Facultade de Informática
UNIVERSIDADE DA CORUÑA

TRABALLO FIN DE GRAO
GRAO EN ENXEÑARÍA INFORMÁTICA
MENCIÓN EN TECNOLOXÍAS DA INFORMACIÓN

Detección de linguaxe misóxino e xenófobo en redes sociais mediante aprendizaxe máquina

Estudante: Laura Rodríguez Fernández
Dirección: Víctor Manuel Carneiro Díaz
Diego Fernández Iglesias

A Coruña, setembro de 2020.

Ao meu eterno compañeiro

Agradecementos

Gustaríame dar as grazas en primeiro lugar aos meus directores de proxecto, Víctor e Diego, por tódalas horas que lle adicaron a este proxecto e pola súa paciencia á hora de explicarme todo.

Tamén quero agradecerlle a miña familia e amigos polo apoio e ánimos que me deron durante todos estes anos.

Resumo

Co incremento do uso das redes sociais, xurde a necesidade de ter todo máis controlado para evitar casos de abuso verbal, discriminación, acoso... Twitter é unha rede social que funciona mediante o envío posts de usuarios, e na que xorden moitos debates e discusións, polo que é bastante habitual ver este tipo de problemáticas.

O obxectivo principal deste traballo é a clasificación de posts de Twitter, para comprobar se conteñen linguaxe despectivo ou expresións de odio cara as mulleres e inmigrantes. Para isto empréganse técnicas de machine learning seguindo a metodoloxía CRISP-DM, a cal consta de 6 fases.

Seguindo as fases desta metodoloxía, analízase e compréndese o dataset que contén os datos, para posteriormente poder obter as características que emprega o algoritmo de Random Forest para a creación do modelo. Para validar este modelo empréganse varios métodos de validación, co fin de obter o modelo que presente mellores resultados.

Despois de todo este proceso e axustar o modelo o mellor posible, chegamos a unha das últimas fases, a avaliación, na cal se aplican distintas métricas para obter os resultados. Cabe destacar que o mellor resultado que se acada é un 78.16% para a métrica de precisión, mellorando ata un 13.16% as precisións obtidas no estado do arte.

Abstract

With the increase in the use of social networks, the need airises to have everything more controlled to avoid cases of verbal abuse, discrimination, harassment... Twitter is a social network that works by sending user posts, and in which many debates an discussions, so it is quite common to see such problems.

The main objective of this work is the classification of Twitter posts, to check if they contain derogatory language or expressions of hatred towards women and immigrants. For this, machine learning techniques are used following the CRISP-DM methodology, which consists of 6 phases.

Following the phases of this methodology, the dataset containing the data is analyzed and understood, in order to subsequently be able to obtain the characteristics used by the Random Forest algorithm for the creation of the model. To validate this model several validation methods are used in order to obtain the model that presents better results.

After all this process and adjusting the model as best as possible, we come to one of the last phases, the evaluation, in which different metrics are applied to get the results. It is worth

noting that the best result is 78.16% for the precision metric, improving the accuracy obtained in the state of the art to 13.16%.

Palabras chave:

- Aprendizaxe máquina
- Random Forest
- Algoritmos de similitude
- Árbores de decisión
- Clasificación
- Validación cruzada

Keywords:

- Machine language
- Random Forest
- Similarity algorithms
- Decision trees
- Classification
- Cross Validation

Índice Xeral

Índice de Figuras

Índice de Táboas

Capítulo 1

Introdución

O uso das redes sociais está cada vez máis presente no noso día a día e con fácil acceso para todos. Por iso, debido á gran cantidade de datos que se manexan é complexo realizar controis para asegurarse de que os usuarios as empregan correctamente, e non de forma daniña, con condutas que poden causar dano a outras persoas. Outro dos motivos polos que é importante ter un control sobre as interaccións nas redes sociais, é evitar que estas condutas influencien a outros usuarios, como poden ser os nenos.

Nas redes sociais podes permanecer anónimo, o que provoca que moita xente acose, insulte, intimide, manipule... a outras persoas chegando a causar danos psicolóxicos.

Figura 1.1: Acoso en redes sociais

En moitas plataformas implantáronse algoritmos para controlar, en certo modo, o seu uso e detectar os posts non axeitados para non permitir a súa publicación e difusión. Nisto é no que se centra este proxecto tamén, na detección de posts non axeitados na rede social Twitter, que conta con uns 340 millóns de usuarios activos.

O que se pretende é analizar o post que envía o usuario para detectar un comportamento non axeitado. Neste caso analizarase a presenza de termos de odio ou comportamento agresivo dirixido a mulleres ou inmigrantes. Isto permitirá realizar unha acción sobre ese post, como pode ser a eliminación, ou sobre o usuario, de ser posible identificalo, para evitar posibles casos de acoso ou contido que incita ao odio e a violencia.

A análise realizarase coa extracción de características do post e con cálculos de similitu-

des entre tódolos tweets do dataset, para posteriormente obter o resultado final empregando técnicas de aprendizaxe máquina.

1.1 Por que aprendizaxe máquina?

Como xa dixemos antes, manéxanse unha gran cantidade de datos, o que fai imposible que se poidan revisar todos os posts manualmente. Por iso unha boa solución é automatizar o proceso facendo uso do Aprendizaxe Máquina (*Machine learning, ML*), do cal falaremos na sección 3.1.

A finalidade deste traballo é detectar se o post é axeitado ou non, é dicir, clasificalo a partir da análise do seu contido, e un dos procedementos máis estendidos para facer isto é empregando unha das técnicas de ML, a de clasificación.

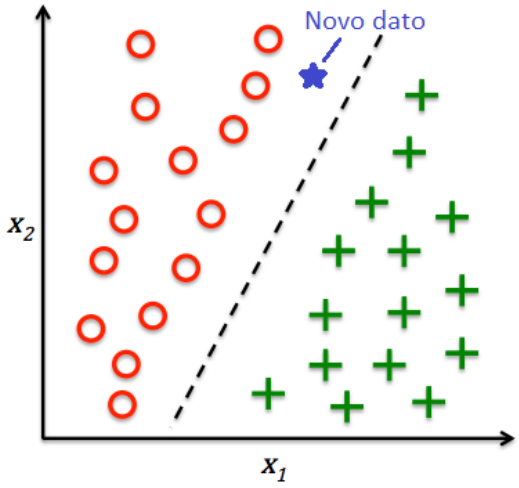

Figura 1.2: Exemplo de clasificación dun novo dato nun conxunto

Un tema que está moi presente actualmente, é o COVID-19, para o cal tamén se fixeron estudos empregando este tipo de técnicas. Así, por exemplo, recentemente publicouse un traballo [1] no que se empregan as técnicas de ML para unha clasificación rápida, escalable e precisa dos xenomas do virus.

Outro traballo que podemos atopar cunha finalidade similar á deste, é un levado a cabo na Universidade Politécnica de Valencia para a detección de linguaxe sexista en publicacións do BOE [2], empregando algoritmos como o Naive Bayes, que é un algoritmo de ML.

1.2 Obxectivos

A continuación, lístanse os obxectivos a alcanzar no proxecto co fin de facilitar a detección en redes sociais deste tipo de comportamentos non apropiados:

- Estudo do estado do arte de técnicas de machine learning aplicadas a detección de usuarios con linguaxe ou comportamentos agresivos. Analizaranse outros traballos que axudarán a determinar as técnicas máis axeitadas.
- Análise do contido dos tweets do dataset: permitirá determinar os tweets máis axeitados e atopar as mellores características para o procesamento en técnicas de aprendizaxe máquina. Para levar a cabo este obxectivo, realizarase un preprocesamento do dataset.
- Análise dos distintos métodos de cálculo de similitude entre vectores resultantes do paso anterior e elección dos máis axeitados en función do contexto e dos algoritmos a empregar para o aprendizaxe. Este paso engloba os cálculos de similitude e a selección de características, empregando as técnicas que permitan o escalado e a optimización destas operacións. Será levado a cabo nun proceso interactivo a través de diversos experimentos ata conseguir unha execución óptima dos indicadores de similitude e os seus estadísticos asociados.
- Análise e selección dos algoritmos de aprendizaxe máquina adecuados ao contexto. Parametrización e desenvolvemento do algoritmo de aprendizaxe máquina seleccionado mediante ferramentas de fontes abertas.
- Análise de resultados e conclusións, onde se fará unha comparativa entre os distintos resultados obtidos na experimentación.

1.3 Estrutura da memoria

A continuación relátase a estrutura deste documento:

- **Introdución** 1: introdución o problema a abordar no traballo e os obxectivos a acadar.
- **Metodoloxía** 2: metodoloxías empregadas no proxecto, a planificación e os costes.
- **Entender o negocio** 3: explicación dos conceptos necesarios para entender o proxecto, das ferramentas empregadas para levalo a cabo e citación de proxectos co mesmo problema a abordar ou similar.
- **Entender os datos** 4: descrición do dataset e o seu contido.

- **Preparar os datos** 5: explicación do tratamento ao que se somenten os tweets do dataset: limpeza, transformación e selección de características.
- **Modelado** 6: explicación do proceso levado a cabo para a creación dos modelos, así como as validacións realizadas.
- **Avaliación** 7: exposición dos resultados obtidos e explicación das probas realizadas para obtelos.
- **Conclusións** 8: trata sobre as conclusión que obtemos despois de realizar o traballo, os obxectivos e leccións aprendidas. Tamén contén unha parte de liñas futuras, que cita funcionalidades que se poderían abordar nun futuro como unha continuación deste proxecto.

