

Universidad Complutense de Madrid

Máster Big Data y Data Science



Tesis del Master

Detección de noticias falsas mediante técnicas de
Machine Learning y Deep Learning

Alejandro Sierra Fernández



https://github.com/Spocklight/Fake_News_Detection

A mi familia.

Agradecimientos

Me gustaría mostrar mi más sincero agradecimiento a todos los profesores y profesoras que han compartido su conocimiento a lo largo del Master. Sin su trabajo este trabajo no habría sido posible. Especial mención a Luis Gascó y a Alberto Ezpondaburu, cuyas técnicas y código compartido sobre el preprocesado de la información y sobre modelos neuronales han sido de gran ayuda a lo largo de todo el estudio.

Índice

- Resumen..... pag1
- ISOT Dataset..... pag2
- Preprocesado de datos..... pag7
- Análisis del sentimiento..... pag8
- Topic Modelling..... pag11
- Modelos no neuronales pag13
- Modelos neuronales pag15
- Conclusión pag20

Nota: La información sobre la productivización del modelo se encuentra en el repositorio de *github* anotado en la portada. Esta se ha hecho empleando *MLFlow* y *Ngrok*.

Resumen

En Brasil, Jair Bolsonaro se vio beneficiado por una campaña de desinformación orquestada contra sus adversarios. En la India, una veintena de personas fue asesinada por bulos transmitidos a través de Whatsapp sobre el secuestro de niños. En Myanmar, el ejército desencadenó el caso más reciente de limpieza étnica al difundir mensajes en Facebook en contra de una minoría musulmana. En Estados Unidos, Donald Trump hizo temblar los cimientos democráticos insinuando en redes sociales un supuesto fraude electoral inexistente. En Ucrania, el ejército ruso afirmó que no había cadáveres en las calles de Bucha cuando abandonaron la ciudad.

La manipulación de la información siempre ha existido: desde las profecías en la antigüedad hasta las conspiraciones de toda índole en la época Moderna. Lo que no tiene precedente es la envergadura masiva y extrema que ha adquirido a través de las plataformas sociales, cuyo modelo de negocio se presta a acelerar la difusión de noticias que acaparen atención, que se consuman y se compartan. En definitiva, que generen beneficios.

A todo esto, hay que añadir que los individuos tienden a aferrarse a sus opiniones, aun sabiendo que no son ciertas, especialmente en el ámbito de la política. Este año, una investigación del MIT publicada en la revista *Science*, afirmó que en Twitter las noticias y rumores de contenido engañoso tienen un 70% más de posibilidades de ser retuiteadas que las imparciales. En las redes sociales la mentira juega con ventaja frente a las afirmaciones verdaderas porque a menudo reafirma aquello en lo que creemos o deseamos creer.

Sin embargo, no todo está perdido. Cada día surgen más personas y plataformas dedicadas a la búsqueda de noticias falsas y campañas de desinformación que luchan por defender una verdad que a veces se vuelve platónica. Estas personas, por suerte, cuentan con una herramienta con la que Platón no contaba: El Big Data y Machine Learning. El enorme volumen de noticias falsas y su gran viralidad exigen la automatización de procesos que validen las noticias que son ciertas y que invaliden y refuten las falsas, haciendo posible soñar con un mundo en el que todavía exista diferencia entre la verdad y la mentira.

En esta tesis trataremos de identificar las diferencias lingüísticas entre los artículos, previamente clasificados como verdaderos o falsos, que conforman la base datos ISOT. Para ello tendremos que preprocesar la información, generar variables que puedan ayudar a identificar a qué clase pertenecen nuestros artículos y definir y entrenar modelos de Machine Learning y de Deep Learning que los clasifiquen.

Área de Estudio: Machine Learning, Deep Learning, Data Mining, Data Visualization

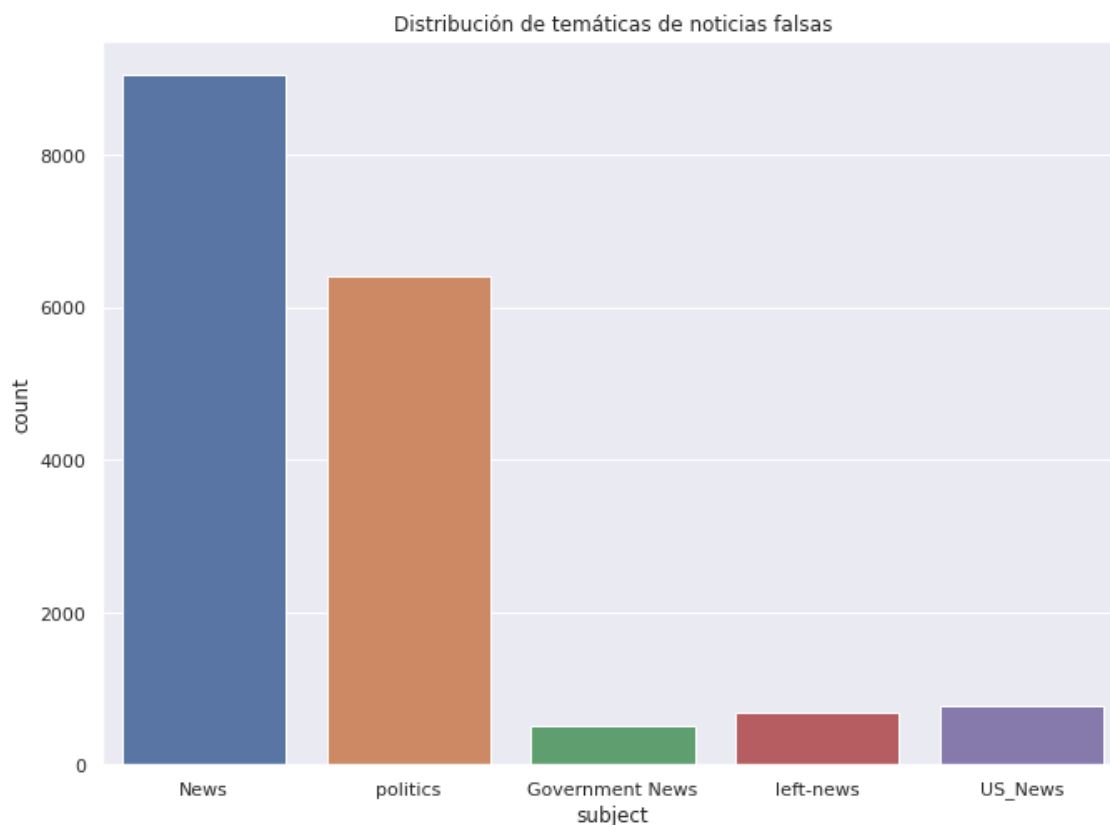
Keywords: Fake news detection, Natural language processing, Classification Models

