamente.

La red neuronal se forma a partir de elementos procesadores conectados de una forma concreta. La forma en la que se conecten estos elementos procesadores, determinará en gran medida la eficacia del modelo.

Los usos de las redes neuronales son amplios, especialmente para el reconocimiento (Julián G., 2014):

- Google Street View cuenta con un tipo de red neuronal que logra el reconocimiento del 96% de los números en las calles
- Reconocimiento de voz
- Ahorro energético en centros de datos
- Determinados juegos de ordenador

6.1.5.6 Máquinas de Vectores de Soporte (Support Vector Machines)

Las Máquinas de Vectores de Soporte (SVM a partir de ahora), son algoritmos pertenecientes al aprendizaje supervisado, muy versátiles ya que se pueden resolver problemas tanto de clasificación como de regresión.

Su funcionamiento básico sería el siguiente: a partir de un conjunto de datos de entrenamiento, se etiquetan las clases y se entrena una SVM para que construya un modelo que sea capaz de predecir la clase de una nueva muestra.

Se denominará atributo a la variable predictora y característica a un atributo transformado que se usa para definir el hiperplano. La elección de la representación más adecuada del universo de datos estudiado, se realizará a través de un proceso que se denomina selección de características.

La mayoría de los métodos de aprendizaje existentes se centran en la minimización de los errores en las predicciones que realiza el modelo, a partir de los datos de entrenamiento. Lo que pretende el SVM es minimizar el *riesgo estructural*, o lo que es lo mismo, que la solución no dependa de la estructura del planteamiento del problema. Por un lado, la idea es minimizar el error en la separación de los objetos dados (error de clasificación) y por otro, maximizar el margen de separación, o lo que es lo mismo: mejorar la generalización del clasificador.

Este tipo de modelos, se utilizan para "la visión artificial, reconocimiento de caracteres, categorización de textos e hipertextos, clasificación de proteínas, procesamiento de lenguaje natural, análisis de series temporales, etc." (Carmona Suárez E. J., 2014, pág. 1)

Dentro de los algoritmos de SVM los hay de varios tipos, dependiendo del tipo de datos con el que se trabaje: de regresión, de clasificación binaria de ejemplos cuasi-separables linealmente, clasificación binaria de ejemplos separables linealmente o de multiclase, entre otros.

6.2 Problemas que pueden presentar los algoritmos de aprendizaje automático

La instalación de Machine Learning dentro de cualquier sistema, es un proceso complejo, tal como se ha visto. Se requieren en primer lugar grandes conocimientos sobre la propia tecnología de ML, lo cual implica dominar numerosos campos dentro de las matemáticas y, además, un nivel alto de conocimientos del funcionamiento del sistema en el que se desea instalar la tecnología: su problemática y sus necesidades. Aunque esas dos variables se den (conocimientos en ML y en el sistema que se pretende instalar), existen una serie de problemáticas asociadas a la implantación de ML en el sistema, en este apartado se explicarán en qué consisten.

Los errores más frecuentes que se dan en la implantación de ML en un sistema y que se analizarán a continuación son los siguientes:

- Sobreentrenamiento y subentrenamiento
- Falta de preparación
- Los algoritmos se equivocan
- Otros errores documentados

6.2.1 Sobreentrenamiento y subentrenamiento

En el primero de los apartados de este estudio se habló sobre el aprendizaje supervisado y no supervisado. Ambos casos, requieren de lo que se llama el entrenamiento de los datos, es en punto donde podría producirse un sobreentrenamiento u *overfitting* de datos o subentrenamiento o *underfitting* de datos.

El proceso denominado entrenamiento de un modelo de ML, no es otra cosa que el abastecimiento de datos de entrenamiento para que aprenda de ellos el algoritmo de aprendizaje de ML (Amazon Machine Learning, Ss. Ff.). Para que el entrenamiento sea efectivo, sus datos deben contener la respuesta correcta, denominada *destino o atributo de destino*. La forma de trabajar será la siguiente: una vez dados los datos para el entrenamiento, el algoritmo de aprendizaje encuentra los patrones en los datos y genera un modelo de ML que captura dichos patrones.

El sobreentrenamiento (o overfitting) es la tendencia que tienen los algoritmos de ML para encontrar la relación entre datos, que realmente sólo se relacionan entre sí a través de la casualidad; es decir que ML encuentra la relación dentro de eventos que realmente no la tienen. Sabiendo que esta situación puede darse, se deberá evitar en la

medida de lo posible, para ello existen dos técnicas: retención de datos y validación cruzada. Suele ocurrir cuando se le proporciona una muestra lo bastante amplia al algoritmo de ML, pero no lo bastante heterogénea.

Otro problema contrario al sobreentrenamiento podría ser el subentrenamiento (o underfitting), suele darse cuando la muestra de datos que se proporciona es demasiado pequeña, el algoritmo no será capaz a proporcionar una respuesta en base a un conocimiento sólido.

6.2.2 Falta de preparación

Con titulares como el de la revista BigDataMagazine "El 58% de las empresas ven el análisis de datos como un activo estratégico para su negocio, cifra que asciende al 70% en el caso de España." (Ramírez V,2018- diciembre) o realizando una simple búsqueda en cualquier buscador de tipo: '¿Estás preparado para el big data?' se encuentran numerosas publicaciones sobre el Big Data y las múltiples ventajas que su análisis puede llevar a cualquier negocio. Esto ha llevado a que muchas empresas se lancen a la conquista del Big Data con la implantación de ML, sin tener claros conceptos como la minería de datos, el Business Intelligence o el Data Science.

Antes de plantearse una implantación de estas magnitudes en el sistema, habría que estudiar primero si la implantación de tecnologías Big Data traerá consigo las ventajas que todo buen comercial y todo titular puntero promete: aumentar el crecimiento del negocio para negocios tradicionales, explotar nuevas líneas de negocio, etc.

La problemática que se plantea en este apartado, trataría no sólo de implantar ML por las ventajas que puede traer a la empresa; sino que también habría que comprender qué es el Big Data y que ventajas reales podría ofrecer. Una vez analizados estos puntos, y realizado el análisis de impacto económico pertinente, ya se podría poner en marcha (o no) la implantación de la tecnología, sin dejarse llevar por impulsos o modas.

6.2.3 - Los algoritmos "se equivocan"

Aunque no es fácil dar con ellas, existen numerosas noticias en las que se habla de los errores cometidos por los algoritmos de ML. Realmente no es que el algoritmo se equivoque, sino más bien que el planteamiento del problema ha sido incorrecto, lo que se traduce en un fallo humano, una vez más.

Mucho se ha hablado sobre el programa COMPAS (Correctional Offender Management Profiling for Alternative Sanctions), cuyo algoritmo es capaz de predecir si un individuo es de 'alto riesgo para la comunidad'. Básicamente y de acuerdo con Julia Angwin, defensora del programa, "es básicamente un cuestionario que se le da a las personas cuando son arrestadas. Y se les pregunta un montón de cosas que terminan concluyendo si esa persona en el futuro podría cometer un crimen" (Maybin S., 2016, p.13). Además del cuestionario, el programa incluye antecedentes familiares en

crímenes, amigos, ubicación habitual de la persona y otra serie de datos sensibles, que aparentemente podrían estar relacionados.

A simple vista, podría parecer que el algoritmo tiene sentido y hasta se podría decir que funciona de forma correcta. Sin embargo, COMPAS también tiene sus prejuicios y también se equivoca.

Pese a que en EEUU muchas de las grandes decisiones se toman en base a programas como COMPAS, se ha observado que éste tiene ciertos errores, por ejemplo, aunque la etnia no es un factor dentro de los que se analizan, las minorías étnicas siempre salen mal paradas en este tipo de análisis, con afirmaciones como: "Si comparas a una persona negra y una blanca que tienen el mismo historial, la misma edad, el mismo género, el mismo pasado judicial y el mismo 'futuro criminal' (las posibilidades de cometer un crimen, dos crímenes o ninguno), el acusado negro tiene un 45% más de posibilidades de obtener un puntaje de riesgo que un acusado blanco" (Maybin S., 2016, p.18). El problema comienza a ser de base cuando se analiza el entorno del acusado y se comprueba que en EEUU la población negra, culpable o inocente, tiene mayor incidencia en los conflictos con la policía.

Se concluye, por lo tanto, que no es que los algoritmos se equivoquen o que ML no funcione para aquello para lo que se ha programado; es que sencillamente el planteamiento de ML lo realizan humanos, y como tales, se equivocan.

6.2.4 Otros errores documentados:

De acuerdo con los artículos que se mencionan a continuación, muchos de ellos escritos por empresas que 'venden' la tecnología de ML, hay una serie de errores típicos que se comenten alrededor de ésta:

- Planteamiento de Machine Learning sin un *data sciencist*[7] o científico de datos (SAS, ss. Ff, p. 2). La empresa SAS, dedicada entre otras, a la implantación de ML, es bastante crítica en cuanto a un estudio de impacto dentro de una empresa para la implantación de ML. No sólo propone que este trabajo lo realicen profesionales dentro de esta rama, sino que además el artículo menciona su mantenimiento en el tiempo a fin de actualizar y mejorar la herramienta. La solución al problema de la carencia de un data sciencist viene de la mano de convenios con universidades y de la creación de un departamento que se dedique exclusivamente a la parte de ML dentro de la empresa. Lo que no se menciona en el artículo, es nada acerca del coste económico que esto supondría para cualquier empresa, grande o pequeña; aunque probablemente y como se ha dicho en ocasiones anteriores, el beneficio de esta inversión será proporcional.
  Es importante en este punto no caer en el error de contratar a cualquier data sciencist (Pranav D., 2018, p. 22-27), se debe tener en cuenta además de las certificaciones pertinentes, que la persona o el equipo tengan experiencia en el campo, aparte de los estudios realizados.

[7] Profesional dedicado a la interpretación de grandes volúmenes de datos, posee conocimientos de alto nivel de estadística y programación, aunque puede abarcar más campos

- Comenzar los entrenamientos con datos ruidosos (SAS, ss. Ff, p. 8): se remarca la importancia de un buen pre-procesamiento de los datos antes de comenzar las labores de entrenamiento.
- Falta de infraestructura (SAS, ss. Ff, p. 24): la implantación de ML requiere máquinas potentes y rápidas y un almacenamiento flexible.
- No tener datos suficientes para proporcionar al modelo de datos (Linthicum D., 2018, p.4-8): el objetivo del ML es que se produzca un aprendizaje, este aprendizaje no es posible si no existe la cantidad de datos suficiente, para hacer que surjan ciertos patrones y que se genere el entrenamiento necesario. Si no se poseen los datos necesarios, la implantación de ML, no tiene sentido puesto que no aportará nada, a excepción de gastos.
- Pretender utilizar ML donde no es necesario (Linthicum D., 2018, p. 9-15): ML no es la solución a todos los problemas, ni una mejora considerable en todos los casos. Un buen análisis de impacto por parte de un equipo de expertos, sería la solución más sencilla a este problema.
- No tomarse el tiempo necesario en (Pranav D., 2018, p. 40-44), la analítica de datos. Para poder aplicar ML, se debe tener un grado de conocimientos de la lógica de negocio muy alto, si no se toma el tiempo necesario en la adquisición de estos conocimientos, la implantación no será eficiente ni aportará una gran información al negocio.
- Resolución de problemas que no existen realmente: ML debe dar solución a un problema, no ser una solución a un problema que no se ha formulado.

6.2.5 Algunos errores cometidos por grandes compañías

Quien tiene boca se equivoca, en este subapartado se analizarán algunos de los errores históricos más sonados en IA.

- 2017: Facebook cierra un chatbot creado con el objetivo de hacerse pasar por humano en una conversación de chat, porque deformó el lenguaje inglés en el que se comunicaba hasta generar un idioma paralelo que a los ojos del humano era un sinsentido (Nieva R., 2017)

ILUSTRACIÓN 32 ERROR FACEBOOK (ELDIARIONUEVODIA.COM)

- 2017: Para el banco HSBC, la voz de los humanos era única e irrepetible, por eso en el 2016 y gracias a la ayuda de algoritmos de ML, generaron un sofisticado sistema de reconocimiento de voz en el que ésta actuaba como una huella digital, sin embargo, esta tecnología no era tan segura como prometía puesto que Dan Simmons, consiguió imitar la voz de su hermano hasta el punto de que el sistema de reconocimiento de su banco lo identificó como tal (BBC, 2017).

- 2017: Alexa, el asistente virtual que se controla a través de la voz lanzado por la multinacional Amazon, tras mantener una conversación con la pequeña de seis años de edad, de la familia Neitzel, en Texas (EEUU), realizó una compra online por valor de 160€ aproximadamente. La compra constaba de una caja de galletas y una casa de muñecas. La madre de la pequeña reparó en el envío cuando descubrió en su cuenta de correo el email de confirmación del mismo. Parece ser que la niña estuvo hablando con Alexa sobre las galletas y la casa de muñecas, entendiendo ésta que debía gestionar la compra. A partir de ese momento, sus padres han activado el control parental en todos los dispositivos de la casa (Antena3, 2017).

ILUSTRACIÓN 33 ERROR ALEXA (ANTENA3.COM)

- 2019: Es bien sabido que no existen sistemas infalibles, sólo falta de interés en las comunidades de hacker para reventarlos. Un grupo de investigadores de la Universidad de Lovaina, en Bélgica, ha ideado una forma de engañar a las cámaras que monitorizan sistemas a través de IA. Descubrieron una debilidad dentro del algoritmo de Redes Neuronales Convolucionales, perteneciente a ML (Merino M. , 2019).

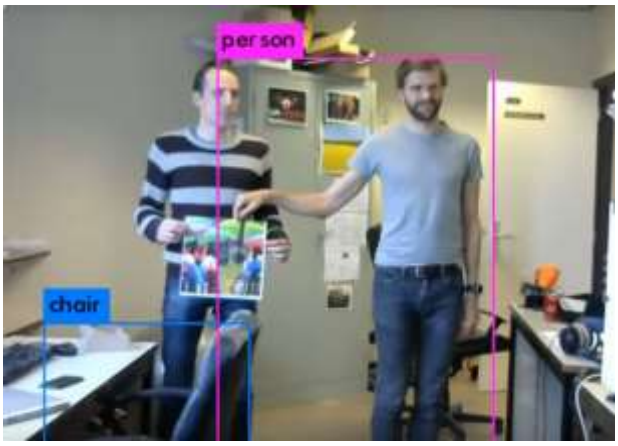


ILUSTRACIÓN 34 ERROR MONITORIZACIÓN (REDUSERS.COM)

6.2.6 Conclusiones sobre los errores ML

Las posibilidades de que la implantación de una tecnología tan compleja como ML fallen, incluso con un buen planteamiento y rodeado de expertos, son tan altas que hasta las grandes multinacionales como Facebook o Amazon han cometido errores que han trascendido.

Pese a los *errores* que puedan darse en la aplicación de ML, a lo largo de este documento se han dado muestras fehacientes de la cantidad de campos que ML abarca y ámbitos en la vida diaria de las personas, que a simple vista pasan desapercibidos.

Viendo su evolución, se podría decir que la tendencia en su utilización es al alza y que los errores que se puedan dar dentro de la aplicación de la tecnología, poco tienen que ver con su efectividad o eficiencia, se trata de errores humanos en algún punto de su desarrollo, entrenamiento o implantación.

Sin embargo, se debería tener cierto grado de escepticismo ante la aplicación de ML a comportamientos humanos. Si bien por un lado tenemos la estadística y la probabilidad, por otro tenemos la capacidad de cambio del ser humano. Porque los patrones no siempre se cumplen (hijo maltratado, padre maltratador) y aunque estadísticamente todas las probabilidades apunten a que un comportamiento se reproducirá, al igual que con la genética, existen las anomalías. Las personas no son máquinas, si algo ha distinguido al ser humano del resto de las especies, es la creatividad, la empatía, la capacidad de perdón, la conciencia de sí mismo… y por muy bueno que sea un algoritmo o una tecnología, todas esas cualidades, no las puede medir ni suplantar; al menos hasta ahora.

6.3 El lenguaje Python

El lenguaje Python es un lenguaje de programación poderoso, intuitivo y de aprendizaje sencillo. En este apartado se explicará muy brevemente el lugar que ocupa Python dentro de este proyecto y se hablará de Anaconda, una plataforma hasta ahora desconocida para la autora de este documento.

Cuando se comenzó esta andadura, se partían de cero conocimientos en el lenguaje, pero ha sido relativamente sencillo hacerse con él, debido en parte a su gran parecido con otros lenguajes de programación y también gracias a la innumerable cantidad de tutoriales que se encuentran en la red.

La forma más sencilla de trabajar con Python es a través de Anaconda, una plataforma gratuita en la que se ahondará en el siguiente apartado.

6.3.1 Trabajando con Python

Python es un lenguaje que cuenta con estructuras de datos eficientes y de alto nivel y orientada a objetos. Su nombre proviene del programa de televisión de la BBC "Monty Python's Flying Circus (Van Rossum G., 2009, pág. 9). Es un lenguaje ideal para el trabajo con scripts, rápido a la hora de procesar grandes volúmenes de datos, ya que realiza operaciones críticas a velocidad máxima.

De gran utilidad resulta la página web https://www.python.org/, en la que se encuentran multitud de tutoriales para empezar a programar en Python, además de ejemplos, preguntas frecuentes y foros de ayuda a usuarios. Además de literatura de ayuda, en la web mencionada también dispone de una amplia variedad de bibliotecas a libre disposición del usuario.

Además de todo lo mencionado, la opción de utilizar Python como lenguaje de extensiones para aplicaciones también existe.

Para poder trabajar en el lenguaje Python con ML, se utilizará *scikit-learn*, una librería muy importante dentro del lenguaje de Python, formada por: NumPy, pandas, SciPy, Matplotlib, IP[y] o SymPy.

ILUSTRACIÓN 35 SCIKIT LEARN PYTHON (LIDGI GONZÁLEZ)

Que se podrían describir, de acuerdo con sus características como:

- *NumPy: librería de matriz n-dimensional base*
- *Pandas: estructura de datos y análisis*
- *SciPy: librería fundamental para la informática científica*
- *Matplotlib: trazado completo 2D*
- *Ipython: consola interactiva mejorada*
- *SymPy: matemática simbólica* (González L., 2018)

Otra biblioteca que puede resultar interesante es TensorFlow, creada por Google. En la página web https://www.tensorflow.org/ hay un mundo de posibilidades para comenzar a trabajar con la biblioteca de TensorFlow: tutoriales, ejemplos, ayuda, preguntas frecuentes, etc. Además desde la Universidad de León se realizaron una serie de tutoriales en YouTube, en los que muestran a esta biblioteca de una forma sencilla y de fácil comprensión (https://www.youtube.com/channel/UCsXbYZd-Yqbws1m8QwH_fgA). A su vez se compone de:

- Keras: Específica para Redes Neuronales
- EagerExecution: Evalúa las opciones existentes de forma inmediata y sin la necesidad de crear grafos.
- ImportingData: Permite la construcción de tuberías de entrada complejas a partir de fragmentos simples.
- Estimators: Los estimadores encapsulan acciones, tales como formación, evaluación, predicción y el exportar para mostrar (construir grafos, cargar datos, inicializar variables, etc.)

6.3.2 Anaconda

Pese a que hay un amplio abanico de posibilidades a la hora de trabajar con Python, la razón por la que se seleccionó Anaconda para trabajar, fue sencillamente

porque era lo más recomendado en los diversos foros y blogs consultados. Otras opciones igual de válidas habrían sido la instalación de un intérprete de Python en la máquina de trabajo habitual o en la máquina de Windows.

Todo el trabajo realizado en este documento a través del lenguaje Python se ha realizado con la plataforma Anaconda, que es de código abierto y, por tanto, gratuita. Una vez instalada, ya se puede empezar a trabajar con el lenguaje Python. La plataforma Anaconda se puede descargar a través del siguiente link: https://www.anaconda.com/distribution/

La apariencia de Anaconda, una vez instalada, será la siguiente:

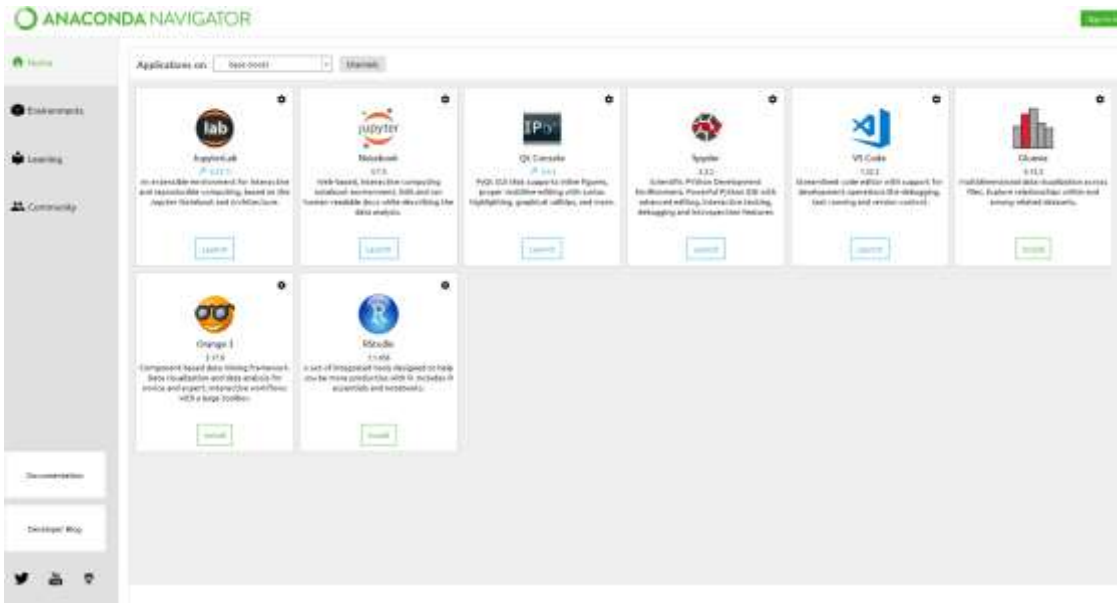


ILUSTRACIÓN 36 ANACONDA (ELABORACIÓN PROPIA)

Y será a través del navegador Jupyter, donde se realizarán las labores propias de programación en Python:

Los datos transformados y limpios se almacenan dentro de un dataframe o dataset, que se podría decir que es un *array* que contiene los datos procesados, aunque también se podrían introducir semiprocesados o en bruto, dependiendo del caso.