Capítulo 2

Metodoloxía

Existen diversas metodoloxías empregadas á hora de levar a cabo un proxecto de minería de datos. As principais técnicas orixináronse na década dos 90, cando o término KDD (Knowledge Discovery in Databases) era usado para referirse ao concepto de atopar coñecemento nos datos, e xa finais dos 90, co obxectivo de normalizalo, xorden dúas metodoloxías principais: SEMMA (Sample, Explore, Modify, Model, ASSESS) e CRIPS-DM (Cross-Industry Standar Process for Data Mining). Ambas metodoloxías describen un proceso, no cal concretan unha serie de fases con obxectivos en cada unha delas, e os pasos a realizar para chegar a acadalo.

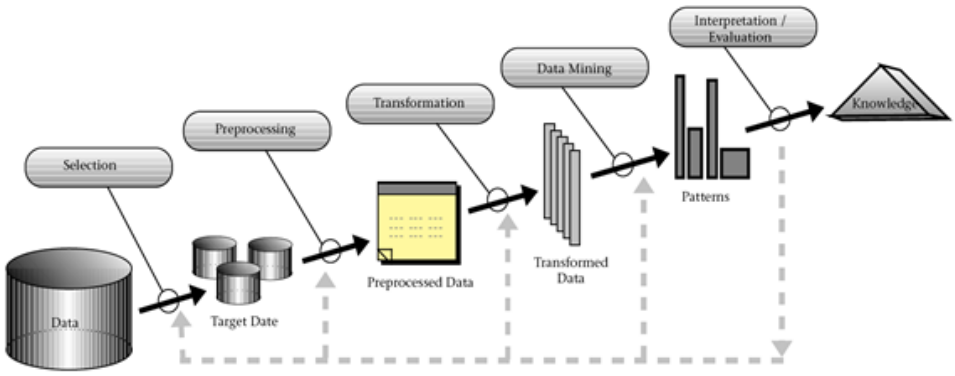

Figura 2.1: Diagrama KDD

Fuente: researchgate.net

Popularmente foi adoptada a metodoloxía CRISP-DM, xa que, aínda que ambas son moi similares, CRISP-DM (figura 2.3) é máis completa e ten en conta a aplicación ao entorno de negocio dos resultados, mentres que SEMMA (figura 2.2) focalízase máis nas tarefas á levar a

cabo para o desenvolvemento do modelo e obvia a parte do entorno de negocio.[3]

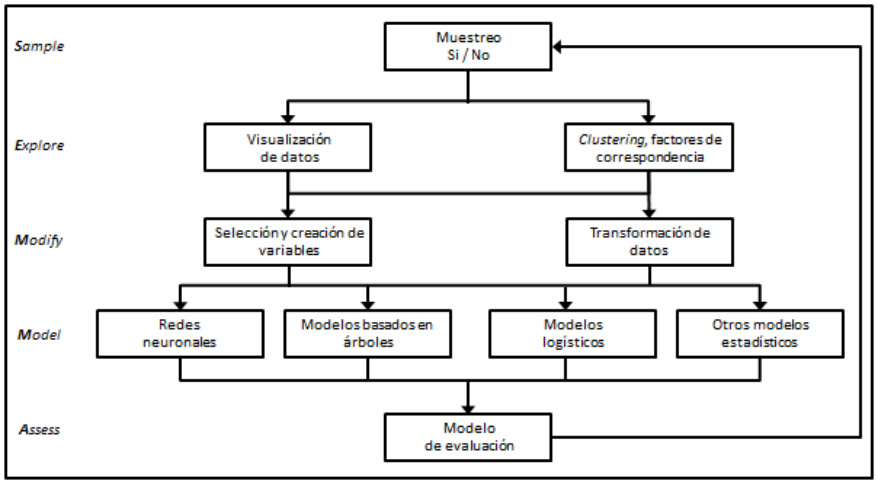

Figura 2.2: Diagrama SEMMA

Fuente: researchgate.net

Debido a que a metodoloxía CRISP-DM é a máis próxima ao concepto real de proxecto, e que as súas fases adecúanse perfectamente a este traballo, será esta a metodoloxía empregada para a realización das tarefas que o comprenden, como a comprensión dos datos ou o modelado.

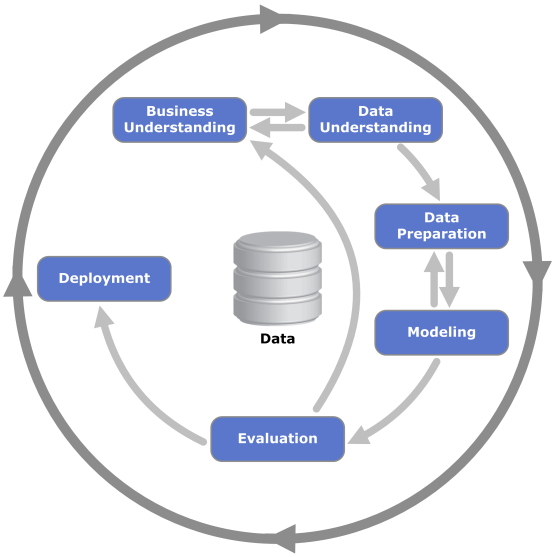

Figura 2.3: Diagrama CRISP-DM

Fuente: wikipedia.org

2.1 CRISP-DM

A metodoloxía CRISP-DM foi deseñada en 1996 por un consorcio de empresas europeas, co obxectivo de crear un esquema que permitise mostrar o ciclo de vida dun proxecto de minería de datos. Tivo unha gran acollida debido a que é independente da ferramenta que se empregue no desenvolvemento do proxecto, e a que a súa distribución é libre e gratuíta.[4]

CRISP-DM esta formada por seis fases e cada unha das fases inclúe unhas serie de tarefas agrupadas en catro niveis de abstracción (figura 2.4, sendo os dous primeiros tarefas xenéricas e o terceiro, tarefas especializadas.

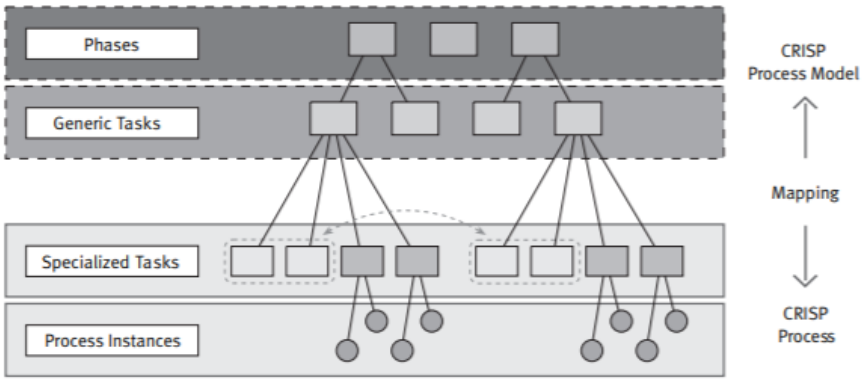

Figura 2.4: Metodología CRISP-DM

Fuente: SPSS

A continuación unha breve descrición das fases: [5]:

- **Entender o negocio**: esta fase inicial está centrada na comprensión dos obxectivos e requisitos do proxecto desde o punto de vista do negocio. Esta fase permitirá definir os requisitos e establecer os obxectivos que se queiran acadar.
- **Entender os datos**: esta fase comprende a recompilación inicial dos datos e as actividades que permiten familiarizarse con eles, co fin de obter un coñecemento e comprensión de calidade sobre os mesmos.
- **Preparar os datos**: esta fase abarca todas as actividades necesarias para crear o conxunto de datos que será empregado para crear o modelo. Inclúe a limpeza de datos e a súa transformación e selección para as ferramentas que modelan.
- **Modelado**: a fase de modelado céntrase nas tarefas que crean o modelo: hiperparametrización, adestramento e validación. O obxectivo desta fase é a obtención dun modelo que cumpra cos requisitos establecidos na primeira fase.

- **Avaliación**: nesta etapa do proxecto xa se obtivo un modelo, o cal hai que avaliar. Esta fase avalía o modelo para asegurarse de que cumpre cos obxectivos.
- **Lanzamento**: esta fase indica a posta en marcha do proxecto dentro da organización, onde se realizará a correspondente documentación, presentación de resultados e o mantemento da aplicación.

Ademais, a secuencia das fases non é ríxida. Como se pode ver na figura 2.3, permítese o movemento cara adiante e cara atrás entre diferentes fases. O resultado de cada fase determina que fase, ou que tarefa concreta dunha fase, hai que facer despois. Tendo en conta que se trata dun proceso iterativo, as frechas indican simplemente as dependencias máis importantes e frecuentes.

2.2 Planificación

Mentres que para a realización das tarefas do proxecto usouse a metodoloxía CRISP-DM 2.1, para a planificación do proxecto escolleuse Scrum, que é unha metodoloxía de desenvolvemento áxil.

A finalidade desta metodoloxía é entregar valor en períodos curtos de tempo (sprint), e está baseada fundamentalmente nos seguintes alicerces [6]:

- Transparencia: existe un coñecemento común do proxecto entre todos os implicados.
- Inspección: existe unha inspección periódica do proxecto para saber que o traballo segue adiante e o equipo funciona correctamente.
- Adaptación: cando existan cambios, o equipo adáptase para conseguir o obxectivo do sprint.