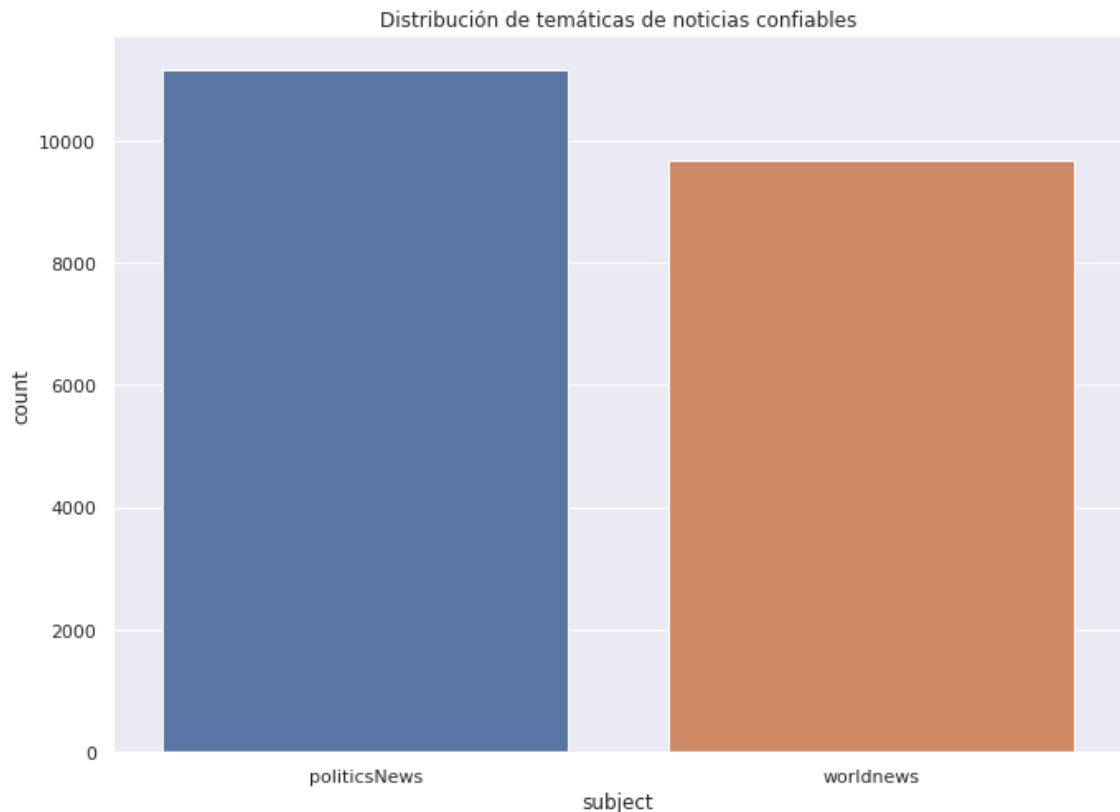
ISOT DATASET

A la hora de desarrollar un proyecto de Machine Learning, es de una importancia crucial trabajar con bases de datos bien estructuradas y que sean muestras lo suficientemente representativas de las clases que representan. De no ser así, nuestros modelos estarían limitados a predecir únicamente en el espacio de la base de datos. Estas son características con las que cuenta el *Information Security and Object Technology (ISOT) Dataset*. En esta sección se resume el contenido y estructura de la base de datos, mostrando gráficas que ayudan a describirla visualmente. Los artículos falsos están etiquetados como 1 y los confiables como 0.

2.1- Descripción

El ISOT Dataset reúne más de 45000 artículos falsos y confiables, obtenidos por sitios webs y periódicos legítimos y por sitios webs y fuentes señaladas como no fiables por la organización Polifact y Wikipedia. La información que recoge, por tanto, es real y pertenece principalmente a los años 2016 y 2017. Esta trata diferentes temas, aunque la gran mayoría podríamos clasificarlos como artículos políticos procedente de los Estados Unidos y que analizan tanto situaciones del propio país como otros sucesos a nivel mundial.





Como vemos en las gráficas superiores, la clasificación que se hace para las noticias falsas y para las confiables es diferente. Esto supone un problema para los modelos que definiremos más adelante porque no podrán emplear el tema que tratan a la hora de predecir si estos no coinciden para los dos tipos de noticia. Solucionaremos este aspecto más adelante empleando la técnica conocida como *Topic Modelling*.

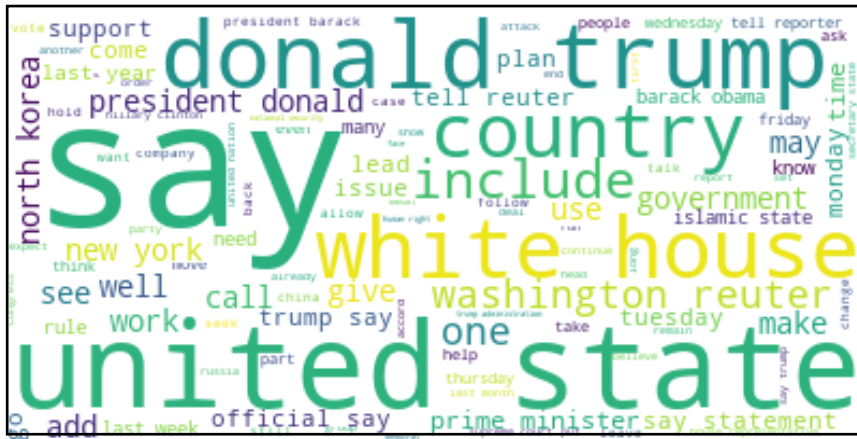
Cada noticia, además, está caracterizada por cuatro variables. Dicho de otro modo, nuestro dataframe dispone de cuatro columnas que incluyen el título del artículo, el cuerpo del artículo, el tema que trata y la fecha en la que fue publicado.

