Trabajo Fin de Máster: Detección de noticias falsas empleando información social de Twitter

Trabajo Fin de Máster

Jesús María Fraile Hernández

Trabajo de investigación para el

Máster en Tecnologías del Lenguaje Universidad

Nacional de Educación a Distancia

Dirigido por los

Prof. Dr. D. Álvaro Rodrigo Yuste
Prof. Dr. D. Roberto Centeno Sánchez

Septiembre 2022

Resumen

La exposición y propagación de noticias falsas ha ido en aumento en los últimos años debido a la rapidez en la transmisión de la información en medios digitales. Debido a la naturaleza de las redes sociales, las noticias falsas se propagan rápidamente y crean un gran daño en la sociedad. Es por ello por lo que estas plataformas están adoptando medidas (Flores, 2022) para luchar contra la desinformación.

A lo largo de esta última década se han propuesto varios modelos para detectar noticias falsas mediante el análisis de características lingüísticas tanto léxicas como semánticas (Horne y Adali, 2017), (Bharadwaj y Shao, 2019), (Kaliyar, Goswami, y Narang, 2021). Sin embargo, debido a la naturaleza engañosa de estas noticias, estos modelos no siempre son suficientes. Por ello la investigación actual se basa en incluir la información social de los usuarios que interactúan con la noticia (Conroy, Rubin, y Chen, 2015), (Buntain y Golbeck, 2017), (Ruchansky, Seo, y Liu, 2017) ,(Shu, Wang, y Liu, 2017). Para realizar esta tarea se han recopilado varios conjuntos de datos obteniendo las interacciones de los usuarios en las redes sociales (Potthast et al., 2018), (Shu et al., 2018). Estos conjuntos de datos con información social no se encuentran disponibles en español.

Con el desarrollo de este trabajo se pretende estudiar el aporte de la información social a la hora de clasificar noticias en español. Se partirá del conjunto de noticias FakeDeS (Posadas-Durán et al., 2019) y se completará extrayendo la información de los usuarios de Twitter que hablan de dichas noticias.

En un primer momento se realizará un análisis exploratorio del conjunto de noticias y del conjunto de información social. Posteriormente se propondrán varios modelos en los cuales se irán combinando diferentes características, tanto textuales como sociales. A continuación, se estudiará el rendimiento de los modelos propuestos y se realizará un estudio sobre la

importancia de la información social en los modelos.

Con este trabajo, se pretende estudiar si la información social permite aportar información útil a la hora de clasificar noticias. Para ello se ha ampliado con la información social el conjunto de noticias en español más relevante y se ha propuesto un modelo clasificador que tiene un buen rendimiento para esta tarea.

Abstract

The exposure and spread of fake news has been increasing in recent years due to the rapid transmission of information on digital media. Due to the nature of social media, fake news spread quickly and create great harm in society. This is why these social media platforms are adopting measures to combat misinformation.

Over the last decade, several models have been proposed to detect fake news by analysing both lexical and semantic linguistic features (Horne y Adali, 2017), (Bharadwaj y Shao, 2019), (Kaliyar, Goswami, y Narang, 2021). However, due to the misleading nature of such news, these models are not always sufficient. Therefore, the methodology of current research included the social information of users interacting with the news (Conroy, Rubin, y Chen, 2015), (Buntain y Golbeck, 2017), (Ruchansky, Seo, y Liu, 2017) ,(Shu, Wang, y Liu, 2017). To perform this task, several datasets have been collected by obtaining user interactions on the social networks (Potthast et al., 2018), (Shu et al., 2018). These datasets with social information are not available in Spanish.

The aim of this work is to study the contribution of social information when classifying news in Spanish. It will start with the set of news FakeDeS (Posadas-Durán et al., 2019) and it will be completed it by extracting information from Twitter users who talk about these news.

At first, an exploratory analysis of the set of news and social information will be carried out. Subsequently, several models will be proposed in which different features, both textual and social, will be combined. Then, the performance of the proposed models will be studied and a study on the importance of social information in the models will be carried out.

The aim of this work is to study whether social information provides useful information when classifying news. For this purpose, the most relevant set of Spanish news items has been extended with social information and a

classifier model has been proposed that performs well for this task.

Índice general

Índice de Figuras

Índice de Tablas

Capítulo 1

Introducción

1.1. Motivación

Debido al aumento en las últimas décadas de los canales de comunicación, la facilidad con la que se pueden difundir bulos o noticias falsas ha aumentado. Desde principios de los 2000 con la creación de las redes sociales más populares como Facebook o Twitter, los usuarios tienen acceso a una cantidad inmensa de información casi instantánea. Este hecho puede verse desde la perspectiva de que las redes sociales han permitido librarse del control de la información por parte de los medios tradicionales, sin embargo, también es relativamente sencillo caer en bulos o en desinformación.

Según un estudio realizado en 2018 por la revista Forbes (McCarthy, 2018), España ocupa el sexto puesto como el país con mayor exposición a noticias falsas. Sin embargo, según el reporte anual de Noticias Digitales de 2019 del Instituto Reuters (Hölig y Hasebrink, 2018), España es el tercer país con mayor preocupación por parte de los ciudadanos sobre la desinformación en internet.

La revista Science (Vosoughi, Roy, y Aral, 2018) publicó en 2018 un artículo donde se investigaba la difusión de las noticias verdaderas y falsas publicadas en Twitter entre 2006 y 2017. Se vio que las noticias falsas son más sencillas de propagarse puesto que suscitan a la interacción de los tuits y el algoritmo premia dicha interacción. En este artículo además destacan que la política es el ámbito en el cual más bulos se difunden, por encima de los negocios o del terrorismo.

Como se ha expuesto, la difusión de noticias falsas afecta a la vida de los ciudadanos y al devenir de los países. Es por ello por lo que sería importante

tener herramientas que permitan distinguir si una noticia es verdadera o es falsa.

1.2. Hipótesis

Dentro del problema de clasificación de la veracidad de las noticias, las hipótesis de las que se parte para la elaboración de este trabajo son:

H1. Es posible distinguir la veracidad de una noticia empleando la información textual de la misma y la información social relacionada con ella.

H2. Añadir la información social a los modelos que emplean solo la información textual de la noticia ofrece una mejora en el rendimiento a la hora de predecir la veracidad de la noticia.

H3. Dentro de la información social relacionada con las noticias, existen características más relevantes a la hora de efectuar la clasificación. Estas características vendrán dadas por lo propenso que es un usuario a compartir noticias falsas, por lo que la información relacionada con el usuario es más importante que la propia información del tuit.

1.3. Propuestas y objetivos

Los objetivos que se persiguen con el desarrollo de este trabajo para corroborar las hipótesis expuestas anteriormente son:

O1. Estudiar si la información social permite aportar información útil para la detección de noticias falsas.

- *SO1.1.* Recopilar la información social de Twitter para ampliar FakeDeS, un corpus relevante de noticias en español.
- *SO1.2.* Entrenar, evaluar y comparar modelos de aprendizaje automático que empleen diferentes características tanto textuales como sociales.

O2. Estudiar qué características sociales son las más relevantes para la clasificación de noticias.

SO2.1. Aplicar técnicas de importancia de características o entrenar modelos que por su construcción permitan obtener la importancia de las mismas.

O3. Construir un modelo que empleando información social e información textual para comparar los resultados de dicho modelo con los resultados de la tarea desarrollada en el congreso IberLEF 2021.

1.4. Estructura del documento

El presente documento se encuentra organizado de la siguiente forma.

Capítulo 1. Introducción. Este capítulo introducirá los principales motivos que han llevado a la realización de este trabajo, así como la problemática y el estado actual de la disciplina. Por último, se presentan las hipótesis que se pretenden probar, los objetivos que se persiguen y las propuestas para lograrlo.

Capítulo 2. Estado del arte. Este capítulo describe en mayor detalle la disciplina que nos ocupa, presentando su origen y su historia hasta el presente. Se muestran las técnicas actuales más utilizadas para resolver las tareas más relevantes del tema abordado, así como sus debilidades.

Capítulo 3. Conjunto de datos. En este capítulo se describe el conjunto de datos realizando un análisis exploratorio del mismo, un resumen del proceso de extracción de información social para completarlo y un análisis de esta información social extraída.

Capítulo 4. Métodos de aprendizaje automáticos. En este capítulo se describen en profundidad todos los métodos y los enfoques de aprendizaje automático empleados para resolver la tarea de clasificar noticias.

Capítulo 5. Evaluación. En este capítulo se describirá tanto la metodología utilizada para evaluar los diferentes modelos entrenados como los resultados de la evaluación.

Capítulo 6. Discusión. Este capítulo analiza y discute en profundidad los resultados obtenidos en la evaluación presentada en el capítulo anterior.

Capítulo 7. Conclusiones y trabajo futuro. Este capítulo recopila las diferentes conclusiones extraídas del trabajo realizado, y propone algunas líneas de trabajo futuro.

Capítulo 2

Estado del arte

Los modelos tradicionales de detección de noticias falsas suponen que los artículos falsos se redactan utilizando un lenguaje engañoso, y que es posible detectar las características lingüísticas tanto léxicas como semánticas que representan a dichas noticias. Con este enfoque en (Horne y Adali, 2017) se emplean 3 categorías de características para clasificar las noticias en tres categorías: verdaderas, falsas o sátiras. Las características estilísticas se basan en el procesamiento del lenguaje para comprender la sintaxis y los elementos gramaticales del texto, para ello se utiliza un etiquetador del discurso (del inglés, *Part Of Speech*) para contar el número de veces que aparece cada etiqueta. Las características de complejidad se dividen en complejidad a nivel de palabras (calculando la legibilidad de cada documento utilizando diferentes índices de legibilidad) y en la profundidad del árbol de sintaxis de las frases. Finalmente, las características psicológicas se basan en recuentos de palabras que se correlacionan con diferentes procesos psicológicos. Posteriormente, en (Bharadwaj y Shao, 2019) se utilizó embeddings preentrenados como Glove que permitían transformar las palabras de un corpus a vectores de un espacio vectorial n-dimensional para posteriormente utilizar una red neuronal recurrente LSTM. Con las palabras expresadas como vectores se puede calcular matemáticamente la distancia y el nivel de cercanía entre varias palabras. Finalmente, con el surgir de los modelos contextuales, (Kaliyar, Goswami, y Narang, 2021) aprovechó el modelo preentrenado BERT, para realizar un aprendizaje transferido e identificar la veracidad de las noticias.

Sin embargo, debido a la dificultad incluso para un humano, de discernir entre una noticia verdadera y una falsa, a veces no es suficiente con

la información textual de la noticia. En (Conroy, Rubin, y Chen, 2015) se propone a nivel teórico la posibilidad de crear un enfoque híbrido que incorporase las características lingüísticas de la noticia y un análisis de las redes que se forman sobre esa noticia. La investigación empezó a centrarse en el comportamiento de los usuarios, principalmente en redes sociales, a los que iban dirigidas estas noticias. Los métodos que emplean la información social pueden clasificarse en métodos basados en contenido (emplean las características textuales y no textuales de la noticia), basados en red (conexiones entre los usuarios que comparten o interaccionan con la noticia), basados en usuario (información del usuario) o híbridos.

En (Buntain y Golbeck, 2017) el autor emplea cuatro tipos de características para identificar noticias falsas en hilos populares de Twitter.

- Características estructurales. Incluyen la cantidad de tuits del hilo, el tiempo de vida del hilo, la profundidad del árbol de conversación, ...
- Características del usuario. Incluyen los atributos del usuario como la edad de la cuenta, el número de seguidos y seguidores, ...
- Características de contenido. Se basan en las características textuales de los tuits, como la polaridad o los sentimientos negativos o positivos del tuit.
- Características temporales. Emplea funciones que describen como las características anteriores varían con el tiempo.

En (Albahar, 2021) se detecta noticias falsas utilizando solamente la información textual extraída. Se emplea un triple encoder para las palabras, las oraciones y los comentarios de la noticia. Uno de los modelos híbridos más conocidos es el modelo CSI propuesto en (Ruchansky, Seo, y Liu, 2017), según los autores las características se pueden dividir en tres módulos.

- Módulo de captura. Este módulo es una red LSTM que recibe como entrada una secuencia de vectores que representan la información de los usuarios en un determinado espacio temporal. Las representaciones textuales se obtienen utilizando un embedding preentrenado doc2vec (Le y Mikolov, 2014).
- Módulo de puntuación. Consta de un perceptrón con dos capas, la primera con una tangente hiperbólica como función de activación y la

segunda con una función sigmoidea. Se construye un grafo de usuario calculando el número de veces que un par de usuarios comentan el mismo artículo. Esta matriz se descompone en valores propios y es usada como entrada a la red anterior. De esta forma se obtiene una puntuación del 0 al 1 para cada usuario.

- Módulo integrador. Este módulo une los módulos anteriores y calcula la predicción final.