En Scrum existen tres roles importantes:

- Product owner: é o responsable de maximizar o valor do traballo do equipo de desenvolvemento e é o único que fala co cliente. Esta persoa pode ser tamén membro do equipo de desenvolvemento.
- Scrum Master: é o responsable de que as técnicas Scrum sexan entendidas e aplicadas.
- Scrum Team: son os responsables de levar a cabo as tarefas priorizadas polo *Product owner*. É un equipo auto-organizado e multifuncional: cada membro é responsable das súas tarefas e debe rematalas no tempo acordado. Non teñen sub-equipos nin especialistas, a responsabilidade do equipo é compartida.

Figura 2.5: Roles

O desenvolvemento de Scrum é iterativo, realízase en iteracións ou sprints, e por cada sprint ocorren os seguintes sucesos:

- Sprint: un spring é cada unha das iteracións que se levan a cabo para finalizar o proxecto e está formado por varias tarefas que teñen como finalidade aportar unha funcionalidade que poida comprobar o cliente. Se o sprint e moi longo, pode supor perda de información para o cliente.

- Sprint Planning: reunión na cal todo o equipo define que tarefas se farán e o obxectivo do sprint.

- Daily Meeting: reunión diaria dentro do sprint. Faise un control dos avances das tarefas e se xurdiron impedimentos.

- Sprint Review: revisión do valor que se entregará. Realízase ao final de cada sprint e preséntaselle o desenvolto ata o momento ao cliente.

- Sprint Retrospective: reunión na que se fai unha avaliación de como se implementou a metodoloxía.

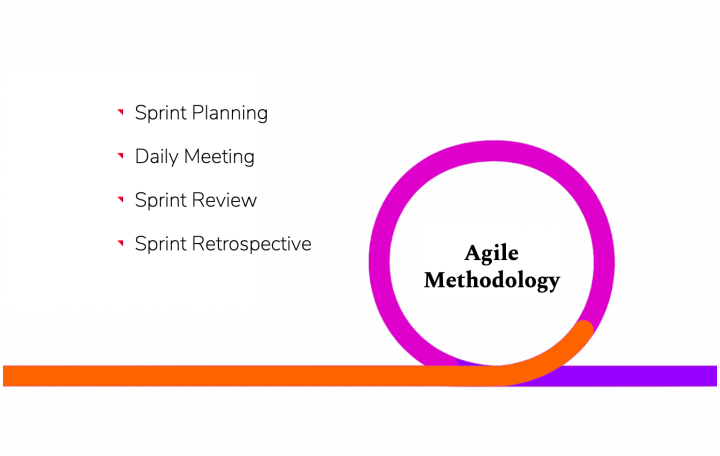

Figura 2.6: Metodoloxía Áxil

Seguindo esta metodoloxía de planificación de Scrum e a metodoloxía empregada para as tarefas, CRISP-DM, realizouse unha estimación do tempo empregado separando as tarefas por sprints, e foi a seguinte:

Tarefa	Tempo en horas
Reunión inicial + Sprints Planning + Sprints Reviews	42
Sprint 1: Entender o negocio	32
Sprint 2: Entender os datos Descrición Exploración	18
Sprint 3: Preparar os datos Limpeza Transformación Selección	60
Sprint 4: Modelado Asunción de modelos Adestramento Validación	48
Sprint 5: Avaliación	40
Sprint 6: Redacción da memoria	54
Sprint 7: Presentación	5
Total	299

Táboa 2.1: Táboa planificación

Agrúpanse todas as reunións realizadas na mesma tarefa para simplificar, e o resto de tarefas, correspondéndose coas fases da metodoloxía CRISP-DM 2.1, forma cada unha un sprint diferente ao que se lle estimou unha duración en horas.

2.3 Presuposto

Neste apartado exponse o custo do traballo, tendo en conta que o software empregado é gratuíto, soamente se terán en conta as horas traballadas.

Consultando os datos do *XVII Convenio colectivo estatal de empresas de consultoría, y estudios de mercados y de la opinión pública* do BOE [7], considérase que un xefe superior cobra uns 15€/h e un analista, 14€/h, e estimando as horas empregadas neste traballo, o custo total sería duns 6406€.

	Horas	Custo por hora	Total
Xefe 1	74	15€/h	1110€
Xefe 2	74	15€/h	1110€
Analista	299	14€/h	4186€
			6406€

Táboa 2.2: Tabla presupostos

Capítulo 3

Entender o negocio

Entender o negocio implica coñecer os fundamentos teóricos e tecnolóxicos e as ferramentas existentes que permiten levar a cabo os obxectivos propostos. A continuación explícanse as ferramentas que foron empregadas durante o traballo.

3.1 Aprendizaxe máquina

O aprendizaxe máquina (Machine Learning) é unha disciplina do ámbito da Intelixencia Artificial que busca desenvolver algoritmos e software que aprendan automaticamente baseándose nos datos, e permitan mellorar o seu rendemento e resultados. Estas técnicas (ML) aprenden a identificar patróns e son capaces de predicir comportamentos futuros. Os algoritmos de ML pódense clasificar atendendo a diferentes tipoloxías, e segundo o tipo de aprendizaxe requerido, pódense clasificar en:

- **Supervisado**: o algoritmo ten información sobre as categorías dos datos e clasifica os novos datos nunha categoría a partir dos patróns que identificou previamente para cada unha delas.

- **Non supervisado**: o algoritmo non ten información sobre as categorías dos datos polo que se vai modificando para ser capaz de recoñecer patróns e clasificar os novos datos.

Debido á natureza do proxecto, vámonos centrar no primeiro tipo, o supervisado, concretamente nun sistema de clasificación, para predicir a que tipo pertencerá o tweet a analizar. Este sistema vai separar os datos en diferentes categorías a partir da variable dos datos que lle indiquemos, no noso caso tratarase dunha clasificación binaria, 2 categorías.
Para levar a cabo dita clasificación, emprégase o algoritmo de *Random Forest.* 3.2

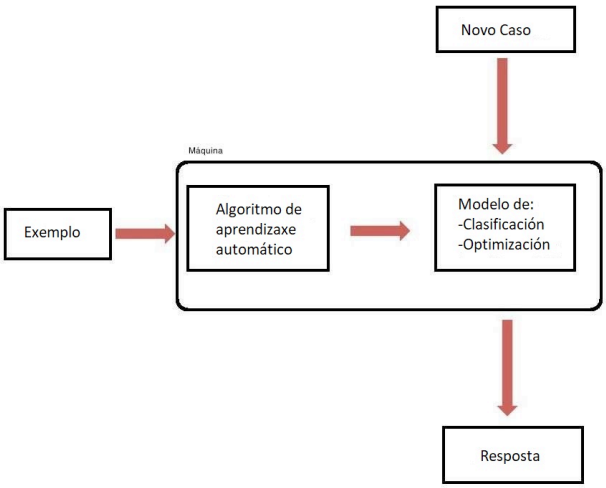

Figura 3.1: Esquema

3.2 Random Forest

O Random Forest (*RF*) é un algoritmo de aprendizaxe automático que basa o seu comportamento en árbores de decisión.

Unha Árbore de Decisión (Decision Tree), como vemos no exemplo da figura 3.2, é un método de predición que axuda a toma de decisións a través da representación dun conxunto xerárquico en forma de árbore. Cada nodo da árbore é una bifurcación que representa unha característica, ata chegar ao final da árbore e obter unha clasificación, é dicir, cada nodo representa a unha pregunta de si ou non e todas elas levan a un resultado final.

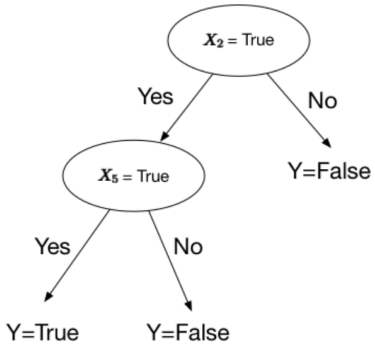

Figura 3.2: Exemplo Árbore de decisión

O principal problema das árbores de decisión é o sobreaxuste, para evitalo o RF emprega varias árbores e introduce a aleatoriedade: crea árbores independentes de mostras distintas, isto é, para crear unha árbore emprega unha porcentaxe das mostras totais, e o resto das mos-

tras son usadas para validala e obter un promedio dos resultados obtidos de tódalas árbores.

Estas mostras empregadas para a creación da árbore son escollidas con reemplazo, denominado como *Bagging*, o que implica e que algunhas das mostras poden ser empregadas varias veces nunha árbore.

Por outra banda, para cada árbore o RF utiliza todas as características, e os seus nodos bifúrcanse levando a cabo unha clasificación binaria. Para escoller a característica empregada en cada nodo, o algoritmo decide entre $\sqrt{n}$ características, sendo n o total de características, e escolle unha delas, a que máis reduce a impureza de Gini [8] (probabilidade de que unha mostra se etiquete incorrectamente).

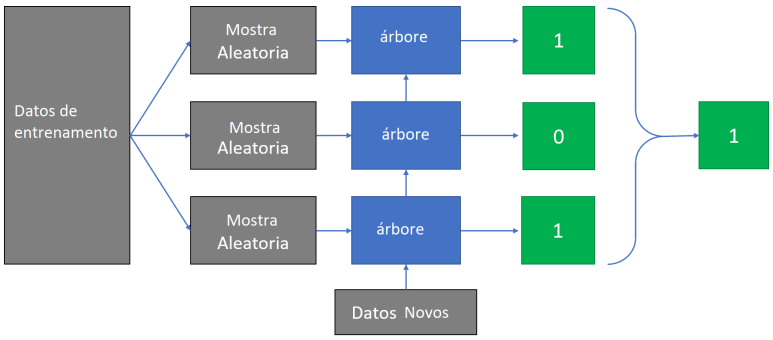

Figura 3.3: Random Forest Esquema

3.3 Validación Cruzada

Os conxuntos de datos que se empregan son creados ao azar, polo tanto é posible que non sempre se obteña o mesmo modelo nin as mesmas precisións. Por isto é unha boa alternativa a validación cruzada, ou *Cross Validation*, xa que obtén resultados máis precisos [9] e nos garantizan que son independentes das particións escollidas.