Para poder crear el dataframe, existen dos opciones: trabajar a partir de un .csv; en el que los datos se dividen con comas y es válido para todos los programas de tratamiento matemático o hacerlo a través de un documento de texto tipo .txt.

ILUSTRACIÓN 37 JUPYTER (ELABORACIÓN PROPIA)

6.3.3 Problemas

El lenguaje está sobradamente explicado y posee muchísimas soluciones prácticas dentro de la red. El principal problema que se ha encontrado en el trabajo con Python es la falta de memoria. Se entiende que el problema no es del lenguaje, sino de la falta de potencia del dispositivo desde el que se ha trabajado, sin embargo, es un punto a tener en consideración, puesto que, a la hora de mostrar las gráficas, el navegador se bloqueaba, provocando multitud de re-arranques hasta conseguir los objetivos que se pretendían.

A parte del problema descrito, el lenguaje de Python es bastante intuitivo y fácil de manejo, además de que, aunque se comentan pequeños errores de sintaxis, el compilador es flexible y no presenta problemática alguna.

7. Presupuestación

Dentro del plan financiero de cada empresa, los presupuestos y su seguimiento suponen una de las decisiones más importantes. La elaboración de presupuestos fiables es clave, para ello se deberá estimar ingresos, realizar una predicción de gastos y una asignación de los recursos necesarios.

De acuerdo con la RAE, presupuestar es "formar el cómputo de los gastos o ingresos, o de ambas cosas que resultan de un negocio público o privado". Básicamente de lo que tratará este apartado es de estudiar el impacto económico que supondría la implantación de ML para el departamento de Mediación, dentro del BSS de la empresa CDS.

Una buena planificación presupuestaria puede determinar la diferencia entre que un proyecto real, con cliente/s real/es, tenga éxito o fracase estrepitosamente. Por la parte que concierne a los empleados, a la hora de plantear un presupuesto de estas características se plantearán tres posibilidades:

- Formación externa para la plantilla basada en ML
- Contratación de servicios de consultoría
- Realización de un curso en la propia sede de la empresa

Si el proyecto se ciñera fielmente a un presupuesto, habría que valorar también los materiales que se necesitarán para la ejecución del mismo, partiendo de que la herramienta que proporciona la información con la que se trabajará (eIUM) está montada y en funcionamiento, habría que calcular también una partida para el coste hardware necesario para que la implementación funcione; además de gastos derivados de luz e instalaciones, conexión a internet, etc.

Un punto a tener en cuenta sería el momento de realización del proyecto. Sin mucho preámbulo o análisis previo, se podría decir que el mejor momento para el estudio e implantación del proyecto de ML aplicado a la lectura de logs en el departamento de Mediación, serían los meses de verano, dado que se produce un descenso en el volumen de trabajo, coincidiendo con las vacaciones de verano.

7.1.1 Formación externa para la plantilla basada en ML

La formación para la plantilla basada en la tecnología de ML, pasaría por dos supuestos: el primero es aquel en el que se contrataría desde cero un grupo especializado en ML, para montar una unidad dedicada a ello; y el segundo, sería a partir de la plantilla que se tiene, ofrecer la posibilidad de una titulación en ML, fuera de su tiempo de trabajo, cuyo coste sería asumido por la propia compañía.

La primera parte que se propone, que consiste en montar una unidad a partir de contrataciones nuevas, no se estudiará debido a la complejidad y a la falta de conocimientos en el área de Recursos Humanos, puesto que son muchas las variables que estudiar en un perfil tan concreto.

La segunda parte que se propone, que consiste en formar a profesionales dentro de esta tecnología, que actualmente forman parte de la plantilla, se estudiará, realizando una pequeña comparativa entre los estudios disponibles dentro del sector. La ventaja que supone que se formen estos profesionales en la materia de ML, es que ya conocen todos los entresijos del trabajo y del funcionamiento del área de Mediación, por lo tanto, es más sencillo perfilar, acotar y definir los límites y la aplicabilidad de la tecnología, sin precisar de un gran apoyo externo.

Aunque existen multitud de empresas que ofrecen información no regalada de Machine Learning, este estudio se centrará en aquella que lleva asociada algún tipo de titulación regalada oficial; por razones obvias. A continuación, el listado de universidades que se han consultado, que poseen una titulación oficial, con el presupuesto solicitado, número de ETCs y duración:

TABLA 7 FORMACIÓN EXTERIOR PARA LA PLANTILLA

Centro	Titulación	Nº ETCs	Duración	Precio
UNIR: Universidad Internacional de La Rioja	Máster en Inteligencia Artificial	60	1 curso académico	6.960€
Google Cloud Certified	Professional Data Engineer Associate Cloud Engineer Professional Cloud Architect Professional Cloud Security Engineer Profesional Cloud Developer Profesional Cloud Network Engineer	-	Variable, basada en la experiencia profesional en las distintas herramientas de google cloud	Variable, a partir de los 125€ y hasta los 300€
UNED: Universidad Nacional de Educación a Distancia	Programa de Postgrado y desarrollo profesional con estructura modular: Data Science	60	7 meses	1.680€
Campus BigData con certificación de la Universidad Católica de Murcia	Master en Machine & Deep Learning e Inteligencia Artificial	-	1 año académico	3.190€
Universidad Internacional de Valencia	Máster en Inteligencia Artificial	60	1 año académico	5.800€

Dados los conocimientos de la plantilla sobre la herramienta eIUM y el funcionamiento del departamento de Mediación, se estima que con dos-cuatro personas que se formen en la materia sería suficiente para ponerla en marcha en el periodo inmediatamente posterior.

7.1.2 Contratación de servicios de consultoría

Para la realización de este punto, se ha procedido directamente al envío de emails a diversas empresas dedicadas a la implantación de Machine Learning. En el email se exponía a grandes rasgos el proyecto y se solicitaba un presupuesto tanto económico como en duración. Muy amablemente, algunas de las empresas consultadas, respondieron a ese email; obteniendo los siguientes datos:

TABLA 8 CONSULTORÍAS CONSULTADAS

Compañía	Presupuesto	Jornadas
 ILUSTRACIÓN 38 LOGOTIPO CONSULTORÍA DECIDE	-	45 jornadas de un Data Science + 10 jornadas de un Arquitecto
 ILUSTRACIÓN 39 LOGOTIPO CONSULTORÍA ACUILAE	12.000 euros y 2 meses y medio de trabajo	-

Se parte de la base de que el equipo tiene una dilatada experiencia en la implantación de ML. Se observa en los datos obtenidos que la información es similar: 45 jornadas propuestas por *Decide*, son más o menos dos meses de trabajo; mientras que *Acuilae* propone dos meses y medio. A continuación, se realizará una escenificación con los datos obtenidos:

Se estima una duración de 2 meses y medio y un total de dos personas implicadas en el proyecto: un Data Science y un Arquitecto.

Línea del tiempo:

TABLA 9 LÍNEA DEL TIEMPO TRABAJO DE CONSULTORÍA

Rol	Mes 1	Mes 2	Mes 3
Data Science	22 jornadas * 8h	22 jornadas *8h	5 jornadas * 8h
Arquitecto	10 jornadas * 8h	-	-

Costes humanos:

TABLA 10 COSTES HUMANOS CONSULTORÍA

Rol	Coste/hora	Número de horas	Total
Data Science	24€	49*8	(9.408)x1persona
Arquitecto	45€	10*8	(3.600)x1persona

=**13.008**€ totales en costes humanos.

7.1.3 Realización de un curso en la propia empresa

Se solicitó información a una empresa formadora proporcionando ésta un presupuesto de 7.600€ para un curso de duración de 10 días dentro de la sede de CDS en León.

La formación se impartiría en una de las salas del centro, habilitada a tales efectos, y que tiene cabida para hasta 15 personas. Por parte de la empresa facilitadora del curso no habría inconveniente en que la formación fuera dirigida hacia las 15 personas que caben holgadamente en la sala.

7.1.4 Conclusiones sobre los presupuestos

En primer lugar, se analizará qué opción es más económica:

TABLA 11 CONCLUSIONES PRESUPUESTOS OPCIONES GENERALES

Opción	Coste	Coste en tiempo
Formación externa	(1.400-6.969€)*3=(4.200-20.907)	6-12 meses + implantación
Servicios de consultoría	12.000-13.008€	2-2.5 meses
Curso en la propia sede	5.000-7.000€ para grupos de hasta 15 personas	10 días + implantación

Dadas las opciones planteadas, se valorarán pros y contras de cada una de ellas:

TABLA 12 RESUMEN VENTAJAS/INCONVENIENTES DE LAS OPCIONES PRESUPUESTARIAS

Metodología	Ventajas	Inconvenientes
Formación externa	Escogiendo “bien”, es la opción más económica. La formación se realizaría fuera de las horas de trabajo. Se crearía un equipo de profesionales con un perfil determinado y que conocen el sistema a la perfección, por lo tanto se podría amortizar más la inversión ampliando las aplicaciones de los conocimientos adquiridos	La opción más económica de las que se ofrecen, sólo estaría disponible para empleados con una titulación universitaria oficial. Se demora mucho en el tiempo. El empleado podría abandonar la empresa. Para que el empleado realice la implantación, deberá abandonar sus tareas habituales
Servicios de consultoría	Es breve en el tiempo. Posee garantías de funcionamiento	No es la opción más económica Los beneficios recaen únicamente sobre la empresa contratada No hay titulación o incentivo para los empleados
Curso en la propia sede	El curso únicamente incluirá las partes que se	No es una titulación oficial No es la opción más

	consideren interesantes para el uso que se le va a dar. Es breve en el tiempo Se crearía un equipo de profesionales con un perfil determinado y que conocen el sistema a la perfección, por lo tanto se podría amortizar más la inversión ampliando las aplicaciones de los conocimientos adquiridos	económica, aunque si se amplía a 15 empleados, el coste de formación/empleado disminuiría considerablemente. Una el empleado es libre de abandonar la empresa Para que el empleado realice la implantación, deberá abandonar sus tareas habituales

La formación externa supone una serie de ventajas para los trabajadores, la más obvia es que esa formación corre a cargo de la empresa y la titulación sería para el trabajador. Tal vez habría que añadir algún tipo de cláusula a la hora de aceptar la formación, a fin de que el empleado no pueda abandonar la empresa hasta pasado un tiempo una vez finalizada la titulación. Con la cláusula que se propone, en caso de que el empleado abandonase la empresa en un plazo inferior al que se firme, deberá asumir la parte proporcional de su formación. Esta opción no se estudiará más allá, ya que supone un desembolso importante, la reestructuración del trabajo y una implantación que se podría dilatar en el tiempo, debido a la falta de experiencia de los trabajadores en el ámbito.

La formación dentro de la sede podría ser una opción mucho más interesante. Los conocimientos que se adquieren son 'a la carta', es decir que, dentro de lo amplio del Big Data, únicamente se estudiarían aquellas partes relevantes para el trabajo que se va a realizar. La formación sería muy breve y a partir de ahí se trabajaría en la implantación. No se podría afirmar a ciencia cierta cuánto tiempo tardaría la implantación, se estima que al no tratarse de profesionales en el sector se podría dilatar más en el tiempo que los dos meses o dos meses y medio que proporcionaron las empresas consultadas.

La opción de dar formación dentro de la empresa a empleados que actualmente trabajan a diario en Mediación supone la grandísima ventaja de que los trabajadores conocen perfectamente la herramienta de trabajo, lógica de negocio y procedimientos específicos. Además, se podrían estudiar propuestas para estudios o proyectos futuros dentro del entorno de trabajo que añadirían valor al servicio.

Si se descartase la opción de formación para los empleados, la única vía que queda disponible es la de contratar los servicios de una consultoría externa. Aunque la cuantía a la que ascienden estos servicios supone una inversión, los trabajadores no verían alteradas sus tareas y estaría acotada a un máximo de dos meses y medio; por lo tanto, a razón de comodidad, no-reestructuración de labores y diferencia de precio, la opción más económica es contratar una compañía externa.

Si se tiene en consideración que se integrarán dentro de la compañía dos personas externas, esto no supondría gastos extra; puesto que, en las oficinas de la ciudad de León, desde donde se plantea este estudio, hay sitio disponible para que se

instalen dos personas, con conexión a internet potente y segura y salas adecuadas para la realización de reuniones en caso de ser necesario. Es por esta razón que no se considerarán gastos de alquiler de oficina, de luz o agua o conexión a internet. Lo único que se debería añadir al presupuesto son las licencias necesarias y como mucho, nuevos ordenadores para llevar a cabo el trabajo:

TABLA 13 PRESUPUESTOS MATERIALES

Material	Precio
Ordenador Portátil HP Probook 640-G4 o superiores	(720€)*2=1.440€

Con todo lo expuesto, el presupuesto total sería el siguiente, aunque quizás habría que añadir un extra para licencias software:

Coste estimado de la implantación: 13.008€

Materiales: 1.440€

TOTAL: **14.448€**

7.1.4.1 Conclusión final:

Partiendo de que CDS es una empresa que cuida a sus empleados y desea conservarlos a lo largo del tiempo, la opción que escogería sería la de formar a los empleados dentro de la propia sede. No es la opción más rápida ni la más económica, sino un punto medio entre las tres opciones disponibles.

La razón en la que baso el criterio expuesto no es otra que la continuidad: ofreciendo proyectos interesantes, se atrae y conserva la excelencia, y, a fin de cuentas, es a lo que se aspira: los empleados contentos trabajan mejor y son más eficientes. Como comenté, el hecho de que los empleados a quienes se ofrece la formación son personas que controlan perfectamente la dinámica de trabajo, el entorno y los entresijos del eIUM podrán aportar ideas sobre cómo mejorar el sistema, el alarmado y la prevención de posibles caídas.

8. Testeo ejemplificado de Machine Learning en Mediación de Telefonía

Habiendo visto cómo funciona ML y todas las ventajas que puede suponer de cara a la obtención de información por parte de los datos, la implantación de ML dentro del BSS/OSS, traería consigo muchas opciones de mejora para el sistema. Sin embargo, la complejidad en el proceso de implantación, hace que antes de dar el paso, la empresa realice un estudio de impacto, tanto económico como en lo que se refiere a la gestión de tiempos.

En este último apartado se tratará de ejemplificar grosso modo, cómo se podría enfocar la implantación del ML dentro del BSS/OSS, en Mediación de Telefonía y a través de la lectura de logs.

Puesto que simular la implantación de ML en un área como la Mediación de Telefonía requeriría conocimientos de experto; lo que se tratará en este apartado serán los pasos a seguir, ejemplificando siempre que se pueda, con los datos que se poseen.

Como bien se especificó en el apartado (*6. Valoración de la implantación de la tecnología Machine Learning en Mediación de Telefonía*), los pasos a seguir a la hora de implantar ML dentro de cualquier sistema son los siguientes:

1. Definición de los objetivos que se pretenden conseguir
2. Selección de los datos aplicables
3. Elección de indicadores de éxito
4. Creación del data mining
5. Estudio de algoritmos y modelos
6. Implantación

Además de todo lo expuesto hasta este momento, dentro de la ejemplificación se expondrán distintas opciones de cara la presupuestación del proyecto, así como las conclusiones que se derivan del estudio realizado.

8.1 Definición de los objetivos

Como se ha explicado durante todo este documento, el objetivo principal que se persigue en este estudio es la instalación de un sistema que funcione con ML y que sea capaz de predecir los errores predecibles de los logs de Mediación antes de que éstos hagan abortar colectores o el sistema completo.

Aunque el sistema eIUM es estable, se encuentra correctamente alarmado y tiene muchos ‘ojos’ chequeando su correcto funcionamiento, el sistema no es infalible.

A la hora de implantar ML, el sistema de archivado de la mediación de telefonía posee una extensa colección de datos de log, de hasta 68 millones de líneas, que se guardan durante un periodo de dos meses.

Puesto que el objetivo de generar una aplicación en ML que sea capaz de predecir errores en los logs de Mediación es demasiado ambicioso y para tener éxito se

requeriría de un grupo de expertos que se dediquen profesionalmente a esta tecnología; a continuación, se definirán los objetivos que se pretenden conseguir para este estudio:

- De cara a los datos:
    1. Conseguir una reducción de los mismos de al menos 70%-80%, que grosso modo, podrían ser valores aceptables tras eliminar aquellos que son redundantes o que no aportan información necesaria de cara a la detección de errores.
    2. Normalizarlos en una matriz de datos para poder trabajar con ella en programas matemáticos
- De cara a la definición del éxito
    1. Determinar qué metodología o metodologías serían las más adecuadas como medida de éxito dentro de los datos de entrenamiento.
    2. Determinar si la medida de éxito que aporta el lenguaje Python refleja la realidad, de cara a los datos que se disponen.
- De cara al algoritmo ML:
    1. A partir de los conocimientos que se tienen de los algoritmos, determinar cuál o cuáles podrían ser los más útiles a la hora de cumplir con los objetivos fijados.

8.2 Selección de los datos aplicables

Dentro del archivado de datos de una de las máquinas de Mediación de la teleco Vodafone se encuentran más de tres mil ficheros de log. Estos ficheros de log contienen información que va desde los propios procesos de colección y distribución de datos, hasta qué colectores modifican sus configuraciones, cuáles son parados o arrancados, ect. De todos ficheros de log, se ha seleccionado un tipo concreto: llamadas de voz y sms de una de las centrales que más tráfico recibe: Ericsson.

El registro total de ficheros de log es inabarcable en un estudio de estas características, así que se centrará en aquellos ficheros de log que presenten problemática, es decir que hayan sufrido algún tipo de caída del proceso, por la razón que sea.

Las caídas en los colectores se pueden dar por múltiples motivos, algunas se pueden ver con cierto margen de tiempo, especialmente aquellas que ralentizan el sistema hasta hacer caer los colectores, como por ejemplo un proceso que se queda atascado y no es capaz de terminar, pero continúa consumiendo recursos. En el otro lado se encuentran las caídas de los colectores que no se podrían predecir, como por ejemplo el procesamiento de un fichero corrupto, que puede hacer que el colector aborte de manera repentina y sea incapaz de rearrancarse hasta que la intervención humana extrae el fichero de forma manual.

Se podrían definir algunas de las causas, debido a las cuales se podría caer un colector, con un margen suficiente de tiempo:

- Máquina al borde del llenado: a través de la experiencia con este tipo de máquinas, se ha visto que cuando su espacio ocupado ronda el 95%, comienzan a darse una serie de errores que se traducen en la ralentización de los procesos. La ralentización es ‘la pescadilla que se muerde la cola’, puesto que, a mayor

latencia, se produce más latencia, hasta que, si no se pone remedio, provoca la caída total del sistema

- Errores de conexión a la Base de Datos (BBDD a partir de ahora): tan sólo se compone de unas pocas trazas antes de que el colector se caiga, lo cual supone unos segundos de antelación. En ocasiones la caída total de la BBDD va precedida de pequeños cortes a tener en consideración.
- Aumento repentino del volumen de datos: en días como por ejemplo Nochebuena o Navidad, se produce un importante incremento en las llamadas telefónicas, no es de extrañar que justo esos días las máquinas funcionen al límite de sus capacidades, puesto que el flujo de datos es mayor, ralentizando los procesos y pudiendo provocar caídas.

Seguidamente, se analizarán si las variables objeto de este estudio son discretas o continuas, con las razones que implica cada una de ellas.