2.2- Visualización de la información

- **Wordclouds:**



Wordcloud perteneciente a los artículos falsos

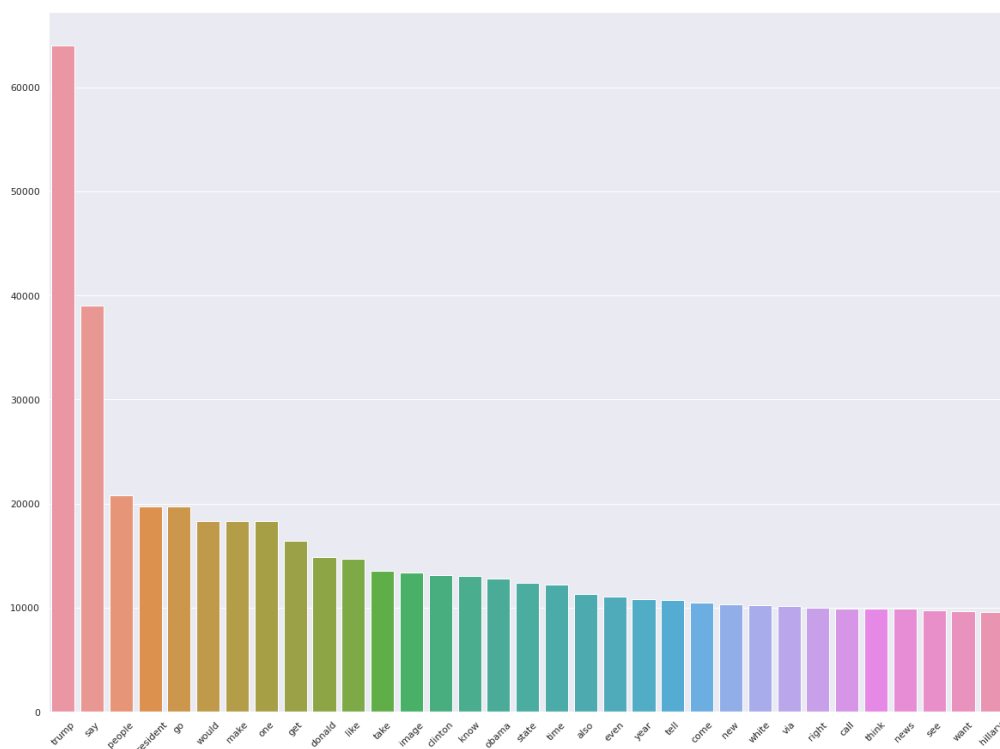


Wordcloud perteneciente a artículos confiables

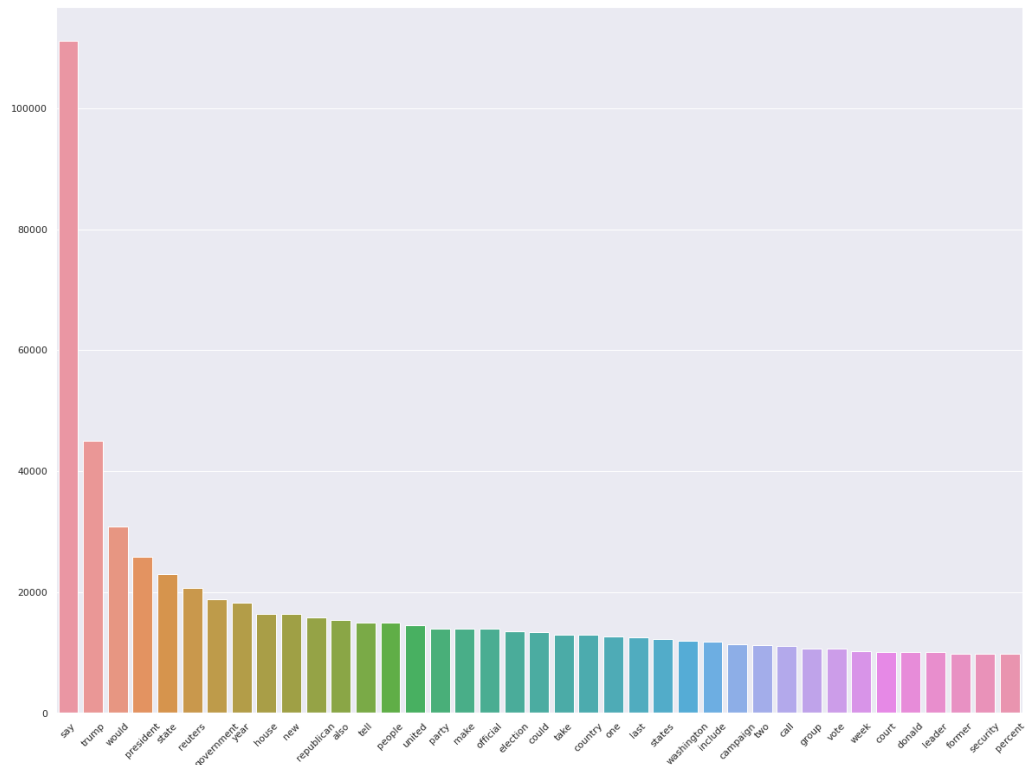
Los Wordclouds fueron generados de forma independiente para las noticias confiables y las noticias falsas tras haber preprocesado la información con el objetivo de poder distinguir a golpe de vista alguna diferencia intuitiva entre ambas clases. Podemos destacar que la frecuencia de los términos más comunes en artículos falsos es más constante que en artículos confiables, en los que hay unas pocas palabras que destacan sobre el resto. También llama la atención como en artículos confiables aparece como destacado el término ‘reuter’, que hace referencia al sitio web de noticias. Podemos intuir la relevancia que tiene la procedencia del artículo o que en él la información se contraste con otras fuentes confiables. Por último tenemos que mencionar la relevancia que tienen términos como ‘make’, ‘fact’, ‘believe’, ‘think’ o ‘show’ en artículos falsos, así como la presencia de un mayor número de alusiones a nombres propios como Clinton u Obama.

- **Histogramas:**

La información anterior puede ser representada de un modo más cuantitativo a través de histogramas.



Histograma de términos más frecuentes en noticias falsas.



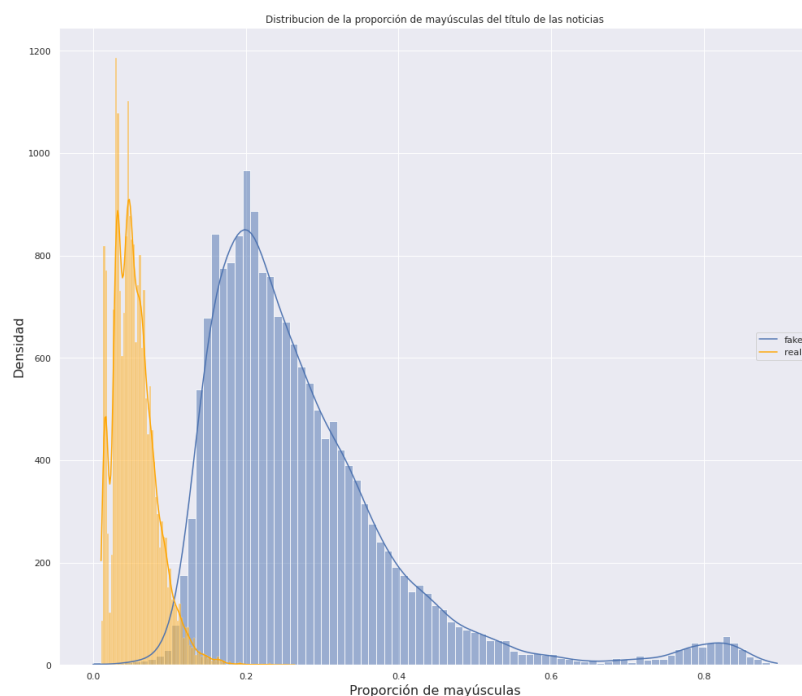
Histograma de términos más frecuentes en noticias confiables.