A validación cruzada consiste na división do conxunto de tódolos datos en K subconxuntos e iterar K veces 3.4.

En cada iteración hai K conxuntos, dos cales 1 será empregado como conxunto de avaliación ou proba e o resto serán empregados como o conxunto de adestramento, sendo o conxunto de proba distinto para cada iteración. A partir do conxunto de adestramento fórmase un modelo que será avaliado co conxunto de proba, obtendo uns resultados para o modelo creado nesa iteración.

Cando rematan as iteracións, calcúlase a media dos resultados obtidos en cada unha delas.

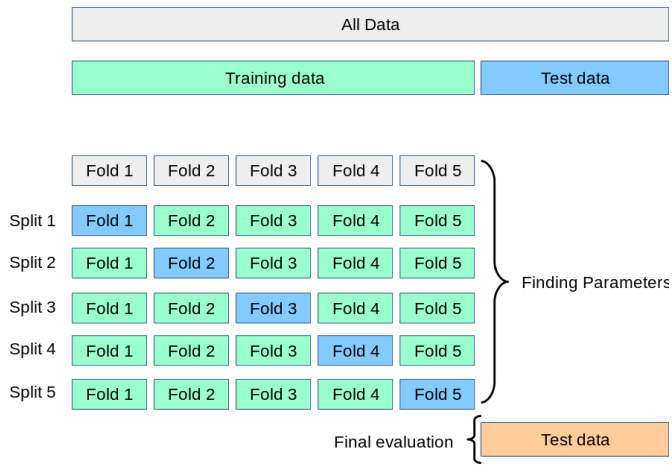

Figura 3.4: Validación cruzada con K=5 subconxuntos

3.4 R e RStudio

Para realizar o desenvolvemento do traballo vamos empregar a ferramenta estadística de R, que é moi común o seu uso en traballos de aprendizaxe máquina. É un programa de software libre e código aberto, presentado no ano 1993 por Robert Gentleman y Ross Ihaka e está enfocado ao cálculo e a realización de gráficas.

Para o seu manexo empregaremos RStudio, un dos entornos de desenvolvemento (IDE) máis coñecidos para esta linguaxe, xa que dispomos de algunha experiencia no seu uso despois de empregalo en materias durante o grao. Outros entornos poderían ser Vim ou RKWard [10].

Unha gran parte das súas funcións están escritas no seu linguaxe, R, e tamén podemos atopar algoritmos máis exixentes computacionalmente escritos en C, C++ ou Fortran [11]. Para ampliar as súas funcionalidades, dispón dunha gran cantidade de paquetes externos que se poden cargar para poder empregarlas.

De todos os paquetes que permite usar R, os máis relevantes que se usaron foron os seguintes:

- **Text2Vec** [12]: ten unha API que permite realizar análise de coleccións de documentos e procesamento de linguaxe natural (NPL). Dentro das súas utilidades, as que máis nos interesan son as ferramentas para vectorización de texto e o cálculo de similitudes. Para o cálculo de similitudes temos varias métodos que podemos escoller. Algúns deles son:

  - *Coseno*: mide a similitude entre dous vectores que conteñen cada un, neste caso, un conxunto de palabras polas que está formado, e empregando o contido dos vectores formado polas palabras que están no post, avalíase o valor do coseno do ángulo comprendido entre eles. Como resultado obtense valor 1 se o ángulo é 0, é

dicir, os vectores son iguais, e se son completamente distintos, apuntan en sentido contrario, o resultado sería -1.

$$similarity(doc_1, doc_2) = cos(\theta) = \frac{doc_1 doc_2}{|doc_1||doc_2|}$$

- *Coseno con LSA (Latent Semantic Analysis)*: xera un conxunto de conceptos relacionados cos documentos que analiza, e cos termos que estes documentos conteñen, asumindo que as palabras con un significado similar, aparecen en partes de texto similares. Estes datos son almacenados nunha matriz creada coa técnica matemática SVD (descomposición de valores singulares) para reducir os termos. Unha vez ten esta matriz, procede a calcular a similitude empregando a función do coseno.
- *Jaccard*: mide o grado de similitude entre dous conxuntos a partir dos elementos que teñen en común, independentemente do tipo destes. Toma valores entre 0, non comparten ningún termo, e 1, sendo 1 a igualdade total.

$$J(doc_1, doc_2) = \frac{doc_1 \cap doc_2}{doc_1 \cup doc_2}$$

- **Random Forest** : conten unha API que permite realizar operación para os problemas de clasificación ou regresión de Random Forest. Coa axuda deste paquete e as suas funcións, crearase o modelo e avalíase a partir das seguintes funcións, respectivamente:

$$randomForest(positivo\ ., data = train, ntree = ntree,$$

$$mtry = mtry, importance = TRUE)$$

$$predict(modelo, data)$$

Sendo *ntree* e *mtry* dous hiperparámetros, é dicir, son uns parámetros axustables que se empregan para a creación do modelo e que poden variar parámetros como o número de árbores crea durante o entrenamento. Ademais, permite obter a importancia de cada característica a partir da saída obtida da creación do modelo.

3.5 Orange

É un programa gratuíto para minería de datos e análisis preditivo creado na Universidade de Liubliana. Permite crear fluxos de traballo interativos para analizar e visualizar os datos. O programa procesa os datos e ofrécenos un listado de operacións para levar a cabo.

Neste proxecto foi empregado para a realización de gráficas a partir do dataset e das características obtidas deste.

3.6 Estado do arte

O ML é unha rama da intelixencia artificial moi estendida, por isto non é raro atopar traballos similares ou iguais a este traballo, aínda que os datos empregados ou as ferramentas non sexan as mesmas.

Neste apartado menciónanse algúns deses traballos similares e os resultados que acadaron.

- IRE-Project-hatEval-2019 [13]: este traballo está realizado en Python empregando técnicas de ML e fixo o estudo para detectar cada unha das características que ten o dataset, é dicir, realizou unha tarefa para detectar expresións de odio e outra para detectar comportamento agresivo ou a quen se dirixe o tweet. O traballo está centrado separou os resultados en tres tarefas: detección de expresións de odio, se vai dirixidigo a un colectivo e detección de comportamento agresivo. Realizou un preprocesado dos datos e creou dous modelos distintos, obtendo, para cada tarefa uns resultados. Nas figuras 3.5 e 3.6 podemos ver os resultados que obtivo para a primeira e a última tarefa, xa que son nas que nos centramos neste traballo:

Precision	[0.71264368 0.64450128]
Recall	[0.7574171 0.59016393]
F1-Score	[0.73434856 0.61613692]
ROC-AUC	0.6737905186965353

Figura 3.5: Resultados detección de expresións de odio

Precision	[0.71264368 0.64450128]
Recall	[0.7574171 0.59016393]
F1-Score	[0.73434856 0.61613692]
ROC-AUC	0.6737905186965353

Figura 3.6: Resultados detección de comportamento agresivo

- SemEval2019-Task5 [14]: deste traballo soamente se atopou o código de execución, sen documentación, pero podemos apreciar que está escrito en Python e realiza unha serie de operacións sobre os datos para obter uns resultados, como por exemplo a transformación dos datos.

- Detección de post de tráfico e contaminación [15]: este traballo, realizado en Python, consiste na detección de posts de Twitter que teñan relación con eventos de tráfico e

contaminación. Neste caso non emprega random forest, pero emprega outros algoritmos como Bayes Naive ou árbores de decisión, obtendo un 85% de exactitude no mellor dos casos. Para acadar o resultado final, vemos que realiza unha serie de pasos como os levados a cabo neste traballo.: recolección e preprocesado dos tweets, procesamento e clasificación.

- Fermi SemEval-2019 Task5 [16]: este traballo está realizado sobre o mesmo dataset que se empregou no noso proxecto. Lograron unha precisión do 65% empregando algoritmos de convinación de incrustación ML, en concreto está precisión foi acadada empregando máquinas de vectores de soporte, SVM.

Capítulo 4

Entender os datos

Esta fase do proxecto focalízase nos datos, para poder coñecelos e tratalos adecuadamente, empezando pola súa orixe.

4.1 Descrición

Os datos obtidos, e cos que se traballa, proveñen do *dataset* de Hate Speech usado no congreso SEMeval 2019 [17] e está composto por 9000 entradas.

Este dataset contén unha serie de posts de Twitter, nos cales, nalgúns recoñécense termos de odio e/ou comportamento agresivo cara aos colectivos de inmigrantes e mulleres, e outros son tweets normais sen estes comportamentos presentes.

A maiores do texto dos posts, hai tres columnas máis, que rexistran se o tweet contén a característica correspondente a esa columna (explicadas a continuación), e as cales serven para clasificalo.
En resumo, o contido do dataset é o seguinte:

- **id**: id numérica do tweet.
- **Text**: texto do tweet en inglés.
- **HS** (hate speech): valor binario (1|0) que indica se contén expresións de odio.
- **TR** (target classification): indica se vai dirixido a un individuo (1) ou a un colectivo (0).
- **AG** (aggressive behaviour): valor binario que indica se contén comportamento agresivo.