En (Shu, Wang, y Liu, 2017) se propone un modelo de detección de noticias que considera la asociación de interacciones de usuarios, el sesgo del editor y la postura de los usuarios ante las noticias. Esto lo llamó marco de embedding de tres relaciones y su esquema puede verse en la Figura 2.1

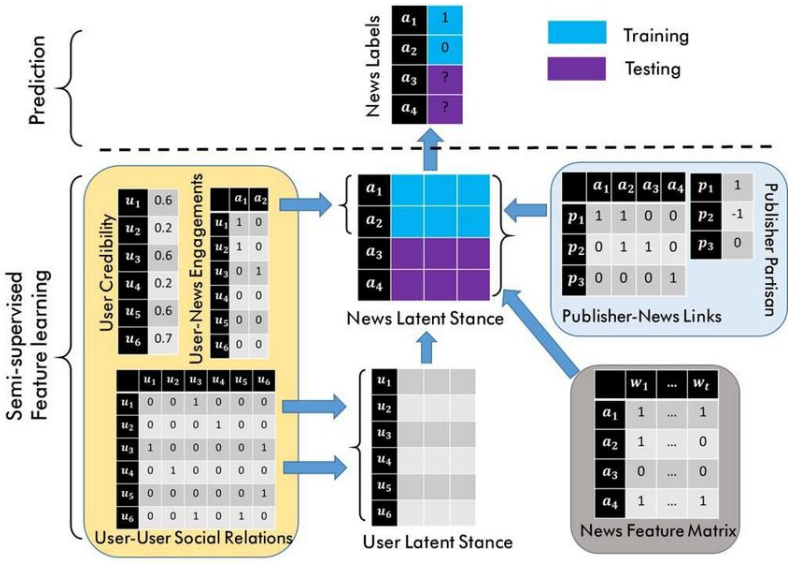


Figura 2.1: Marco de embedding de tres relaciones.

2.1. Modelos contextuales

En (Vaswani et al., 2017) se presenta la arquitectura Transformer, un modelo de Procesamiento de Lenguaje Natural que tenía como principal innovación la sustitución de las redes recurrentes. Las redes recurrentes presentaban el problema de que los pesos de las primeras palabras del entrenamiento iban disminuyendo frente a las últimas puesto que la entrada de la frase se efectuaba palabra a palabra. Para solventar este problema, en esta nueva arquitectura se recibe como entrada la oración completa y se

emplean las capas de atención junto a múltiples bloques codificadores (Nx en la Figura 2.2). Estas capas de atención codifican cada palabra de la frase en función del resto de la frase, permitiendo así introducir de una manera matemática el contexto. En el paper se presenta una arquitectura enfocada en la traducción automática, es por ello por lo que cuenta con una estructura encoder-decoder. Esta nueva arquitectura es evaluada en tareas de traducción obteniendo resultados mucho mejores que los métodos anteriores.

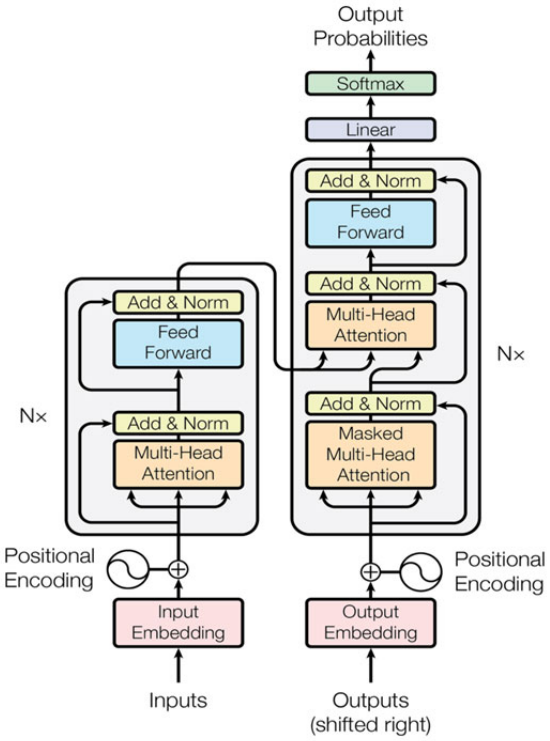


Figura 2.2: Arquitectura red Transformer.

2.1.1. Modelo BERT

Publicado en (Devlin et al., 2018) este modelo recibe su nombre por sus siglas en Inglés Bidirectional Enconder Representations from Transformers, es decir es un codificador que obtiene representaciones bidireccionales a partir de una Red Transformer. Este modelo fue creado con el objetivo de tener un modelo base preentrenado capad de interpretar el lenguaje en general y posteriormente poder especializarlo para cada tarea en concreto.

En el paper se proponen dos estructuras de BERT diferentes, $\text{BERT}_{\text{base}}$ Figura 2.3 que cuenta con 12 codificadores y 110 millones de parámetros y

$\text{BERT}_{\text{large}}$ con 24 codificadores y 340 millones de parámetros. Estos modelos se entrenan en varios pasos.

1. En esta primera fase *(Masked LM)*, BERT es entrenado usando dos corpus de gran tamaño con más de 3000 millones de palabras, Wikipedia y Google Books. Con este entrenamiento BERT aprende a analizar el texto teniendo en cuenta todo el contexto.

   Para lograr esto se enseña al modelo a completar una palabra faltante en una frase. Esta palabra puede encontrarse en cualquier ubicación de la frase, y con esto se fuerza al modelo a analizar el contexto de manera global.

2. En esta segunda fase se enseña al modelo a predecir la continuación de una frase *(Next Sentence Prediction)*. Para ello, dadas dos frases A y B el modelo tiene que responder a la pregunta de clasificación binaria ¿la frase B sigue a la frase A o es una frase aleatoria? Durante el entrenamiento la mitad de las veces la frase B sí seguirá a la frase A y la otra mitad no.

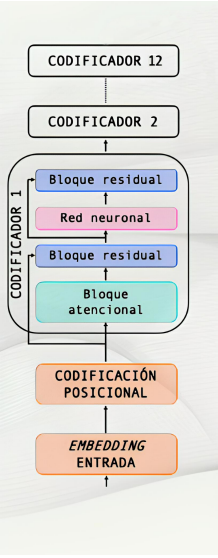


Figura 2.3: Arquitectura de BERT base.

2.2. Colecciones de evaluación

Recoger y crear grandes corpus de noticias es una tarea que requiere una gran cantidad de tiempo y de recursos. Además, presenta la desventaja de que con el tiempo se van creando nuevos términos o conceptos que no eran recogidos por la colección. A esto se le añade los impedimentos que las redes sociales ponen para descargar y compartir la información de sus usuarios. Es por ello por lo que no hay una gran cantidad de conjuntos de datos que contengan dicha información. A estos inconvenientes hay que añadir que si se quiere trabajar en otro idioma que no sea el inglés nos encontraremos que no existen tales conjuntos de datos. Por ejemplo, para el español el único corpus de noticias clasificadas que es reconocido es el Spanish Fake News Corpus (FakeDeS) creado en (Posadas-Durán et al., 2019). Este conjunto de datos contiene noticias verdaderas y falsas en español, sin embargo, no cuenta con ninguna información social asociada a las noticias.

En inglés podemos encontrarnos con el conjunto BuzzFeed-Webis Fake News Corpus 16 (Potthast et al., 2018). Con este conjunto se analizó 2282 publicaciones de Facebook provenientes de 9 medios políticos durante siete días de la semana cercana a las elecciones estadounidenses. Todos los medios se encontraban verificados en Facebook y las publicaciones fueron verificadas con posterioridad por periodistas profesionales de BuzzFeed. El conjunto de datos está formado por el ID de cada publicación, el ID del usuario de Facebook, la puntuación otorgada por BuzzFeed, el número de veces que se ha compartido, las reacciones y los comentarios. Posteriormente, en (Shu et al., 2018) se publicó BuzzFace, una extensión de BuzzFeed que contenía todas las respuestas de las publicaciones originales de Facebook. En total se recopiló 1.7 millones de publicaciones.

También en (Zhang et al., 2018) se presenta FakeNewsNet, un conjunto de datos que contiene noticias verdades y falsas. Para cada noticia contiene los tuits y los retuits relacionados con la misma, la información de los perfiles de usuarios involucrados (incluyendo la cronología y los datos de los seguidores y de los seguidos). En total se recopiló casi dos millones de tuits relacionados con la noticia y la información social de más de mil millones de usuarios.

2.3. IberLEF 2021

En el congreso IberLEF 2021 se propuso una tarea compartida cuyo objetivo consistía en clasificar una serie de noticias en verdaderas o falsas. Para ello se empleó el corpus FakeDeS expuesto anteriormente. Los participantes en esta tarea tenían que subir sus modelos y enfoques de resolución de la tarea. En Gómez-Adorno (McCarthy, 2018), se publicó un informe que recogía las características más importantes de los modelos con mejores rendimientos. En la Figura 2.4 pueden verse los resultados de esta tarea por los distintos grupos que enviaron un modelo.

Team	Fake	True	F_{macro}	Accuracy
GDUFS_DM	**0.7666**	**0.7649**	**0.7666**	**0.7657**
Haha	0.7548	0.7522	0.7548	0.7535
Chats_	0.7514	0.7690	0.7514	0.7605
SINAI	0.7385	0.7821	0.7385	0.7622
baseline-BERT	0.7321	0.7432	0.7321	0.7378
baseline-BOW-SVM	0.7217	0.7359	0.7217	0.729
Lcad_UFES	0.7102	0.6837	0.7102	0.6976
CiTIUS-NLP	0.7098	0.4940	0.7098	0.6311
baseline-CHAR-3-GRAMS-SVM	0.7063	0.6883	0.7063	0.6976
zk15120170770	0.7053	0.6053	0.7053	0.6626
ForceNLP	0.6925	0.4739	0.6925	0.6119
GRX	0.6915	0.5624	0.6915	0.6381
TSIA	0.6860	0.5263	0.686	0.6224
FREE	0.6855	0.6519	0.6855	0.6696
LIMCA	0.6812	0.7027	0.6812	0.6923
ZZWEI	0.6737	0.6794	0.6737	0.6766
Premjithb	0.6576	0.7177	0.6576	0.6906
Sdamian	0.6542	0.75	0.6542	0.7098
Yeti	0.6316	0.609	0.6316	0.6206
Gulu	0.6226	0.476	0.6226	0.5612
Nicksss	0.6119	0.7592	0.6119	0.7028
Bribones tras la esmeralda perdida	0.5835	0.5878	0.5835	0.5857
WSSC	0.5118	0.6657	0.5118	0.6031
Skblaz	0.4838	0.649	0.4838	0.5822

Figura 2.4: Resultados IberLEF 2021 sobre el conjunto de test.

Entre los enfoques empleados para su resolución cabe mencionar que los participantes del equipo GDUFS_DM, equipo que consiguió la mejor exactitud, empleó un modelo BERT y memoria de muestras con un mecanismo de atención. El método consistía en tomar los primeros y últimos segmentos de los textos y alimentarlos en un sistema BERT, obteniendo dos incrustaciones (cabeza y cola). Además, se cuenta con una matriz denominada 'memoria de muestra', que se obtiene tomando una muestra aleatoria de las incrustaciones de la cabeza y la cola; esta matriz se utiliza en un mecanismo de atención

con el resto de los textos. En contraposición al enfoque de GDUFS_DM, los participantes del equipo Haha, equipo que obtuvo la segunda posición, emplearon una selección de características con un pesado tf-idf y un perceptrón multicapa. Este modelo no solamente analizaba el contenido de la noticia, sino que combinaba información como el editor de la noticia o el tema de la misma.

Capítulo 3

Conjunto de datos

En este capítulo se describe en profundidad el conjunto de datos con el que se trabajará. En un primer momento se hará un análisis exploratorio del conjunto de datos. Posteriormente, se expondrá el proceso de extracción de la información social y se realizará un análisis de la información social extraída.

3.1. Análisis exploratorio de los datos

El conjunto de datos con el que trabajaremos es el Spanish Fake News Corpus (FakeDeS), este contiene pares de publicaciones en español falsas y verdaderas sobre diferentes eventos que fueron recopiladas desde noviembre de 2020 hasta marzo de 2021. Para recopilar la información se emplearon principalmente sitios web de periódicos y sitios web de verificación de hechos.

El conjunto de datos está dividido en 3 archivos con un total de 1543 noticias. Los archivos `train.xlsx`, `development.xlsx` y `test.xlsx`. Por la metodología empleada se ha decidido unir los archivos `train.xlsx` y `development.xlsx` obteniendo lo que denominaremos conjunto de entrenamiento. Cada uno de estos archivos tiene una serie de columnas que se describen a continuación:

- Id *(int)*: asigna un identificador a cada instancia.
- Categoría *(str)*: indica la categoría de la noticia (verdadera o falsa)
- Tema *(str)*: indica el tema relacionado con la noticia.
- Fuente *(str)*: indica el nombre de la fuente de la noticia.

- Titular *(str)*: contiene el titular de la noticia.

- Texto *(str)*: contiene el texto sin procesar de la noticia.

- Enlace *(str)*: contiene la url de la noticia en la fuente.

El conjunto de entrenamiento tiene en total 971 noticias, de las cuales 480 son falsas y 491 son verdaderas. Por su parte, el conjunto de test está formado por 572 noticias de las cuales la mitad son verdaderas y la mitad son falsas. Por tanto, estamos ante conjuntos de datos balanceados.

Los temas tratados en el corpus de entrenamiento abarcan desde política hasta deporte. El número de noticias sobre estos temas puede verse en la Figura 3.1.

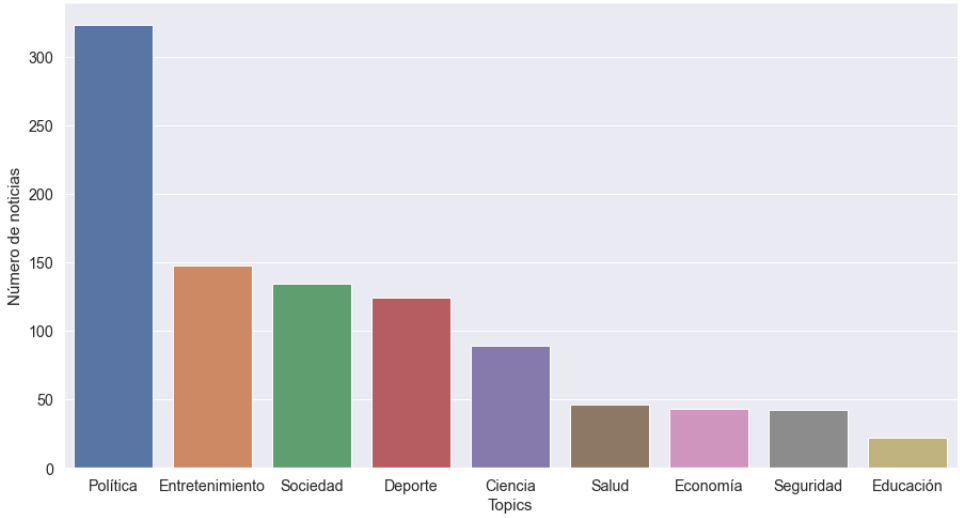


Figura 3.1: Número de noticias por temas

Como puede observarse, existe un gran número de noticias relacionadas con la política. A continuación, la Figura 3.2 muestra la distribución de noticias verdaderas y falsas para cada uno de los diferentes temas.