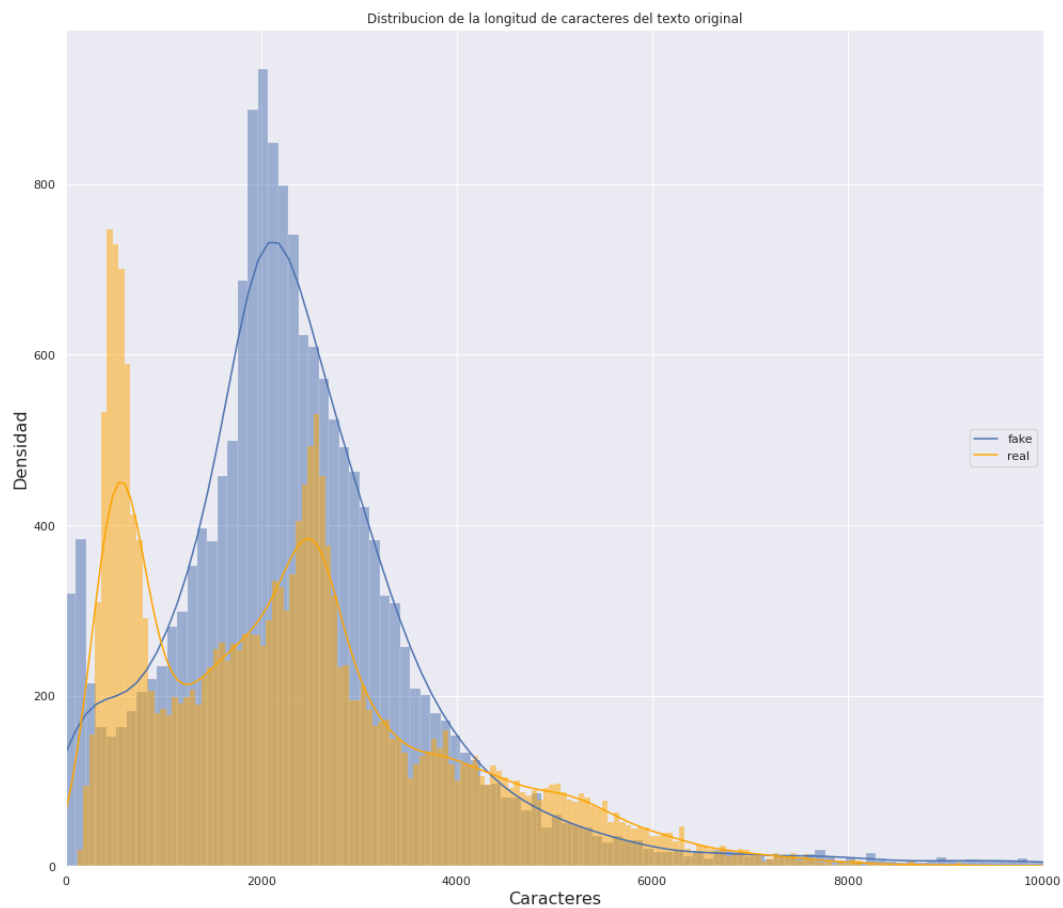
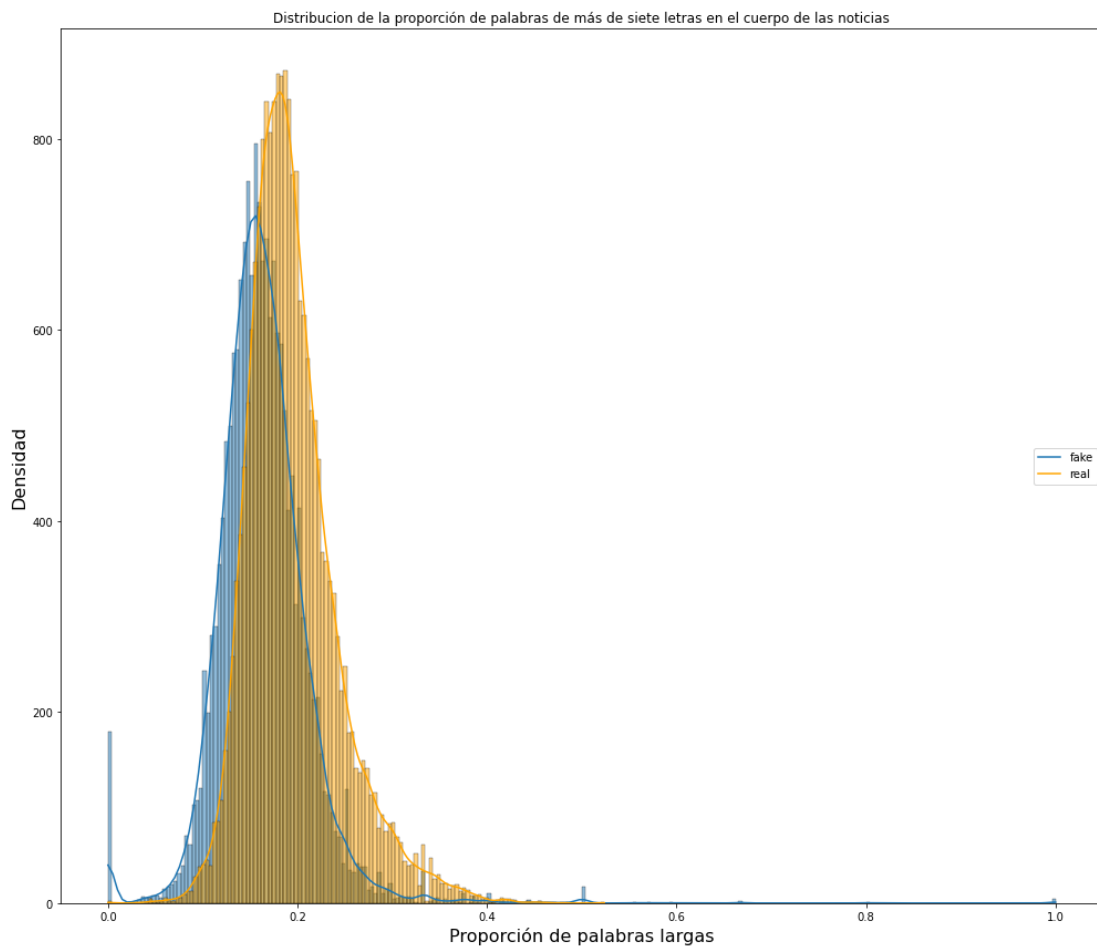
- **Ingeniería de variables:**