Hai que ter en conta que as columnas TR e AG dependen do valor da columna HS, é dicir, se a columna HS non ten valor 1, a TR e a AG terán valor 0 sempre, pero se a columna HS ten valor 1, estas dúas poden ter valor 0 ou 1.

4.2 Exploración

O dataset consta de 9000 filas, das cales podemos observar uns exemplos na figura 4.1. Aquí vemos un par de exemplos, pero en todo o dataset, ao ser linguaxe usado en redes sociais, e polo tanto, unha linguaxe máis coloquial, podemos atopar tamén algúns caracteres diferentes como emoticóns, siglas para abreviar expresións ou letras repetidas para mostrar máis énfase.

id	tweet	HS	TR	AG
31568	Bro is a bitch, fucking cunt https://t.co/dQ8JagmNCZ	0	0	0
30734	I'm rewatching Breaking Bad & I could see why I hated this bitch the first time I watched this show; Skyler is such a cunt & a fake as wife	1	1	0
31227	@charliekirk11 @asia742 364 down only about 12,499,636 more to go!#BuildThatWall	0	0	0
31224	>Harasses women and calls them bitch and crabby >They block him >Plays the victim https://t.co/gmraGDtcQm	0	0	0
31139	Bitch you ain't gotta call my phone, matter fact all you hoes can leave me alone!! ??	0	0	0
34076	@ABCPolitics A illegal aliens showing exactly what they were coming to this country for in the first place. #sendthemback!!!!!	1	0	1
32479	Bitch....how the fuck. Siri you a dumb ass hoe https://t.co/nldwbLrK10	1	1	0
32985	#FactCheck true #TuesdayThoughts #WalkAwayFromDemocrats #WakeUpAmerica #VoteDemsOut #RedWaveRising2018 #BuildTheWall #	0	0	0
30284	@GMA @TVMarci His own fault #SENDTHEMBACK	0	0	0
30072	@christianllamar @LindseyGrahamSC stop w 'we have to worry about the children' No we do not-many R >20yrs old Go home and make	1	0	0
34098	@FoxNews @BarackObama Like he ever kept out any threats. He's lying as usual. #BuildThatWall	0	0	0
33818	Im going to explode if I have to listen to these entitled, privileged cunts bitch for ONE MORE SECOND about their inconvenient FREE flight	1	0	0
34138	He is a snake ass bitch. He is a fugly slut who shouldn't be trusted and I'm patiently waiting for him to be 18 cuz https://t.co/NbezBfDPQ4	0	0	0
34165	You seem like a hoe Ok bitch? Did I ever deny that? Nope, Next.	0	0	0

Figura 4.1: Exemplo de tweets do dataset

Neste fase do traballo levouse a cabo unha revisión inicial do dataset, na cal se comprobou que non existise ningún post que estivese baleiro ou fose ilexible.

Na figura 4.2 móstrase a relación de tweets entre os dous conxuntos determinados 5.2: 42,03% do conxunto dos verdadeiros *contra* 57,97% do conxunto dos falsos; e na figura 4.3 apréciase a relación da característica HS coa AG, a partir da cal se determina que unicamente os tweets con expresións de odio (HS) poden conter comportamento agresivo (AG).

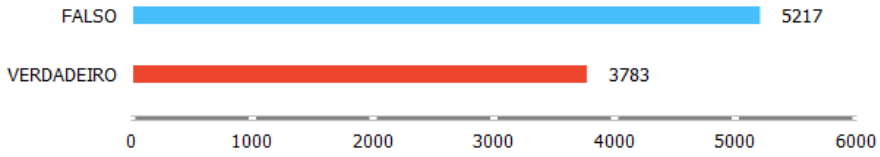

Figura 4.2: Diagrama de caixas dos dous conxuntos

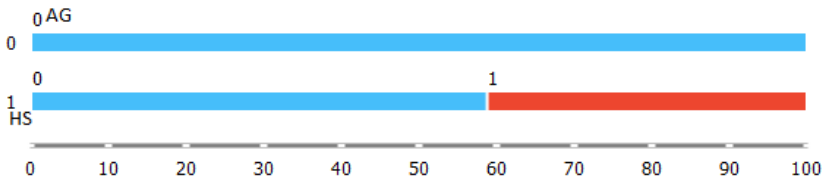

Figura 4.3: Diagrama de caixas

Como se pode ver na figura 4.3, os tweets do dataset soamente presentan comportamento agresivo se conteñen expresións de odio, polo que se pode ver que a característica AG depende do valor da característica HS.

No diagrama de caixa da figura 4.4, podemos ver unha comparación do número de letras que conteñen os tweets de cada un dos conxuntos, explicados na sección 5.2. A pesar de que o tweet con maior cantidade de letras pertence ao conxunto dos falsos, vemos que no conxuntos dos verdadeiro hai máis posts que están compostos por unha maior cantidade de letras: os valores do 1º cuartil (mediana da primeira metade da mostra), da mediana, da media e do 3º cuartil (mediana da segunda metade) son un pouco máis altos, sobre todo este último valor.

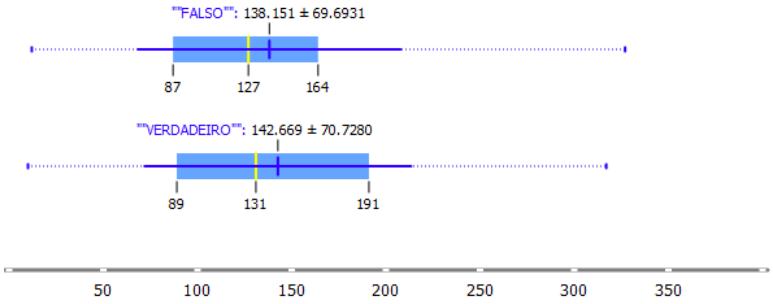

Figura 4.4: Diagrama de caixa coa característica do número de letras para cada conxunto

Na seguinte figura 4.5 móstrase o diagrama de caixa do número de veces que aparece unha letra maiúscula nos tweets de cada conxunto. Neste caso, os tweets que presentan maiores valores para esta característica, o número de maiúsculas que contén o posts, son tamén os do conxunto dos verdadeiros, a simple vista vese que estes acadan un maior número de tweets, é dicir, mentres que o valor máximo para esta característica no conxunto dos falsos non chega aos 150, no outro conxunto seguen habendo tweets con valores ata case as 250 letras maiúsculas, tamén vemos unha media de 13.28, mentres que no conxunto dos falsos é de 9.89.

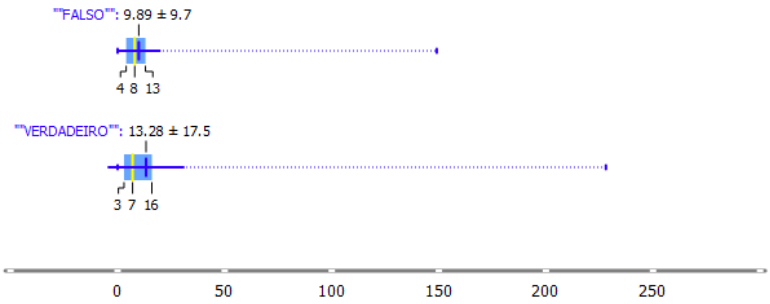

Figura 4.5: Número de exclamacións que conteñen os tweets en relación aos conxuntos

Capítulo 5

Preparar os datos

A continuación relátanse as tarefas levadas a cabo nesta terceira fase, as cales son necesarias para orixinar posteriormente o modelo.

5.1 Limpeza

Ante a posible presenza de tweets anómalos, é dicir, que estiveran baleiros ou que o seu contido fose ilexible, revisouse o dataset para que, de existir, estes posts fosen eliminados. Esta revisión do dataset pódese facer empregando o método Isolation forest, que detecta os posts con valores anómalos, empregando o uso de árbores, e elimínaos. Este paso lévase a cabo para evitar que os resultados se vexan afectados ante a presenza deste tipo de tweets anómalos e como consecuencia, redúcese o volume dos datos, o que provoca unha redución no tempo de execución nos seguintes pasos desta fase e nas seguintes. Coa cantidade de posts que ten este dataset resulta asumible facer esta revisión a man, polo que neste caso non se empregou o algoritmo de Isolation forest para revisar os datos.

A continuación, sobre os tweets orixinais do dataset, recoléctanse algunhas características que se obteñen do post sen necesidade de realizar ningunha operación sobre el, e que despois da limpeza non serían posibles de obter, como a cantidade de veces que contén un signo de exclamación ou unha letra maiúscula. Outras destas características son o número de letras que conten, o número de palabras, a cantidade de veces que aparece o carácter @ ou o #, ou número de veces que contén unha url, entre outras que se citan no apartado 5.3.

Seguidamente aplícanse unha serie de operacións individuais a cada tweet, limpándoos e procesándoos coa finalidade de eliminar caracteres innecesarios e crear un vocabulario máis preciso. As operacións realizadas son as seguintes:

- Minúsculas: convértense os tweets a minúsculas para tratar a todas as palabras por igual. Por exemplo: a palabra “people” e a palabra “People” serán tratadas como a

mesma.

- Eliminación de caracteres e díxitos mediante expresións regulares: elimínanse os caracteres (,.:;¿?!¬¡|—&$-(_)<=>+-/”%) e os díxitos que non aportan información relevante. Así, o carácter @ ou o # non se eliminan xa que serán empregados máis adiante.