8.2.1 Variable discreta vs variable continua

En las siguientes líneas se darán las razones por las que se ha seleccionado la variable continua en lugar de la discreta para los valores que se pueden dar dentro de los

```
head -5  ER4RE01A.logOLD.20190205053037 ; tail -5
ER4RE01A.logOLD.20190205053037

02/05/2019 05:05:43.215 CET main (BusinessRule) INFORMATIVE: CDR
filtered

02/05/2019 05:05:43.224 CET main (BusinessRule) INFORMATIVE: CDR
filtered

02/05/2019 05:05:43.257 CET main (BusinessRule) INFORMATIVE: CDR
filtered

02/05/2019 05:05:43.295 CET main (07.003.11) INFORMATIVE:
Aggregated 8 NMEs at a rate of 14 NMEs/sec

02/05/2019 05:05:43.366 CET main (07.018.01) INFORMATIVE: Scheme S1
tree contained 0 nmes, 0 flushed, 0 remain.

02/05/2019 05:30:21.xxx CET DistributorMove.sh - invalid_reproc
(24914) INFO: Distribution Starting for file
'ER4REPROC:MBC201618405:ER4RE01A:INVALID_REPROC:977919'

02/05/2019 05:30:21.451 CET main (06.024.05) INFORMATIVE: command
'/home/siuvoice/release/scripts/handlers/DistributorMove.ksh
ericssonR4 fdtvm_aida
/var/opt/SIUVOIa/config/handlers/DistributorMove.cfg' completed
with status: 0

02/05/2019 05:30:21.453 CET main (06.024.04) INFORMATIVE: Datastore
waiting for command,
'/home/siuvoice/release/scripts/handlers/DistributorMove.ksh
ericssonR4 hot112
/var/opt/SIUVOIa/config/handlers/DistributorMove.cfg', to complete

02/05/2019 05:30:21.xxx CET DistributorMove.sh - hot112 (24976)
```

logs de Mediación. Se pondrá por ejemplo un log de un colector (ER4RE01A) perteneciente al flujo de Ericsson (llamadas de voz y sms) del mes de febrero de 2019:

Se podría decir que las líneas de los logs de mediación Vodafone, no se lanzan de forma aleatoria, sino que se rigen por una serie de normas. Dentro de estas reglas se podrían diferenciar, por ejemplo:

1. CRITICAL: Informa sobre el aborto de un colector que sucede de forma abrupta: "ER401A.logOLD.20190216122204:02/16/2019 09:50:50.662 CET main (03.003.18) CRITICAL: Collector ER401A exiting due to errors."
2. ACCOUNTING: Informa sobre una parada sobre el colector: "02/14/2019 07:58:28.883 CET Thread-1 (02.010.24) ACCOUNTING: Process ER404A(40952) Exiting By Request." En este caso concreto, se ha lanzado el proceso de parada del colector, de forma manual y voluntaria, fuera cual fuera la razón de la misma.
3. WARNING: Avisa sobre una acción inesperada que podría suponer o no el aborto del colector: "01/25/2019 20:43:46.408 CET main (06.016.07) WARNING: File /var/opt/SIUVOIa/work/ready/ER401A/READY_reproc/ERMBC201:MBC201752858:ER401A:INVALID_REPROC:249327 from scheme S1inv_rep was not removed during aging."
4. INFORMATIVE: Avisa sobre el lanzamiento de determinados procesos habituales dentro de la colección, entre ellos:
   - lanzamiento de procesos para el movimiento de ficheros: "command '/home/siuvoice/release/scripts/handlers/DistributorMove.ksh ericssonR4 ono /var/opt/SIUVOIa/config/handlers/DistributorMove.cfg' completed with status: 0"
   - realización de cierre de ficheros, como por ejemplo: "02/05/2019 05:05:44.554 CET main (07.003.19) INFORMATIVE: Flushed 19 NMEs at a rate of 0 NMEs/sec"
   - movimiento de ficheros entre directorios del propio sistema: "02/05/2019 05:05:43.625 CET main (06.016.04) INFORMATIVE: Wrote NME file: /var/opt/SIUVOIa/work/ready/ER4RE01A/READY_invalid/ER4REPROC:MBC201616953:ER4RE01A:INVALID:433621 for scheme S1invalid."
   - velocidad en la lectura o envío de ficheros "02/05/2019 05:06:41.842 CET main (07.003.11) INFORMATIVE: Aggregated 6 NMEs at a rate of 10 NMEs/sec"
5. INFO: Monitoriza el buen hacer de uno de los procesos que mueven ficheros dentro del sistema. Muy útil a la hora de localizar archivos.

Si a cada una de las categorías se le asigna un 'peso' o valor numérico, podríamos asemejarlas, por ejemplo, a las caras de un dado: en un intervalo de valores en el que:

1. CRITICAL: 4
2. ACCOUNTING: 3
3. WARNING: 2
4. INFORMATIVE: 1
5. INFO: 0

Los logs únicamente podrían tomar valores entre los (0-4) asignados. Por lo tanto y sin lugar a dudas, estaríamos hablando de un tipo de variable: discreta.

8.2.2 Aprendizaje supervisado vs aprendizaje no supervisado

A lo largo de este trabajo se ha ido viendo la diferencia entre el aprendizaje supervisado y no supervisado. Puesto que en el caso de los logs se dispone de un histórico más o menos amplio de datos de entrada, con sus respectivas salidas; se podría decir que se trataría de un aprendizaje supervisado.

8.2.3 Conclusiones para las variables y el aprendizaje

Dadas las características de los datos que se obtienen en los logs, se trataría de un aprendizaje supervisado, con variables discretas; y por lo tanto, y basando el estudio en el gráfico (Ilustración 26 Tipos de algoritmos ML) habría que buscar entre los algoritmos de clasificación.

Concretamente el análisis se realizará sobre los ficheros de log de Ericsson para una de las máquinas del sistema de mediación de la compañía Vodafone, que contienen datos sobre llamadas telefónicas y envíos de sms. El total de líneas que se dispone es de: 68.060.962 que se corresponden con los logs generados entre los días: 01/28/2019 y 02/28/2019.

```
[2019-03-05 17:04.56]  /drives/C/Users/50026713/Desktop/archive
[50026713.HCESV5CG824639Q] ➤ cat ER*[0-9] | wc -l
68060962
```

ILUSTRACIÓN 40 LÍNEAS TOTALES DISPONIBLES LOGS ERICSSON (MOBA) (ELABORACIÓN PROPIA)

Durante el tiempo transcurrido en el histórico de logs, se poseen de entre los poco más de sesenta y ocho millones de líneas, un total de 692 se identifican con la traza CRITICAL que provoca el aborto del colector. Lo que se tratará de encontrar en los archivos históricos de los logs, es si existe de relación entre las trazas que tiene el log y el error que provoca la caída del colector y quizás se puedan predecir futuras caídas en el sistema.

```
[2019-03-05 18:03.20]  /drives/C/Users/50026713/Desktop/archive
[50026713.HCESV5CG824639Q] ➤ grep CRITICAL ER*[0-9] | wc -l
692
```

ILUSTRACIÓN 41 CRITICAL TOTALES DISPONIBLES FLUJO ERICSSON (MOBA) (ELABORACIÓN PROPIA)

8.3 Elección de los indicadores de éxito

Puesto que ya se ha determinado que se trata de un algoritmo de clasificación, se aplicarán los indicadores de éxito de esta metodología que se describieron en el punto *Elección de indicadores de éxito*, como la matriz de confusión. Aunque también se utilizarán los indicadores que posee el lenguaje de Python, que resultan igual de útiles.

8.4 Creación del Data Mining

La creación del data mining, se puede plantear de muchas formas, en esta ejemplificación se verán y estudiarán varias, de forma que se podrá observar el comportamiento de diferentes algoritmos a la hora de introducir los datos con las distintas *limpiezas*.

8.4.1 Limpieza de datos

Se posee una amplia variedad de ficheros de logs, correspondientes al archivado de Vodafone. No toda la información que poseen estos logs es relevante, así que tal y como se ha comentado en el apartado *(6.1.4.1Limpieza de datos*), se procede a su limpieza.

8.4.1.1 Selección de "líneas sobrantes"

Dados que los datos que se encuentran en los logs no son sencillos de parametrizar en variables, puesto que no son medibles como 'el precio de una vivienda' o 'el tamaño de un pulmón', sino que se trata de valorar cuál de la información que aportan es útil.

En base a la experiencia en la lectura y análisis de logs, se realizará la siguiente reducción de datos.

Se toma como ejemplo, la traza de CRITICAL perteneciente al log ER401A.logOLD.20190216122204, formado por un total de 54.037 líneas:

```
[2019-03-07 23:03.43]  /drives/C/Users/50026713/Desktop/archive
[50026713.HCESV5CG8246390] ➤ cat ER401A.logOLD.20190216122204 | wc -l
54037
```

ILUSTRACIÓN 42 NÚMERO DE LÍNEAS LOG (MOBA) (ELABORACIÓN PROPIA)

El *CRITICAL* que se menciona pertenece a la siguiente línea: "`02/16/2019 09:50:50.662 CET main (03.003.18) CRITICAL: Collector ER401A exiting due to errors`".

Antes de entrar de lleno en el análisis de las razones por las cuales se produce el *CRITICAL* que provocó el aborto del colector, será necesaria la eliminación de determinadas trazas que no resultan relevantes. La experiencia y los conocimientos del sistema en el que se está trabajando una posible implantación, son claves a la hora de realizar esta detección en la irrelevancia de los datos. El criterio a seguir ha sido el siguiente: detección de trazas que en sí mismas no generan errores, es decir que el hecho de que no aparezcan, no va a generar que el sistema se paralice. A continuación, se realizará un análisis bastante exhaustivo sobre los tipos de líneas de logs que se pueden encontrar. En primer lugar, se descartarán las siguientes:

1. *02/16/2019 08:55:44.xxx CET DistributorMove.sh - dbintra (55992) INFO: file 'ERMBC201:MBC201834340:ER401A:DBINTRA:670864' moved to '/var/opt/SIUVOIa/work/input/dbcall/dbcall3/ds4/ERMBC201:MBC2018343*

40:ER401A:DBINTRA:670864' ➔ Indica un movimiento entre directorios. No resulta relevante.

```
[2019-03-07 23:10.36]  /drives/C/Users/50026713/Desktop/archive
[50026713.HCESV5CG824639Q] ▸ grep "DistributorMove.sh -" ER401A.logOLD.20190216122204 |  grep "moved to" | wc -l
2072
```

ILUSTRACIÓN 43 TRAZA IRRELEVANTE 1 (MOBA) (ELABORACIÓN PROPIA)

2. "*02/16/2019 08:55:44.256 CET main (06.024.04) INFORMATIVE: Datastore waiting for command, '/home/siuvoice/release/scripts/handlers/DistributorMove.ksh ericssonR4 rcf /var/opt/SIUVOIa/config/handlers/DistributorMove.cfg', to complete"* ➔ Espera un fichero para completer un proceso, en caso de error. Se descartarán.

```
[2019-03-07 23:07.10]  /drives/C/Users/50026713/Desktop/archive
[50026713.HCESV5CG824639Q] ▸ grep "INFORMATIVE: Datastore waiting for command"  ER401A.logOLD.20190216122204 | wc -l
3775
```

ILUSTRACIÓN 44 TRAZA IRRELEVANTE 2 (MOBA) (ELABORACIÓN PROPIA)

3. "*02/16/2019 08:55:44.xxx CET DistributorMove.sh - rcf (56034) INFO: Mandatory parameter DIR_DESTINATION defined with value '/var/opt/SIUVOIa/archive/output/o%3%/rcf/dia16/hora08"* ➔ Indica que el fichero ha sido guardado en el archivado de forma correcta. El archivado no generará problemas salvo que la máquina esté llena, cosa que sucede muy rara vez, debido a que se encuentra debidamente alarmada; por lo tanto esta línea será descartada.

```
[2019-03-07 23:07.26]  /drives/C/Users/50026713/Desktop/archive
[50026713.HCESV5CG824639Q] ▸ grep "Mandatory parameter DIR_DESTINATION" ER401A.logOLD.20190216122204 | wc -l
2363
```

ILUSTRACIÓN 45 TRAZA IRRELEVANTE 3 (MOBA) (ELABORACIÓN PROPIA)

4. *"02/16/2019 00:46:35.xxx CET DistributorMove.sh - ono1 (134277) INFO: There are no files to distribute"* ➔ Indica que para una salida concreta, para un tipo de datos de envío, no existen esos datos y por lo tanto no se realizará el envío.

```
[2019-03-07 23:15.58]  /drives/C/Users/50026713/Desktop/archive
[50026713.HCESV5CG824639Q] ▸ grep "There are no files to distribute" ER401A.logOLD.20190216122204 | grep DistributorMove.sh | wc -l
2186
```

ILUSTRACIÓN 46 TRAZA IRRELEVANTE 4 (MOBA) (ELABORACIÓN PROPIA)

5. *"02/16/2019 08:00:19.065 CET main (06.016.04) INFORMATIVE: Wrote NME file: /var/opt/SIUVOIa/work/ready/ER401A/READY_dbintra/ERMBC201:MBC201834274:ER401A:DBINTRA:176281 for scheme S1dbintra."* ➔ Su significado es que se ha escrito y cerrado un fichero para su possible envío. No es relevante.

```
[2019-03-07 23:44.18]  /drives/C/Users/50026713/Desktop/archive
[50026713.HCESV5CG824639Q] ▸ grep "Wrote NME file"  ER401A.logOLD.20190216122204 | wc -l
2190
```

ILUSTRACIÓN 47 TRAZA IRRELEVANTE 5 (MOBA) (ELABORACIÓN PROPIA)

6. "*02/16/2019 02:09:01.xxx CET DistributorMove.sh - bdt (135841) INFO: Optional parameter ENVIO defined with value 'SELECTED'"* ➔ Informa sobre un parámetro definido en una configuración, no es relevante.

```
[2019-03-07 23:40.31]  /drives/C/Users/50026713/Desktop/archive
[50026713.HCESV5CG824639Q] ➤ grep "Optional parameter ENVIO defined with value" ER401A.logOLD.20190216122204 | wc -l
341
```

ILUSTRACIÓN 48 TRAZA IRRELEVANTE 6 (MOBA) (ELABORACIÓN PROPIA)

7. *"02/16/2019 06:21:32.xxx CET DistributorMove.sh - bdt (5166) INFO: Branching execution for file /var/opt/SIUVOIa/work/ready/ER401A/READY_bdt/ERMBC201:MBC201834157:ER401A:BDT_ESPAT:286028"* ➔ Informa sobre un movimiento de ficheros interno. No es relevante.

```
[2019-03-07 23:50.15]  /drives/C/Users/50026713/Desktop/archive
[50026713.HCESV5CG824639Q] ➤ grep " Branching execution "  ER401A.logOLD.20190216122204 | wc -l
332
```

ILUSTRACIÓN 49 TRAZA IRRELEVANTE 7 (MOBA) (ELABORACIÓN PROPIA)

8. *"02/16/2019 01:38:47.302 CET main (06.005.01) INFORMATIVE: No value for GSMipADR. Using default"* ➔ Indica que falta un parámetro dentro de una configuración y por lo tanto se utilizará otro por defecto. No resulta relevante.

```
[2019-03-07 23:57.39]  /drives/C/Users/50026713/Desktop/archive
[50026713.HCESV5CG824639Q] ➤ grep "Using default" ER401A.logOLD.20190216122204 | wc -l
56
```

ILUSTRACIÓN 50 TRAZA IRRELEVANTE 8 (MOBA) (ELABORACIÓN PROPIA)

9. "*02/16/2019 00:50:37.xxx CET DistributorMove.sh - vms (109149) INFO: Insertion in IUM_VOZ_INTERNAL_AUDIT committed*" ➔ Indica que el fichero de auditoria ha sido actualizado. No resulta relevante.

```
[2019-03-08 10:47.47]  /drives/C/Users/50026713/Desktop/archive
[50026713.HCESV5CG824639Q] ➤ grep "Insertion in IUM_VOZ_INTERNAL_AUDIT"  ER401A.logOLD.20190216122204 | wc -l
86
```

ILUSTRACIÓN 51 TRAZA IRRELEVANTE 9 (MOBA) (ELABORACIÓN PROPIA)

10. *"02/16/2019 01:21:48.xxx CET DistributorMove.sh - fintra (140086) INFO: file 'ERMBC101:MBC101467350:ER401A:FINTRA:451715' copied to '/var/opt/SIUVOIa/work/input/intra/intra1/ds_fintra2/ERMBC101:MBC101467350:ER401A:FINTRA:451715'"* ➔ Movimiento entre ficheros dentro de la propia máquina. No resulta relevante.

```
[2019-03-08 10:51.10]  /drives/C/Users/50026713/Desktop/archive
[50026713.HCESV5CG824639Q] ➤ grep "copied to" ER401A.logOLD.20190216122204 | wc -l
173
```

ILUSTRACIÓN 52 TRAZA IRRELEVANTE 10 (MOBA) (ELABORACIÓN PROPIA)

11. "*02/16/2019 01:21:49.xxx CET DistributorMove.sh - unbill (140373) INFO: File 'ERMBC101_MBC101467350_ER401A_613250' compressed in directory '/var/opt/SIUVOIa/archive/output/o3/unbill/dia16/hora01/'"* ➔ Compresión satisfactoria de un fichero dentro del archivado. Irrelevante.

```
[2019-03-08 10:55.59]  /drives/C/Users/50026713/Desktop/archive
[50026713.HCESV5CG824639Q] ▸ grep "compressed in directory" ER401A.logOLD.20190216122204 | wc -l
203
```

ILUSTRACIÓN 53 TRAZA IRRELEVANTE 11 (MOBA) (ELABORACIÓN PROPIA)

12. "*02/16/2019 10:29:29.054 CET main (39.009.03) INFORMATIVE: Detected Plugin: com.hp.usage.fst.datastore version 5.0.0*" ➔ Se generan al inicio de los colectores, indicant versiones de elementos que utilizan. No es relevante.

```
[2019-03-08 11:07.07]  /drives/C/Users/50026713/Desktop/archive
[50026713.HCESV5CG824639Q] ▸ grep version ER401A.logOLD.20190216122204 | wc -l
7359
```

ILUSTRACIÓN 54 TRAZA IRRELEVANTE 12 (MOBA) (ELABORACIÓN PROPIA)

13. Otras trazas informativas e irrelevantes serían aquellas que tienen que ver con la versión de java que se utiliza, el sistema operativo, la versión de sistema operativo, definición de path, usuario, zona horaria…

```
[2019-03-08 11:13.25]  /drives/C/Users/50026713/Desktop/archive
[50026713.HCESV5CG824639Q] ▸ cat  ER401A.logOLD.20190216122204 | grep -E "Operating|Path|Version|Home|PWD|ROOT|User|Timezone" | wc -l
561
```

ILUSTRACIÓN 55 TRAZA IRRELEVANTE 13 (MOBA) (ELABORACIÓN PROPIA)

14. "*02/16/2019 10:23:23.372 CET main (02.021.17) INFORMATIVE: Patch Name is: /opt/SIUVOIa/lib/siusession.jar"* ➔ Pertenece a las trazas de inicialización del colector, no es relevante.

```
[2019-03-08 11:18.59]  /drives/C/Users/50026713/Desktop/archive
[50026713.HCESV5CG824639Q] ▸ grep -i patch ER401A.logOLD.20190216122204  | wc -l
5940
```

ILUSTRACIÓN 56 TRAZA IRRELEVANTE 14 (MOBA) (ELABORACIÓN PROPIA)

15. *" 02/16/2019 10:40:10.827 CET main (BusinessRule) INFORMATIVE: CONSTANTS [/CommonConstants]: eco__DB_ADMUSER_ERICSSON_CONSOL_TABLE*
02/16/2019 10:40:10.827 CET main (BusinessRule) INFORMATIVE: CONSTANTS [/CommonConstants]: eco__DB_DRIVER_ERICSSON_CONSOL_TABLE" ➔ Pertenece a las trazas de inicialización del colector, concretamente las que se corresponden con la carga de tablas. No es relievante.

```
[2019-03-08 11:27.02]  /drives/C/Users/50026713/Desktop/archive
[50026713.HCESV5CG824639Q] ▸ grep CONSTANTS  ER401A.logOLD.20190216122204 | wc -l
15941
```

ILUSTRACIÓN 57 TRAZA IRRELEVANTE 15 (MOBA) (ELABORACIÓN PROPIA)

16. *"02/16/2019 09:54:41.788 CET main (BusinessRule) INFORMATIVE: LoadCommonConstants.configureTableConnections[MEDRHP - MEDRHP]:*
02/16/2019 09:54:41.789 CET main (BusinessRule) INFORMATIVE: CONNECTDATA [/CommonConstants/MEDDATA/Connections/MEDRHP]: DRIVER "

"02/16/2019 09:54:41.783 CET main (BusinessRule) INFORMATIVE: TABLELOCATION [/CommonConstants/MEDDATA]: CLIENTES_PLANPRECIOS_IPCM" ➔ Son trazas resultantes del inicio del colector, irrelevantes en el análisis de errores.

```
[2019-03-08 12:07.21]  /drives/C/Users/50026713/Desktop/archive
[50026713.HCESV5CG824639Q] ➤  grep -E "TABLELOCATION|CONNECTDATA|LoadCommonConstants"  ER401A.logOLD.20190216122204  | wc -l
1254
```

ILUSTRACIÓN 58 TRAZA IRRELEVANTE 16 (MOBA) (ELABORACIÓN PROPIA)

17. *"02/16/2019 10:40:07.754 CET main (01.006.05) WARNING: Attribute CATEGORY already exists in the nme schema of type com.hp.siu.utils.IntegerAttribute, now getting overriden with type com.hp.siu.utils.StringAttribute"* ➔ Aporta información sobre determinados problemas o asignaciones de valores a atributos. No aporta información valiosa para el análisis de errores.

```
[2019-03-08 12:48.15]  /drives/C/Users/50026713/Desktop/archive
[50026713.HCESV5CG824639Q] ➤ grep Attribute  ER401A.logOLD.20190216122204 | wc -l
429
```

ILUSTRACIÓN 59 TRAZA IRRELEVANTE 17 (MOBA) (ELABORACIÓN PROPIA)

18. *"02/16/2019 10:40:43.438 CET main (DBTable) INFORMATIVE: DBTable.createDBTable() Loading hashTable with select VALUE,START_DATE,END_DATE from medadm. REFERENCES where CODE='ACT_SUNRISE_ROUTES'"*➔ Información relativa a cargas de DBTables, no resulta relevante salvo que lleve asociada alguna mención al tiempo. Los tiempos son importantes, ya que cuando la duración en la carga de algún tipo de dato se prolonga, suele llevar asociado algún tipo de problema relevante que puede finalizar con el aborto del colector.

```
[2019-03-08 13:10.45]  /drives/C/Users/50026713/Desktop/archive
[50026713.HCESV5CG824639Q] ➤ grep  DBTable ER401A.logOLD.20190216122204 |  grep -v milliseconds | wc -l
283
```

ILUSTRACIÓN 60 TRAZA IRRELEVANTE 18 (MOBA) (ELABORACIÓN PROPIA)

A partir de aquí, habría que calcular el porcentaje de reducción de la limpieza aplicada. No hay una regla universal que diga qué porcentaje de limpieza es el más adecuado, todo dependerá de cómo de *limpios* estén los datos previamente tratados.