La ingeniería de variables la podemos definir como el proceso de usar el conocimiento que se tienen sobre los datos para crear variables que pueden mejorar el desempeño de los algoritmos de *machine learning*, resultando un paso muy importante a la hora de mejorar la calidad y la precisión de las predicciones.

En este caso algunas de las que hemos decidido añadir son la longitud de caracteres, el número de términos que permanecen tras el procesado de la información (tokens), la proporción de mayúsculas o la proporción de palabras compuestas por siete o más letras. Además, estas funciones las aplicamos tanto al cuerpo de las noticias como a los títulos y tanto a las noticias y títulos originales como a los que surgen después del procesamiento de estos.

Veamos las distribuciones de algunas de estas variables:





En la primera gráfica vemos la distribución de las proporciones de mayúsculas en los títulos de las noticias para noticias falsas y confiables, en la segunda la proporción de palabras de más de siete letras en el cuerpo de las noticias y en la tercera la longitud de caracteres del texto original.

Lo que perseguimos en este tipo de análisis gráficos es identificar aquellas variables que vayan a ser de utilidad a los modelos a la hora de predecir. Para ello buscamos distribuciones que sean lo más diferente posible para las noticias confiables y las falsas, ya que estas diferencias les permitirán emplear la variable como identificador a la hora de clasificar los datos. Es por esto por lo que sabemos de antemano que la proporción de mayúsculas en el título de las noticias será una variable de gran relevancia, puesto que permite distinguir de forma bastante limpia ambas clases. En este caso, distribución de las noticias falsas se encuentra desplazada a la derecha, presentando una proporción de mayúsculas en sus títulos superior a la de las noticias confiables.

Preprocesado de datos

Desde los datos originales que contienen las noticias hasta la información que los modelos recogen como *input* existe un proceso de transformación que, a pesar de no ser la parte más vistosa de un proyecto, es la que más tiempo consume. Además, realizar un buen preprocesado es fundamental para mejorar la calidad de las predicciones de los modelos, siendo igual o más importante que el propio ajuste de parámetros de los algoritmos.

Para realizar estas transformaciones hemos empleado un *pipeline*, término que en informática se refiere a la cadena de procesos conectados de tal forma que la salida de cada elemento es la entrada de la próxima, permitiendo la comunicación y sincronización entre procesos. Este pipeline, por tanto, está compuesto por varias etapas en las que se van realizando las transformaciones de las que mencionábamos anteriormente. Estas etapas son:

- **Borrado de espacios extras**
- **Conversión de mayúsculas a minúsculas**
- **Borrado de comillas**
- **Borrado de caracteres no-ascii**
- **Eliminar las contracciones del inglés** (e.g: Don't ----> Do not)
- **Tokenización**
- **Borrado de stopwords**
- **Borrado de signos de puntuación**
- **Lematización**

Todos estos son elementos comunes y frecuentes de la rama que se conoce como ‘procesamiento del lenguaje natural’ (NLP) y que se ocupa de cómo programar los ordenadores para procesar y analizar grandes cantidades de datos del lenguaje natural o humano. Entre ellos, los menos intuitivos quizás sean la tokenización, el borrado de *stopwords* y la lematización. La Tokenización consiste en dividir los textos o cadenas de *strings* en elementos individuales como pueden ser oraciones o en palabras y signos de puntuación. Por otra parte, con el borrado de *stopwords* nos referimos al borrado de palabras redundantes o que no transmiten mucha información por la alta frecuencia con la que aparecen y la poca carga de significado que tienen. En este sentido podríamos argumentar, por ejemplo, que los artículos no van a tener a penas peso a la hora de clasificar las noticias en comparación con los adjetivos, por lo que serán tratados como *stopwords* y eliminados. Finalmente, la lematización es un proceso lingüístico que consiste en, dada una forma flexionada (es decir, en plural, en femenino, conjugada, etc.) hallar el lema correspondiente o la forma que por convenio se acepta como representante de todas las formas flexionadas de una misma palabra. De esta forma, el término *going* se transformaría en *go*.