- Función de Stemming: aplícase unha función de Stemming, usando a linguaxe de Porter, coa finalidade de reducir as palabras á súa raíz e así, as palabras derivadas, como os plurais ou as conxugacións de verbos, redúcense á súa raíz e serían a mesma palabra. Véxase a palabra ”study” e ”student”, se se lle aplica a función de Porter Stemming ambas palabras quedan reducidas a súa raíz, ”stud”, e serían a mesma palabra. Deste xeito esta función axuda a reducir o número de palabras coas que se formará o vocabulario [18].

- Eliminación de *stopwords*: este tipo de palabras non aportan un significado ao texto por si mesmas e soen ser artigos, pronomes, preposicións, etc. Por exemplo ‘it’, ‘is’, que son irrelevantes para o noso obxectivo e por esta razón as eliminamos. Para desfacerse delas, faise uso do paquete de R *tidytext* [19].

5.2 Transformación

Neste paso explícanse as operacións levadas a cabo para obter as características necesarias para a posterior creación do modelo.

Primeiramente sepárase o dataset en dous conxuntos a partir das características propias que posúe.

Para a separación dos conxuntos, engádese unha columna ao dataset, *OR*, na cal se reflexa o conxunto ao que pertence cada tweet (táboa 5.1) e a que será unha das características empregadas para o modelo.

Os conxuntos elixidos preséntanse a continuación:

- Un conxunto de **”positivos”** ou **”verdadeiros”**: tweets que no dataset teñen a característica HS e/ou a característica AG con valor 1.

- Un conxunto de **”negativos”** ou **”falsos”**: tweets que teñen as características HS e AG con valor 0.

	id	tweet	HS	TR	AG	OR
1	201	hurray save us in so mani way @potus @realdonal...	1	0	0	1
2	202	whi would young fight age men be the vast major of...	1	0	0	1
3	203	@kamalaharri illeg dump their kid at the border like...	1	0	0	1
4	204	ny time near all white state pose an array of problem...	0	0	0	0
5	205	orban in brussel european leader are ignor the will of...	0	0	0	0
6	206	@kurtschlicht legal is not illeg #buildthatwal	1	0	0	1
7	207	@ritapanahi @maureen @realcandaceo antifa are just...	0	0	0	0
8	208	exteach plead not guilti to rape charg httpstcodmguvtg	0	0	0	0
9	209	still place on our bengali sylheti class it london nd lan...	0	0	0	0
10	210	dfid africa region profil juli httpstconpfzcriww	0	0	0	0

Táboa 5.1: Vista do dataset cos tweets clasificados na columna *OR*

Unha vez finalizou o preprocesado dos tweets, créase o vocabulario de todo o dataset coa axuda da librería *Text2Vec* de R 3.4.

O vocabulario fórmase coas palabras extraídas de cada un dos tweets do dataset, pero soamente teremos en conta as que se repiten dúas ou máis veces, eliminando así palabras sen peso que só están presentes unha vez en todo o dataset e que poderían prexudicar na clasificación final.

Co vocabulario creado, xérase unha matriz, denominada DTM (*Document-Term Matrix*), tamén coa axuda da librería Text2Vec 3.4. Esta matriz representa a frecuencia coa que os termos (palabras do vocabulario) aparecen nun documento (o dataset), sendo os termos as columnas e os documentos, ou tweets, as filas. Se un termo aparece nun documento, terá valor positivo, do contrario o seu valor será 0.

	Termo 1	Termo 2	Termo 3
Tweet 1	1	0	1
Tweet 2	0	1	1

Figura 5.1: Exemplo matriz DTM

Para rematar con este paso, aplícaselle á matriz denominada como DTM o algoritmo de similitude co método elixido: jaccard, coseno ou lsa; dos cales se falou anteriormente na sección 3.4.

$$sim2(dtm, null, method, norm)$$

Unha vez executada esta función, obtense unha matriz cos resultados individuais para cada tweet comparado cos demais, é dicir, para cada post obtense o valor del mesmo comparado, un a un, con todos os demais.

A partir desta matriz son obtidas algunhas características, as denominadas como carac-

terísticas adicionais, citadas no apartado 5.3. Ao conter as comparacións de todos os tweets, permite obter valores de cada post en relación a cada un dos dous conxunto, dos que se falou na sección 5.2.

Un dos valores que se obteñen é o valor máximo, para un tweet obtense o valor máximo de similitude que obtivo en relación ao conxunto dos verdadeiros e ao conxunto dos falsos, e así poderíase saber a que conxunto ten máis parecido este tweet e polo tanto máis posibilidade de pertencer a el.

Pero para que estas características sexan válidas, anúlase a diagonal da matriz, para obviar o valor dun tweet comparado con el mesmo. Outras características que se obteñen desta matriz son o valor mínimo, a varianza, a media e a media, obtendo valores relativos a cada conxunto.

Con todas estas características o algoritmo do RF poderá crear un modelo.

5.3 Selección

No último paso desta fase, selecciónanse as características que lle permitirán ao modelo conseguir unha maior precisión e que se foron obtendo nos pasos anteriores. Este paso é importante, xa que a clasificación depende das características que se escollan e do relevantes que son.

Unha cousa que se tivo en conta á hora de seleccionar as características foi que estas non tivesen relación, xa que RF traballa escollendo as características aleatoriamente para evitar sobreaxustes. Se metemos características correlacionadas, a súa importancia pode variar e podría darnos resultados irreais. Debido a isto, a característica TR que traía o propio dataset foi eliminada, pola súa correlación coa característica HS.

En resumo, as características totais obtidas son as seguintes:

- **OR**: é a característica que determina o grupo ao que pertence o post.
- **Características adicionais**: estas características son extraídas a partir de cálculos estatísticos dos resultados obtidos a partir do cálculo da similitude 5.2:
  - Valor mínimo en relación a cada conxunto (verdadeiros e falsos)
  - Valor máximo en relación a cada conxunto
  - Varianza en relación a cada conxunto
  - Media en relación a cada conxunto
  - Mediana en relación a cada conxunto
- **Características básicas**: son características obtidas a partir do texto do tweet orixinal:

- Número de letras
- Número de palabras
- Cantidade de veces que aparece o carácter @
- Cantidade de veces que aparece o carácter #
- Cantidade de veces que aparece un signo de exclamación
- Número de espazos en branco
- Número de veces que contén unha url
- Número de veces que hai letras maiúsculas
- Número de veces que aparece un díxito

Na figura 5.2 pódense ver o tipo e algúns dos valores que conteñen as características de ambos conxuntos:

```
$ minF       : num  0 0 0 0 0 0 0 0 0 0 ...
$ minV       : num  0 0 0 0 0 0 0 0 0 0 ...
$ maxF       : num  0.284 0.394 0.286 0.354 0.316 ...
$ maxV       : num  0.704 0.309 0.459 0.365 0.365 ...
$ meanF      : num  0.00962 0.04234 0.01098 0.02843 0.02027 ...
$ meanV      : num  0.0195 0.0324 0.0302 0.0216 0.0199 ...
$ medianF    : num  0 0 0 0 0 0 0 0 0 0 ...
$ medianV    : num  0 0 0 0 0 0 0 0 0 0 ...
$ varF       : num  0.000936 0.003459 0.001063 0.003141 0.002351 ...
$ varV       : num  0.00302 0.00262 0.00375 0.00245 0.00223 ...
$ positivo   : Factor w/ 2 levels "FALSO","VERDADEIRO": 2 2 2 1 1 2 1 1 1 1 ...
$ letters    : int  121 300 255 126 122 53 212 68 159 63 ...
$ arroba     : int  2 0 1 0 0 1 3 0 3 0 ...
$ hashtag    : int  5 0 4 0 0 1 1 0 0 0 ...
$ words      : int  15 48 40 15 18 6 32 8 18 7 ...
$ exclamacion: int  0 0 1 0 0 0 0 0 2 0 ...
$ blank      : int  14 47 39 14 17 5 31 7 17 6 ...
$ url        : int  0 1 0 2 1 0 0 1 1 1 ...
$ mayusc     : int  20 7 36 25 8 11 9 12 11 11 ...
$ number     : int  0 1 0 2 0 0 3 3 2 5 ...
```

Figura 5.2: Contido características

Capítulo 6

Modelado

O modelado consiste na selección e aplicación do algoritmo para crear o modelo e no axuste das características para obter os mellores resultados posibles.

6.1 Asunción de modelos

O obxectivo principal deste proxecto é a clasificación dos tweets, por isto o algoritmo que se seleccionou é o de Random Forest 3.2, xa que é un dos máis empregados para este tipo de problemas.

6.2 Adestramento

Neste paso realizáronse varias probas para intentar acadar o mellor modelo posible.

Partindo da existencia dos conxuntos de características vistos na sección 5.3, crearanse dous modelos:

- Un modelo construído soamente empregando das características adicionais.
- Outro modelo construído empregando das características básicas e as adicionais.

Co conxunto de datos das características referentes a cada tweet, obtéñense dous subconxuntos:

- **Adestramento**: neste conxunto están o 80% dos tweets, e serán os cales crearán o modelo a validar.
- **Proba**: neste conxunto están o restante dos tweets, os cales serán empregados para verificar a eficacia do modelo.

Para determinar os tweets que forman cada conxunto hai dúas opcións:

- Escoller sempre os mesmos tweets.

- Escoller os tweets aleatoriamente.