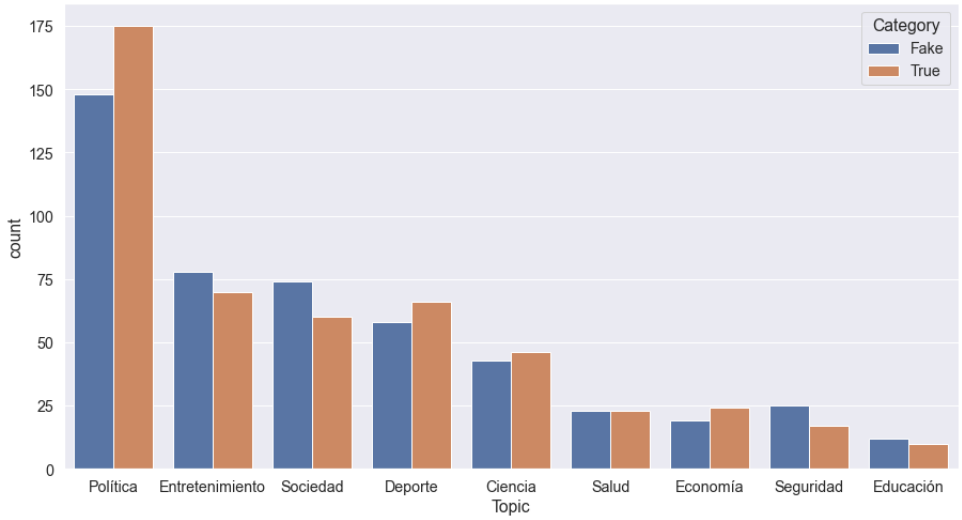


Figura 3.2: Distribución de noticias por temas

A la vista del gráfico anterior, dentro de cada tema volvemos a encontrarnos con un conjunto balanceado. Como muestra la Tabla 3.1, el conjunto de test dentro de cada tema también se encuentra balanceado. Además, es necesario señalar que el conjunto de test cuenta con el tema Covid-19, mientras que el conjunto de entrenamiento no presenta ninguna noticia de dicha temática (lo más similar son las noticias de salud pero en ningún caso relacionadas con el Covid-19). Por lo tanto, los modelos que se planteen tendrán que clasificar correctamente esta temática sin haberla visto en el entrenamiento.

Tema	Falso	Verdadero	Total
Ambiental	2	2	4
Ciencia	7	6	13
Covid-19	119	118	237
Deporte	1	1	2
Internacional	7	7	14
Política	54	53	107
Sociedad	96	99	195
Total	286	286	572

Tabla 3.1: Distribución de noticias en el conjunto de test

3.2. Extracción de información social

El principal objetivo de del trabajo es estudiar la información que aporta la información social a la hora de detectar noticias falsas, y como se ha comentado en el Capítulo 2, no hay ningún corpus en español que contenga esta información. Es por ello por lo que se ha decidido extraer esta información de la red social Twitter. Para ello se ha hecho uso de la API que la plataforma proporciona junto con el paquete Tweepy de Python (Roesslein, 2020).

Con el fin de evitar errores en las consultas se ha codificado tanto la url como el titular de cada noticia al formato Uniform Resource Identifier (URI). Para cada noticia se ha buscado aquellos tuits que contengan el titular de la noticia o el enlace a la misma. Sin embargo, se han dado errores debido a que la longitud máxima de las consultas a la API es de 1024 caracteres. Para solventar este problema se han eliminado caracteres especiales de los titulares de las noticias.

Según los artículos (Buntain y Golbeck, 2017) y (Albahar, 2021) existe una serie de metadatos de los tuits que permiten extraer información sobre si el usuario puede ser propenso a la propagación de noticias falsas o el tuit puede contener información no verídica. De esta forma, se ha decidido extraer los siguientes metadatos de cada uno de los tuits.

- **Tuit.** Texto del tuit *(str)*, id del autor *(int)*, id del tuit *(int)*, número de retuits *(int)*, número de respuestas al tuit *(int)*, número de me gusta *(int)*, número de citas del tuit *(int)*.

- **Usuario.** Nombre de usuario *(str)*, fecha de creación del usuario *(date) ISO 8601*, usuario verificado *(bool)*, número de seguidores*(int)*, número de seguidos *(int)*, número de tuits *(int)*, número de veces listado *(int)*.

La información social recopilada se guarda en una lista donde cada elemento es un diccionario con la siguiente estructura.

{'tweet_info': {'text':, 'author_id':, $'$id':, 'public_metrics': {'retweet_count':, 'reply_count':, 'like_count':, 'quote_count':}}, 'author_info': {'username':, 'verified':, 'name':, 'id':, 'public_metrics': {'followers_count': , 'following_count':, 'tweet_count':, 'listed_count':}, 'created_at':}}

Figura 3.3: Estructura diccionario metadatos de Twitter.

A continuación, haremos un pequeño estudio de los tuits recopilados. De las 1543 noticias, 643 no se han podido extraer ningún tuit, bien debido a que ninguna persona ha publicado sobre ello o bien por errores en las consultas a la API. La distribución por temas de estas noticias sin tuits puede verse en la Figura 3.4. Como puede apreciarse en la imagen, son ligeramente más las noticias falsas las que no reciben comentarios en Twitter.

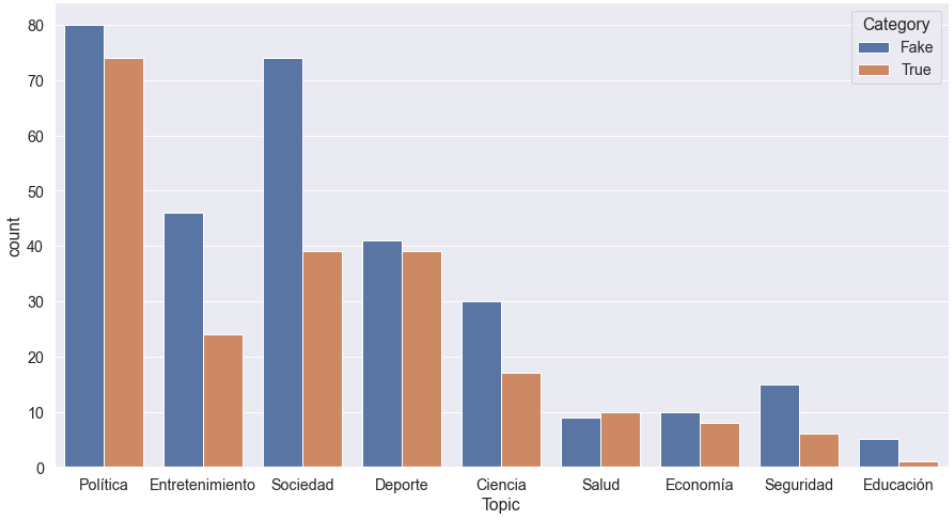


Figura 3.4: Distribución por temas de las noticias sin tuits.

Las noticias las cuales sí se ha podido extraer tuits sobre ellas presentan una distribución del número de tuits recopilados como puede verse en la Figura 3.5 . Esta distribución presenta una gran concentración en el intervalo de $(0, 200)$, concretamente un 86 % de las noticas. En este intervalo, la tendencia es que las noticias verdaderas reciban más interacción. Sin embargo, a medida que aumenta el número de tuits sobre cierta noticia, se puede ver que son las noticias falsas las que reciben más interacción. Esto puede apreciarse en el diagrama de violín de la Figura 3.6.

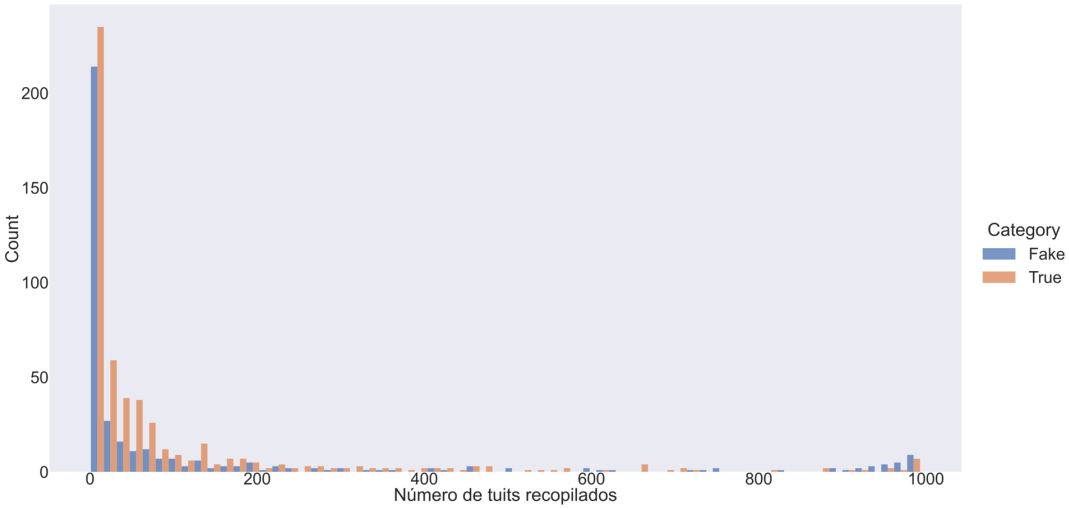


Figura 3.5: Distribución del número de tuits recopilados.

Figura 3.6: Diagrama de violín del número de tuits recopilados.

Cabe destacar que, aunque las noticias están escritas en español, existen tuits en inglés o en francés que hablan sobre dichas noticias. Esto se presenta especialmente en noticias relacionadas con el Covid-19.

Capítulo 4

Modelos de Aprendizaje Automático

En este capítulo se presentarán los modelos de machine learning empleados para comparar la importancia de la información social. Se ha decidido crear modelos usando varios enfoques diferentes: empleando la información textual de la noticia, empleando la información social de los tuits que hablan de dicha noticia y empleando métodos híbridos que emplean ambas características.

4.1. Métodos textuales

En esta sección se expondrán los métodos textuales empleados para la clasificación binaria de las noticias. Se ha empleado el texto completo de la noticia por lo que se ha tenido que realizar un preprocesado de la misma. Para los modelos no contextuales se han eliminado las urls, los emoticonos o expresiones no textuales, las stopwords (de, la, que, ...), se ha convertido el texto a minúsculas y se han aplicado los procesos de lematización y stemming. Sin embargo, para los modelos contextuales tan solo se han eliminado las urls.

Posteriormente se ha empleado 5 enfoques diferentes.

1. Modelo de espacio vectorial basado en bolsas de palabras (BoW).
2. Modelo de espacio vectorial mediante un pesado tf-idf.
3. Conteo de bigramas.
4. Redes Neuronales y aprendizaje profundo.

5. Modelos contextuales.

Para los enfoques 1, 2 y 3, puede verse en la Tabla 4.1 los diferentes modelos empleados con los hiperparámetros elegidos.

Para el entrenamiento de estos modelos se ha empleado el paquete de Python scikit-learn (Buitinck et al., 2013).

Modelos	**Hiperparámetros**	**Modelos entrenados**
Naive Bayes	*alpha*: $1, 0.1, 0.01$	3
SVM	C: $1, 10, 100$ *kernel*: linear	6
	C: $10, 100, 1000$ *gamma*: 0.0001 *kernel*: rbf	
Regresión Logística	*penalty*: l1, l2 C: $0.001, 0.01, 0.1, 1, 10, 100, 1000$	14
Árboles de decisión	*criterion*: gini, entropy *max_depth*: $4, 6, 10$ *min_samples_split*: $4, 6, 8$ *min_samples_leaf*: $3, 5$	36
Random Forest	*n_estimators*: $200, 500, 700$ *max_depth*: $5, 10, 30, 50$ *criterion*: gini, entropy	24

Tabla 4.1: Modelos empleados para enfoques BoW y TF-IDF

4.1.1. Redes neuronales y aprendizaje profundo

Se han entrenado múltiples modelos de redes neuronales. Los enfoques de estos pueden resumirse en:

- Perceptrones multicapa con entrada el vector del pesado tf-idf.
- Perceptrones multicapa y redes convolucionales con una capa de embedding.
- Perceptrones multicapa, redes convolucionales, redes LSTM, GRU y bidireccionales con una capa de embedding preentrenado.

Cada modelo presenta una topología diferente con el fin de optimizar los resultados y reducir el sobreajuste. Para el entrenamiento de estos modelos se ha empleado el paquete de Python Keras (Chollet y others, 2015).

4.1.2. Modelos contextuales

Se ha decidido seleccionar un modelo contextual preentrenado para el español. En particular se ha seleccionado el modelo BETO: Spanish BERT (Canete et al., 2020), este modelo es un modelo BERT entrenado con la técnica de enmascaramiento de palabras completas, sobre un gran corpus de más de tres billones de palabras en español. Se encuentra disponible tanto para palabras que contienen mayúsculas como tan solo para minúsculas. Al emplear solo palabras minúsculas se empeora ligeramente el resultado a cambio de una menor cantidad de parámetros. Se ha decidido emplear el modelo que emplea solo minúsculas.