Otro punto que hay que tener en consideración sobre el log de análisis ER401A.logOLD.20190216122204, es que algunas de las trazas podrían solaparse, así que el porcentaje podría variar algunos puntos, aunque está suficientemente acotado.

Restando todas las líneas que se van a eliminar del total de líneas del log ejemplo, se obtienen las siguientes cifras:

TABLA 14 REDUCCIÓN OBTENIDA EN LIMPIEZA EJEMPLO

Reducción	Número de coincidencias	Búsqueda
1	2072	“moved to”
2	3775	“Datastore waiting for command”
3	2363	“DIR_DESTINATION”
4	2106	“There are no files to distribute”
5	2190	“Wrote NME file”
6	341	“Optional parameter ENVIO”
7	332	“Branching execution for file”
8	56	“Using default”
9	86	“Insertion in IUM_VOZ_INTERNAL_AUDIT committed”
10	173	“copied to”
11	203	“compressed in”
12	7359	“version”
13	561	“Operating|Class Path|Version|Home|PWD|ROOT| User|Timezone“
14	5940	“patch|Patch”
15	15941	“CONSTANTS”
16	1254	“TABLELOCATION|CONNECTDATA|LoadCommonConstants”
17	429	“Attribute”
18	283	“Dbtable” sin “milliseconds”

Dentro del algoritmo de entrenamiento que se creará a partir de esta y otra información, se realizará la limpieza de los datos.

Toda la batería de pruebas que se realizará, se hará en base a la información que se tiene del archivado de datos, sin embargo, de realizarse la implantación real, se debería hacer de forma que el algoritmo recibiera los datos en tiempo real.

8.4.2 Clasificación de los datos

En el apartado (*6.1.4.4 Clasificación de los datos)* se dieron una serie de pautas a la hora de clasificar los datos. Esas pautas, se ejemplificarán a continuación:

1. División de las líneas de código en campos
2. Modificación del formato de fecha, de forma que sea fácilmente legible
3. Volcado de los datos obtenidos en una matriz

8.4.2.1 División de las líneas de código en campos

La división de los campos dentro de una línea de log, se puede realizar de diversas formas, en este caso se abogará por la forma que se considera más sencilla, aunque quizás la cantidad de información que aporta, sea escaso. Dentro de una línea ejemplo, como puede ser: “*02/16/2019 00:46:35.xxx CET DistributorMove.sh - virtuales*

(133948) INFO: Distribution Starting for file 'ERMBC201:MBC201833754:ER401A:VIRTUALES:352977"'

Puesto que de lo que se trata esta prueba es de la determinación de si una traza lleva o no al colector al aborto, se planteará una variable *booleana* que indique si la traza genera que el colector se pare definitivamente. Esta variable será la *instancia* del algoritmo.

Se pueden obtener varios valores:

TABLA 15 VALORES ANALIZABLES EN LOGS

Valor	Tipo	Ejemplo
Fecha	Fecha	02/16/2019 00:46:35
Grado de información	String, aunque se le podría aplicar un valor numérico entero	INFO
Tipo de información	String, aunque se le podría aplicar un valor numérico entero.	Distribution Starting
Aborto del colector	Boolean	CRITICAL = true Cualquier otra traza = false

Se considera el nombre de los ficheros irrelevante, puesto que lo único que interesa, de cara al análisis de errores que provocan el aborto del colector, es si fallan o no; y en este caso concreto, no fallan.

Realizar esta operación para todas las líneas de código con las que se va a trabajar, sería un primer paso a fin de obtener datos comprensibles para la máquina que los va a analizar.

A continuación, un listado de los tipos de líneas de log que se pueden encontrar:

TABLA 16 EJEMPLO TRAZAS DISPONIBLES EN LOG

Nº	Línea ejemplo	Significado
1	02/16/2019 00:46:35.xxx CET DistributorMove.sh - virtuales (133948) INFO: Distribution Starting for file	Comienzo de la distribución a otros directorios de la máquina, del fichero en cuestión
2	02/16/2019 00:46:35.xxx CET DistributorMove.sh - virtuales (133948) INFO: End distribution of file 'ERMBC201:MBC201833754:ER401A:VIRTUALES:352977'	Fin de distribución del fichero dentro de la propia máquina
3	02/16/2019 00:46:35.231 CET main (06.024.05) INFORMATIVE: command '/home/siuvoice/release/scripts/handlers/DistributorMove.ksh ericssonR4 virtuales /var/opt/SIUVOIa/config/handlers/DistributorMove.cfg' completed with status: 0	Ejecución satisfactoria del script de distribución
4	02/16/2019 00:50:05.535 CET main (08.003.11) INFORMATIVE: Setting next file	Comienzo de procesamiento de un fichero de datos

	to: /var/opt/SIUCOLa/work/output/ericsson/ER401A/ERMBC201.MBC201833758#155027460041	procedente de la central
5	02/16/2019 00:50:05.537 CET main (hp) WARNING: GSMCaller2 source_: ERMBC201 02/16/2019 00:50:05.537 CET main (hp) WARNING: GSMCaller2 fileinfo.sourceFileName(): MBC201833758 02/16/2019 00:50:05.537 CET main (hp) WARNING: GSMCaller2 fileinfo.qualifier(): 694697	Aunque vengan marcados como WARNING, la información que contienen es informativa: tipo de central e identificador del proceso de colección
6	02/16/2019 01:21:44.752 CET main (07.003.11) INFORMATIVE: Aggregated 4,158 NMEs at a rate of 2 NMEs/sec	Velocidad y número de registros agregados
7	02/16/2019 01:21:45.583 CET main (07.018.01) INFORMATIVE: Scheme S1 tree contained 0 nmes, 0 flushed, 0 remain.	Informacion sobre el número de registros que se han añadido a una de las salidas a otros sistemas
8	02/16/2019 01:21:49.548 CET main (08.025.01) INFORMATIVE: DeleteFile.cleanup(): Deleted file /var/opt/SIUCOLa/work/output/ericsson/ER401A/ERMBC101.MBC101467350#155027648018	Fin de procesamiento de un fichero, y eliminación del mismo de directorio de entrada en máquina de procedencia
9	02/16/2019 09:49:42.651 CET main (148.002.06) INFORMATIVE: SFTPSource.executeFTPOp(): Sleeping for 2,000 milliseconds, retried 0 times 02/16/2019 09:50:04.654 CET main (148.002.06) INFORMATIVE: SFTPSource.executeFTPOp(): Sleeping for 2,000 milliseconds, retried 1 times 02/16/2019 09:50:26.655 CET main (148.002.06) INFORMATIVE: SFTPSource.executeFTPOp(): Sleeping for 2,000 milliseconds, retried 2 times 02/16/2019 09:50:48.656 CET main (148.002.06) INFORMATIVE: SFTPSource.executeFTPOp(): Sleeping for 2,000 milliseconds, retried 3 times	La máquina está intentando hacer un SFTP sin éxito. Aunque sea INFORMATIVE, es una señal clara de que algo no va bien
10	02/16/2019 09:50:50.656 CET main (148.002.05) WARNING: Could not perfrom FTP operation to Host : colmenaa.prod.airtel.es , after trying for 3 number of times	Tras tres intentos de SFTP, el resultado no ha sido el esperado, por lo tanto se lanza el aviso
11	02/16/2019 09:50:50.657 CET main (101.002.06) WARNING: DecoderStreamEncapsulator.getNME(): Caught exception: java.io.IOException: timeout: socket is not established	Se lanza una excepción puesto que el error con el SFTP persiste
12	02/16/2019 09:50:50.660 CET main	Fallo del collector que provoca

	(101.002.09) WARNING: DecoderStreamEncapsulator.getNME(): Stopping collector 02/16/2019 09:50:50.660 CET main (06.016.06) WARNING: FileJDBCDatastore is aborting the current transaction.	su aborto
13	02/16/2019 09:50:50.662 CET main (08.036.02) INFORMATIVE: End of Input Data. Collection Terminated.	Se da por terminado el proceso
14	02/16/2019 09:50:50.662 CET main (03.003.18) CRITICAL: Collector ER401A exiting due to errors.	El collector ha abortado
15	02/16/2019 09:50:50.663 CET Thread-1 (02.010.25) ACCOUNTING: Process ER401A(122152) Exiting.	Fin del procesamiento, el colector finaliza inesperadamente
16	02/16/2019 09:51:32.046 CET main (02.021.01) INFORMATIVE: ==== ER401A Starting ====	Reinicio del collector
17	02/16/2019 09:56:08.468 CET main (02.007.04) INFORMATIVE: * Creation Date: 2014-04-15 19:44:24 CEST	Información dentro del reinicio. No es muy relevante aunque de momento, se dejarán aquí
18	02/16/2019 09:56:08.753 CET main (LicenseManager) INFORMATIVE: com.hp.siu.protocolengine.radiusprotocolengine.RadiusProtocolConnector component does not have premium license and will be disabled	Información sobre licencias y clases. No es muy relevante aunque de momento, se dejarán aquí
19	02/16/2019 09:56:08.755 CET main (LicenseManager) WARNING: Denied premium com.hp.usage.rtp.IcapConnector not found in component factory	Información dentro del reinicio. No es muy relevante aunque de momento, se dejarán aquí
20	02/16/2019 09:56:10.667 CET main (24.001.01) INFORMATIVE: Created channel: Default	Información dentro del reinicio. No es muy relevante aunque de momento, se dejarán aquí
21	02/16/2019 09:56:10.753 CET main (178.004.02) WARNING: Service with short name ManagementService already exists. Remove it from short name list.	Pequeño error que se produce en el reinicio, no resulta de gran relevancia, se dejará aquí de momento.
22	02/16/2019 09:55:03.138 CET main (03.003.13) CRITICAL: Problem configuring collector: timeout: socket is not established	Error que provoca el aborto del colector.

Aunque podrían existir más trazas relevantes dentro de la monitorización y analítica de los logs, se considera que con la información aportada es suficiente para comenzar; más adelante se podrá comprobar si ha sido o no suficiente, si hay que añadir o quitar parámetros dentro del análisis.

8.4.2.2 Modificación del formato fecha para hacerlo legible

Antes de entrar de lleno en la transformación de las líneas, se ha observado un tipo de líneas que podrían requerir de un trato especial, y son aquellas que informan por el inicio y el fin de la distribución de ficheros, como se puede observar a continuación:

"*02/16/2019 03:57:28.**xxx** CET DistributorMove.sh - unbill (15829) INFO: Distribution Starting for file 'ERMBC101:MBC101467538:ER401A:UNBILL:795751'*

*02/16/2019 03:57:28.**xxx** CET DistributorMove.sh - unbill_aida (15834) INFO: Distribution Starting for file 'ERMBC101:MBC101467538:ER401A:UNBILL_AIDA:432258'*

02/16/2019 03:57:29.xxx CET DistributorMove.sh - unbill_aida (15834) INFO: End distribution of file 'ERMBC101:MBC101467538:ER401A:UNBILL_AIDA:432258'"

Estas trazas son relevantes, puesto que se trata de información acerca del inicio y el fin de procesamiento de ficheros, de su movilidad a través de los directorios pertenecientes a la máquina. El trato especial del que se ha hablado, será el siguiente:

- ⇨ Partiendo de un formato de fecha: 02/16/2019 03:57:29.xxx
- ⇨ Se podrían realizar modificaciones como las siguientes:
  - 02/16/2019 03:57:29.321 ➔ valor de la traza inmediatamente anterior
  - 02162019035729 ➔ ignorando la triple x del final
  - \- ➔ asignando un número secuencial en el orden normal de las trazas.

Por otro lado, existen múltiples formas de transformar este tipo de datos, algunas de ellas podrían ser las siguientes:

A partir de este formato: 02/16/2019 09:50:50.660, se podría obtener:

- 216095050660
- 02162019095050660 ➔ en este caso habría que pensar qué hacer con las trazas que tienen formato .xxx
- 02162019095050
- Sencillamente numerar las trazas por orden de secuencial, del 1 a 'fin de fichero'

8.5 Estudio de algoritmos y modelos

Aunque se podría hacer un estudio pormenorizado de los algoritmos existentes dentro del aprendizaje supervisado y los algoritmos de clasificación, en base a los estudios realizados con anterioridad para datos similares a los que se están analizando, no es el método más efectivo.

Para saber qué algoritmos son los más idóneos, bastará con utilizar las librerías de las que dispone el lenguaje Python. Gracias a estas librerías, se pueden probar diferentes algoritmos y observar los resultados que obtienen.

8.6 Pruebas

Se realizarán varias pruebas con datos provenientes de un mismo fichero, pero procesando estos datos de diferentes formas, a fin de ver cuál de ellas obtiene mejores resultados. Para ello se explicará cómo se realizarán las pruebas.

1. Definición del tipo de limpieza de datos que se va a realizar
2. Resultados de la limpieza de datos
3. Realización de prueba en modelo
4. Conclusiones de la prueba

8.6.1 Prueba número 1

Esta será la más sencilla de todas las pruebas que se realizarán. La prueba consiste en una sencilla limpieza de datos, la misma que se describe en el apartado *(8.4.1 Limpieza de datos*) en el que se eliminan las trazas *irrelevantes*. Para el resto de trazas que componen el log, se asignará una serie de 'pesos' o 'valores' en los que se determinará cuánto de cerca (4) o lejos (0) está la traza de provocar un error grave, capaz de provocar el aborto del mismo.

8.6.1.1 Definición y resultado de los datos de trabajo

A continuación, los valores con los que se ha trabajado en esta prueba:

Limpieza de datos:

- Número de ficheros de log: UNO
- Nombre de fichero: ER401A.logOLD.20190216122204
- Número de líneas iniciales del fichero: 54037
- Tipo de limpieza de datos aplicado: Apartado 8.4.1.1
- Número de líneas obtenidas post-limpieza: 12760
- Porcentaje de reducción de datos: 76'38%

Formato de fecha:

- Modificación aplicada al formato de fecha:

Definición de variables: 02162019035729

- Número de variables en la muestra: dos
- Mensajes que se analizan: 5
- Definición de las variables:

TABLA 17 VARIABLES IDENTIFICABLES EN LOG

Nombre	Tipo	Ejemplo
Fecha	String	02162019035729
Tipo de información	Integrer	CRITICAL: 4 ACCOUNTING: 3 WARNING: 2 INFORMATIVE: 1 INFO: 0

8.6.1.2 Gráfica de datos

Gracias a una serie de pequeños scripts generados bajo la Shell de Unix, se ha generado la limpieza de trazas especificada en los apartados anteriores. Sin embargo, a partir de este momento se ha tomado la determinación de no trabajar más esta metodología, puesto que el procesamiento de grandes volúmenes de datos, resulta muy lento y las ejecuciones pueden demorarse horas.

Con la limpieza que se ha realizado, se ha obtenido la siguiente gráfica de datos (a través del programa MATLAB de tratamiento matemático); teniendo como eje x la línea de tiempo y como eje y, la valoración asignada a las diferentes trazas de log analizadas:

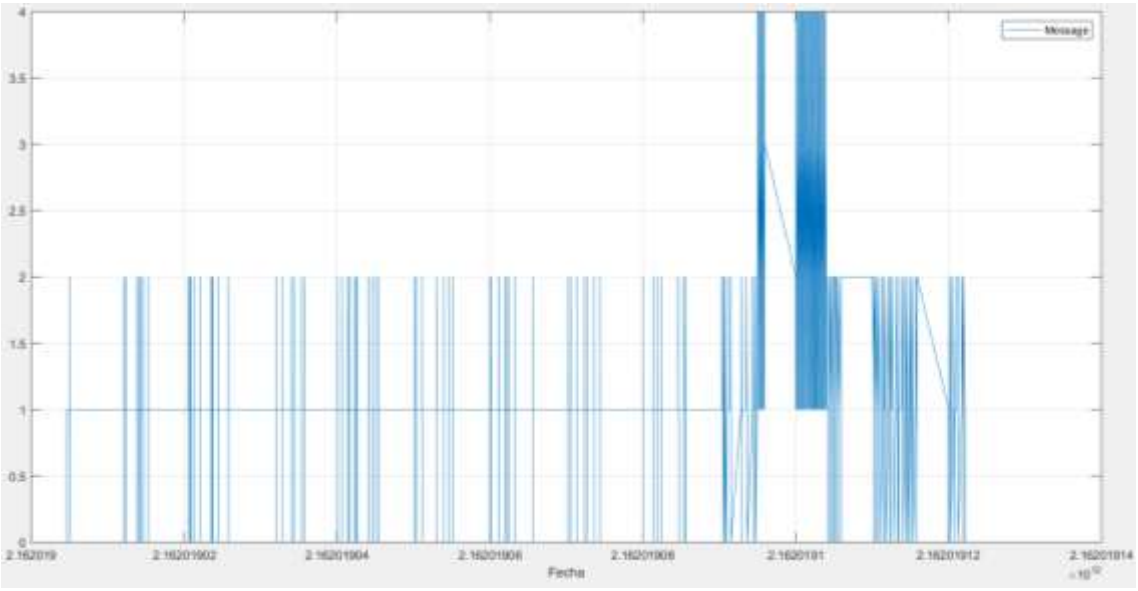


ILUSTRACIÓN 61 PRUEBA 1 (ELABORACIÓN PROPIA)

Los datos se encuentran excesivamente dispersos, se observa que claramente se han eliminado ciertas trazas.

8.6.1.3 Conclusiones de la prueba número 1

El número de variables y valores que pueden tomar éstas, resulta insuficiente. No se continuará el análisis para estos datos, puesto que, aunque la limpieza resulte válida, la gráfica muestra una escasez en los datos en la que se hace difícil ninguna predicción. Se precisa que los datos tengan asignados una variable más que determine si la traza provoca o no el aborto ya que, de la forma en la que se encuentran en este momento, es imposible determinar para cualquier sistema si ha habido o no caída del mismo.

8.6.2 Prueba número 2

Para esta prueba se realizan una serie de variaciones relevantes, con respecto a la anterior. En primer lugar, la asignación de una variable más: Aborto-NoAborto, tipo boolean, que determine si la traza provoca aborto del colector (true) o no (false). En segundo lugar, está la cuestión sobre los pesos asignados a las distintas trazas, que en

este caso poseen una complejidad mayor al anterior, con valores que van desde el 0 al 6; siendo este último resultado de la caída del colector.

En este caso, la limpieza de datos es también diferente al de la Prueba 1, puesto que en esta ocasión se analizarán las trazas relevantes, en lugar de eliminar las que no lo son.

Además de lo aportado en las líneas anteriores, cabe destacar que de ahora en adelante, se elimina la fecha para colocar en su lugar sencillamente el orden secuencial de las trazas. Esto se debe a que no resulta necesario saber cuántas líneas irrelevantes se han eliminado, con saber en qué orden están las que son importantes para el análisis, es más que suficiente.

8.6.2.1 Definición de los datos de trabajo

A continuación, los valores con los que se ha trabajado en esta prueba:

Limpieza de datos:

- Número de ficheros de log: UNO
- Nombre de fichero: ER401A.logOLD.20190216122204
- Número de líneas iniciales del fichero: 54037
- Tipo de limpieza de datos aplicado: Apartado 8.4.1.1
- Número de líneas obtenidas post-limpieza: 7531
- Porcentaje de reducción de datos: 86.06%
- Fichero de análisis: Analisis.P2_SeisValores.txt
- Fichero Resultado: Resultado_Prueba2.txt

Formato de fecha:

- Modificación aplicada al formato de fecha: orden secuencial

Definición de variables:

- Número de variables en la muestra: tres
- Mensajes que se analizan: 26
- Definición de las variables:

TABLA 18 DEFINICIÓN DE VARIABLES EN LA PRUEBA 2

Valor del mensaje	Aborto	Trazas
0	0	INFO: Distribution Starting for file
0	0	INFO: End distribution of file
1	0	completed with status: 0
0	0	INFORMATIVE: Setting next file to
2	0	WARNING: GSMCaller2 source_:
2	0	WARNING: GSMCaller2 fileinfo.sourceFileName()
2	0	WARNING: GSMCaller2 fileinfo.qualifier()
1	0	INFORMATIVE: Aggregated
1	0	INFORMATIVE: Scheme S1 tree contained 0 nmes, 0 flushed, 0 remain.

1	0	INFORMATIVE: DeleteFile.cleanup
2	0	INFORMATIVE: SFTPSource.executeFTPOp(): Sleeping for 2,000 milliseconds, retried
3	1	WARNING: Could not perfrom FTP operation to Host :
4	0	WARNING: DecoderStreamEncapsulator.getNME(): Caught exception:
5	1	java.io.IOException: timeout: socket is not established
6	1	WARNING: DecoderStreamEncapsulator.getNME(): Stopping collector
6	1	WARNING: FileJDBCDatastore is aborting the current transaction.
6	1	INFORMATIVE: End of Input Data. Collection Terminated.
6	1	CRITICAL: Collector ER401A exiting due to errors
6	1	ACCOUNTING: Process ER401A(122152) Exiting.
1	0	INFORMATIVE: ==== ER401A Starting ====
1	0	INFORMATIVE: * Creation Date: 2014-04-15 19:44:24 CEST
1	0	INFORMATIVE: com.hp.siu.protocolengine.radiusprotocolengine.RadiusProtocolConnector component does not have premium license and will be disabled
2	0	WARNING: Denied premium com.hp.usage.rtp.IcapConnector not found in component factory
1	0	INFORMATIVE: Created channel: Default
2	0	WARNING: Service with short name ManagementService already exists. Remove it from short name list.
6	1	CRITICAL: Problem configuring collector: timeout: socket is not established

Para esta prueba se ha realizado el procesamiento de los datos a través del lenguaje Python, en lugar de Shell script, que es muy rápido procesando grandes volúmenes de datos.