O modelo resultante, entre outras cousas, móstranos unha saída como a da figura 6.1. Esta saída é resultado de aplicar o seguinte código á seguinte función do paquete de Random Forest 3.4:

$$randomForest(positivo\ ., data = train, ntree = 500, mtry = 4, importance = TRUE)$$

Nela podemos observar os seguintes datos:

- A chamada feita á operación *randomForest* a partir da cal se crea o modelo.

- Tipo de random forest: o problema para o que está sendo empregado o algoritmo, clasificación ou regresión.

- Número de árbores creadas e de características que empregou en cada nodo da árbore.

- OOB estimate of erro rate: para cada árbore que se crea, colle un subconxunto dos datos que se lle proporcionan, e os datos restantes son empregados para validar esa árbore. Este valor obtense calculando a media dos erros de predición obtidos con estes datos restantes aplicados a tódalas árbores.

- Matriz de confusión: mostra o número de tweets que clasificou en cada categoría, así como a cantidade deles que clasificou correctamente ou non. Para entender mellor os datos da matriz de confusión, móstrase a figura 6.2.

```
Call:
 randomForest(formula = positivo ~ ., data = train, importance = TRUE)
               Type of random forest: classification
                     Number of trees: 500
No. of variables tried at each split: 3

        OOB estimate of  error rate: 26.48%
Confusion matrix:
           FALSO VERDADEIRO class.error
FALSO       1040        284   0.2145015
VERDADEIRO   314        620   0.3361884
```

Figura 6.1: Saída mostrada polo paquete randomForest tras crear o modelo

		Predición	
		Positivos	**Negativos**
Observación	**Positivos**	Verdadeiros Positivos (VP)	Falsos Negativos (FN)
	Negativos	Falsos Positivos (FP)	Verdadeiros Negativos (VN)

Figura 6.2: Explicación matriz de confusión

Por outra banda, tamén podemos obter do modelo as importancias de cada unha das características empregadas para o adestramento. En ambos casos, canto maior é o valor, maior é a importancia desa variable para a clasificación [20].

- **Mean Drecrease Accuracy**: obtén un resultado a partir das veces que se usou esa característica no modelo e a precisión mellorou.
- **Mean Drecrease Gini**: cada vez que un nodo elixe a característica que emprega, faino mediante a impureza Gini, esta medida da importancia ten en conta cada vez que usa a característica e a impureza do nodo diminúe, é dicir, o modelo mellora.

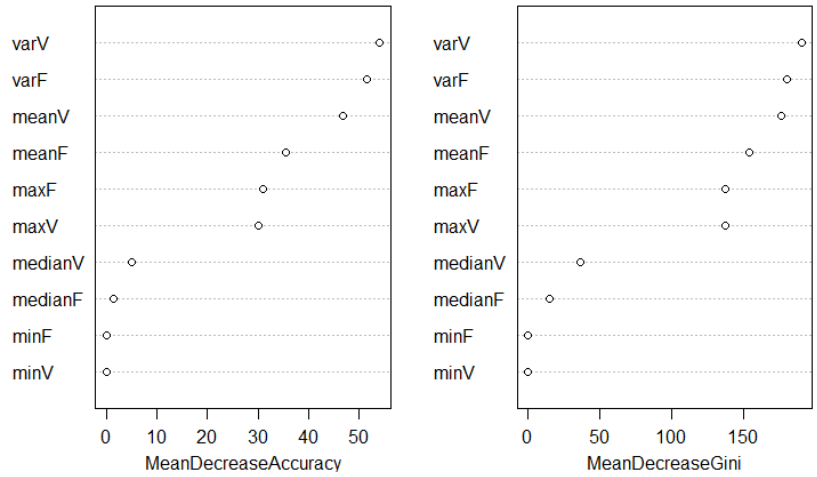

Figura 6.3: Gráficas que mostran importancia das características

A parte das características do dataset que se aportan para a creación do modelo, o algoritmo dispón duns parámetros internos, os chamados *hiperparámetros*, que son empregados para axustar o modelo e así obter unha maior precisión.

O número de árbores que crea e o número de características que emprega en cada nodo, das cales falamos na sección 3.2 de como se empregan, son exemplos de hiperparámetros que

podemos empregar para influír na creación das árbores ou na cantidade delas que se crean, o que alterará a creación do modelo. Neste traballo probáronse con varias opcións de número de árbores (200, 300, 500) e de número de características (3,4), e finalmente foi seleccionada na que se apreciaron mellores resultados: 500 árbores e 4 características. Estes parámetros son pasados a función de randomForest, do paquete Random Forest de R citado no apartado 3.4, a través dos atributos ntree e mtry respectivamente.

6.3 Validación

Para validar un modelo, existen varios tipos de validacións que levar a cabo:

- **Validación OOB** (Out Of Bag): cando se crea o modelo a partir do conxunto de entrenamento, cada árbore de decisión é creada a partir dun subconxunto aleatorio dos datos aportados, deixando fora un pequeno porcentaxe das mostras, mostras de OOB. Con estas mostras é coas que se valida a árbore creada. O RF obtén o OOB rate como métrica da precisión dos resultados obtidos.

- **Validación simple** (VS): obtense o modelo a partir do conxunto de adestramento, e valídase empregando o conxunto de proba facendo uso da función *predict* da que dispón o paquete de radomforest en R 3.4.

- **Validación cruzada**: empregando o conxunto total dos datos, crea varios subconxuntos e segue o método explicado anteriormente na sección 3.3. Cando se teñen os resultados das predicións obtidas de todos os subconxuntos, calcúlase a media deles para obter un valor do porcentaxe de acerto deste modelo.

A partir de cada validación obtemos unha porcentaxe de erro, que será a que indique a efectividade do modelo. No caso de obter malos resultados, repetiríase o proceso ata obter un modelo máis axeitado.

Na seguinte táboa 6.1, pódese ver un exemplo dos resultados obtidos a partir da validación simple e da validación OOB para un modelo creado a partir do conxunto de características adicionais, e outro, a partir de ambos conxuntos de características, dos cales se fala no apartado 5.3. Como se menciona anteriormente, hai dúas formas para escoller os tweets pertencentes a cada conxunto 6.2, e aínda que ambos métodos foron probados, vamos a mostrar os resultados do segundo: no que son escollidos aleatoriamente. No caso da validación simple, obtivéronse resultados en función de catro métricas, das que se falará na sección 7.

	Jaccard	Coseno	LSA	Conxunto Características
VS-Precisión	0.7683333	0.7605556	0.7516667	Adicionais
	0.7694444	0.7894444	0.7738889	Adicionais + Básicas
VS-Exactitude	0.7327824	0.7195467	0.7199413	Adicionais
	0.7616959	0.7696710	0.7281831	Adicionais + Básicas
VS-Sensibilidade	0.7046358	0.6855601	0.6572959	Adicionais
	0.6739974	0.7116402	0.7011019	Adicionais + Básicas
VS-F1	0.7184335	0.7021424	0.6871938	Adicionais
	0.7151682	0.7395189	0.7143860	Adicionais + Básicas
OOB error rate	23.76%	24.56%	24.74%	Adicionais
	21.99%	22.22%	23.44%	Adicionais + Básicas

Táboa 6.1: Validación simple e OOB

Capítulo 7

Avaliación

Nesta fase revisaranse os resultados obtidos e verase en que grao cumpren coa finalidade do proxecto, que é a detección de tweets misóxinos e xenófobos.

Unha vez se obteñen os resultados obtidos coa validación, existen varias métricas que se poden empregar para calcular a porcentaxe de acerto alcanzada [21], pero para isto hai que ter en mente o esquema da matriz de confusión (figura 6.2), xa que os resultados obtidos na predición serán transformados e mostrados como unha matriz de confusión para poder levar a cabo os cálculos das métricas máis facilmente.

As métricas que se empregaron neste traballo foron as seguintes:

- **Precisión** (Accuracy): representa a porcentaxe de elementos clasificados correctamente.

$$(VP + VN)/(VP + FP + FN + VN)$$

- **Exactitude** (Precision): representa ao número de elementos clasificados correctamente como positivos do total dos clasificados como positivos.

$$VP/(VP + FP)$$

- **Sensibilidade** (Recall): tamén coñecida como taxa de verdadeiros positivos, representa ao número de elementos clasificados correctamente como positivos do total dos verdadeiros positivos.

$$VP/(VP + FN)$$

- **F1**: esta métrica fusiona dúas das anteriores: exactitude e sensibilidade, tendo en conta que ambas teñen a mesma importancia.

$$2 * ((exactitude * sensibilidade)/(exactitude + sensibilidade))$$

Na seguinte táboa vese un resumo dos resultados obtidos en función do conxunto empregado.

Para cada un dos métodos empregados no algoritmo de similitude 3.4, están listados na táboa 7.1 os resultados obtidos da avaliación empregando validación cruzada. Para cada modelo composto polos conxuntos de características, indicados na táboa, obtéñense os resultados en función das distintas métricas explicadas.