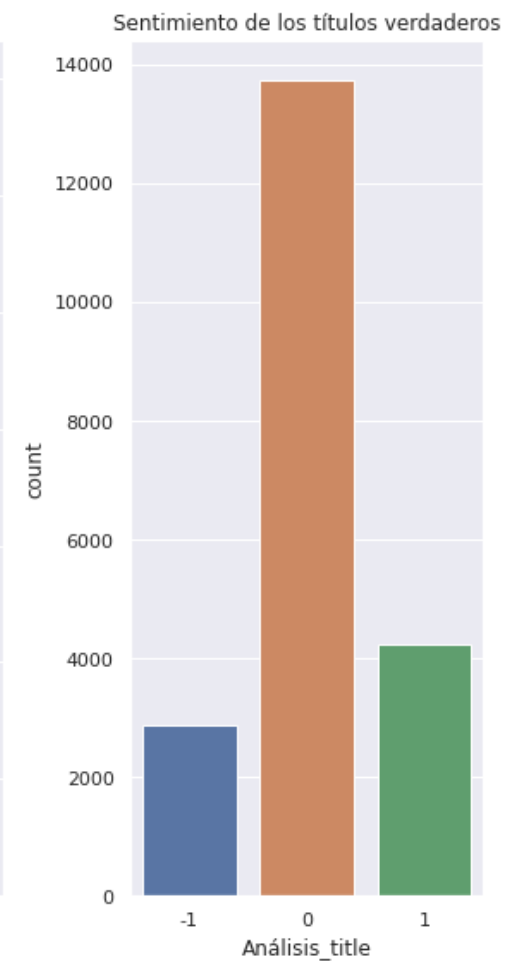
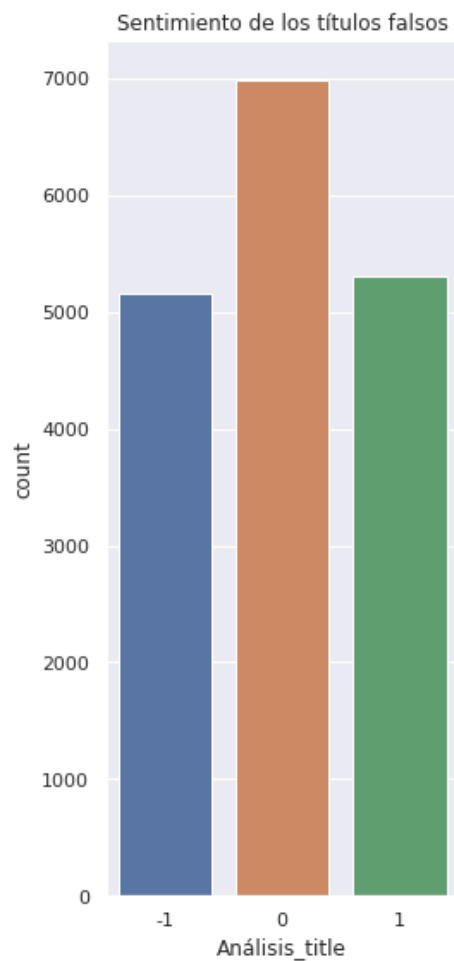
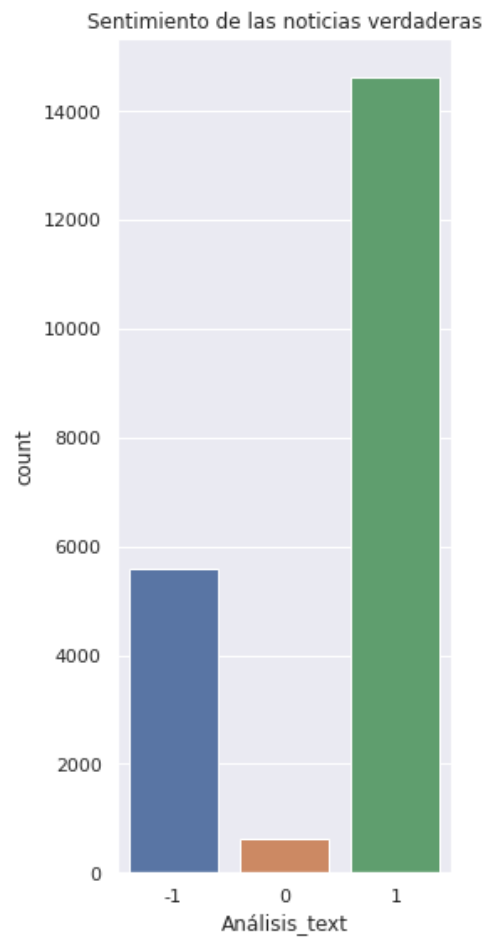
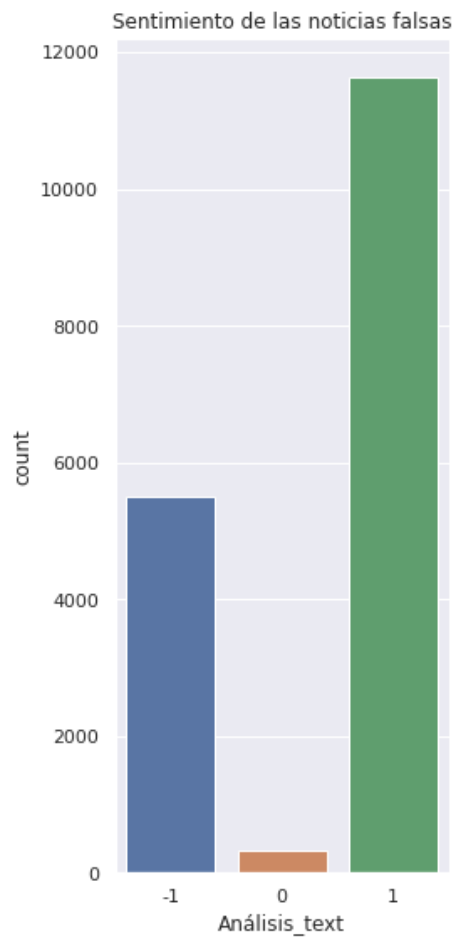
Análisis del sentimiento

El análisis del sentimiento es un subcampo del Text Mining cuyas herramientas se utilizan para detectar, extraer y analizar la opinión expresada por el autor de un texto. En este campo, el concepto de opinión aglutina otros como los sentimientos, valoraciones, actitudes y emociones que un sujeto presenta hacia una persona, objeto, institución o evento.

En este trabajo hemos hecho uso de la librería *TextBlow*, que utiliza un modelo pre-entrenado que nos devolverá dos parámetros al aplicarla sobre un texto, la polaridad y la subjetividad. El primero tiene un dominio $D=\{-1,+1\}$ y mide el grado de positividad y negatividad de un conjunto de palabras. El segundo tiene un dominio $D=\{0,1\}$ y mide el grado de subjetividad, opinión, emoción o juicio de un mensaje. Los valores menores para cada variable se corresponden con los elementos más subjetivos y negativos. Además de estos dos parámetros, hemos decidido introducir una tercera variable binaria que mide si la polaridad es negativa o positiva a la que denominamos ‘análisis’.