Este modelo tiene una arquitectura como la que puede verse en la Figura 2.3. Está formado por una capa de embedding que reduce la dimensionalidad del espacio vectorial a 768 dimensiones, contiene 12 bloques codificadores y cada codificador posee 12 bloques atencionales.

Finalmente, para este modelo se le añade a la última capa de BETO una capa formada por 768 neuronas y una capa de salida formada por 2 neuronas con una función de activación SoftMax.

Para la importación y el entrenamiento de este modelo se ha empleado el paquete de Python Transformers (Wolf et al., 2020) y el repositorio Huggingface.

4.2. Métodos información social

Los métodos que emplean únicamente la información social de las noticias recopiladas emplean los siguientes 9 metadatos de cada tuit publicado: número de retuits, número de respuestas, número de me gusta del tuit, número de citas del tuit, usuario verificado, número de seguidores, número de seguidos, número de tuits del autor, número de veces que ha sido listado el autor. A continuación, para dejar constancia de la repercusión en redes sociales que ha tenido la noticia se añade el número de tuits recopilados para dicha noticia.

Para representar todos los tuits que hablan de cierta noticia se ha realizado una media de las anteriores características de cada tuit. Finalmente se ha añadido la desviación típica de cada característica. De esta forma se obtiene una matriz de datos con 20 columnas (dónde la columna relativa a la desviación del número de tuits de la noticia siempre es 0).

Una vez obtenida la matriz de características se ha empleado diferentes modelos de aprendizaje con diferentes exploraciones de hiperparámetros como puede verse en la Tabla 4.2. Toda la información sobre los modelos entrenados (hiperparámetros, tiempos de entrenamiento, precisión en 5 fold CV, ...) puede verse en los archivos `AutoML_social_info.ipynb`, `Modelos_ML_si.ipynb` y `pipeline_social_info.pkl`. Hay que tener en cuenta que se ha realizado el entrenamiento y la evaluación con aquellas noticias de las cuales se ha podido extraer tuits.

Para hacer un entrenamiento automático de algunos modelos se ha empleado el paquete de Python Auto-Sklearn (Feurer et al., 2020), este es un framework de Auto Machine Learning. Este paquete busca seleccionar algoritmos y optimizar sus hiperparámetros. En la Figura 4.1, puede ver una representación de auto-sklearn proporcionada por sus autores.

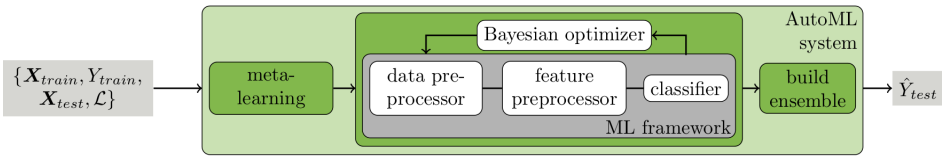


Figura 4.1: Flujo de trabajo de auto-sklearn

Modelos	**Modelos entrenados**
Random Forest	260
Árboles de decisión	65
SVM	40
Gradient Boosting	35
Adaptive Boosting (AdaBoost)	29
Perceptrón Multicapa (MLP)	24
Clasificador Pasivo-Agresivo	23
Bernouilli Naive Bayes	22
Análisis Discriminante Lineal (LDA)	21
Perceptrón con dos capas ocultas	24
Extremely Randomized Trees (Extra Trees)	20
K-Nearest	17
Análisis Discriminante Cuadrático	17
Regresión Logística	14
Multinomial Naive Bayes	6
Perceptrón con una capa oculta	4
Total	**621**

Tabla 4.2: Modelos entrenados con información social

4.3. Métodos híbridos

Se han desarrollado varios modelos híbridos que buscan aprovechar tanto la información textual que proporciona el texto de la noticia como la información social extraída de los datos de Twitter (tanto la información no textual de la sección 4.2 como el texto de los tuits recopilados). Para ello se han desarrollado 4 enfoques diferentes.

- Modelo HY1: Modelo que emplea la información social de la noticia como probabilidades y el texto de la noticia.
- Modelo HY2: Modelo que emplea la información social de la noticia y el texto de la noticia.
- Modelo HY3: Modelo que emplea la información social de la noticia como probabilidades, el texto de los tuits recopilados y el texto de la noticia.
- Modelo HY4: Modelo que emplea la información social de la noticia, el texto de los tuits recopilados y el texto de la noticia.

4.3.1. Modelo Híbrido 1 (HY1)

En este modelo, para cada noticia se emplea un modelo especializado en clasificar las noticias empleando información social. Para ello se selecciona el mejor modelo de la sección 4.2 (Random Forest). De este modelo, para cada noticia se extraen las probabilidades de ser verdadera o falsa utilizando como entrada la fila correspondiente de la matriz de características sociales con desviación típica descrita en dicha sección. En el caso de que en una noticia no se hubiera podido extraer ningún tuit, la salida sería un vector de dos ceros.

De forma paralela se procesa el texto de la noticia utilizando el modelo BETO: Spanish BERT descrito en la sección 4.1.2. Obteniendo como salida un vector de dimensión 768.

A continuación, se unen los dos vectores para obtener un vector de dimensionalidad 770. El esquema de trabajo puede verse en la Figura 4.2

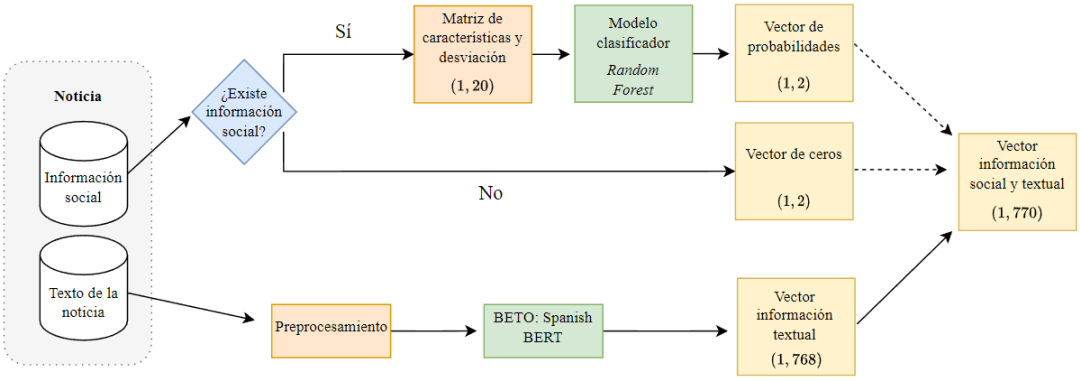


Figura 4.2: Flujo de trabajo del modelo HY1.

Una vez procesadas todas las noticias siguiendo el diagrama anterior, se ha empleado varios modelos de Machine Learning (mediante búsqueda manual y Auto-Sklearn) como puede verse en la Tabla 4.3. Toda la información sobre los modelos entrenados (hiperparámetros, tiempos de entrenamiento, precisión en 5 fold CV, ...) puede verse en los archivos `HY1_AutoML.ipynb`, `HY1_ML.ipynb` y `pipeline_RF_noticia.pkl`.

Modelos	**Modelos entrenados**
Análisis Discriminante Lineal (LDA)	484
Clasificador Pasivo-Agresivo	106
Análisis Discriminante Cuadrático	98
SVM	90
Random Forest	80
Árboles de decisión	77
Bernouilli Naive Bayes	63
Gradient Boosting	50
K-Nearest	48
Adaptive Boosting (AdaBoost)	47
Extremely Randomized Trees (Extra Trees)	42
Perceptrón Multicapa (MLP)	39
Perceptrón con dos capas ocultas	20
Regresión Logística	14
Multinomial Naive Bayes	10
Perceptrón con una capa oculta	5
Total	**1273**

Tabla 4.3: Modelos entrenados para el modelo HY1

4.3.2. Modelo Híbrido 2 (HY2)

En este modelo, para cada noticia se emplea la información social no textual y el texto de la noticia. Para cada noticia se extrae la fila correspondiente de la matriz de características sociales con desviación típica descrita en la sección 4.2. En el caso de que en una noticia no se hubiera podido extraer ningún tuit, la salida sería un vector de ceros de dimensión 20.

De forma paralela se procesa el texto de la noticia utilizando el modelo BETO: Spanish BERT descrito en la sección 4.1.2. Obteniendo como salida un vector de dimensión 768.

A continuación, se unen los dos vectores para obtener un vector de dimensionalidad 788. El esquema de trabajo puede verse en la Figura 4.3

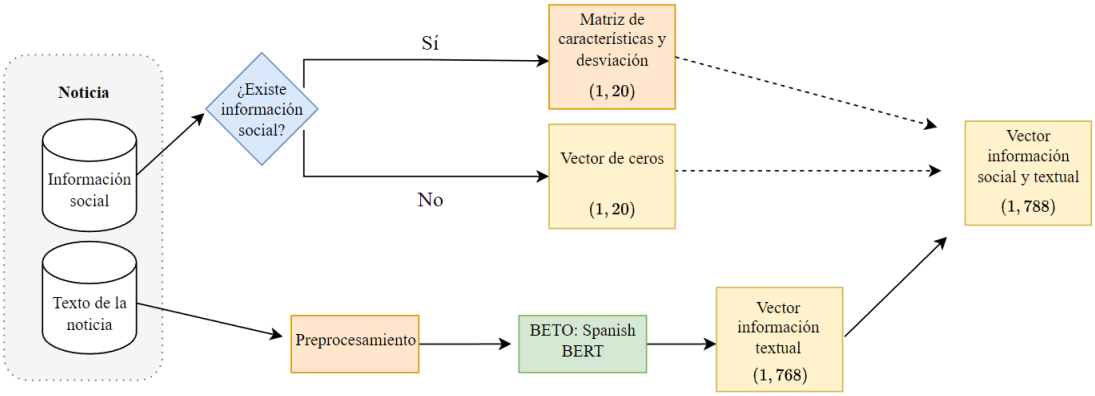


Figura 4.3: Flujo de trabajo del modelo HY2.

Una vez procesadas todas las noticias siguiendo el diagrama anterior, se ha empleado varios modelos de Machine Learning (mediante búsqueda manual y Auto-Sklearn) como puede verse en la Tabla 4.4. Toda la información sobre los modelos entrenados (hiperparámetros, tiempos de entrenamiento, precisión en 5 fold CV, ...) puede verse en los archivos `HY2_AutoML.ipynb`, `HY2_ML.ipynb` y `pipeline_matriz_texto.pkl`.

Modelos	**Modelos entrenados**
Random Forest	230
Árboles de decisión	49
Perceptrón con tres capas ocultas	40
SVM	32
Bernouilli Naive Bayes	28
Clasificador Pasivo-Agresivo	23
Análisis Discriminante Lineal (LDA)	20
Perceptrón con dos capas ocultas	20
K-Nearest	17
Extremely Randomized Trees (Extra Trees)	16
Análisis Discriminante Cuadrático	15
Gradient Boosting	15
Adaptive Boosting (AdaBoost)	15
Perceptrón multicapa (MLP)	12
Regresión Logística	14
Multinomial Naive Bayes	5
Perceptrón con una capa oculta	5
Total	**556**

Tabla 4.4: Modelos entrenados para el modelo HY2

4.3.3. Modelo Híbrido 3 (HY3)

En este modelo se emplea la información social no textual, el texto de los tuits recopilados y el texto de la noticia.

Para la información social y para el texto de la noticia se emplea el mismo flujo de trabajo que en el modelo HY1. Para la información social de cada noticia se obtiene un vector de longitud 2 y para la noticia un vector de longitud 768.

De forma paralela a estos dos procesos, para cada noticia con tuits recopilados se preprocesa cada uno de los tuits (eliminando URLs y tokenizando) y se procesa utilizando el modelo preentrenado XLM-roBERTa-base (Barbieri, Espinosa-Anke, y Camacho-Collados, 2022).

Este modelo transformer ha sido entrenado sobre un corpus de unos 198 millones de tuits en 8 diferentes lenguajes (español, árabe, inglés, francés, alemán, hindú, portugués e italiano) y se ha especializado en la clasificación de sentimientos (positivo, negativo o neutral). En nuestro caso se eliminará la última capa del modelo obteniendo como salida un vector de longitud 768 que representará las características más relevantes del texto del tuit.

Para cada tuit disponible se ha procesado por el modelo anterior obteniendo un vector de longitud 768. Finalmente se ha hecho una media de todos los vectores de los tuits de la noticia para obtener un vector que represente a los tuits de dicha noticia. En el caso de que la noticia no tuviera información social se devuelve un vector de ceros.