8.6.2.2 Metodología de trabajo

Para esta y futuras pruebas, la metodología de trabajo ha sido la siguiente:

1. Carga de los datos de trabajo en un fichero llamado *Analisis.txt*
2. Volcado del fichero de trabajo *Analisis.txt* en un array de trabajo
3. Análisis de las trazas del fichero de log que se encuentran dentro del fichero de *Analisis.txt*
4. Volcado de las líneas resultantes del análisis del paso anterior en un array y un fichero de trabajo.
5. Exposición de gráficas
6. Valoración del modelo aplicable
7. Conclusiones

Todos estos pasos se podrían explicar de la siguiente manera, tanto para ésta, como para pruebas posteriores:

1. Cargar en un fichero llamado *Analisis.txt* los datos referentes a Traza_de_log#Valor_asignado_a_la_traza#Aborto/no_aborto_del_log, obteniendo una línea por cada una de las trazas que se han considerado relevantes de la forma que se muestra en el cuadro de texto (Ejemplo fichero de Analisis.txt)

TABLA 19 EJEMPLO FICHERO DE ANALISIS.TXT

```
INFO: Distribution Starting for file#0#0
INFO: End distribution of file#0#0
completed with status: 0#1#0
INFORMATIVE: Setting next file to#0#0
WARNING: GSMCaller2 source_:#1#0
WARNING: GSMCaller2
fileinfo.sourceFileName()#1#0
```

2. Una vez cargados los datos dentro del fichero *Analisis.txt*, se ha procedido a cargarlos dentro de una lista de listas; es decir, un array que contiene la información para cada una de las trazas.

3. Cuando ya está cargada la información en el array, se ha recorrido el fichero de log sobre el que se ha estado trabajando (ER401A.logOLD.20190216122204), de forma que se han seleccionado los datos relevantes, asignado el valor correspondiente y volcado esta información en un fichero de texto, de forma que el fichero tendría con el siguiente formato: Numero_de_línea_por_orden_secuencial,Valor_asignado_al_mensaje_de_log,Aborto(1)_no_aborto(0), tal y como se observa en el cuadro de texto Ejemplo fichero de trabajo

TABLA 20 EJEMPLO FICHERO DE TRABAJO

```
Linea,Mensaje,Aborto
1,1,0
2,9,0
3,2,0
4,1,0
5,9,0
```

4. El siguiente paso será realizar una serie de gráficas para observar los datos cargados. Esto se hará a través de una serie de librerías que posee Python. Por ejemplo, se puede ver el histograma de datos, que no es otra cosa que la visualización de los valores que más se repiten dentro de la muestra de datos:

```
In [109]: dataframe.drop(['Linea'],1).hist()
          plt.show()
```

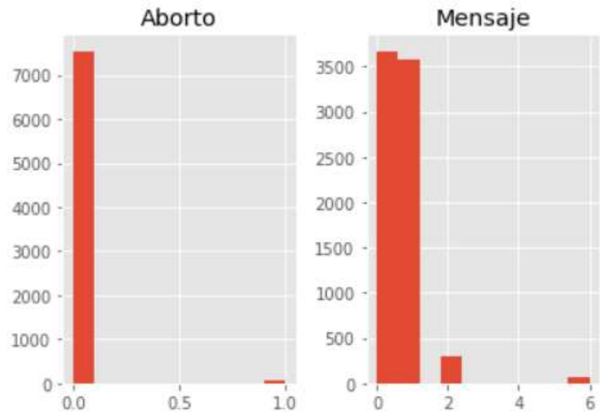


ILUSTRACIÓN 62 HISTOGRAMA PRUEBA 2 (ELABORACIÓN PROPIA)

A simple vista en la imagen *Histograma Prueba 2* se observa que hay una mayor predominancia de los valores 0 y 1 para las trazas (Mensaje) y que obviamente, hay una visible diferencia entre el porcentaje de trazas que no generan aborto del colector (0) que entre las que sí lo generan (1).

8.6.2.3 Modelo de Regresión Logística

Una vez realizados los pasos que se acaban de describir, se procede a la realización de pruebas con el algoritmo seleccionado, que no es otro que el modelo logístico.

```
In [152]: predictions = model.predict(X)
          print(predictions)[0:400]

[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]

In [153]: model.score(X,y)

Out[153]: 0.999734431018457
```

ILUSTRACIÓN 63 REGRESIÓN LOGÍSTICA RESULTADO PRUEBA2 (ELABORACIÓN PROPIA)

Para que se entienda, lo que muestra la pantalla inmediatamente anterior (*Regresión Logística resultado Prueba2*) a estas líneas es la aplicación del modelo de regresión logística. El valor denominado con Aborto, se corresponde con un 0, que significa que la traza que se corresponde con ese valor asignado no provoca el aborto del colector, mientras que el valor Aborto=1, significaría lo contrario.

8.6.2.4 Modelo kNN

Uno de los campos de posible aplicación para el algoritmo kNN es la búsqueda de anomalías, y dada la estabilidad del sistema eIUM, ésta podría ser una buena opción a la hora de experimentar con un modelo válido de predicción de datos.

A la hora de trabajar con un modelo de Knn, se debe determinar el valor de k, a fin de que el algoritmo tenga la mejor tasa de acierto. El valor k dentro del algoritmo kNN, es el número de elementos en el que se agrupará la muestra para determinar el resultado. Por ejemplo, si se coge una muestra de 6 trazas y solo una provoca un 'aborto' (equivale al valor 5), el algoritmo entonces determina que, por aproximación, la pequeña muestra tomada provoca el 'no aborto' ya que, de los 6 vecinos tomados, 5 de ellos son trazas correspondientes al no aborto:

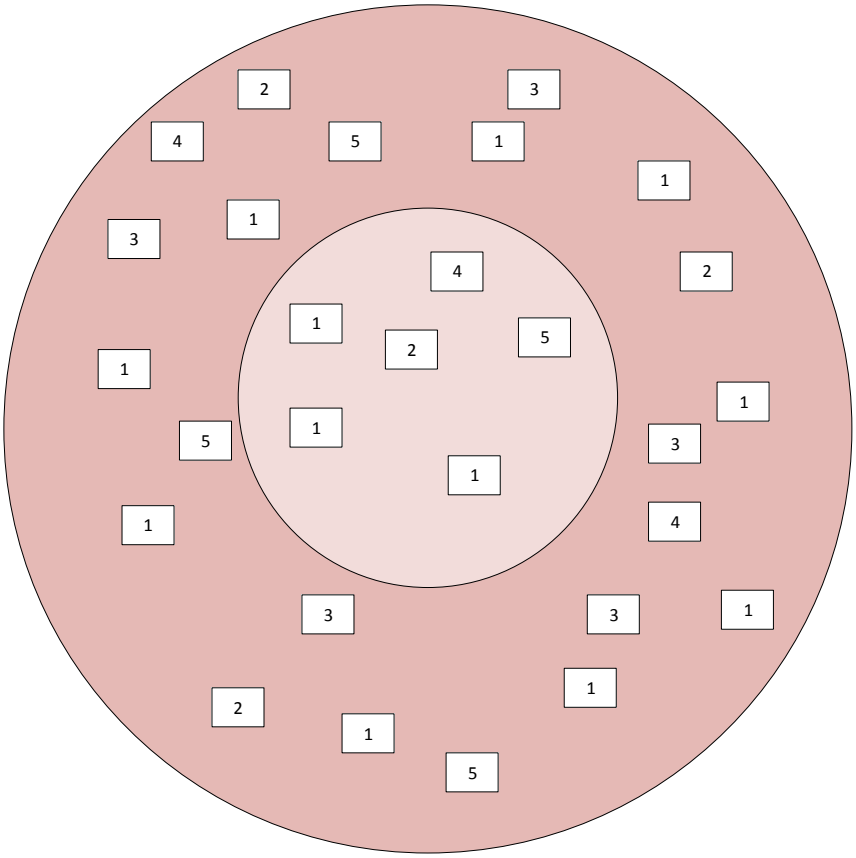


ILUSTRACIÓN 64 EJEMPLIFICACIÓN FUNCIONAMIENTO DE KNN (ELABORACIÓN PROPIA)

Para determinar los valores de k que resultan más óptimos, Python cuenta con una sentencia dentro de la librería sklearn (sklearn.neighbors). De forma que al lanzar:

```
In [11]: k_range = range(1, 100)
         scores = []
         for k in k_range:
             knn = KNeighborsClassifier(n_neighbors = k)
             knn.fit(X_train, y_train)
             scores.append(knn.score(X_test, y_test))
         plt.figure()
         plt.xlabel('k')
         plt.ylabel('accuracy')
         plt.scatter(k_range, scores)
         plt.xticks([0,5,10,15,20])
```

ILUSTRACIÓN 65 LANZAMIENTO GRADOS DE LIBERTAD KNN EN PRUEBA 2 (ELABORACIÓN PROPIA)

Se obtiene:

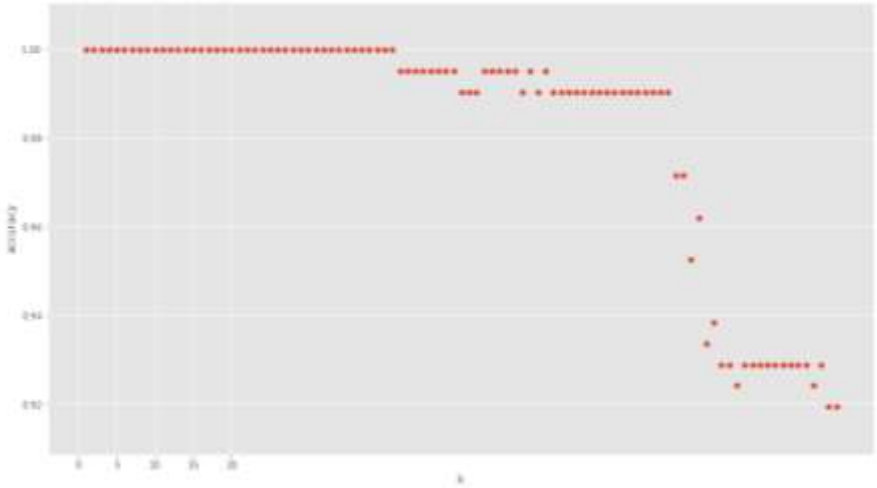


ILUSTRACIÓN 66 GRÁFICA GRADOS DE LIBERTAD KNN EN PRUEBA 2 (ELABORACIÓN PROPIA)

Donde (Ilustración 66 Gráfica grados de libertad kNN en Prueba 2) se observa que a partir de un valor superior a 40, la exactitud del algoritmo varía negativamente. Es por esta razón, por la que se escoge un valor de N=7, aunque daría lo mismo colocar cualquier valor entre 0-40 aproximadamente.

```
In [26]: n_neighbors = 7

         knn = KNeighborsClassifier(n_neighbors)
         knn.fit(X_train, y_train)
         print('Accuracy of K-NN classifier on training set: {:.2f}'
               .format(knn.score(X_train, y_train)))
         print('Accuracy of K-NN classifier on test set: {:.2f}'
               .format(knn.score(X_test, y_test)))

Accuracy of K-NN classifier on training set: 1.00
Accuracy of K-NN classifier on test set: 1.00
```

ILUSTRACIÓN 67 VALORES DE N PARA PRUEBA NÚMERO 2 (ELABORACIÓN PROPIA)

Para este caso y a modo curiosidad, se mostrará a continuación la matriz de confusión para este modelo, con la limpieza de datos descrita.

```
In [21]: pred = knn.predict(X_test)
         print(confusion_matrix(y_test, pred))
         print(classification_report(y_test, pred))

[[2394    1]
 [   0   17]]
              precision    recall  f1-score   support

           0       1.00      1.00      1.00      2395
           1       0.94      1.00      0.97        17

   micro avg       1.00      1.00      1.00      2412
   macro avg       0.97      1.00      0.99      2412
weighted avg       1.00      1.00      1.00      2412
```

ILUSTRACIÓN 68 MATRIZ DE CONFUSIÓN PARA KNN EN PRUEBA 2 (ELABORACIÓN PROPIA)

8.6.2.5 Conclusiones de la prueba número 2

A partir de los datos de Aborto/NoAborto, el modelo de datos puede predecir si la traza generará o no la caída del sistema.

Los datos correspondientes a model.score(X,y), muestran el índice de acierto que consigue a aplicación del modelo de regresión logarítmica es de un 99'97%. Se observa que para el modelo kNN, es incluso mayor, llegando al 100%.

La razón de que el algortimo kNN consiga unos resultados tan buenos, se debe a la ausencia de datos de no-aborto, es relativamente sencillo determinar que no habrá fallo cuando se agrupa en pequeños tramos las variables. Esto explica que a partir de un valor de k>50 (concretamente k=80), la exactitud del mismo varíe, como se observa en la *Variación valores de k en prueba 2:*

```
In [12]: n_neighbors = 80

         knn = KNeighborsClassifier(n_neighbors)
         knn.fit(X_train, y_train)
         print('Accuracy of K-NN classifier on training set: {:.2f}'
               .format(knn.score(X_train, y_train)))
         print('Accuracy of K-NN classifier on test set: {:.2f}'
               .format(knn.score(X_test, y_test)))

Accuracy of K-NN classifier on training set: 0.97
Accuracy of K-NN classifier on test set: 0.95
```

ILUSTRACIÓN 69 VARIACIÓN VALORES DE K EN PRUEBA 2 (ELABORACIÓN PROPIA)

8.6.3 Prueba número 3

En esta prueba se comprobará cuál es el valor que hace que el porcentaje de acierto varíe dentro de la aplicación de los modelos de regresión logística. Para ello, se ha determinado una nueva valoración de las trazas de log, en lugar de valorarse del 1 al 6, como en la prueba anterior (Prueba 2); se valorarán del 1 al 20.

8.6.3.1 Definición de datos de trabajo

Limpieza de datos:

- Número de ficheros de log: UNO
- Nombre de fichero: ER401A.logOLD.20190216122204
- Número de líneas iniciales del fichero: 54037
- Tipo de limpieza de datos aplicado: Apartado 8.4.1.1
- Número de líneas obtenidas post-limpieza: 7531
- Porcentaje de reducción de datos: 86.06%
- Fichero de análisis: Analisis.P3_20Valores.txt

TABLA 21 DEFINICIÓN DE VARIABLES EN LA PRUEBA 3

Valor del mensaje	Aborto	Trazas
0	0	INFO: Distribution Starting for file
1	0	INFO: End distribution of file
1	0	completed with status: 0
4	0	INFORMATIVE: Setting next file to
7	0	WARNING: GSMCaller2 source_:
7	0	WARNING: GSMCaller2 fileinfo.sourceFileName()
6	0	WARNING: GSMCaller2 fileinfo.qualifier()
2	0	INFORMATIVE: Aggregated
5	0	INFORMATIVE: Scheme S1 tree contained 0 nmes, 0 flushed, 0 remain.
3	0	INFORMATIVE: DeleteFile.cleanup
14	0	INFORMATIVE: SFTPSource.executeFTPOp(): Sleeping for 2,000 milliseconds, retried
15	1	WARNING: Could not perfrom FTP operation to Host :
16	0	WARNING: DecoderStreamEncapsulator.getNME(): Caught exception:
17	1	java.io.IOException: timeout: socket is not established
18	1	WARNING: DecoderStreamEncapsulator.getNME(): Stopping collector
19	1	WARNING: FileJDBCDatastore is aborting the current transaction.
20	1	INFORMATIVE: End of Input Data. Collection Terminated.
20	1	CRITICAL: Collector ER401A exiting due to errors
20	1	ACCOUNTING: Process ER401A(122152) Exiting.
13	0	INFORMATIVE: ==== ER401A Starting ====
9	0	INFORMATIVE: * Creation Date: 2014-04-15 19:44:24 CEST
8	0	INFORMATIVE: com.hp.siu.protocolengine.radiusprotocolengine.RadiusProtocolConnector component does not have premium license and will be disabled
11	0	WARNING: Denied premium com.hp.usage.rtp.IcapConnector not found in component factory
10	0	INFORMATIVE: Created channel: Default
12	0	WARNING: Service with short name ManagementService already exists. Remove it from short name list.
19	1	CRITICAL: Problem configuring collector: timeout: socket is not established

8.6.3.2 Aplicación del modelo

A partir de esta nueva valoración de datos, y siguiendo los pasos que se especificaron en la Prueba 2 para Python, se obtienen los siguientes histogramas, visiblemente diferentes a los de la prueba anterior. En ellos se observa, especialmente la variación en los datos para la variable *Mensaje*:

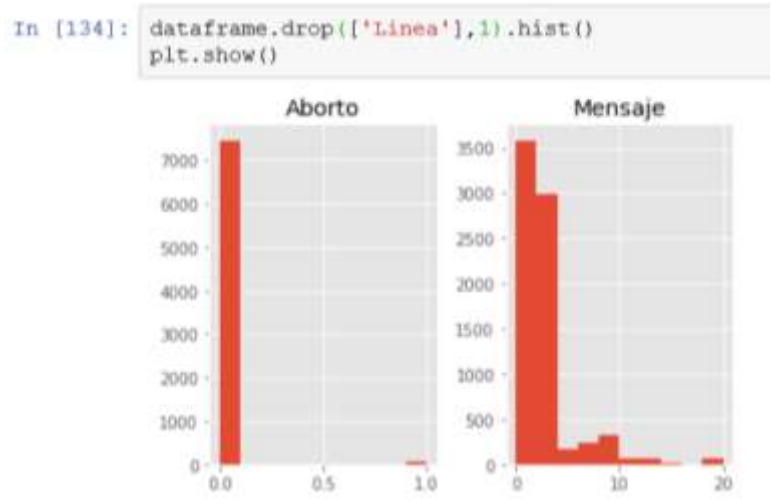


ILUSTRACIÓN 70 HISTOGRAMAS PRUEBA 3 (ELABORACIÓN PROPIA)

A continuación, un histograma más detallado, para que observe con más detalle, la diversidad en los 'pesos' de la variable *Mensaje*:

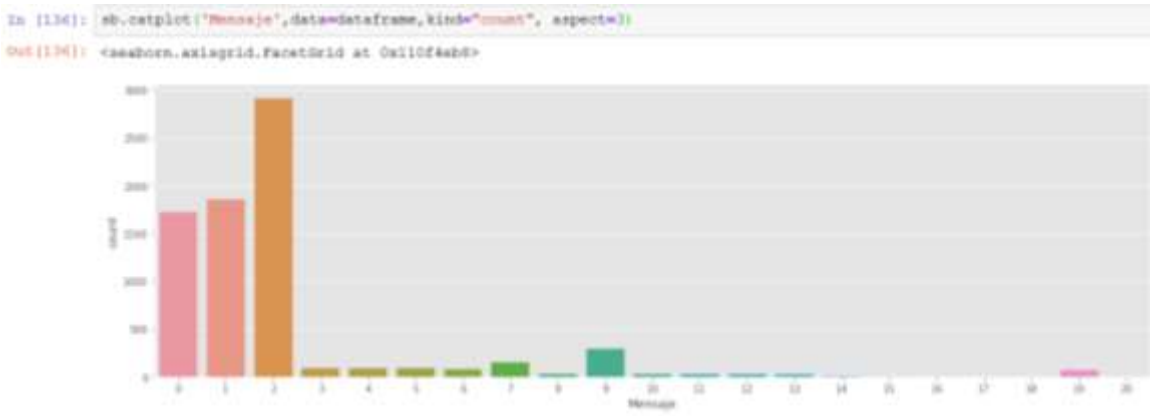


ILUSTRACIÓN 71 HISTOGRAMA MENSAJE PRUEBA 3

Se observa en este caso que el volumen de mensajes que obtuvieron valores 1 y 3 son más significativos que el resto, mientras que el histograma de *Aborto*, es igual que el anterior, ya que la ponderación asignada a las trazas, es la misma.

8.6.3.3 Modelo de Regresión Logística

A continuación, si se procesan los datos a través del modelo de regresión logística, se obtiene lo siguiente:

```
In [142]: model.score(X,y)
Out[142]: 0.999734431018457
```

ILUSTRACIÓN 72 RESULTADO REGRESIÓN LOGÍSTICA PRUEBA 3 (ELABORACIÓN PROPIA)

El índice de acierto es exactamente el mismo que se observó en la prueba anterior. Se puede concluir, por lo tanto, que la ponderación que se asigna a las trazas, poco tiene que ver con el resultado que obtiene el modelo aplicado.

8.6.3.4 Modelo kNN

Al igual que en la prueba anterior (*8.6.2 Prueba número 2*), se debe determinar el valor de K, a fin de que el algoritmo tenga la mejor tasa de acierto. Para ello Python cuenta con una sentencia dentro de la librería sklearn (sklearn.neighbors).