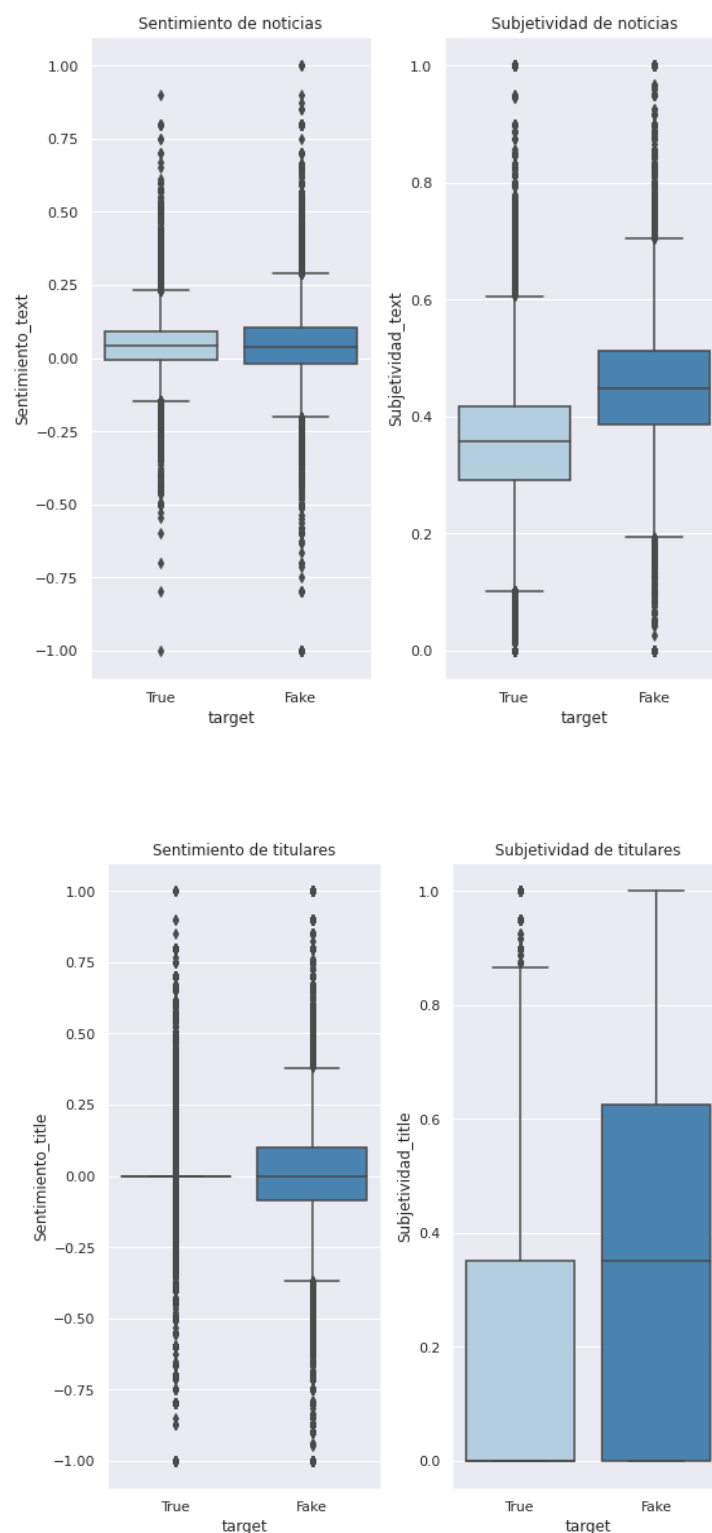
De esta forma obtenemos dos nuevas variables para cada una de nuestras instancias, que serán representadas en nuestro dataframe como dos nuevas columnas. Teniendo esto en consideración, se puede entender el análisis del sentimiento como una etapa más dentro de la ingeniería de variables. Sin embargo, hemos decidido considerarlo de forma separada por emplear otras librerías y modelos pre-entrenados para obtener los resultados.

Nuevamente, buscamos distribuciones lo más diferente posibles entre las variables asociadas a textos verdaderos y falsos. Para la variable ‘análisis’ obtenemos los siguiente *countplots* sobre los titulares y las noticias, gráficos que miden el número de instancias pertenecientes a cada clase:



Como podemos observar, la variable 'análisis' correspondiente a los titulares de las noticias dará un mayor rendimiento que la aplicada a las noticias al mostrar los titulares verdaderos un tono neutro con mucha mayor frecuencia.

Con respecto al sentimiento y a la subjetividad obtenemos el comportamiento inverso, siendo variables que van a resultar más provechosas al aplicarse sobre el cuerpo de las noticias que sobre los titulares. Podemos ver dicho comportamiento en los siguientes *boxplots*:



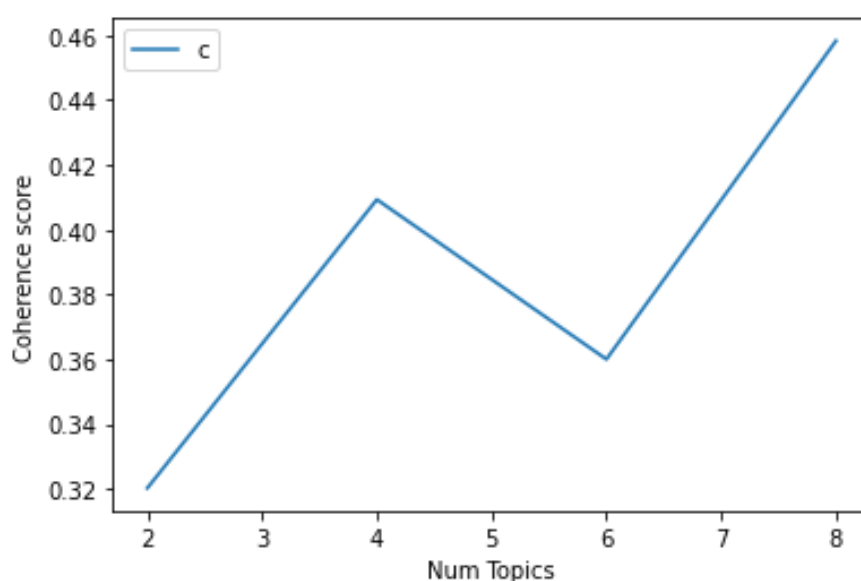
Topic Modelling

En esta fase le pondremos una solución al hecho de no tener una correspondencia entre la variable que define las temáticas de las noticias confiables y las falsas. Lo que haremos será sustituir esta columna por una nueva aplicando la técnica conocida como '*Topic Modelling*'.

Esta herramienta consiste en identificar, sin ayuda de ningún diccionario, los temas principales que encierra un texto a través del conteo de tokens que aparecen en cada texto y las relaciones semánticas que existen entre ellos. Para ello empleamos la librería '*Gensim*' y los modelos LDA (*Latent Dirichlet Allocation*). Los modelos LDA son modelos generativos que permiten que conjuntos de observaciones puedan ser explicados por grupos no observados que explican por qué algunas partes de los datos son similares. Por ejemplo, si las observaciones son palabras en documentos, presupone que cada documento es una mezcla de un pequeño número de categorías y la aparición de cada palabra en un documento se debe a una de las categorías a las que el documento pertenece.