A continuación, se unen los tres vectores para obtener un vector de dimensionalidad 1538. El esquema de trabajo puede verse en la Figura 4.4

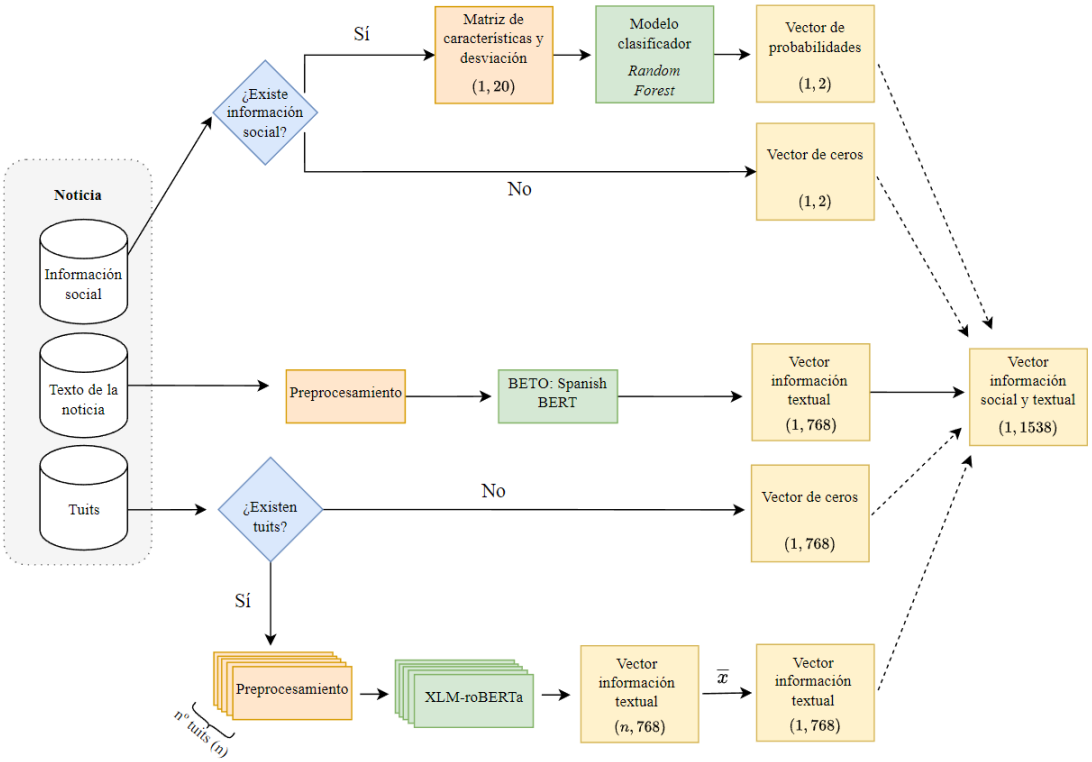


Figura 4.4: Flujo de trabajo del modelo HY3.

Una vez procesadas todas las noticias siguiendo el diagrama anterior, se ha empleado varios modelos de Machine Learning (mediante búsqueda manual y Auto-Sklearn) como puede verse en la Tabla 4.5. Toda la información sobre los modelos entrenados (hiperparámetros, tiempos de entrenamiento, precisión en 5 fold CV, ...) puede verse en los archivos `HY3_AutoML.ipynb`, `HY3_ML.ipynb` y `pipeline_RF_Texto_Tuits.pkl`.

Modelos	Modelos entrenados
SVM	229
Análisis Discriminante Lineal (LDA)	132
Árboles de decisión	118
Random Forest	77
Gradient Boosting	57
Clasificador Pasivo-Agresivo	48
Adaptive Boosting (AdaBoost)	48
Extremely Randomized Trees (Extra Trees)	47
Perceptrón con tres capas ocultas	40
Bernouilli Naive Bayes	39
Análisis Discriminante Cuadrático	39
Perceptrón multicapa (MLP)	31
K-Nearest	30
Perceptrón con dos capas ocultas	20
Regresión Logística	14
Multinomial Naive Bayes	7
Perceptrón con una capa oculta	5
Total	**982**

Tabla 4.5: Modelos entrenados para el modelo HY3

4.3.4. Modelo Híbrido 4 (HY4)

En este modelo se emplea la información social no textual, el texto de los tuits recopilados y el texto de la noticia.

Para la información social y para el texto de la noticia se emplea el mismo flujo de trabajo que en el modelo HY2. Para la información social de cada noticia se obtiene un vector de longitud 20 y para la noticia un vector de longitud 768.

De forma paralela a estos dos procesos, para cada noticia con tuits recopilados se realiza el mismo procesamiento que en el modelo HY3. De esta forma se obtiene para cada noticia un vector de longitud 768.

A continuación, se unen los tres vectores para obtener un vector de dimensionalidad 1556. El esquema de trabajo puede verse en la Figura 4.5

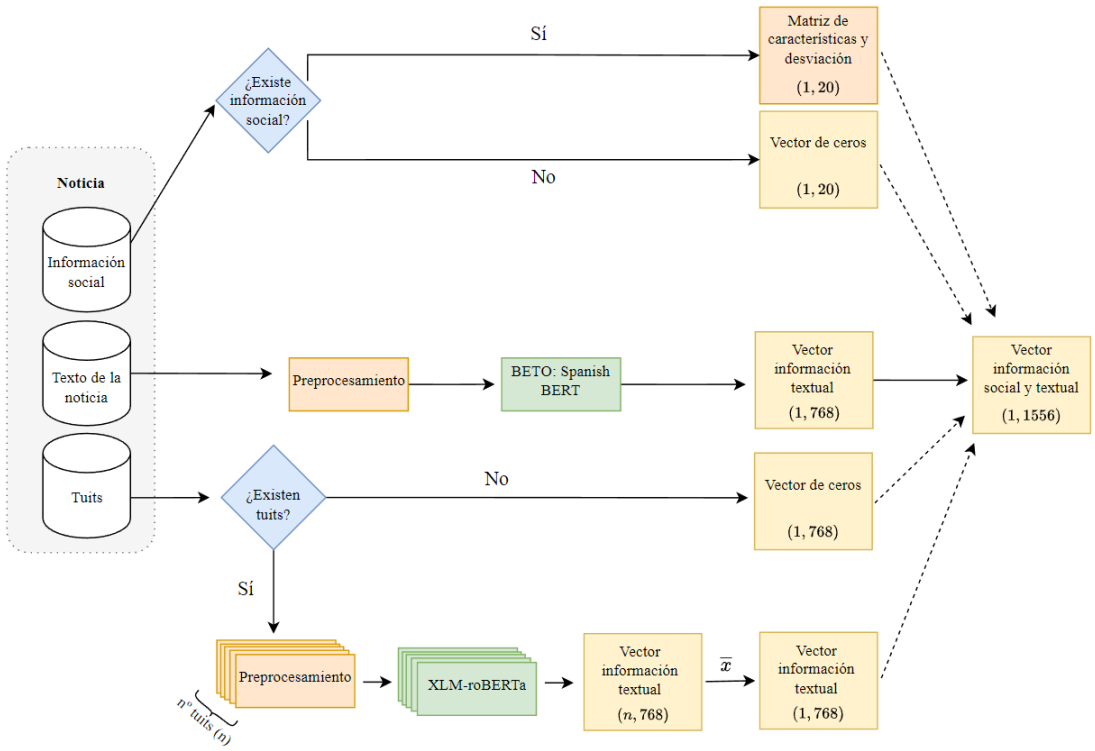


Figura 4.5: Flujo de trabajo del modelo HY4.

Una vez procesadas todas las noticias siguiendo el diagrama anterior, se ha empleado varios modelos de Machine Learning (mediante búsqueda manual y Auto-Sklearn) como puede verse en la Tabla 4.6. Toda la información sobre los modelos entrenados (hiperparámetros, tiempos de entrenamiento, precisión en 5 fold CV, ...) puede verse en los archivos `HY4_AutoML.ipynb`, `HY4_ML.ipynb` y `pipeline_matriz_tuits_texto.pkl`.

Modelos	**Modelos entrenados**
Gradient Boosting	142
Árboles de decisión	45
Random Forest	43
Perceptrón con tres capas ocultas	40
SVM	20
Perceptrón con dos capas ocultas	20
Clasificador Pasivo-Agresivo	18
Análisis Discriminante Lineal (LDA)	15
Regresión Logística	14
Perceptrón multicapa (MLP)	13
Adaptive Boosting (AdaBoost)	12
Extremely Randomized Trees (Extra Trees)	12
Bernouilli Naive Bayes	12
Análisis Discriminante Cuadrático	10
K-Nearest	8
Multinomial Naive Bayes	5
Perceptrón con una capa oculta	5
Total	**434**

Tabla 4.6: Modelos entrenados para el modelo HY4

Capítulo 5

Evaluación

En este capítulo se describe la metodología y las métricas empleadas para evaluar los modelos de aprendizaje automático del Capítulo 4. A continuación, se presentan los resultados obtenidos empleando en la evaluación.

5.1. Metodología de evaluación

Como se ha descrito en la sección 3.1, se han unido los archivos `train.xlsx` y `development.xlsx` obteniendo un único conjunto que se denominará conjunto de entrenamiento. A la hora de evaluar los modelos se han empleado dos metodologías diferentes.

Validación cruzada k-fold

La validación cruzada es uno de los métodos más utilizados para estimar el error de predicción de un modelo con un conjunto determinado de hiperparámetros. Existen diferentes métodos de validación cruzada pero el más común y el que se ha empleado se conoce como k-fold. (o validación cruzada de k pliegues).

La validación cruzada $k-$fold, Figura 5.1, divide al conjunto de datos en k partes iguales $\{P_1, \ldots, P_k\}$. Para cada P_n se entrena el modelo utilizando las otras $k-1$ partes y se calcula el error a la hora de predecir los datos de P_n (datos que nunca ha visto este modelo). Haciendo esto para las k partes obtenemos un conjunto de errores. Con estos k errores se calcula su media y su desviación para obtener una medida del error medio de ese modelo con esos hiperparámetros.

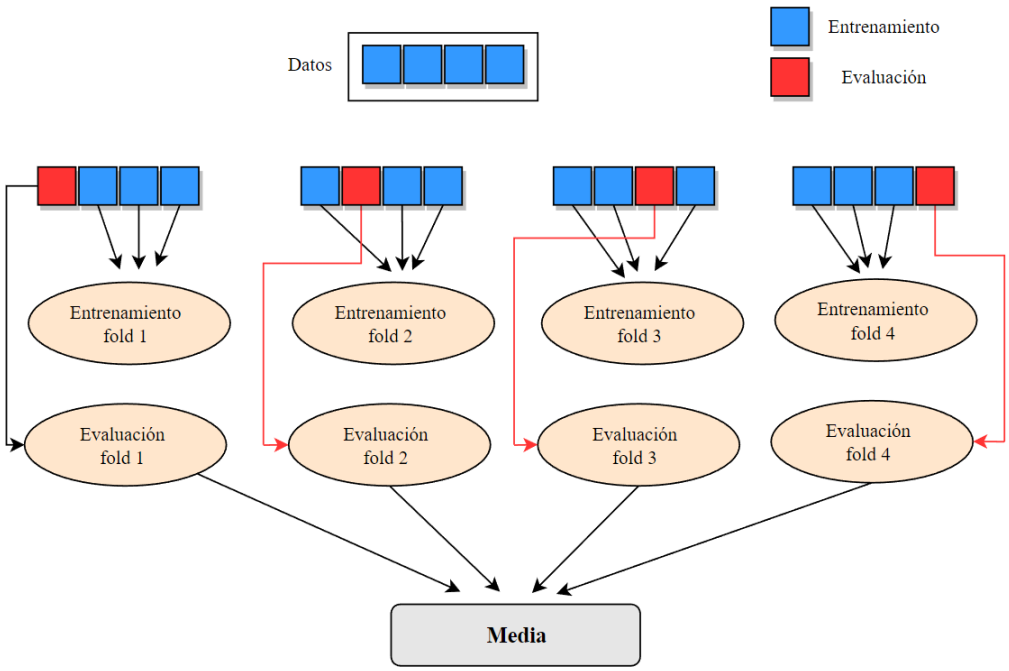


Figura 5.1: Método de validación cruzada k-fold.

Hay que tener en cuenta que este método requiere una carga computacional bastante grande, puesto que para una validación cruzada de k-folds se necesitaría entrenar k modelos. Por norma general se suele escoger entre el valor 5 o 10 como un buen compromiso entre el sesgo y la varianza. En nuestro caso se ha empleado una validación cruzada de 5 pliegues.

Evaluación conjunto de test

Finalmente, para el modelo que mejores resultados haya tenido en las evaluaciones anteriores se evaluará sobre el conjunto de test. Este conjunto nunca será visto por el modelo y permitirá obtener una representación de la capacidad de generalización del modelo.

5.2. Métricas de evaluación

Las métricas de evaluación son medidas que permiten valorar el rendimiento de un modelo de aprendizaje.

Matriz de confusión

Una matriz de confusión es una representación matricial de los resultados de las predicciones que se utiliza a menudo para describir el rendimiento de un modelo clasificador. Para ello se evalúa el modelo sobre un conjunto de datos de prueba (conocido como conjunto de test) cuyos valores reales se conocen.

Cuando se trata de problemas de clasificación binaria se suelen etiquetar a las clases objetivo como positivo y negativo. De esta forma cada predicción puede tener uno de estos cuatros resultados:

- Verdadero positivo (TP). Cuando la clase predicha es verdadera y la clase real es verdadera.
- Verdadero negativo (TN). Cuando la clase predicha es negativa y la clase real es negativa.
- Falso positivo (FP). Cuando la clase predicha es verdadera y la clase real es falsa.
- Falso negativo (FN). Cuando la clase predicha es falsa y la clase real es verdadera.

Estos cuatro valores quedan reflejados visualmente en la Tabla 5.1.

		Predicción	
		Positivo	**Negativo**
Realidad	**Positivo**	Verdadero positivo *TP*	Falso negativo *FN*
	Negativo	Falso Positivo *FP*	Verdadero negativo *TN*

Tabla 5.1: Matriz de confusión binaria

Se empleará la matriz de confusión para mostrar los resultados del mejor modelo, sobre el conjunto de test, de cada uno de los enfoques del Capítulo 4.

Exactitud

A lo hora de evaluar y seleccionar los modelos durante el entrenamiento se ha empleado la exactitud (en inglés *Accuracy*). Esta medida indica cuanto de alejados están los valores predichos por el modelo de su valor real.