De forma que al lanzar:

```
In [11]: k_range = range(1, 100)
         scores = []
         for k in k_range:
             knn = KNeighborsClassifier(n_neighbors = k)
             knn.fit(X_train, y_train)
             scores.append(knn.score(X_test, y_test))
         plt.figure()
         plt.xlabel('k')
         plt.ylabel('accuracy')
         plt.scatter(k_range, scores)
         plt.xticks([0,5,10,15,20])
```

ILUSTRACIÓN 73 EXACTITUD KNN PRUEBA 3 (ELABORACIÓN PROPIA)

Se obtiene:

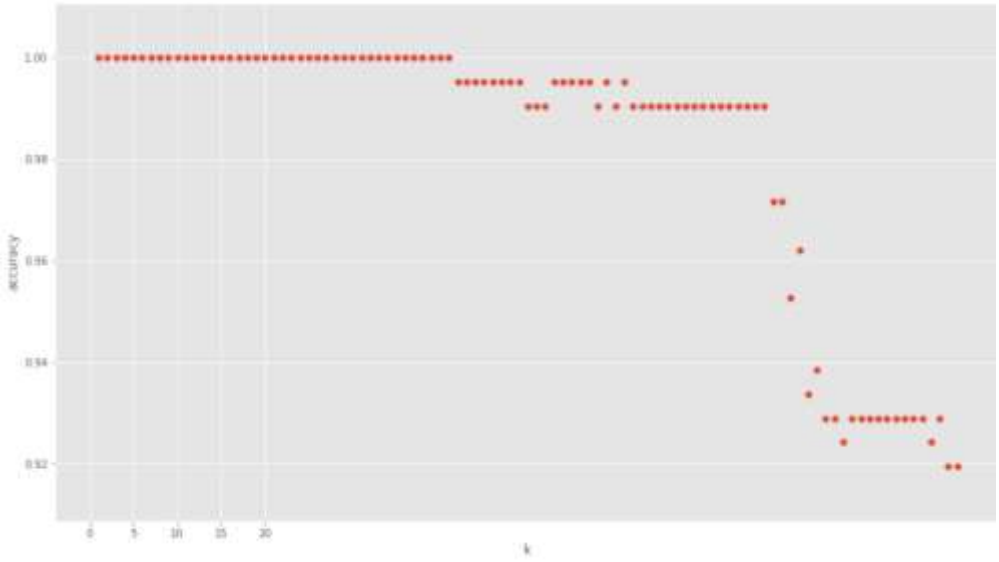


ILUSTRACIÓN 74 GRÁFICA GRADOS DE LIBERTAD DE K. PRUEBA 3 (ELABORACIÓN PROPIA)

Donde (*Ilustración 74 Gráfica grados de libertad de K. Prueba 3*) se observa que al igual que en la prueba anterior (*8.6.2 Prueba número 2*); partir de un valor superior a 40, la exactitud del algoritmo varía negativamente. Es por esta razón, por la

que se escoge un valor de N=7, aunque daría lo mismo colocar cualquier valor entre 0-40 aproximadamente.

```
In [44]: n_neighbors = 7

         knn = KNeighborsClassifier(n_neighbors)
         knn.fit(X_train, y_train)
         print('Accuracy of K-NN classifier on training set: {:.2f}'
               .format(knn.score(X_train, y_train)))
         print('Accuracy of K-NN classifier on test set: {:.2f}'
               .format(knn.score(X_test, y_test)))

         Accuracy of K-NN classifier on training set: 1.00
         Accuracy of K-NN classifier on test set: 1.00
```

ILUSTRACIÓN 75 EXACTITUD KNN EN PRUEBA 3 (ELABORACIÓN PROPIA)

Y nuevamente, se observa que la exactitud de este algoritmo de predicción es del 100%.

8.6.3.5 Conclusiones prueba número 3

Se observa que la aproximación en la predicción es muy cercana al 100% para los dos modelos aplicados, aunque el que obtiene mejores resultados es el kNN. Esto se debe a razones similares a las de la prueba anterior, existen muy pocas trazas en las que se demuestre que el colector se cae, por lo tanto, es lógico que los mecanismos de predicción sean tan exactos, al ser un sistema tan estable.

8.6.4 Prueba número 4

Con el fin de obtener un resultado diferente y comprobar qué es lo que provoca la variación de los datos en cuanto a las predicciones; para este caso, se han eliminado una serie de trazas y valorado cada una de ellas de forma diferente a la anterior. Las trazas eliminadas han sido consideradas las menos relevantes de las que se encontraban dentro del análisis de datos objeto de este estudio.

8.6.4.1 Definición de datos de trabajo

Limpieza de datos:

- Número de ficheros de log: UNO
- Nombre de fichero: ER401A.logOLD.20190216122204
- Número de líneas iniciales del fichero: 54037
- Tipo de limpieza de datos aplicado: Apartado 8.6.2.1
- Número de líneas obtenidas post-limpieza: 758
- Porcentaje de reducción de datos: 98’6%
- Fichero de análisis: Analisis.P4.txt
- Fichero Resultado: Resultado_Prueba4.txt

Formato de fecha:

- Modificación aplicada al formato de fecha:

Definición de variables: orden secuencial

- Número de variables en la muestra: tres
- Definición de las variables:

TABLA 22 DEFINICIÓN DE VARIABLES EN LA PRUEBA 4

Valores	Aborto	Trazas
1	0	INFORMATIVE: Setting next file to
2	0	INFORMATIVE: Scheme S1 tree contained 0 nmes, 0 flushed, 0 remain.
3	0	INFORMATIVE: ==== ER401A Starting ====
4	0	INFORMATIVE: * Creation Date: 2014-04-15 19:44:24 CEST
5	0	INFORMATIVE: com.hp.siu.protocolengine.radiusprotocolengine.RadiusProtocolConnector component does not have premium license and will be disabled
6	0	WARNING: Service with short name ManagementService already exists. Remove it from short name list.
7	0	WARNING: DecoderStreamEncapsulator.getNME(): Caught exception:
8	0	WARNING: Denied premium com.hp.usage.rtp.IcapConnector not found in component factory
9	0	WARNING: GSMCaller2 source_:
10	0	INFORMATIVE: SFTPSource.executeFTPOp(): Sleeping for 2,000 milliseconds, retried 0 times
11	1	WARNING: Could not perfrom FTP operation to Host :
12	1	java.io.IOException: timeout: socket is not established
13	1	WARNING: DecoderStreamEncapsulator.getNME(): Stopping collector
14	1	WARNING: FileJDBCDatastore is aborting the current transaction
15	1	INFORMATIVE: End of Input Data. Collection Terminated
16	1	CRITICAL: Problem configuring collector: timeout: socket is not established
17	1	CRITICAL: Collector ER401A exiting due to errors.
18	1	ACCOUNTING: Process ER401A(122152) Exiting

8.6.4.2 Aplicación del modelo

En este caso, el número de trazas analizadas y procesadas es muy inferior a las pruebas anteriores, de forma que el fichero de datos procesado, en lugar de tener 7531 líneas, posee únicamente 758 líneas.

```
In [158]: dataframe.drop(['Linea'],1).hist()
          plt.show()
```

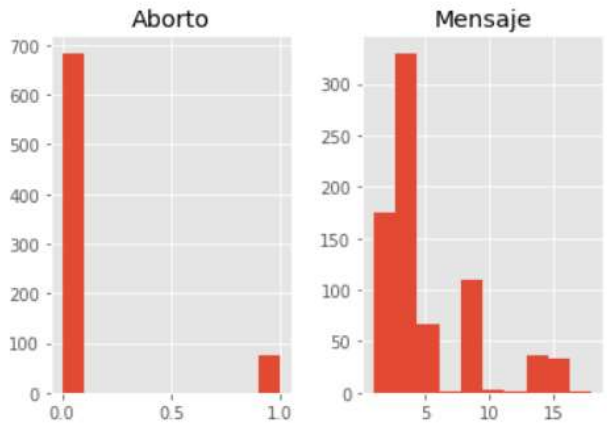


ILUSTRACIÓN 76 HISTOGRAMAS PRUEBA 4 (ELABORACIÓN PROPIA)

Siendo la división por trazas:

Mensaje	
1	87
2	88
3	33
4	297
5	33
6	33
7	1
8	33
9	77
10	2
11	1
12	1
13	1
14	36
15	1
16	32
17	1
18	1

Al haberse conservado las líneas que se corresponden con el aborto del log y haberse eliminado muchas de las que no, se observa que, en el histograma de Aborto, el porcentaje de abortos (1) es superior al de las gráficas de pruebas anteriores.

Por la parte que corresponde a los Mensajes, también se podría decir que ha habido variaciones, menor presencia de valores predominantes (en este caso el 4), y mayor diversidad en los valores que en histogramas de pruebas anteriores.

8.6.4.3 Modelo de Regresión Logística

Para finalizar, si se pasan los datos a través del modelo de regresión logística, se observa lo siguiente:

```
In [166]: predictions = model.predict(X)
          print(predictions)[0:400]

[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0
 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1
 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0
 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0
 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0
 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0]

In [167]: model.score(X,y)

Out[167]: 0.9986807387862797
```

Ilustración 77 Matriz de predicción Prueba 3, Modelo de Regresión Logística (Elaboración propia)

El valor que se obtiene predictivo es ligeramente inferior al de pruebas anteriores, aunque sigue teniendo una tasa de acierto de prácticamente el 100%.

8.6.4.4 Modelo kNN

Al igual que en la prueba anterior (*8.6.3 Prueba número 3)*, se debe determinar el valor de K, a fin de que el algoritmo tenga la mejor tasa de acierto. Para ello Python cuenta con una sentencia dentro de la librería sklearn (sklearn.neighbors).

De forma que al lanzar:

```
In [60]: k_range = range(1, 100)
         scores = []
         for k in k_range:
             knn = KNeighborsClassifier(n_neighbors = k)
             knn.fit(X_train, y_train)
             scores.append(knn.score(X_test, y_test))
         plt.figure()
         plt.xlabel('k')
         plt.ylabel('accuracy')
         plt.scatter(k_range, scores)
         plt.xticks([0,5,10,15,20,25,30,35,40,45,50])
```

Ilustración 78 Lanzamiento gráfica valores n, Prueba 4 (Elaboración propia)

Se obtiene:

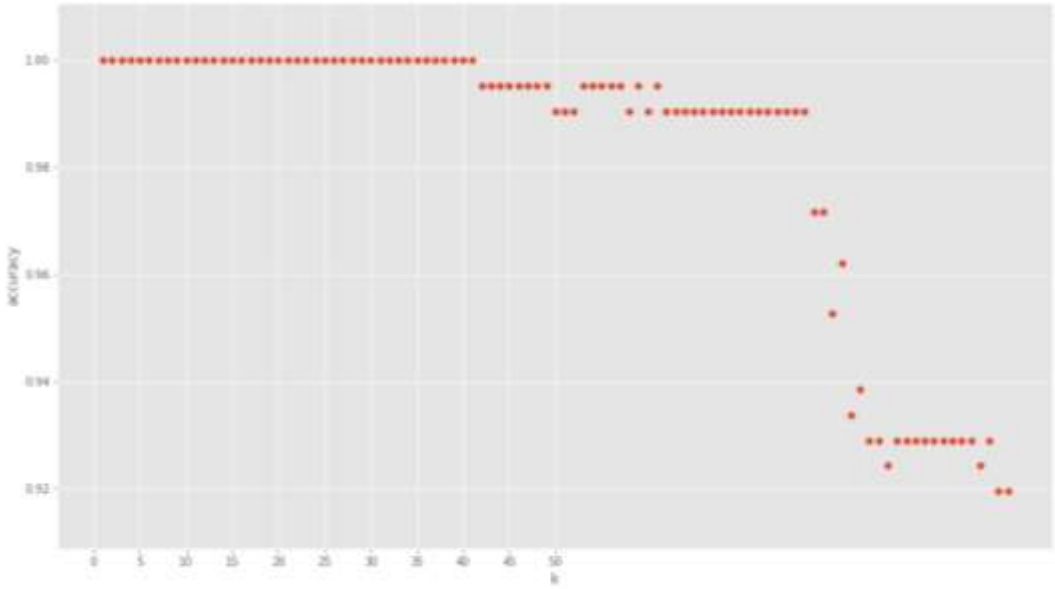


ILUSTRACIÓN 79 VALORES DE N PARA LA PRUEBA 4 (ELABORACIÓN PROPIA)

Donde (Ilustración 79 Valores de N para la prueba 4) se observa que al igual que en la prueba anterior (Prueba número 3); partir de un valor ligeramente inferior a 50, la exactitud del algoritmo varía negativamente. Es por esta razón, por la que se escoge un valor de *n=7*, aunque daría lo mismo colocar cualquier valor entre 0-48 aproximadamente.

Y nuevamente, se observa que la exactitud de este algoritmo de predicción es del 100%.

```
In [55]: n_neighbors = 7

         knn = KNeighborsClassifier(n_neighbors)
         knn.fit(X_train, y_train)
         print('Accuracy of K-NN classifier on training set: {:.2f}'
               .format(knn.score(X_train, y_train)))
         print('Accuracy of K-NN classifier on test set: {:.2f}'
               .format(knn.score(X_test, y_test)))

         Accuracy of K-NN classifier on training set: 1.00
         Accuracy of K-NN classifier on test set: 1.00
```

ILUSTRACIÓN 80 EXACTITUD DE PREDICCIÓN PRUEBA 4, KNN

8.6.5 Prueba número 5

A fin de probar un escenario con más errores y ver cómo se comporta el modelo, se probará a 'falsear' las trazas dentro del log. Una de las trazas habituales, será considerada 'aborto' aunque en la realidad no lo es.

8.6.5.1 Definición de datos de trabajo

Limpieza de datos:

- Número de ficheros de log: UNO
- Nombre de fichero: ER401A.logOLD.20190216122204
- Número de líneas iniciales del fichero: 54037
- Tipo de limpieza de datos aplicado: Apartado 8.6.2.1
- Número de líneas obtenidas post-limpieza: 758
- Porcentaje de reducción de datos: 98'6%

Dados los datos de partida de la Prueba 4, lo que se hará será observar el histograma de datos:

Una vez obtenida la gráfica que observa que la traza más común es una meramente informativa, concretamente se corresponde con "INFORMATIVE: Scheme S1 tree contained 0 nmes, 0 flushed, 0 remain". Esta traza será considerada como 'Aborto', estableciéndose de esa forma en el fichero de texto Analisis.txt.

Por último, se pondrá el valor correspondiente en la tabla de datos:

TABLA 23 DEFINICIÓN DE VARIABLES EN LA PRUEBA 5

Valores	Aborto	Trazas
1	0	INFORMATIVE: Setting next file to
2	0	INFORMATIVE: Scheme S1 tree contained 0 nmes, 0 flushed, 0 remain.
3	0	INFORMATIVE: ==== ER401A Starting ====
4	0	INFORMATIVE: * Creation Date: 2014-04-15 19:44:24 CEST
5	**1**	INFORMATIVE: com.hp.siu.protocolengine.radiusprotocolengine.RadiusProtocolConnector component does not have premium license and will be disabled
6	0	WARNING: Service with short name ManagementService already exists. Remove it from short name list.
7	0	WARNING: DecoderStreamEncapsulator.getNME(): Caught exception:
8	0	WARNING: Denied premium com.hp.usage.rtp.IcapConnector not found in component factory
9	0	WARNING: GSMCaller2 source_:
10	0	INFORMATIVE: SFTPSource.executeFTPOp(): Sleeping for 2,000 milliseconds, retried 0 times
11	1	WARNING: Could not perfrom FTP operation to Host :
12	1	java.io.IOException: timeout: socket is not established
13	1	WARNING: DecoderStreamEncapsulator.getNME(): Stopping collector
14	1	WARNING: FileJDBCDatastore is aborting the current transaction
15	1	INFORMATIVE: End of Input Data. Collection Terminated
16	1	CRITICAL: Problem configuring collector: timeout: socket is not established
17	1	CRITICAL: Collector ER401A exiting due to errors.
18	1	ACCOUNTING: Process ER401A(122152) Exiting

8.6.5.2 Aplicación del modelo

Como se puede observar, el histograma de la variable *Aborto* tiene variaciones con respecto a la prueba anterior, habiendo más abundancia del valor true (aborto) para esta variable:

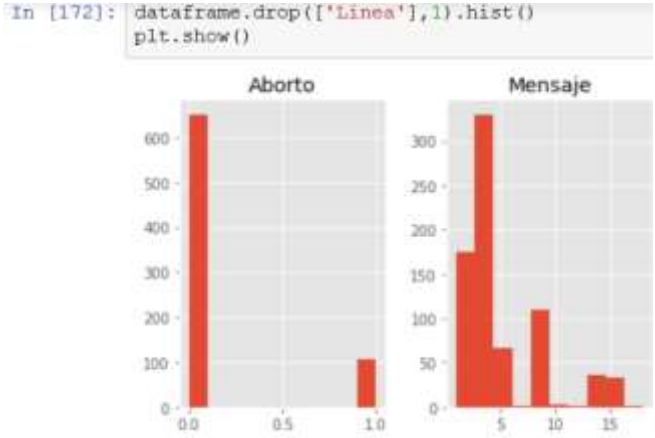


ILUSTRACIÓN 81 HISTOGRAMA ABORTO Y MENSAJE EN PRUEBA 5 (ELABORACIÓN PROPIA)

8.6.5.3 Modelo de Regresión Logística

Al aplicar el modelo de Regresión Logística, se observa el siguiente resultado:

```
In [178]: model.score(X,y)
Out[178]: 0.9551451187335093
```

ILUSTRACIÓN 82 EXACTITUD MODELO DE REGRESIÓN LOGÍSTICA EN PRUEBA5 (ELABORACIÓN PROPIA)

Se observa y concluye, por lo tanto, que a una mayor presencia de trazas consideradas 'aborto' (Aborto=1), la predicción del algoritmo disminuye; aunque sigue siendo muy alta la tasa de acierto, que en este caso es del 95.51%.

8.6.5.4 Modelo kNN

A la hora de aplicar el modelo Knn lo primero será comprobar qué valores son los más óptimos:

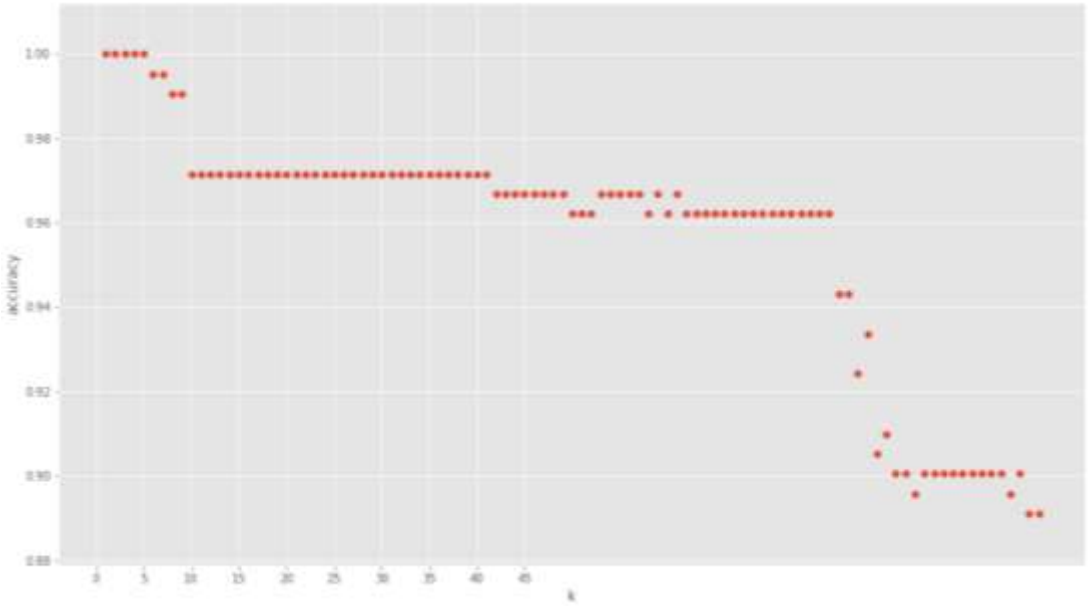


ILUSTRACIÓN 83 VALORES DE K PARA PRUEBA 5 (ELABORACIÓN PROPIA)

Como se ve en la gráfica, en este caso concreto y para la nueva limpieza que se realiza sobre los datos, la exactitud del algoritmo varía sustancialmente a partir de un valor *k>5*. De forma que al ejecutar el comando que calcula la exactitud, con un *k=12*, en la predicción se obtiene:

```
In [20]: n_neighbors = 12

         knn = KNeighborsClassifier(n_neighbors)
         knn.fit(X_train, y_train)
         print('Accuracy of K-NN classifier on training set: {:.2f}'
               .format(knn.score(X_train, y_train)))
         print('Accuracy of K-NN classifier on test set: {:.2f}'
               .format(knn.score(X_test, y_test)))

         Accuracy of K-NN classifier on training set: 0.97
         Accuracy of K-NN classifier on test set: 0.97
```

ILUSTRACIÓN 84 VALOR PREDICTIVO KNN PRUEBA 5 (ELABORACIÓN PROPIA)

Como se puede comprobar a la vista de los datos obtenidos, al hacer una limpieza de datos diferente, las variaciones dentro de las predicciones son sustanciales. Independientemente de que sigan teniendo unos niveles de predicción cercanos al 100%, se puede observar cómo se ven afectados al realizar los diferentes tratamientos; de forma que se demuestra la importancia de la limpieza en los mismos.

8.7 Resumen de pruebas

A continuación, una tabla en la que se exponen las pruebas realizadas con sus respectivos resultados.