Métrica	Jaccard	Coseno	LSA	Conxunto Características
Precisión	0.7643343	0.7592235	0.7604434	Adicionais
	0.7816664	0.7814392	0.7706679	Adicionais + Básicas
Exactitude	0.6862329	0.6706482	0.6761849	Adicionais
	0.6930994	0.6928215	0.6883465	Adicionais + Básicas
Sensibilidade	0.7356765	0.7339593	0.7331639	Adicionais
	0.7659159	0.7657555	0.7466294	Adicionais + Básicas
F1	0.7098740	0.7005221	0.7032766	Adicionais
	0.7273556	0.7268784	0.7161502	Adicionais + Básicas

Táboa 7.1: Validación cruzada

Podemos apreciar que os mellores resultados obtéñense cando se usan os dous conxuntos de características, obtendo para o mellor dos casos un resultado de 78.16% na métrica de precisión, superando ata nun 13.16% os valores das métricas de precisión atopadas no estado do arte para este dataset, e cumprindo cos obxectivos do proxecto. Aínda así, non se observa moita diferencia entre os resultados empregados con distintos conxuntos, xa que as características do conxunto das básicas non eran moi distintivas, pero non se atoparon outras que puideran selo máis, isto queda pendente para as liñas futuras.

Por outro lado, comparando os tres algoritmos, vemos que empregando o método de jaccard para calcular a similitude, os resultados melloran lixeiramente. Isto pode ser por como traballo algoritmo, do que se fala na sección 3.4, que se adapta mellor a linguaxe empregada ou a cantidade de datos que contén o noso dataset.

Unha das probas que se levou a cabo, foi introducir a característica TR, e si que se viu unha mellora bastante notable na predición, obtendo resultados que superaban o 80% de acertos, pero non foi considerada como unha boa característica, xa que soamente ten valor nos casos en que a característica HS tiña valor 1.

Unha das características do RF é que reduce o sobreaxuste escollendo as mostras e as características que emprega aleatoriamente. Por iso, se nos introducimos unha característica correlacionada pode dar resultados falsos, por exemplo, a característica TR so ten a posibilidade de ter valor cando HS é igual a 1 e pode ocorrer que sempre que TR non teña valor a a asigne a un conxunto, sendo incorrecto nalgúns casos, ou o contrario, se ten valor a asigne sempre a o outro conxunto.

Capítulo 8

Conclusións

Para concluír con este traballo e despois de acadar todos os obxectivos, vexo que me foi de utilidade para mellorar as competencias adquiridas durante o estudo do grado, así como para comprender moitos aspectos do ML, un campo moi amplo e que era practicamente descoñecido para min.

Ao mesmo tempo, tamén me foi moi útil para poder ver dun xeito práctico e máis real o desenvolvemento dun proxecto, a planificación, todas as tarefas, a investigación... e ver que, aínda que ao principio pódense cometer moitos erros, pois era a primeira vez que realizaba un proxecto deste ámbito, tamén se poden adquirir coñecementos e sacar conclusións destas probas erradas.

Este proxecto está centrado en posts de Twitter, pero podería ser empregado para calquera rede social ou aplicación na que se vexa necesaria a súa función. Tamén podería ser empregado para outra finalidade, sempre e cando se especifiquen as características adecuadas (por exemplo, para a detección de enfermidades a partir de síntomas).

Un dos obxectivos era a análise do contido do dataset e a extracción de características dos posts, o cal inclúe varios pasos que vamos comentar seguidamente.

- Atopar características extraídas directamente do post: obtivéronse características que deron bos resultados, pero foi algo que me pareceu algo difícil, pois soamente tiñamos o texto do tweet.

- Cálculo de similitude: o complexo deste paso é a obtención do vocabulario, pois para acadar mellores resultados necesitamos que sexa o máis preciso posible, para isto, un dos pasos que fixemos foi realizar un preprocesado previo dos posts. Aínda así, non é un texto oficial, polo que a xente pode escribir empregando abreviaturas, repetindo letras para mostrar énfase... o que dificulta este procesado, pero aínda así acadáronse bos resultados, pois temos precisións do 78.16% de acertos.

Para a obtención dos resultados finais, fixéronse varias probas empregando distintos méto-

dos para o cálculo da similitude e distintos conxuntos de características para crear un modelo que foi probado a partir de distintos métodos de validación. Para poder interpretar mellor estes resultados e determinar cal foi o modelo máis preciso, empregamos distintas métricas sobre os resultados obtidos para obter a porcentaxe de acerto e poder comparar máis facilmente.

8.1 Posibles liñas futuras

Por varias circunstancias, como a falta de tempo ou de datos aportados, non se puido desenvolver máis o traballo, pero algunhas funcionalidades que se poderían levar a cabo son as seguintes:

- Realización de clasificación con usuarios: identificar un usuario que ten estes comportamentos a través dos seus tweets.
- Emprego de máis características: atopar algunha característica máis distinta que permita mellorar os resultados.
- Varias clasificacións: neste traballo o obxectivo era clasificar os tweets diferenciando aos que contiñan expresións de odio e comportamento agresivo, pero xa que o contido destes tweets é misóxino ou xenófobo, poderíase intentar diferencias entre ambos.
- Integración nun entorno de probas e nun real.

Apéndices

Bibliografía

[1] G. S. Randhawa, M. P. M. Soltysiak, H. E. Roz, C. P. E. de Souza, K. A. Hill, and L. Kari, "Machine learning using intrinsic genomic signatures for rapid classification of novel pathogens: Covid-19 case study," [Accedido 23-05-2020]. [En liña]. Dispoñible en: https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0232391

[2] P. R. Talens, "Detección de lenguaje sexista en documentos," 01 2018. [En liña]. Dispoñible en: http://hdl.handle.net/10251/93786

[3] A. Azevedo, "Kdd, semma and crisp-dm: A parallel overview," [Accedido 16-12-2019]. [En liña]. Dispoñible en: https://pdfs.semanticscholar.org/7dfe/3bc6035da527deaa72007a27cef94047a7f9.pdf

[4] C. L. Hernández G. and M. X. Dueñas R., "Hacia una metodología de gestión del conocimiento basada en minería de datos," [Accedido 16-12-2019]. [En liña]. Dispoñible en: http://repositorio.uigv.edu.pe/bitstream/handle/20.500.11818/982/COMTEL-2009-80-96.pdf?sequence=1&isAllowed=y

[5] P. Chapman, J. Clinton, R. Kerber, T. Khabaza, T. Reinartz, C. Shearer, and R. Wirth, "Crisp-dm 1.0. step-by-step data mining guide," [Accedido 18-12-2019].

[6] "Metodología scrum: qué es y cómo funciona," [Accedido 13-02-2019]. [En liña]. Dispoñible en: https://www.wearemarketing.com/es/blog/metodologia-scrum-que-es-y-como-funciona.html#:~:text=Scrum%20es%20una%20metodolog%C3%ADa%20de,en%20iteraciones%20cortas%20de%20tiempo.&text=Esto%20permite%20al%20cliente%2C%20junto,obtener%20ventas%20(Sales%20enablement).

[7] "Xvii convenio colectivo." [En liña]. Dispoñible en: https://www.boe.es/boe/dias/2018/03/06/pdfs/BOE-A-2018-3156.pdf

[8] W. Koehrsen, "An implementation and explanation of the random forest in python," [Accedido 15-02-2020]. [En liña]. Dispoñible en: https://towardsdatascience.com/an-implementation-and-explanation-of-the-random-forest-in-python-77bf308a9b76

[9] "Introducción a la validación cruzada (k-fold cross validation) en r," [Accedido 21-04-2020]. [En liña]. Dispoñible en: https://rpubs.com/rdelgado/405322

[10] "What are the best r ides?" [En liña]. Dispoñible en: https://www.slant.co/topics/2897/~best-r-ides

[11] "R (lenguaje de programación." [En liña]. Dispoñible en: https://es.wikipedia.org/wiki/R_(lenguaje_de_programaci%C3%B3n)

[12] *Documents Similarity*, 2018. [En liña]. Dispoñible en: http://text2vec.org/similarity.html#documents_similarity

[13] H. Mahajan and R. Saha, "Ire-project-hateval-2019," 11 2018, [Accedido 24-03-2020]. [En liña]. Dispoñible en: https://github.com/ash0904/IRE-Project-hatEval-2019

[14] V. Basile, "Semeval2019-task5 gitlab," 09 2018, [Accedido 24-03-2020]. [En liña]. Dispoñible en: https://github.com/msang/hateval/blob/master/SemEval2019-Task5/evaluation/evaluation.py

[15] V. L. C. Alvarado, "Clasificación de tweets mediante modelos de aprendizaje supervisado," pp. 1–79, 09 2018. [En liña]. Dispoñible en: https://eprints.ucm.es/49774/1/TFM%20Veronica%20Chamorro%20Alvarado.pdf

[16] V. Indurthi, B. Syed, M. Shrivastava, N. Chakravartula, M. Gupta, and V. Varma, "Fermi at semeval-2019 task5," pp. 1–5, 07 2019. [En liña]. Dispoñible en: https://www.aclweb.org/anthology/S19-2009.pdf

[17] "Semeval 2019 task 5 - shared task on multilingual detection of hate," [Accedido 12-12-2019]. [En liña]. Dispoñible en: https://competitions.codalab.org/competitions/19935#learn_the_details

[18] "Stem pre-proccesed," 2014. [En liña]. Dispoñible en: https://www.r-bloggers.com/r-stem-pre-processed-text-blocks/

[19] "Package 'tidytext'," [Accedido 15-01-2020]. [En liña]. Dispoñible en: https://cran.r-project.org/web/packages/tidytext/tidytext.pdf

[20] "Machine learning - random forest," [Accedido 11-05-2020]. [En liña]. Dispoñible en: https://wiki.q-researchsoftware.com/wiki/Machine_Learning_-_Random_Forest

[21] “Machine learning: Seleccion metricas de clasificacion,” [Accedido 14-05-2020]. [En liña]. Dispoñible en: https://sitiobigdata.com/2019/01/19/machine-learning-metrica-clasificacion-parte-3/#