En un problema de clasificación se define la exactitud como el número de clasificaciones correctas entre el número de clasificaciones totales. Si estamos ante un problema de clasificación binaria podemos definir la exactitud en términos de su matriz de confusión de la forma

$$Exactitud = \frac{TP + TN}{TP + TN + FP + FN}$$

Esta métrica no funciona bien cuando las clases están desbalanceadas puesto que para el modelo tendería a clasificar todos los datos con la etiqueta del conjunto mayoritario.

Precisión

La precisión (en inglés *Pecision*) es una medida que indica la proporción de datos clasificados como una clase y que fueron clasificados correctamente. De esta forma, en una clasificación binaria podemos tener la precisión para cada una de las dos clases de modo que

$$Precisión_{Positivo} = \frac{TP}{TP + FP} \qquad Precisión_{Negativo} = \frac{TN}{TN + FN}$$

Exhaustividad

La exhaustividad (en inglés *Recall*) es una medida que indica la proporción de datos que eran de una clase y fueron clasificados correctamente. De esta forma, en una clasificación binaria podemos tener la exhaustividad para cada una de las dos clases de modo que

$$Exh_{Positivo} = \frac{TP}{TP + FN} \qquad Exh_{Negativo} = \frac{TN}{TN + FP}$$

Valor F1

El valor F1 es una medida que permite combinar la precisión y la exhaustividad en un solo valor. Empleando esta medida se puede comparar el rendimiento combinado de la precisión y la exhaustividad entre varios modelos.

El valor F1 se calcula haciendo la media armónica entre la precisión y la exhaustividad. Al igual que en las medidas anteriores, podemos tener el valor F1 de cada una de las dos clases.

$$F1 = 2 \cdot \frac{Precisión \cdot Exhaustividad}{Precisión + Exhaustividad}$$

5.3. Resultados

En esta sección se expondrán los resultados de los diversos modelos entrenados. Para cada enfoque del Capítulo 4 se mostrarán los resultados de los siguientes métodos de evaluación.

- Dentro del entrenamiento de un enfoque en particular, se mostrará la exactitud de los mejores algoritmos empleados. Quedará reflejada la media de las exactitudes empleando una validación cruzada 5-fold (5.1).
- Para cada enfoque se seleccionará el modelo con mejor exactitud durante el entrenamiento. Posteriormente, se evaluará sobre el conjunto de test. Se expondrá la matriz de confusión, la exactitud, la precisión, la exhaustividad y el valor F1 sobre dicho conjunto.

5.3.1. Métodos textuales

Los resultados de los entrenamientos de los métodos descritos en la sección 4.1 quedan recogidos en la Tabla 5.2.

Métodos textuales	**Mejor modelo**	**Exactitud**
Pesado TF-IDF	Random Forest	**0.849**
Bolsa de palabras (BoW)	Random Forest	0.825
Bigramas	Random Forest	0.822
Perceptrón (Embedding)	Embedding dimensión 50 1 capa oculta 32 neuronas	0.786
Perceptrón (TF-IDF)	1 capa oculta 32 neuronas	0.751
Red convolucional	Embedding dimensión 50	0.740
Modelos Contextuales	BETO: Spanish BERT	0.727
Red recurrentes	Embedding dimensión 300 Bidireccional GRU	0.678

Tabla 5.2: Resultados entrenamiento modelos textuales

En la tabla se puede apreciar como aquellos modelos no neurales destacan sobre los que emplean redes neuronales. Este hecho podría deberse a que los modelos que se están empleando tienen una gran cantidad de parámetros a optimizar y contamos con una serie de datos bastante limitada. Cabe destacar que el uso de embedding preentrenados ha resultado en un menor

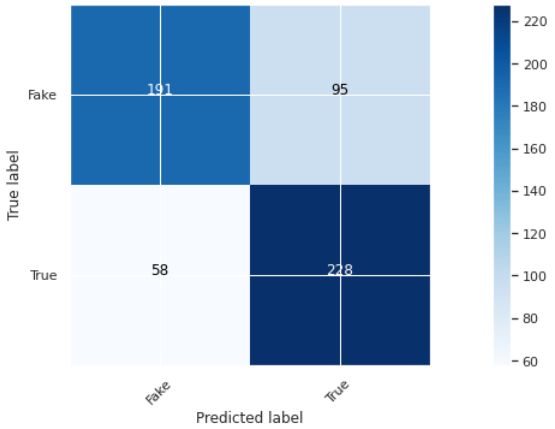
rendimiento que entrenar los embeddings desde cero. También es reseñable el poco rendimiento que se obtiene con las redes recurrentes, modelos que han requerido una gran cantidad de tiempo de entrenamiento y que comúnmente se han empleado para los problemas de procesamiento del lenguaje.

El enfoque que mejores resultados ha obtenido ha sido utilizando un pesado tf-idf junto con un modelo Random Foret. Además, para este modelo se ha realizado una selección de características empleando el test estadístico Chi-Cuadrado con el objetivo de reducir la dimensionalidad del problema. Esta se ha reducido de 24223 a 5000.

Los hiperparámetros de este modelo han sido:

criterion: gini, max_depth: 30, max_features: auto, n_estimators: 700

Evaluando este modelo sobre el conjunto de test obtenemos las métricas y la matriz de confusión de la Figura 5.2. Se puede apreciar como el modelo tiene una exhaustividad para la clase Fake de 0.67, que junto con el 0.80 de la clase Real refleja que el modelo tiende a clasificar considerablemente peor las noticias falsas que las verdaderas. Este presenta un valor F1 de 0.73 y una exactitud de 0.733.



	Fake	Real
Precisión	0.77	0.71
Exhaustividad	0.67	0.80
F1	0.71	0.75
Media F1	0.73	
Exactitud	0.733	

Figura 5.2: Métricas de evaluación modelos textuales

5.3.2. Métodos información social

Los resultados de los entrenamientos de los métodos descritos en la sección 4.2 quedan recogidos en la Tabla 5.3.

Modelos Información Social	**Exactitud**
Random Forest	**0.845**
Gradient Boosting	0.834
Adaptive Boosting (AdaBoost)	0.826
Extremely Randomized Trees (Extra Trees)	0.817
Árboles de decisión	0.797
K-Nearest	0.788
Perceptrón Multicapa (MLP)	0.787
SVM	0.785
Clasificador Pasivo - Agresivo	0.785
Perceptrón con dos capas ocultas	0.783
Análisis Discriminante Lineal (LDA)	0.781
Multinomial Naive Bayes	0.781
Perceptrón con una capa oculta	0.781
Bernouilli Naive Bayes	0.779
Análisis Discriminante Cuadrático	0.776
Regresión Logística	0.703

Tabla 5.3: Resultados entrenamiento modelos sociales

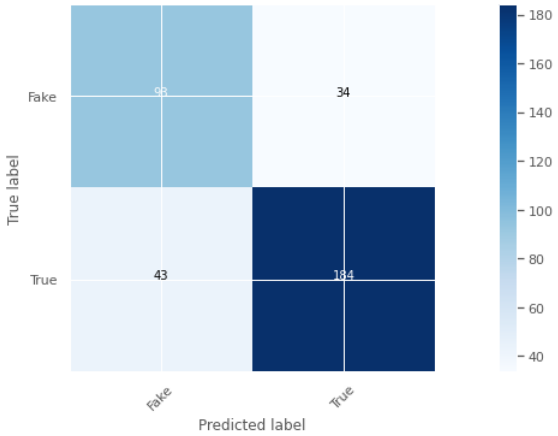
Podemos apreciar que los modelos presentan una exactitud bastante elevada. Los modelos basados en árboles ocupan las 5 primeras posiciones de la lista. Además, aquellos basados en árboles destacan sobre los árboles de decisión individuales.

El enfoque que mejores resultados ha obtenido ha sido un modelo Random Forest. Los hiperparámetros de este modelo han sido:

criterion: entropy, max_depth: 50, max_features: auto, n_estimators: 500

Evaluando este modelo sobre el conjunto de test obtenemos las métricas y la matriz de confusión de la Figura 5.3. Se puede apreciar que el modelo está bastante equilibrado en cuanto a los errores a la hora de clasificar. Presenta un valor F1 de 0.77 y una exactitud de 0.783.

Sin embargo, hay que recordar que este modelo tan solo ha sido entrenado y evaluado con aquellas noticias de las cuales se ha podido extraer información social, por lo que el conjunto de entrenamiento y test es más reducido que en el resto de casos.



	Fake	Real
Precisión	0.68	0.84
Exhaustividad	0.73	0.81
F1	0.71	0.83
Media F1	0.77	
Exactitud	0.783	

Figura 5.3: Métricas de evaluación modelos sociales

Después de entrenar estos modelos se ha realizado un estudio de qué características de información social son las más relevantes para el modelo. Pare ello se ha empleado la importancia de la permutación expuesto en (Altmann et al., 2010). El método consiste en entrenar el modelo con el conjunto de entrenamiento y obtener una puntuación sobre el conjunto de test. Esta será nuestra línea base. A continuación, se mezclan los valores de una variable y se calcula el resultado con esta mezcla. Si la característica que se acaba de mezclar es importante, el modelo debería sufrir una bajada considerable de rendimiento. Por otro lado, si la característica no es importante, el modelo debería de verse muy poco afectado. Este proceso aleatorio se repite un gran número de veces para reducir la incertidumbre. Los resultados de este proceso en el modelo Random Forest que mejores resultados ha dado puede verse en la Figura 5.4

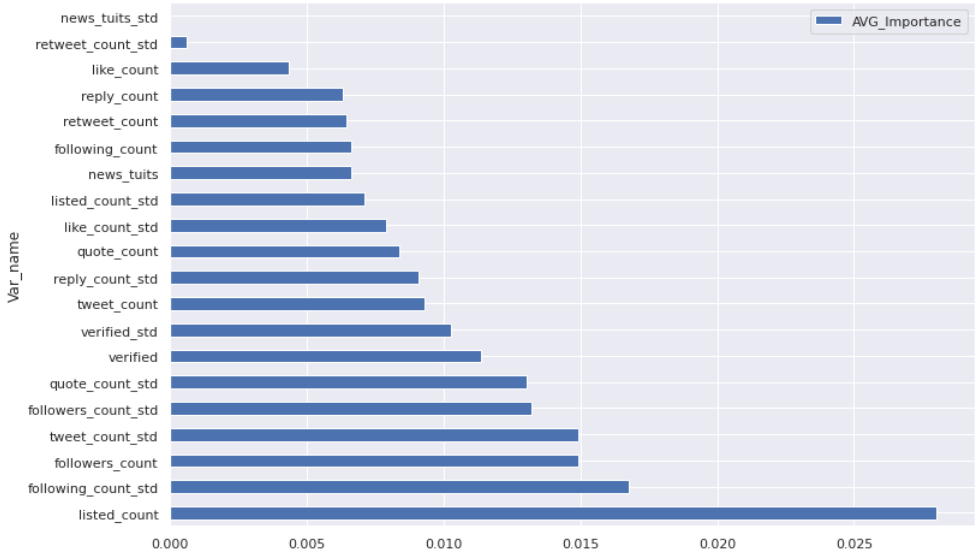


Figura 5.4: Importancias de las características sociales.

5.3.3. Modelo híbrido HY1

Los resultados de los entrenamientos de los métodos descritos en la sección 4.3.1 quedan recogidos en la Tabla 5.4.

Modelos HY1	**Exactitud**
Random Forest	**0.818**
Árboles de decisión	**0.818**
Regresión Logística	**0.818**
Perceptrón con una capa oculta	0.810
Perceptrón con dos capas ocultas	0.810
Análisis Discriminante Lineal (LDA)	0.809
Clasificador Pasivo-Agresivo	0.809
Análisis Discriminante Cuadrático	0.809
SVM	0.809
Bernouilli Naive Bayes	0.809
Gradient Boosting	0.809
K-Nearest	0.809
Adaptive Boosting (AdaBoost)	0.809
Extremely Randomized Trees (Extra Trees)	0.809
Perceptrón Multicapa (MLP)	0.809
Multinomial Naive Bayes	0.809

Tabla 5.4: Resultados entrenamiento modelo híbrido HY1

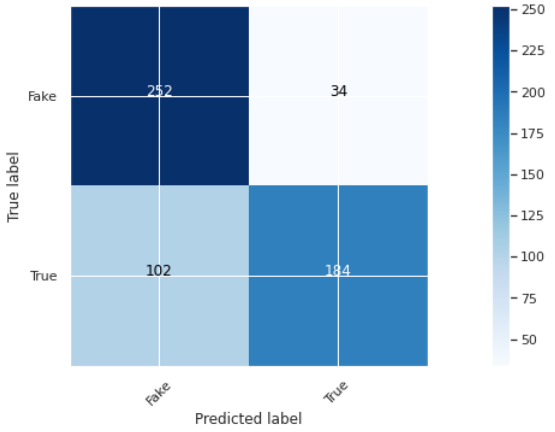
A la vista de los resultados del entrenamiento cualquiera de los 3 primeros

modelos sería válido para su elección. El resto de modelos presentan una exactitud muy similar a los tres primeros. Se ha decidido seleccionar la regresión logística sobre los árboles de decisión y el Random Forest puesto que es un algoritmo más simple, con un menor número de hiperparámetros y con un menor coste computacional.

Los hiperparámetros de este modelo han sido:

C: 0.01, penalty: l2

Evaluándolo sobre el conjunto de test obtenemos las métricas y la matriz de confusión de la Figura 5.5. Puede apreciarse que aunque el modelo presenta una exactitud y un valor F1 alto, el modelo obtiene esta puntuación debido a que clasifica muy bien las noticias falsas. Sin embargo, las noticias verdaderas tan solo son clasificadas correctamente el 64 % de las mismas.