TABLA 24 RESUMEN DE LAS PRUEBAS REALIZADAS

	Prueba 1	**Prueba 2**	**Prueba 3**	**Prueba 4**	**Prueba 5**
Número de líneas tras la reducción	12760	7531	7531	758	
Porcentaje reducido	76’38%	86.06%	86.06%	98’6%	98’6%
Número de trazas relevantes	4	22	24	18	18
Valores asignados a las trazas	5, de 0-4	6, de 0-5	21, de 0 a 20	18, de 1 a 18	18, de 1 a 18; con un aborto más que en la prueba 4
Existencia de la variable Aborto/No-Aborto	NO	SI	SI	SI	SI
Regresión logística	-	Acierto predictivo del: 99.97%	Acierto predictivo del: 99.97%	Acierto predictivo del: 99.87%	Acierto predictivo del: 95.51%
Knn	-	Acierto predictivo del: 100%	Acierto predictivo del: 100%	Acierto predictivo del: 100%	Acierto predictivo del: 97%

La primera de las pruebas, resultó ser más una toma de contacto que una prueba en sí. Los errores a la hora de procesar los datos antes de pasarlos a través del algoritmo, provocaron que éstos resultasen poco útiles.

Con respecto al resto de pruebas:

- Se observa que el acierto predictivo es muy alto, es decir que las posibilidades tener unos excelentes resultados son muy altas. Esto se debe a que los resultados en su mayoría son positivos (no aborto) y a que los posibles resultados son únicamente dos: aborto/no_aborto
- A más porcentaje de abortos (más pesos considerados aborto), menos probabilidad de acierto
- A mayor número de trazas, menor probabilidad de acierto.

9. Conclusiones del estudio

Las conclusiones a las que se llegan a partir del estudio realizado, se abordarán desde distintas perspectivas:

- Respecto a los datos
- Respecto al modelo
- Respecto a los conocimientos
- Respecto a las extensiones de este trabajo

9.1 Respecto a los datos

El análisis de datos ocupa la totalidad de las pruebas que se han realizado. Sólo se ha tomado una "pequeña" pero significativa muestra de los mismos, puesto que el sistema eIUM para la mediación de Vodafone, genera aproximadamente 100.000.000 de líneas mensualmente. Se puede calificar la muestra como representativa, ya que, desde el comienzo de esta investigación, no se han sucedido más errores fatales dentro de los procesos de colección.

Se ha podido observar la relevancia del análisis de datos, puesto que de él depende el éxito o el fracaso en la predicción de los mismos. Inicialmente, en las primeras pruebas, se simplificó en demasía la complejidad del sistema, obteniendo unos resultados tan buenos como irreales. Los resultados obtenidos en las últimas pruebas, siguen siendo satisfactorios; la problemática que presentan, es que se necesitaría una muestra más amplia de datos, para que se ajustaran completamente a la realidad del sistema.

¿Por qué no se ha realizado una muestra más amplia para hacer de este análisis algo más tangible, más cercano a la realidad o de mayor utilidad? Las razones son varias:

- Falta de transparencia: el eIUM posee un código propio al que no se tiene acceso, de forma que no se puede acceder a la totalidad de los errores que pueden darse; salvo que se trate de un desarrollador de este sistema, y no es el caso.
- Falta de potencia: se necesita un ordenador más potente que el que se posee para poder trabajar con Machine Learning. Los volúmenes de datos que se manejan son relativamente grandes y, aunque Python es un lenguaje muy potente y rápido, el ordenador no lo es, por lo tanto, se producen bloqueos constantes.
- Falta de trazas que contengan errores fatales: actualmente y gracias a los constantes cambios producidos, las máquinas de Mediación tan solo tienen abortos graves en contadas ocasiones; lo cual dificulta la tarea si lo que se pretende es la prevención de estos errores. Se necesitarían ejemplos de aborto por diferentes errores en las BBDD, llenados del sistema, procesos que se quedan corriendo inexplicablemente, etc.

Pese a los problemas presentados en las líneas anteriores, se podría concluir que el resultado del estudio es muy satisfactorio, si se analiza de cara a encontrar una

tecnología prácticamente infalible en la detección línea a línea de posibles abortos en los colectores.

9.2 Respecto al modelo

La selección de los modelos kNN y Regresión Logística, se realizó en base a la utilización de éstos para volúmenes de datos y tipos de datos similares a los analizados, a través de estudios científicos y blogs tecnológicos. Se podría haber intentado con otros muchos, dentro de los algoritmos de clasificación, puesto que el lenguaje Python incluye muchos de ellos; sin embargo, la variedad de tipos de algoritmos y modelos personalizados o no, es muy alta, y hay que tomar decisiones al respecto o este estudio podría haberse prolongado demasiado.

Se consideran satisfactorios los resultados obtenidos por los dos modelos, ya que el propio Python indica que la probabilidad de que fallen en sus predicciones es baja.

9.3 Respecto a los conocimientos

Cuando comencé esta andadura Machine Learning sonaba del mismo modo que Skynet, a la IA aplicada en la tecnología del futuro. Poco a poco mis conocimientos han ido en aumento, hasta dar por finalizado este estudio.

Para realizar una implantación como la que se plantea y que ésta sea fiable, el planteamiento debería ser diferente al que se ha realizado. Este estudio es un buen ejemplo de cara a un análisis previo, pero tal como se ha comprobado, la forma en la que se normalizan y reducen los datos, determina en gran medida su resultado; por lo tanto, se precisaría de un mayor conocimiento sobre la tecnología y una batería de pruebas más potente, que contemple muchos más escenarios (trazas) para realizar una implantación real. Estos conocimientos los posee, por ejemplo, un científico de datos gracias a su experiencia.

9.4 Respecto a la ampliación de este trabajo

Para que este trabajo pudiera resultar de utilidad para la compañía HPE CDS, habría que realizar una serie de modificaciones en su planteamiento inicial. Este estudio se podría considerar una toma de contacto sobre Machine Learning, sin embargo, si se quisiera abarcar más que eso, se deberían analizar varios frentes.

1. Automatización

Una herramienta de automatización para aplicación de la tecnología de Machine Learning, sería de gran utilidad. Se trataría de una aplicación capaz de analizar los datos de los logs en tiempo real. Esta herramienta se debería instalar en cada una de las cuatro máquinas que posee la compañía poseedora de los logs con los que se han realizado las pruebas (Vodafone) y tendría acceso a todos los logs del sistema. De esta forma, la aplicación tendría acceso a los datos en tiempo real, lanzando los avisos pertinentes en caso de que se produjeran problemas. El hecho de tener un volumen tan alto de datos,

mejoraría la eficiencia del algoritmo, puesto que, a más datos, más posibilidades de fallo y con ellas, más posibilidades de encontrar una solución.

Para un planteamiento de trabajo en tiempo real, habría que sentar las bases de forma diferente a la planteada en este estudio. Se debería lanzar una aplicación que permanentemente rastreara los datos de los logs en tiempo real, buscando en ellos, la posibilidad de predicción de fallos. Se entiende con esto, que la variedad de trazas a las que se tendría acceso, sería más amplia que la que se ha visto, abarcando todo tipo de tráfico de datos y en cada una de las máquinas.

El volumen de datos a procesar sería altísimo, como ya se ha comentado. El número de líneas mensuales (100.000.000) habría que multiplicarlo mínimo por cuatro, por cada una de las máquinas que procesan estos datos. De los datos obtenidos se buscaría relación con los distintos errores que se pueden dar y la aplicación debería determinar si hay probabilidades de que los colectores o la máquina en general aborten.

2. Búsqueda de información

Otro punto que se podría abordar, sería la búsqueda de manuales antiguos. Desempolvando antiguas wikis y repositorios propios del equipo de trabajo. No sería un trabajo especialmente gratificante éste, puesto que las búsquedas se realizarían sobre manuales antiguos y desfasados, con probabilidades de encontrar información útil, pero sin garantías reales de ello. La obtención de esta información podría proporcionar mayor traceado de lo que en los logs se puede encontrar, de cara a una mejor identificación de los posibles errores que tengan como consecuencia el aborto del sistema.

3. Acceso a herramientas propias

Como es lógico, el equipo de Mediación de telefonía, para cualquier operadora es permanente. En todo momento hay al menos una persona de guardia, dispuesta a solucionar cualquier problema que pudiera darse en cualquiera de las máquinas que conforman el sistema de colección y procesamiento de datos. Las incidencias que se producen, se tratan con una aplicación vulgarmente conocida como *Remedy*, cuyo nombre real es el *BMC Remedy Service Desk.* El *Remedy*, no es otra cosa que un gestor de incidencias en el que todos los equipos de trabajo que de alguna manera gestionan el sistema de Mediación, están interconectados. De esta forma, si, por ejemplo, sucede un error dentro de una de las máquinas, una persona (*Service Desk*) asigna ese error al equipo que considera responsable y éste deberá investigar a qué se debe el error, solucionarlo, escalarlo o si no le concierne, enviarlo al grupo que sepa cómo atajarlo. La cuestión es que las caídas dentro de los colectores o del sistema sean las mínimas posibles, para que en todo momento los datos fluyan de forma correcta.

Otra posible aplicación de ML dentro de Mediación, podría ser que la aplicación mencionada en el punto 1, tuviera acceso al Remedy. En esta aplicación se encuentra todo el histórico de incidencias y de cómo fueron solventadas, de forma que se daría acceso a una cantidad de trazas importantes y se podrían automatizar soluciones.

La aplicación propuesta requeriría de la comprensión por parte de la aplicación de ML, del Lenguaje Natural, puesto que las incidencias se redactan por las personas que las solucionan.

Este último punto, dentro del apartado, se podría considerar ‘castillos en el aire’, puesto que la complejidad de las extensiones que se proponen no es poca y el trabajo para poder llevarlas a cabo, requeriría personal con una formación muy específica y cierta experiencia. Para realizar todo este trabajo, se requeriría de más estudios, además de unos conocimientos en la materia y una experiencia muy superiores a los que posee la autora. Y por último y no menos importante, sería muy interesante que analizar muy en detenimiento si las ventajas de todas las extensiones que se proponen, valen el coste que pueda suponer su realización. A priori, se podría decir que el costo sería altísimo; sin embargo, si se considera de cuántas horas de guardia se evitaría a los trabajadores con la automatización, la situación probablemente cambiaría de forma radical.

10. Bibliografía

Amazon Machine Learning (Sin fecha). "*Entrenamiento de modelos de ML*". Recuperado de: https://docs.aws.amazon.com/es_es/machine-learning/latest/dg/training-ml-models.html

López Takeyas Bruno (2005). "*Procesamiento de lenguaje natural*". Apuntes del Instituto tecnológico de Nuevo Laredo, Méjico. Recuperado de: http://www.itnuevolaredo.edu.mx/takeyas/Apuntes/Inteligencia%20Artificial/Apuntes/tareas_alumnos/PLN/PLN(2005-II).pdf

Anónimo (2016). "*Algunos conceptos acerca de biometría*". Universidad Internacional de Valencia. Recuperado de: https://www.universidadviu.es/conceptos-acerca-biometria/

Antena3 (2017). *"Una niña de seis años encarga accidentalmente una casa de muñecas y galletas mientras hablaba con el asistente de voz de Amazon*". Antena 3 noticias. Recuperado de: https://www.antena3.com/noticias/mundo/nina-seis-anos-encarga-accidentalmente-casa-munecas-galletas-mientras-hablaba-asistente-voz-amazon_201701095873bf750cf2187c0d4e327e.html

Alonso Alegre Díez Maria Begoña (2016). "*Gestión de logs*". Máster universitario en Seguridad Informática. Universidad Internacional de la Rioja. Recuperado de: https://reunir.unir.net/bitstream/handle/123456789/3618/ALONSO-ALEGRE%20DIEZ%2C%20MARIA%20BEGO%C3%91A.pdf?sequence=1

Álvarez Eduardo (2018). *"Vodafone y Huawei comienzan a desplegar antenas de 5G NSA en España*". Revista digital ComputerHoy. Recuperado de: https://computerhoy.com/noticias/industria/vodafone-huawei-comienzan-desplegar-antenas-5g-nsa-espana-282631

Arrabales Raúl (2016). *"Deep Learning: qué es y por qué va a ser una tecnología clave en el futuro de la inteligencia artificial"*. Xataka, contenido tecnológico. Recuperado de: https://www.xataka.com/robotica-e-ia/deep-learning-que-es-y-por-que-va-a-ser-una-tecnologia-clave-en-el-futuro-de-la-inteligencia-artificial

Arriagada Rodríguez Miguel (2015). "*Comparación de métricas de distancia en el algoritmo K-Vecinos Más Cercanos para el problema de Reconocimiento Automático de Dígitos Manuscritos*" Informe del Proyecto para optar al Título Profesional Ingeniero de Ejecución en Informática. Universidad Católica Pontificia de Valparaíso. Recuperado de: http://opac.pucv.cl/pucv_txt/txt-3000/UCD3128_01.pdf

Bagnato Juan Ignacio (2017). "*Regresión logística paso a paso con Python*". AprendeMachineLearning.com. Recuperado de: http://www.aprendemachinelearning.com/regresion-logistica-con-python-paso-a-paso/

Balari Sergio (2014). "*Teoría de lenguajes formales*". Centro de lingüística teórica de la Universidad Autónoma de Barcelona. Recuperado de: https://ddd.uab.cat/pub/llibres/2014/116304/teolenfor_a2014p1iSPA.pdf

BBC (2017). *"Cómo un reportero de la BBC y su hermano mellizo lograron engañar al sistema de reconocimiento de voz del banco HSBC*". Recuperado de: https://www.bbc.com/mundo/noticias-39975337

Berlanga Silvente Vanesa, Rubio Hurtado Maria Jose y Vila Baños Ruth (2013). "*Cómo aplicar árboles de decisión en SPSS*" Recuperado de: http://diposit.ub.edu/dspace/bitstream/2445/43762/1/618361.pdf

Borrajo Millán Daniel, González Boticario Jesús, Isasi Viñuela Pedro (2006). "*Aprendizaje Automático*". Ed. Sanz y Torres, Madrid.

Bownlee Jason (2017). "*How to Clean Text for Machine Learning with Python*". Machinelearningmastery.com. Recuperado de: https://machinelearningmastery.com/clean-text-machine-learning-python/

Boyán Ivanov Bonev (2014) "*Depuración y gestión de logs*". Curso de Aplicaciones y Especialista en Aplicaciones y Servicios Web con Java Enterprise, Universidad de Alicante. Recuperado de: http://www.jtech.ua.es/j2ee/publico/lja-2012-13/sesion06-apuntes.html#Gesti%C3%B3n+de+logs+con+Log4Java

Brunet Robert (2015) "*¿Qué es la minería de datos?*". Revista online MuyInteresante. Recuperado de: https://www.muyinteresante.es/tecnologia/preguntas-respuestas/que-es-la-mineria-de-datos-311477406441

Cal González Abel (2015). "*Propuesta de arquitectura distribuida para la gestión de logs*". Proyecto de fin de carrera, Universidad Carlos III de Madrid. Recuperado de: https://e-archivo.uc3m.es/bitstream/handle/10016/22278/PFC_Abel_Cal_Gonz%C3%A1lez.pdf

Cárdenas Juan Pablo, Olivares Gastón, Alfaro Rodrigo (2014). "*Clasificación automática de textos usando redes de palabras*". Revista Signos nº 87, vol. 46. Estudios de lingüística. Recuperado de: https://scielo.conicyt.cl/scielo.php?script=sci_arttext&pid=S0718-09342014000300001

Carlos Torres Luis (sin fecha). "*Inteligencia Artificial*". Universidad Nacional de Colombia. Recuperado de: http://disi.unal.edu.co/~lctorress/iartificial/IAc016.pdf

Carmona Suárez Enrique J. (2014) "*Tutorial sobre Máquinas de Vectores de Soporte*". Universidad Nacional de Educación a Distancia, departamento de Inteligencia Artificial. Recuperado de: http://www.ia.uned.es/~ejcarmona/publicaciones/%5B2013-Carmona%5D%20SVM.pdf

CDS (2018). "*CDS Spain*". Recuperado de: https://www.hpcds.com/es/about-us.html

Cho Sung-Bae (2016). "*Explote Machine Learning Techniques for location, recognition and prediction with smartphone logs*" Universidad de Yonsey, Seúl, Corea del Sur. Elsevier. Recuperado de: http://sclab.yonsei.ac.kr/publications/Papers/IJ/2016_Neurocomputing_SBC.pdf

Cohen Gerry (2012). "*Predicciones y tendencias en Business Inteligence para 2013*". Revista Mundocontact Recuperado de: https://mundocontact.com/predicciones-y-tendencias-en-business-intelligence-para-2013/

Comisión Nacional de los Mercados y la Competencia (2017). "*Códigos de operadores de portabilidad*". Recuperado de: https://sede.cnmc.gob.es/sites/default/files/2017-01/CodOperadoresPortabilidad.pdf

Corbin Juan Armando (2018). "*12 tipos de lenguaje y sus características*". Psicología y mente: Psicología y relaciones personales. Recuperado de: https://psicologiaymente.com/social/tipos-de-lenguaje

Coronado Alberto (2017). "*Formación de Machine Learning: Algoritmos de Machine Learning por tipo de aplicación*" Recuperado de: https://www.albertcoronado.com/2017/01/04/formacion-machine-learning-algoritmos-de-machine-learning-por-tipo-de-aplicacion/

Cortés Vázquez Augusto, Vega Huerta Hugo y Pariona Quispe Jaime (2009). *"Procesamiento de Lenguaje Natural"*. Facultad de Ingeniería de Sistemas e Informática. Universidad Mayor de San Marcos, Perú. Recuperado de: http://revistasinvestigacion.unmsm.edu.pe/index.php/sistem/article/view/5923/5121

Cossío Alfonso (2018). "*Bots, Machine Learning, Servicios Cognitivos. Realidad y perspectivas de la Inteligencia Artificial en España, 2018". Revista pwc, Marzo de 2018.* Recuperado de: https://www.pwc.es/es/publicaciones/tecnologia/assets/pwc-ia-en-espana-2018.pdf

Cyert Richard M., Simon Herbert A. y Trow Donald B. (1956). "The Journal of Busines". Universidad de Chicago

Despujol Zabala Ignacio (2018). "*Aprendizaje automático. Ejemplo de matriz de confusión no binaria*". Universidad Politécnica de Valencia. Recuperado de: http://www.upv.es/visor/media/d0835a70-f4d9-11e8-9264-b32f4dcc5875/c

Diccionaro de la Lengua Española (sin fecha). Definición de "*lenguaje*". Recuperado de: https://dle.rae.es/?id=HTxyZDZ|HTy5CnJ

Diccionario de la Lengua Española (sin fecha). Definición de "*Lingüística*". Recuperado de: https://dle.rae.es/?id=NNPFPOl

Diccionario de términos (sin fecha) "*Qué es gramática*". Significados.com. Recuperado de: https://www.significados.com/gramatica/

DigitalHouse (2018). "*Cómo la inteligencia artificial puede aumentar la productividad de la empresa*". DigitalHouse. Recuperado de: https://www.digitalhouse.com/dh-blog/la-inteligencia-artificial-puede-aumentar-la-productividad-la-empresa/

Durant Kathleen T. y Smith Michael D. (2006). "*Meaning sentiment classification from political web logs*". Division of Engineering and Applied Sciences Cambridge, MA USA. Recuperado de: https://www.researchgate.net/profile/Kathleen_Durant/publication/228337184_Mining_sentiment_classification_from_political_web_logs/links/5667bc1008ae8905db8bc965/Mining-sentiment-classification-from-political-web-logs.pdf

Estévez Macarena (2017). "*Machine Learning (Píldora de conocimiento)*". Recuperado de: https://inteligencia-analitica.com/machine-learning/

Estévez Macarena (2018). "*Machine Learning y matemáticas*". Inteligencia Analítica. Recuperado de: https://inteligencia-analitica.com/machine-learning-matematicas/

Euroforum (2018). "*Inteligencia Artificial y lenguaje natural ¿Cuál es la conexión?"*. Recuperado de: https://www.euroforum.es/blog/inteligencia-artificial-y-lenguaje-natural-cual-es-la-conexion/

Fernando Negrete Jorge (2014). "*OSS/BSS: sistemas para operadores pensando en los usuarios*". Revista online MediaTelecom. Recuperado de: https://www.mediatelecom.com.mx/2014/10/27/oss-bss-sistemas-para-operadores-pensando-en-los-usuarios/

Fradkin Dmitry y Mörchen Fabian (2015). "*Minig secuential patterns for classification*" Knowledge and Information Systems, nº 45. Recuperado de: https://www.researchgate.net/publication/270292710_Mining_sequential_patterns_for_classification

Francois Pujet Jean (2016). "*What is Machine Learning?*". Ibm.com. Recuperado de: https://www.ibm.com/developerworks/community/blogs/jfp/entry/What_Is_Machine_Learning?lang=en

Galarza Hernández Javier (2017) "*Reducción de dimensionalidad en Machine Learning. Diagnóstico de cáncer de mama basado en datos genómicos y de imagen*". Universidad Politécnica de Valencia. Recuperado de: https://riunet.upv.es/bitstream/handle/10251/92565/GALARZA%20-%20Reducci%C3%B3n%20de%20dimensionalidad%20en%20Machine%20Learning.%20Diagn%C3%B3stico%20de%20c%C3%A1ncer%20de%20mama%20bsado%20e....pdf?sequence=1

Gámez Granados Juan Carlos (2017). "*Uso de técnicas de aprendizaje para clasificación ordinal y regresión*". Tesis doctoral. Universidad de Granada, Ciencias de la Computación e Inteligencia Artificial. Recuperado de: www.unav.edu/departamento/preventiva/files/file/cap12.doc

García A. (2000). "*Administración de redes: utilitarios, SNMP y RMON*". Universidad Tecnológica de Perú. Recuperado de: http://repositorio.pucp.edu.pe/index/bitstream/handle/123456789/28691/Redes_Cap23.pdf?sequence=23