	Fake	Real
Precisión	0.71	0.84
Exhaustividad	0.88	0.64
F1	0.79	0.73
Media F1	0.76	
Exactitud	0.762	

Figura 5.5: Métricas de evaluación modelo híbrido HY1

5.3.4. Modelo híbrido HY2

Los resultados de los entrenamientos de los métodos descritos en la sección 4.3.2 quedan recogidos en la Tabla 5.5.

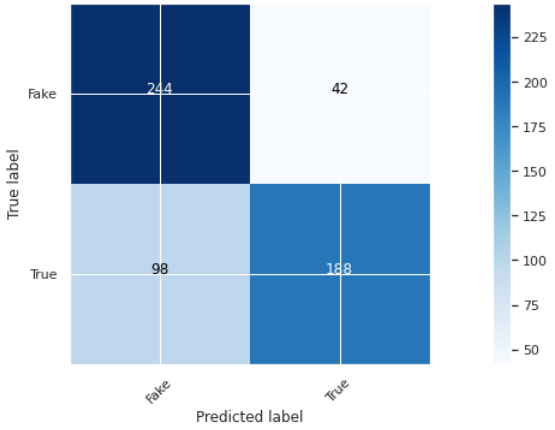
Modelos HY2	**Exactitud**
Random Forest	**0.720**
Gradient Boosting	0.704
Adaptive Boosting (AdaBoost)	0.703
Árboles de decisión	0.695
SVM	0.693
Análisis Discriminante Lineal (LDA)	0.688
Clasificador Pasivo-Agresivo	0.686
Extremely Randomized Trees (Extra Trees)	0.685
Análisis Discriminante Cuadrático	0.685
Perceptrón con dos capas ocultas	0.684
Perceptrón con tres capas ocultas	0.683
Perceptrón multicapa (MLP)	0.682
Perceptrón con una capa oculta	0.681
Multinomial Naive Bayes	0.678
Bernouilli Naive Bayes	0.676
K-Nearest	0.674
Regresión Logística	0.666

Tabla 5.5: Resultados entrenamiento modelo híbrido HY2

Los modelos que mejor rendimiento presentan son aquellos que están basados en árboles de decisión y más concretamente en bosques aleatorios. El modelo con el que se ha obtenido una mayor exactitud ha sido un Random Forest cuyos hiperparámetros han sido:

criterion: gini, max_depth: 30, max_features: auto, n_estimators: 700

Evaluando este modelo sobre el conjunto de test obtenemos las métricas y la matriz de confusión de la Figura 5.6. A la vista de estos resultados, el modelo presenta una mayor predisposición a la clasificación correcta de noticias falsas y comete más error en la clasificación de noticias verdaderas.



	Fake	Real
Precisión	0.71	0.82
Exhaustividad	0.85	0.66
F1	0.78	0.73
Media F1	0.75	
Exactitud	0.756	

Figura 5.6: Métricas de evaluación modelo híbrido HY2

5.3.5. Modelo híbrido HY3

Los resultados de los entrenamientos de los métodos descritos en la sección 4.3.3 quedan recogidos en la Tabla 5.6.

Modelos HY3	**Exactitud**
Árboles de decisión	**0.818**
Regresión Logística	**0.818**
SVM	0.809
Análisis Discriminante Lineal (LDA)	0.809
Random Forest	0.809
Gradient Boosting	0.809
Clasificador Pasivo-Agresivo	0.809
Adaptive Boosting (AdaBoost)	0.809
Extremely Randomized Trees (Extra Trees)	0.809
Análisis Discriminante Cuadrático	0.809
Perceptrón multicapa (MLP)	0.809
K-Nearest	0.809
Perceptrón con tres capas ocultas	0.809
Perceptrón con dos capas ocultas	0.808
Perceptrón con una capa oculta	0.808
Multinomial Naive Bayes	0.631
Bernouilli Naive Bayes	0.607

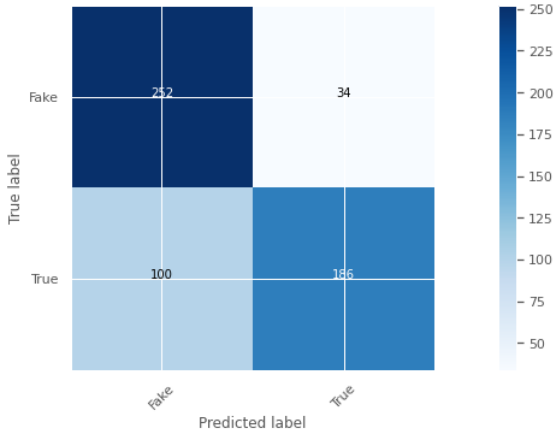
Tabla 5.6: Resultados entrenamiento modelo híbrido HY3

Como ocurría anteriormente, tenemos dos modelos que presentan los mismos resultados en la evaluación. El resto de modelos presentan una exactitud bastante similar, con excepción de los clasificadores Naive Bayes. Al

igual que antes, se ha decidido seleccionar la regresión logística debido a su simplicidad. Los hiperparámetros de este modelo son:

C: 0.1, penalty: l2

Evaluando este modelo sobre el conjunto de test obtenemos las métricas y la matriz de confusión de la Figura 5.7. Nos encontramos un modelo que presenta un valor alto tanto de exactitud como de valor F1. Sin embargo, presenta una exhaustividad de 0.65 para la clase Real por lo que la mayor parte de su rendimiento se encuentra clasificando noticias falsas.



	Fake	Real
Precisión	0.72	0.85
Exhaustividad	0.88	0.65
F1	0.79	0.74
Media F1	0.76	
Exactitud	0.7657	

Figura 5.7: Métricas de evaluación modelo híbrido HY3

5.3.6. Modelo híbrido HY4

Los resultados de los entrenamientos de los métodos descritos en la sección 4.3.4 quedan recogidos en la Tabla 5.7.

Modelos HY4	**Exactitud**
Gradient Boosting	**0.711**
Perceptrón con dos capas ocultas	0.703
Perceptrón con tres capas oculta	0.703
Regresión Logística	0.702
Extremely Randomized Trees (Extra Trees)	0.701
Random Forest	0.695
Perceptrón con una capa oculta	0.691
Adaptive Boosting (AdaBoost)	0.688
Análisis Discriminante Lineal (LDA)	0.686
Árboles de decisión	0.684
Clasificador Pasivo-Agresivo	0.673
SVM	0.665
Perceptrón multicapa (MLP)	0.658
K-Nearest	0.658
Análisis Discriminante Cuadrático	0.654
Bernouilli Naive Bayes	0.570
Multinomial Naive Bayes	0.570

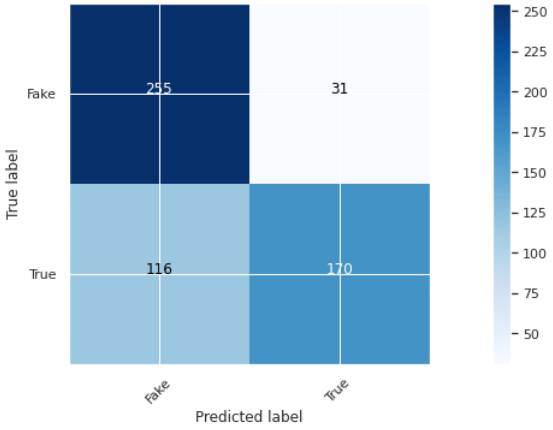
Tabla 5.7: Resultados entrenamiento modelo híbrido HY4

A la vista de la tabla anterior el modelo que mejores resultados ha obtenido ha sido un Gradient Boosting. No obstante, los perceptrones con dos y tres capas ocultas obtienen una exactitud bastante similar.
Hay que tener en cuenta que el modelo HY4 es el modelo que presenta un mayor número de características (concretamente 1556 por cada noticia), por lo que estos perceptrones con topologías profundas podrían presentar sobreajuste.

Los hiperparámetros empleados para el entrenamiento del Gradient Boosting han sido:

l2_regularization: 0, learning_rate: 0.08198, loss: auto, max_bins: 255, max_depth: None, max_leaf_nodes: 63, min_samples_leaf: 6, n_iter_no_change: 11, scoring: loss, tol:0

Evaluando este modelo sobre el conjunto de test obtenemos las métricas y la matriz de confusión de la Figura 5.8. Este modelo presenta una exhaustividad muy baja para las noticias verdaderas, tan solo un 59 % de ellas son clasificadas correctamente. Es por este hecho por lo que el modelo presenta un valor de F1 de 0.74 y una exactitud de 0.743.



	Fake	Real
Precisión	0.69	0.85
Exhaustividad	0.89	0.59
F1	0.78	0.7
Media F1	0.74	
Exactitud	0.743	

Figura 5.8: Métricas de evaluación modelo híbrido HY4

Capítulo 6

Discusión

En esta sección se presenta una discusión sobre los resultados obtenidos. Se analizará la consecución de los objetivos y el cumplimiento de las hipótesis del Capítulo 1.

6.1. Análisis de los resultados

En las hipótesis H1 y H2 del Capítulo 1 se afirmaba que era posible detectar la veracidad de las noticias empleando la información textual y social y que además, el uso de la información social aumentaba el rendimiento de los modelos.

A la vista de los resultados descritos en la sección 5.3 se puede observar que el enfoque que más exactitud obtiene es un modelo que emplea únicamente información textual, más concretamente un Random Forest con un pesado tf-idf. Este enfoque obtiene una mayor exactitud frente a otro tipo de modelos que incluyen información social, por lo que a priori se podría pensar que la información social no aporta una información relevante.

Sin embargo, podemos apreciar como sobre el conjunto de test es el método que obtiene los peores resultados comparándolos con aquellos que sí emplean la información social. Esto es debido a al emplear un pesado tf-idf es posible que existan palabras en el corpus sobre el cual se le aplica el pesado (corpus de las noticias de entrenamiento) y que no existan sobre el conjunto de test. Es por ello por lo que modelos como las redes transformers preentrenadas sobre grandes corpus tendrán más capacidad de generalización y, por tanto, serán capaces de obtener mejores resultados.

En la Tabla 6.1 se puede ver la comparativa del número de noticias mal

clasificadas con el modelo de pesado TF-IDF y el modelo HY3 en función de su temática. Entre paréntesis se indica el porcentaje de noticias mal clasificadas de esa temática.

Tema	Modelo pesado TF-IDF	Modelo HY3
Ambiental	2 (50 %)	1 (25 %)
Ciencia	4 (31 %)	3 (23 %)
Covid-19	64 (27 %)	56 (24 %)
Internacional	2 (14 %)	2 (14 %)
Política	28 (26 %)	30 (28 %)
Sociedad	49 (47 %)	42 (40 %)
Deporte	2 (100 %)	0 (0 %)

Tabla 6.1: Número de noticias mal clasificadas en función de su temática

A pesar de que las noticias con temática Covid-19 representan un 41 % del conjunto de test y esta temática nunca se ha visto en el conjunto de entrenamiento, la diferencia de los dos modelos es bastante pequeña.

Una vez introducida la información social a los modelos puede verse un aumento significativo de los resultados. Esto es debido a que por una parte el texto se está procesando empleando modelos transformers con una capacidad de generalización muy grande y que la información social no textual extraída de Twitter es la misma independientemente de la temática.

Los modelos que mejores resultados han obtenido han sido los modelos HY1 y HY3. Ambos constan de un Random Forest especializado en la clasificación de noticias empleando su información social. El modelo HY1 empleaba la información textual de la noticia y el modelo HY3 empleaba la información textual de la noticia y de los tuits recopilados. La exactitud de los dos modelos en la validación cruzada ha sido la misma y en el conjunto de test es que el modelo HY3 ha clasificado bien 2 noticias más que el modelo HY1.

En la Figura 6.1 se puede observar el porcentaje de importancia de las diez características más relevantes empleadas en la regresión logística del modelo HY3. Para calcular la importancia de cada característica, f_i, se han extraído los coeficientes de la regresión, w_i, y se ha realizado la operación

$$f_i = e^{w_i}$$

Finalmente se ha calculado el porcentaje de cada una de ellas. Como se observa, la característica más relevante para el modelo ha sido la variable 1,

que corresponde con la probabilidad devuelta por el Random Forest de que una noticia sea verdadera empleando la información social de la noticia.

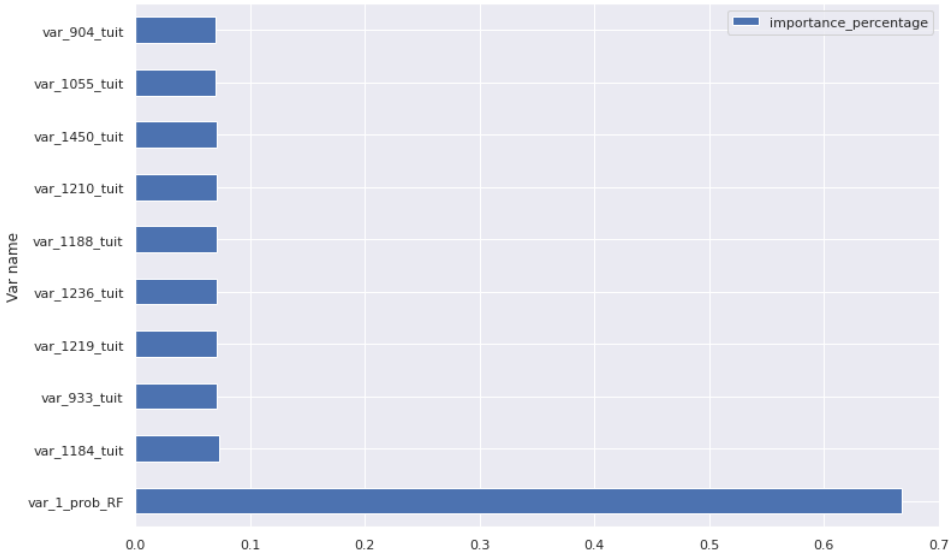


Figura 6.1: Importancias de características de regresión logística en HY3.

Aunque el añadir la información social es un factor muy relevante para el rendimiento del modelo no es el único factor determinante. De las 134 noticias mal clasificadas por el modelo HY3, 75 de ellas sí que tenían información social mientras que 59 no.

Como puede verse en la Tabla 6.1, el tema que peor clasifican los modelos es el tema Sociedad. A continuación, se expondrá dos de las noticias mal clasificadas de dicho tema por el modelo HY3.

Noticia verdadera etiquetada como falsa

La primera fase del censo será virtual y solo hasta abril próximo iniciarán las encuestas puerta a puerta. Este martes 9 de enero iniciará la primera etapa correspondiente a la recolección de información del Censo Nacional de Población y Vivienda de 2018. La inversión para la realización del Censo es de 350.000 millones de pesos, con vigencias futuras del 2016 y 2017, informó este lunes festivo Caracol Radio...

Noticia falsa etiquetada como verdadera

> Mientras la Minga se desplazó hacia la capital de la República las fuerzas de seguridad del Estado han aprovechado para ingresar a los territorios Cauca de dominio indígena para destruir más de 63 laboratorios de procesamiento de coca protegidos por los indígenas y pertenecientes a los grupos armados ilegales que se encuentran en la zona del Cauca y Caquetá. Según lo reveló la Dirección Antinarcóticos de la Policía, los laboratorios intervenidos se encontraron en los municipios de Piamonte (Cauca), San José del Fragua, Puerto Rico, Valparaíso y Milán (Caquetá), zona limítrofe de los dos departamentos y de dominio indígena. La operación del grupo antinarcóticos se denominó 'Resplandor II'Y con todo con el apoyo aéreo de comandos operativos desplegados desde Florencia por parte de la aviación de la Policía Nacional...

Como puede apreciarse, incluso para un humano, es posible no discernir la veracidad de las noticias.

6.2. Análisis de las características sociales

Las hipótesis H3 afirmaba que las características sociales más importantes para predecir la veracidad de las noticias no dependen tanto del contenido del tuit (texto, numero de me gusta, numero de retuits, ...) si no de la información del usuario.

La Figura 5.4 describe la importancia de las características sociales empleando la importancia de la permutación. Se puede observar como 8 de las 9 características más relevantes solamente dependen de la información del autor y no del contenido o de la información del tuit.

Además, dentro de estas características destacan la información que aportan aquellas obtenidas a partir de la desviación típica del conjunto de tuits recopilados para cada noticia.

6.3. Comparación de resultados con IberLEF 2021

Como se ha expuesto en la sección 2.3, este mismo conjunto de noticias fue usado como tarea en el congreso IberLEF 2021. Comparando los resultados del mejor modelo propuesto con los modelos expuestos en la Figura 2.4, el equipo que mejor rendimiento obtuvo, consiguió una precisión de 0.7657,

la misma precisión que el modelo HY3. Cabe señalar que con cualquier modelo híbrido se habría obtenido una puntuación suficiente estar en el top 5 de resultados de la tarea. Además, si en lugar de seleccionar el modelo HY3 se hubiera seleccionado el modelo HY2 se habría estado en segunda posición.

Estos resultados se han obtenido agregando la información social de las noticias y ensamblando distintos modelos. Esto contrasta con la complejidad de la propuesta del equipo con mayor puntuación en la competición.

Capítulo 7

Conclusiones y trabajo futuro

En este capítulo se exponen las diferentes conclusiones extraídas del trabajo realizado, y se propone algunas líneas de trabajo futuro.

7.1. Conclusiones

A lo largo del desarrollo del trabajo se ha observado como la introducción de información social ha permitido, junto con la información textual a clasificar las noticias ayudando a la mejora del rendimiento de los modelos (Objetivo *O1*). Es por ello por lo que a la hora de resolver un problema sería útil añadir la información social al conjunto de datos. Sin embargo, la obtención de esta información es bastante costosa tanto a nivel económico como a nivel de tiempo puesto que tienes que tener acceso a la API de la red social en cuestión.

Además, se ha estudiado la importancia de las características sociales en los modelos clasificadores, concluyendo que aquellas características relacionadas con el autor adquieren un mayor peso que las relacionadas con el tuit (Objetivo *O2*).

También se ha desarrollado un modelo que junte todas las características textuales y sociales para compararlo con el rendimiento de los modelos de la tarea del congreso IberLEF 2021 (Objetivo *O3*).

7.2. Trabajo futuro

Como línea de trabajo futuro sería interesante seleccionar alguno de los modelos que mejores resultados obtuvieron en la tarea IberLEF 2021 y añadirle la información social, con el objetivo de aumentar el rendimiento en esta tarea. Además se podría haber empleado el conjunto de datos FakeNewsNet expuesto en la sección 2 para entrenar con un mayor conjunto de información social a los clasificadores de la sección 4.2 que empleaban dicha información.

También sería un buen enfoque no solo estudiar los metadatos sociales individuales de cada usuario, como el número de seguidores y de seguidos, sino que se podría estudiar un grafo social de los seguidores o seguidos para ver las relaciones sociales que existen entre ellos.

Además, en el caso de tener un mayor número de datos se podría ordenar cronológicamente las noticias y puntuar a los usuarios en función del número de veces que ha compartido anteriormente noticias falsas. De esta forma, para una nueva noticia podríamos saber si ese usuario es propenso a compartir noticias falsas.

Bibliografía

Bibliografía

Albahar, Marwan. 2021. A hybrid model for fake news detection: Leveraging news content and user comments in fake news. *IET Information Security*, 15(2):169–177.

Altmann, André, Laura Toloşi, Oliver Sander, y Thomas Lengauer. 2010. Permutation importance: a corrected feature importance measure. *Bioinformatics*, 26(10):1340–1347.

Barbieri, Francesco, Luis Espinosa-Anke, y Jose Camacho-Collados. 2022. Xlm-t: Multilingual language models in twitter for sentiment analysis and beyond. *Proceedings of the LREC, Marseille, France*, páginas 20–25.

Bharadwaj, Pranav y Zongru Shao. 2019. Fake news detection with semantic features and text mining. *International Journal on Natural Language Computing (IJNLC) Vol*, 8.

Buitinck, Lars, Gilles Louppe, Mathieu Blondel, Fabian Pedregosa, Andreas Mueller, Olivier Grisel, Vlad Niculae, Peter Prettenhofer, Alexandre Gramfort, Jaques Grobler, Robert Layton, Jake VanderPlas, Arnaud Joly, Brian Holt, y Gaël Varoquaux. 2013. API design for machine learning software: experiences from the scikit-learn project. En *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*, páginas 108–122.

Buntain, Cody y Jennifer Golbeck. 2017. Automatically identifying fake news in popular twitter threads. En *2017 IEEE International Conference on Smart Cloud (SmartCloud)*, páginas 208–215. IEEE.

Canete, José, Gabriel Chaperon, Rodrigo Fuentes, Jou-Hui Ho, Hojin Kang, y Jorge Pérez. 2020. Spanish pre-trained bert model and evaluation data. *Pml4dc at iclr*, 2020:1–10.

Chollet, François y others. 2015. Keras. https://keras.io.

Conroy, Nadia K, Victoria L Rubin, y Yimin Chen. 2015. Automatic deception detection: Methods for finding fake news. *Proceedings of the association for information science and technology*, 52(1):1–4.

Devlin, Jacob, Ming-Wei Chang, Kenton Lee, y Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Feurer, Matthias, Katharina Eggensperger, Stefan Falkner, Marius Lindauer, y Frank Hutter. 2020. Auto-sklearn 2.0: Hands-free automl via meta-learning.

Flores, Jose Enrique Palomeque. 2022. Twitter fusiona sus equipos de lucha contra el spam y prevención de la desinformación, Aug.

Hölig, Sascha y Uwe Hasebrink. 2018. Reuters institute digital news report 2019. *Ergebnisse für Deutschland. Arbeitspapiere des Hans-Bredow-Instituts*, 44.

Horne, Benjamin D y Sibel Adali. 2017. This just in: Fake news packs a lot in title, uses simpler, repetitive content in text body, more similar to satire than real news. En *Eleventh international AAAI conference on web and social media*.

Kaliyar, Rohit Kumar, Anurag Goswami, y Pratik Narang. 2021. Fakebert: Fake news detection in social media with a bert-based deep learning approach. *Multimedia tools and applications*, 80(8):11765–11788.

Le, Quoc y Tomas Mikolov. 2014. Distributed representations of sentences and documents. En *International conference on machine learning*, páginas 1188–1196. PMLR.

McCarthy, Niall. 2018. Where exposure to fake news is highest [infographic], Jun.

Posadas-Durán, Juan-Pablo, Helena Gómez-Adorno, Grigori Sidorov, y Jesús Jaime Moreno Escobar. 2019. Detection of fake news in a new corpus for the spanish language. *Journal of Intelligent & Fuzzy Systems*, 36(5):4869–4876.

Potthast, Martin, Johannes Kiesel, Kevin Reinartz, Janek Bevendorff, y Benno Stein. 2018. A Stylometric Inquiry into Hyperpartisan and Fake News. En Iryna Gurevych y Yusuke Miyao, editores, *56th Annual Meeting of the Association for Computational Linguistics (ACL 2018)*, páginas 231–240. Association for Computational Linguistics, Julio.

Roesslein, Joshua. 2020. Tweepy: Twitter for python! *URL: https://github.com/tweepy/tweepy*.

Ruchansky, Natali, Sungyong Seo, y Yan Liu. 2017. Csi: A hybrid deep model for fake news detection. En *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, páginas 797–806.

Shu, Kai, Deepak Mahudeswaran, Suhang Wang, Dongwon Lee, y Huan Liu. 2018. Fakenewsnet: A data repository with news content, social context and spatialtemporal information for studying fake news on social media. *arXiv preprint arXiv:1809.01286*.

Shu, Kai, Suhang Wang, y Huan Liu. 2017. Exploiting tri-relationship for fake news detection. *arXiv preprint arXiv:1712.07709*, 8.

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, y Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Vosoughi, Soroush, Deb Roy, y Sinan Aral. 2018. The spread of true and false news online. *Science*, 359(6380):1146–1151.

Wolf, Thomas, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, y others. 2020. Transformers: State-of-the-art natural language processing. En *Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations*, páginas 38–45.

Zhang, Jiawei, Limeng Cui, Yanjie Fu, y Fisher B Gouza. 2018. Fake news detection with deep diffusive network model. *arXiv preprint arXiv:1805.08751*.

Apéndice A

Estructura Archivo .zip

La estructura del archivo .zip subido es la siguiente.

- `Programas`
  - `API_Twitter`
    - `Tweets_Automaticos.ipynb`
      (Programa extracción tuits empleando la API de Twitter)
  - `dataset`
    - `tuits_corregidos`
    - `dataset_social_info_v3.csv`
    - `dataset_social_information_included.pkl`
      (Dataset completado con la información social extraída)
    - `dev.csv`
    - `test.csv`
    - `train.csv`
    - `development.xlsx`
    - `test.xlsx`
    - `train.xlsx`
  - `Modelos_hibridos`
    - `HY1_AutoML.ipynb`
      (Modelos entrenados con el paquete Auto-Sklearn para el enfoque HY1)
    - `HY2_AutoML.ipynb`
    - `HY3_AutoML.ipynb`

  - `HY4_AutoML.ipynb`
  - `HY1_Matriz_Caracteristicas.ipynb`
    (Creación de la matriz de características del modelo HY1)
  - `HY2_Matriz_Caracteristicas.ipynb`
  - `HY3_Matriz_Caracteristicas.ipynb`
  - `HY1_ML.ipynb`
    (Modelos entrenados manualmente para el enfoque HY1)
  - `HY2_ML.ipynb`
  - `HY3_ML.ipynb`
  - `HY4_ML.ipynb`
  - `matrix_features_variance.csv`
  - `pipeline_matriz_texto.pkl`
    (Pipeline con los modelos entrenados con Auto-Sklearn para el enfoque HY2)
  - `pipeline_matriz_tuits_texto.pkl`
    (Pipeline con los modelos entrenados con Auto-Sklearn para el enfoque HY4)
  - `pipeline_RF_noticia.pkl`
    (Pipeline con los modelos entrenados con Auto-Sklearn para el enfoque HY1)
  - `pipeline_RF_Texto_Tuits.pkl`
    (Pipeline con los modelos entrenados con Auto-Sklearn para el enfoque HY3)
  - `RF_Bert_Features_Matrix.csv`
    (Matriz de características del modelo HY1)
  - `RF_Tuits_Text_Features_Matrix.csv`
    (Matriz de características del modelo HY3)
  - `Varianza_Bert_matrix_features.csv`
    (Matriz de características del modelo HY2)

- `Modelos_informacion_social`
  - `AutoML_social_info.ipynb`
    (Modelos entrenados con el paquete Auto-Sklearn para los métodos de información social)
  - `Modelos_ML_si.ipynb`

(Modelos entrenados manualmente para los métodos de información social)

- `pipeline_social_info.pkl`
  (Pipeline con los modelos entrenados con Auto-Sklearn para los métodos de información social)
- `random_forest_model.sav`
  (Modelo con mejor rendimiento empleando la información social)
- `social_info_df_test.pkl`
- `social_info_df_train.pkl`
- `test_matrix_si_features_variance.csv`
  (Matriz de características de información social para el conjunto de test)
- `train_matrix_si_features_variance.csv`

- `Modelos_textuales`
  - `modelos_contextuales.ipynb`
    (Modelo BERT para los métodos textuales)
  - `Modelos_textuales_1.ipynb`
    (Modelos entrenados de los métodos textuales)