García Cambronero Cristina (2012). "*Algoritmos de aprendizaje KNN y KMEANS*" Inteligencia en redes de comunicación. Universidad Carlos III de Madrid. Recuperado de: http://blogs.ujaen.es/barranco/wp-content/uploads/2012/02/Algoritmos-de-aprendizaje-knn-y-kmeans.pdf

García Montalvo José (2017) "*Crédito con 'c' mayúscula*". Periódico La Vanguardia. Recuperado de: https://www.lavanguardia.com/economia/20171007/431873420798/credito-con-c-mayuscula.html

Gerhards R. (2009). *"The Syslog Protocol*". RCF Editor. Recuperado de: https://www.rfc-editor.org/info/rfc5424

Girardin Luc and Brodbeck Dominique (1998). "A *visual approach for monitoring logs*". Veinteaba conferencia de Sistemas de Administración de la Universidad de Masachusets. Recuperado de: http://static.usenix.org/events/lisa98/full_papers/girardin/girardin.pdf

González Andrés (2014). "*¿Qué es Machine Learning*?". Cleverdata.io, blog tecnológico. Recuperado de: https://cleverdata.io/que-es-machine-learning-big-data/

González Ligdimar (2018). "*Regresión Logística*". Inteligencia Artificial. Recuperado de: http://ligdigonzalez.com/aprendizaje-supervisado-logistic-regression/

Gonzalo de Alba Álvaro (2018). "*Tipos de aprendizaje automático*". MachineLearningparatodos. Recuperado de: http://machinelearningparatodos.com/tipos-de-aprendizaje-automatico/

Gómez Rodríguez Valentín (2011) "*Propuesta de un sistema de distribución para mercancías para la pequeña y mediana empresa*"Instituto Politécnico Nacional, Tesis para Ciencias de la Computación, México DF. Recuperado de: https://tesis.ipn.mx/bitstream/handle/123456789/8440/C2.316.pdf?sequence=1

Gershgorn Dave (2018) *"Banks are already bumping up against the limits of AI in lending decisions*". Recuperado de: https://qz.com/1277305/ai-for-lending-decisions-us-bank-regulations-make-that-tough/

Hallam-Baker Phillip M., Behlendorf Brian (1996) "*Extended log file format*". Instituto de Tecnología de Israel. Recuperado de: https://www.w3.org/TR/WD-logfile.html

Herrero José (2013). "*La telefonía móvil cumple 40 años, repasamos su historia*". Ideal.es, edición de Jaén. Recuperado de: https://www.ideal.es/jaen/20130406/mas-actualidad/tecnologia/telefonia-movil-cumple-anos-201304060134.html

Hewlett Packard (sin fecha). *"HPE eIUM – enhanced Interactive Unified Mediation. Real-Time Mediation, Policy and Charging platform*". Recuperado de: https://www.hpe.com/ca/en/solutions/csp/enhanced-internet.html

Hewlett-Packard Development Company (2004). "*HP eIUM Overview Guide*". Recuperado de: https://support.hpe.com/hpsc/doc/public/display?docId=emr_na-a00025680en_us

Instituto Nacional de Ciberseguridad (Sin Fecha). "*Gestión de los log. Políticas de seguridad para la Pyme*". Recuperado de: https://www.incibe.es/sites/default/files/contenidos/politicas/documentos/gestion-logs.pdf

Inzaurralde Martín, Isi Jorge, Garderes Javier (2016). "*Telefonía Celular*". Universidad de la República de Montevideo, Uruguay. Recuperado de: https://cmapspublic2.ihmc.us/rid=1KLHSC8DS-1VL4W8V-2HP9/Trabajo%20sobre%20telefonia%20celular.pdhttp://www.antonioflores.es/antonioflores/articulos/consultateleco.pdf

ITlligent (2017). "*Procesamiento del lenguaje natural y sus aplicaciones*". Empresa de base tecnológica dedicada a la inteligencia web. Recuperado de: https://www.itelligent.es/es/procesamiento-del-leguaje-natural-aplicaciones/

Jieming Zhu, Pinjia He, Qiang Fu, Hongyu Zhang, Michael R. Lyu y Dongmei Zhang (2015). "*Learning to Log: Helping Developers Make Informed Logging Decisions*". Departamento de Informática e Ingeniería de la Universidad China de Hong Kong.

Johnsonbaugh Richard. (2005) "*Matemáticas discretas*". Sexta Edición. Pearson Educacion. Recuperado de: https://catedras.facet.unt.edu.ar/lad/wp-content/uploads/sites/93/2018/04/Matem%C3%A1ticas-Discretas-6edi-Johnsonbaugh.pdf

Julián Guillermo (2014) "*Redes neuronales: qué son y porqué están volviendo*". Comunidad Xataca. Recuperado de: https://www.xataka.com/robotica-e-ia/las-redes-neuronales-que-son-y-por-que-estan-volviendo

Keneth E. Nawyn (2003). *"A Security Analysis of System Event Login with Syslog*". Universidad PennState, Pensilvania, Estados Unidos. Recuperado de: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.198.7053&rep=rep1&type=pdf

Knepp Dustin (2010). "*La lingüística*" Universidad del Estado de California, Bakersfield. Departamento de Lenguajes Modernos y Literatura. Recuperado de: https://www.csub.edu/modlang/department/Spanish/LINGUISTICS/TEMA%201%20MA.pdf

Marín Diazaraque Juan Miguel (2014). *"Tema 3: Análisis de componentes principales*". Universidad Carlos III. Recuperado de: http://halweb.uc3m.es/esp/Personal/personas/jmmarin/esp/AMult/tema3am.pdf

Maverick Lin (2017). *"How does Machine Learning work?*". Quora, conocimiento compartido. Recuperado de: https://www.quora.com/How-does-machine-learning-work

Maybin Simon (2016). *"¿Cómo en EEUU las matemáticas te pueden meter en prisión*?". BBC. Recuperado de https://www.bbc.com/mundo/noticias-37679463

Merino Marcos (2019). *"Un simple parche de colores permite engañar a las cámaras de vigilancia con IA para que no detecten personas en una imagen*". Xataca.com. Recuperado de: https://www.xataka.com/inteligencia-artificial/simple-parche-colores-permite-enganar-a-camaras-vigilancia-ia-no-detecten-personas-imagen

Mitchell M. Tom (1997). "*Machine Learning*". McGraw-Hill Science/Engineering/Math. Recuperado de: http://profsite.um.ac.ir/~monsefi/machine-learning/pdf/Machine-Learning-Tom-Mitchell.pdf

Mizumoto Tomoya, Komachi Mamoru, Nagata Masaaki, Matsumoto Yuji (2011). "*Mining Revision Log of Language Learning SNS for Automated Japanese Error Correction of Second Language Learners*". 27 Conferecia Internacional del Ingeniería del Software, volumen 1, Florencia, Italia. Recuperado de: http://www.aclweb.org/anthology/I11-1017

Mohd Helmy Abd Wahab, Mohd Norzali Haji Mohd, Hafizul Fahri Hanafi y Mohamad Farhan Mohamad Mohsin (2008). *"Data Pre-Procesing on Web Server Logs for generalized association rules mining Algoritm*". World Academy of Science, Engineering and Technology 48 2008. Recuperado de: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.140.5102&rep=rep1&type=pdf

Molina López Jose Manuel y García Herrero Jesús (2006). *"Técnicas de Análisis de Datos. Aplicaciones prácticas utilizando Microsoft Excel y Weka*". Universidad de Jaén. Recuperado de: http://matema.ujaen.es/jnavas/web_recursos/archivos/weka%20master%20recursos%20naturales/apuntesAD.pdf

Montero Juan Manuel (sin fecha). "*Lenguajes naturales y lenguajes formales*". Universidad Politécnica de Madrid. Recuperado de: http://lorien.die.upm.es/juancho/pfcs/DPF/capitulo2.pdf

Moujahid Abdelmalik, Inza Iñaquiy Larrañaga Pedro (sin fecha). "*Tema 5: Clasificadores KNN*". Computación e Inteligencia Artificial. Universidad del País Vasco-Euskal Herriko Unibersitatea. Recuperado de: http://www.sc.ehu.es/ccwbayes/docencia/mmcc/docs/t9knn.pdf

Muñoz Pérez José (sin fecha). "*Sistemas Autoorganizativos*". Universidad de Málaga, departamento de Lenguajes y Ciencias de la Computación. Recuperado de: http://www.lcc.uma.es/~munozp/documentos/modelos_computacionales/temas/Tema6MC-05.pdf

National Institute of Standards and Technology (2006). "*Guide to Computer Security Log Management*". Recuperado de: https://www.govinfo.gov/content/pkg/GOVPUB-C13-52c3b5520393598b18782a7b55fde7e6/pdf/GOVPUB-C13-52c3b5520393598b18782a7b55fde7e6.pdf

Nieva Richard (2017). "*Facebook cierra chatbots que crearon un lenguaje secreto*". Cnet.com. Recuperado de: https://www.cnet.com/es/noticias/facebook-cierra-chatbots-que-crearon-un-lenguaje-secreto/

Nodus Trends (2017). "*Qué es y para qué sirve un servidor 'proxy"*". Periódico online Europapress. Recuperado de: https://www.europapress.es/portaltic/internet/noticia-sirve-servidor-proxy-20170403085932.html

Lafuente Ainoha (2018*). "Reducción de la dimensionalidad (o porqué más datos no siempre es mejor"*. Blog de divulgación Aureka. Recuperado de: https://aukera.es/blog/reduccion-dimensionalidad/

Lahoz-Beltrá Rafael (2004) "*Bioinformática, simulación, vida artificial e inteligencia artificial"*. Ediciones Díaz de Santos, Madrid.

Lara Velázquez Pedro, Gallardo López Lizbeth, Gutiérrez Miguel Ángel (2000). "*Asignación de frecuencias en telefonía celular aplicando el problema de coloración robusta*". Revista Matemática: Teoría y Aplicaciones, edición 2000, 16(2): 231–239.

Levín Mangin Jean Pierre, Flórez López Raquel y Fernández José Miguel (2008) "*Las redes Neuronales Artificiales, fundamentos teóricos y aplicaciones prácticas"*. Universidad de León, Netbiblio, León, España.

Linthicum David (2018). *"3 common machine learning mistakes to avoid*". Revista digital InfoWorld, artículo 29 de Junio de 2018. Recuperado de: https://www.infoworld.com/article/3285969/3-common-machine-learning-mistakes-to-avoid.html

Longvic C. (2001). "*The BSD Syslog Protocol*". Cisco Systems. Recuperado de: https://www.rfc-editor.org/rfc/pdfrfc/rfc3164.txt.pdf

Ortega Priego Jose Luis (2004). "*Análisis del consumo de Información de una revista electrónica: análisis de ficheros log de cibermetrics*". Revista Española de Documentación Científica Vol 27, nº 4, año 2004. Recuperado de: http://redc.revistas.csic.es/index.php/redc/article/view/159/213

Olarte William, Botero Marcela, Cañon Benhur (2010). Revista Scientia et Technica, Nº 45, Agosto de 2010. Universidad Tecnológica de Pereira. ISSN 0122-1701. Recuperado de: https://dialnet.unirioja.es/descarga/articulo/4546591.pdf

Ozertem Umit, Chapelle Olivier, Velipasaog Emre y Donmez Pinar (2012) "*Learning to Suggest: A Machine Learning framework for Ranking Query Suggestions*". Publicado en la 35 international ACM SIGIR conference on Research and

development in information retrieval en Oregón, Estados Unidos, páginas 25-34. Recuperado de: http://olivier.chapelle.cc/pub/fp487-ozertemPS.pdf

Pardo Vasallo Luis Miguel y Gómez Pérez Domingo (2010). "*Las gramáticas formales. Cómo describir un lenguaje formal*". Universidad de Cantabria, Asignatura de *Teoría de Autómatas y Lenguajes Formales*. Recuperado de: https://ocw.unican.es/pluginfile.php/1516/course/section/1946/1-3_Gramaticas_formales.pdf

Paruchuri Vik (2013). "*Natural Language Processing Tutorial"*. Recuperado de: http://www.vikparuchuri.com/blog/natural-language-processing-tutorial/

Peluffo Ignacio, Capobianco Marcela, Echaiz Javier (2014). *"Machine Learning aplicado a la detección de intrusos*". Universidad Nacional de la Plata, Argentina. Recuperado de: http://sedici.unlp.edu.ar/handle/10915/43264

Pérez César y Santín Daniel (2007). *"Minería de Datos: Técnicas y Herramientas"*. Madrid: Ediciones Paraninfo, S.A.

Pértega Díaz Sonia, Pita Fernández Salvador (2004). *"Asociación de variables cualitativas: test de Chi-Cuadrado*". Fisterra, atención primaria de salud en la red. Recuperado de: https://www.fisterra.com/mbe/investiga/chi/chi.asp

Pranav Dar (2018). "*13 Common mistakes Amateur Data Scientists make and how to avoid them*". Blog analítico. Recuperado de: https://www.analyticsvidhya.com/blog/2018/07/13-common-mistakes-aspiring-fresher-data-scientists-make-how-to-avoid-them/

Raghunarayan Rajiv (2018). "*Antivirus is dead: How AI and machine learning will drive cybersecurity*". Revista Techbeacon. Recuperado de: https://techbeacon.com/antivirus-dead-how-ai-machine-learning-will-drive-cybersecurity

Ramírez Vicente (2018) "*BigDataPerdida ¿Qué es el Deep Learning? Diferencias con el Machine Learning y la Inteligencia Artificial*". Revista digital BigDataMagazine. Recuperado de: https://bigdatamagazine.es/que-es-el-deep-learning-diferencias-con-el-machine-learning-y-la-inteligencia-atificial

Ramírez Vicente (2018 Diciembre). "*El 58% de las empresas ven el análisis de datos como un activo estratégico para su negocio, cifra que asciende al 70% en el caso de España.*". Revista digital BigDataMagazine. Recuperado de: https://bigdatamagazine.es/el-87-de-las-grandes-empresas-ya-han-invertido-en-machine-learning-o-piensan-hacerlo-en-los-proximos-dos-anos

Ramón Gustavo S. (sin fecha) "*Correlación entre variables*". Universidad de Antioquía. Recuperado de: http://viref.udea.edu.co/contenido/menu_alterno/apuntes/ac36-correlacion-variables.pdf

Redusers (2019). "Investigadores belgas descubren un método para engañar a las cámaras". Redusers.com, Comunidad de tecnología. Recuperado de: http://www.redusers.com/noticias/investigadores-belgas-descubren-metodo-enganar-camaras-ia/

Rodríguez Gámez Orlando, Hernández Perdomo Reynaldo, Torno Hidalgo Leonardo, García Escalona Leonid y Rodríguez Romero Roland (2005). "*Telefonía móvil celular: origen, evolución, perspectivas*". Sistemas de información científica: Red de revistas científicas de América Latina y el Caribe, España y Portugal. Revista trimestral: Enero-Marzo de 2005. Recuperado de: https://www.redalyc.org/html/1815/181517913002/

Rodríguez Olim Daniel José (2017). "*Business Process Transformation to Dynamize and Improve Fulfilment and Assurance Agility in CSPs*"Tesis doctoral de Informática y Telecomunicaciones de la Universidad Técnica de Lisboa. Recuperado de: https://fenix.tecnico.ulisboa.pt/downloadFile/281870113703808/METI-73971-ist-thesis-msc.pdf.pdf

Rodríguez Rama José Manuel (2018). "*Aplicación de técnicas de Machine Learning a la prevención de ataques*". Trabajo de fin de Máster en Seguridad de las TIC, para la Universidad de la Universidad Oberta de Catalunya. Recuperado de: http://openaccess.uoc.edu/webapps/o2/bitstream/10609/81126/11/jmrodriguez85TFM0618memoria.pdf

Roman Victor (2019). *"Introducción a Machine Learning, una guía desde cero"*. Recuperado de: https://medium.com/datos-y-ciencia/introduccion-al-machine-learning-una-gu%C3%ADa-desde-cero-b696a2ead359

Rubio Pablo Juan (2018) "*Cómo evitar crear un monstruo de Machine Learning*". GoodRebels. Recuperado de: https://www.goodrebels.com/es/como-evitar-crear-un-monstruo-de-machine-learning/

Salas Velasco Manuel (1996) "Estadística española" Vol. 38, Núm. 141, págs. 193 a 217. Universidad de Granada. Departamento de Economía Aplicada.

Sánchez Muñoz Jose Manuel (2016). "*Análisis de calidad cartográfica mediante el estudio de la matriz de confusión*". Revista de Investigación, Volum VI, número 2 009-026, ISSN 2174-0410. Recuperado de: https://dialnet.unirioja.es/descarga/articulo/6522539.pdf

SAS (sin fecha). *"5 Machine Learning Mistakes and how to avoid them*" Empresa dedicada a Machine Learning, entre otras soluciones IT. Recuperado de: https://www.sas.com/en_us/insights/articles/big-data/5-machine-learning-mistakes.html#mdd-industries

Schoenbaum Dan (2018). "*How to Leverage AI to Predict (and Prevent) Customer Churn".* Towards Data Science. Recuperado de: https://towardsdatascience.com/how-to-leverage-ai-to-predict-and-prevent-customer-churn-f84d653a76fb

Seif George (2018). "*An easy introduction to Natural Languaje Processing. Using computers to understand human language*". Towards data science. Recuperado de: https://towardsdatascience.com/an-easy-introduction-to-natural-language-processing-b1e2801291c1

Sims Scott (2015). *"Procesamiento del lenguaje natural*". Blog_Big Data. Recuperado de: https://madridschoolofmarketing.es/blog/big-data/pln-procesamiento-del-lenguaje-natural

Sosa Eduardo (1997). "*Procesamiento del lenguaje natural: revisión del estado actual, bases teóricas y aplicaciones (Parte I)".* El profesional de la información, revista internacional, científica y profesional. Recuperado de: http://www.elprofesionaldelainformacion.com/contenidos/1997/enero/procesamiento_del_lenguaje_natural_revisin_del_estado_actual_bases_tericas_y_aplicaciones_parte_i.html

Van Rossum Guido (2009)*"Tutorial de Python"*. Python Fundation. Recuperado de: http://docs.python.org.ar/tutorial/pdfs/TutorialPython2.pdf

Veeramachaneni Kalyan, Arnaldo Ignacio, Vamsi Korrapati (2016). "*AI^2: Training a Big Data Machine to Defend*". Segunda conferencia Internacional de Seguridad en Big Data en la Nube, Nueva York, Estados Unidos. Recuperado de: https://people.csail.mit.edu/kalyan/AI2_Paper.pdf

Vicente Carlos (2008). "*Gestión de logs*" Universidad de Oregon. Recuperado de: https://nsrc.org/workshops/2008/walc/presentaciones/monitorizacion_pasiva.pdf

Walid Ghobar Enio (2017).*"Un sistema de recomendación basado en perfiles generados por agrupamiento y asociaciones*". Trabajo de fin de Máster en Ingeniería y Tecnología de Sistemas Software para la Universidad Politécnica de Valencia. Recuperado de: https://riunet.upv.es/bitstream/handle/10251/94049/WALID%20-%20Un%20sistema%20de%20recomendaci%C3%B3n%20basado%20en%20perfiles%20generados%20por%20agrupamiento%20y%20asociaciones.pdf?sequence=1

Referencia de imágenes

Antena3.com. (s.f.). *https://www.antena3.com/noticias/mundo/nina-seis-anos-encarga-accidentalmente-casa-munecas-galletas-mientras-hablaba-asistente-voz-amazon_201701095873bf750cf2187c0d4e327e.html.*

CDS official website. (s.f.). *https://www.hpcds.com/es/.*

Citeseerx website. (s.f.). *http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.140.5102&rep=rep1&type=pdf.*

Citeseerx Website. (s.f.). *http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.140.5102&rep=rep1&type=pdf.*

Culturacion website. (s.f.). *http://culturacion.com/la-historia-del-telefono-movil-origen-pasado-y-presente/.*

Elaboración propia. (s.f.).

Elaboración propia. (s.f.). *Jerarquía Noam Chomsky.*

EldiarioNuevoDia.com. (s.f.). *https://www.eldiarionuevodia.com.ar/tecnologia/2017/7/28/facebook-perdi-control-sistema-inteligencia-artificial-tom-vida-propia-43468.html.*

Hallam-Baker P. M. (1996). *Behlendorf B.*

HP eIUM Overview Guide. (s.f.). *https://support.hpe.com/hpsc/doc/public/display?docId=emr_na-a00025680en_us.* pág. 17.

HP eIUM Overview Guide, p. 2. (Third Edition). *http://h20628.www2.hp.com/km-ext/kmcsdirect/emr_na-a00026059en_us-1.pdf.*

HP Official Website. (s.f.). *https://www8.hp.com/es/es/home.html.*

HPE official website. (s.f.). *https://www.hpe.com/es/es/support.html.*

Intple website. (s.f.). *http://www.intple.com/product-page_13/MasterOne_en_copy.html.*

Lidgi González. (s.f.). *http://ligdigonzalez.com/libreria-scikit-learn-de-python/introduccion-a-la-libreria-scikit-learn/.*

Redusers.com. (s.f.). *http://www.redusers.com/noticias/investigadores-belgas-descubren-metodo-enganar-camaras-ia/.*

Stadista portal de estadísticas. (s.f.). *https://es.statista.com/estadisticas/545506/redes-moviles-numero-de-estaciones-base-en-espana/.*

Technoistoria Website. (s.f.). *https://sites.google.com/site/tecnohistoria1975/AlexanderGBell.*

Xatakamovil. (s.f.). *https://www.xatakamovil.com/movil-y-sociedad/martin-cooper-inventor-del-movil-en-una-entrevista-en-la-television-norteamericana.*