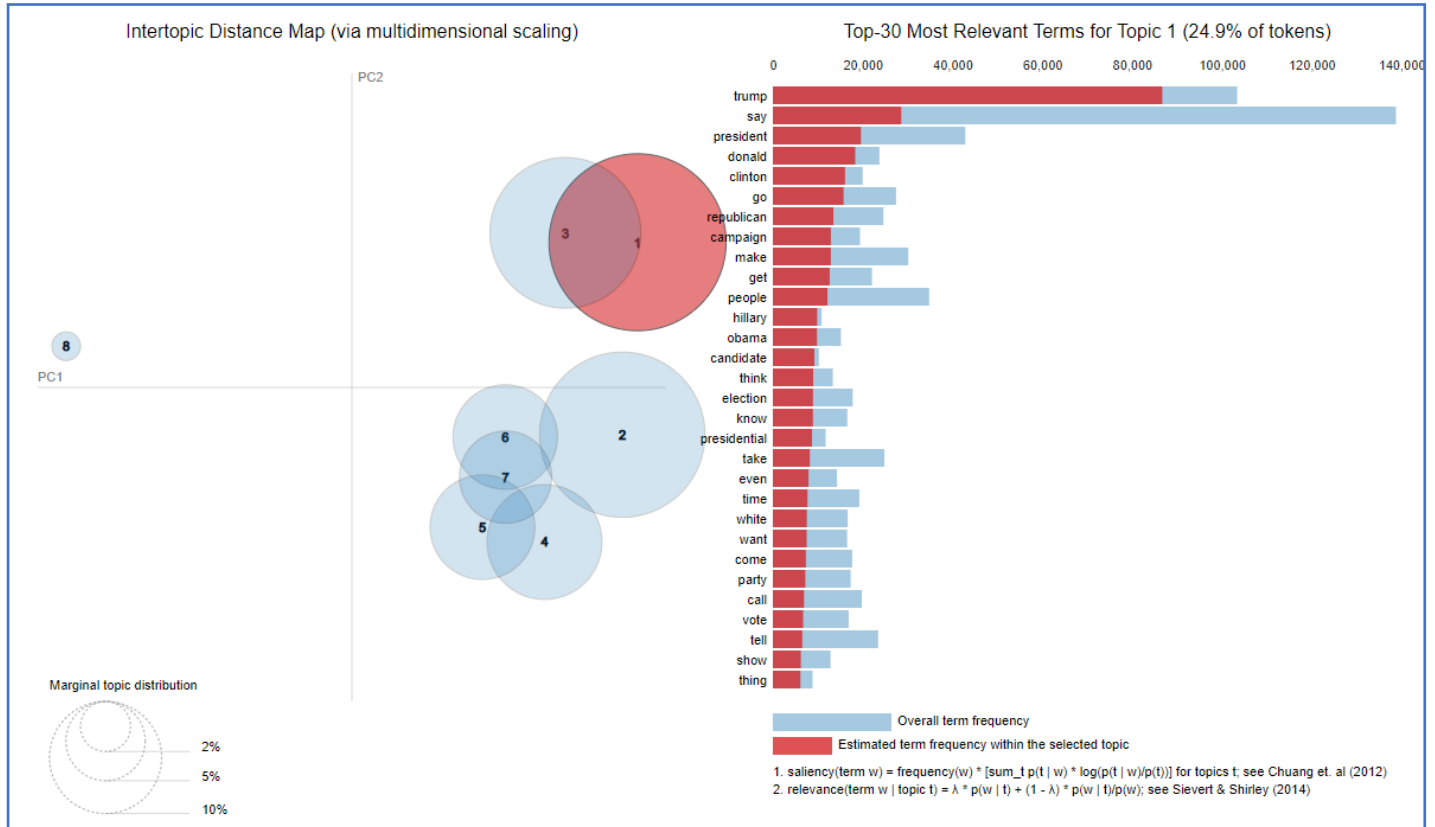
Estos modelos, además de los vectores de entrada, necesitan un número de tópicos en los que clasificar nuestros documentos para ser entrenados. Como a priori no sabemos cuál es el número de temas óptimo en el que clasificar las noticias, lo que hacemos es definir diferentes modelos que dividan los datos en un número diferente de categorías y comparar sus 'coherencias', siendo la coherencia una métrica que nos da una idea de cómo de coherente es un modelo en cuanto a la distribución de sus *topics*: cuanto más distintas sean las palabras de los topics entre sí, menos relacionados estarán los topics y más coherente será el modelo.

En la siguiente gráfica podemos ver para ocho topics la coherencia alcanza un máximo, siendo este un valor óptimo y que vamos a elegir para entrenar nuestro modelo LDA.



Coherencia del modelo LDA en función del número de topics.

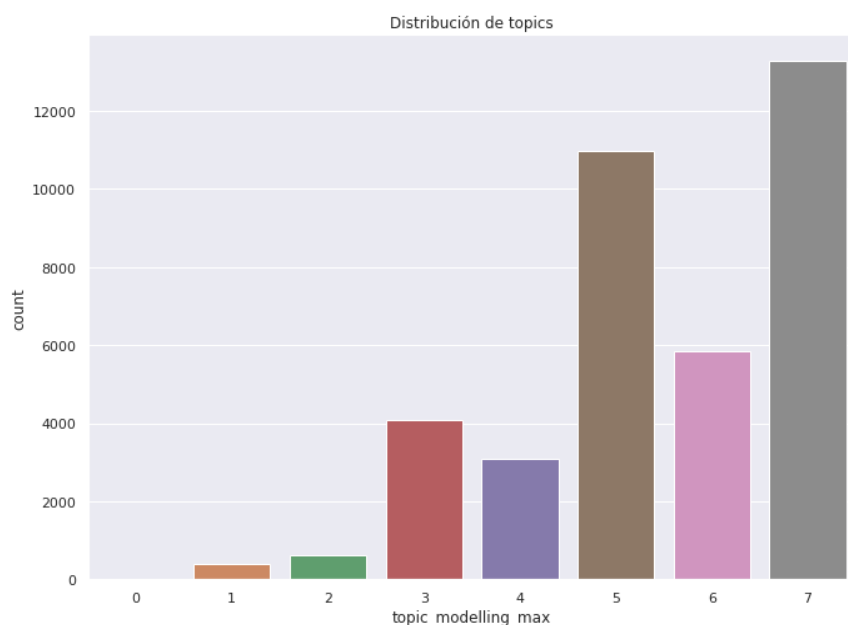
En la siguiente imagen podemos ver los *topics* representados en un espacio vectorial bidimensional dependiendo de las palabras que los representan. Estas palabras, a su vez, tienen una representación n-dimensional y dependiendo de la relación que exista entre ellas tendrán una posición más cercana o lejana en dicho espacio n-dimensional. El diámetro de los círculos que representan los topics depende del número de palabras que lo representan.



Representación de los topics en un espacio bidimensional y de los términos que los definen.

Cabe mencionar que este proceso también podría considerarse como parte de lo que conocemos como 'ingeniería de datos', puesto que estamos añadiendo una nueva variable a nuestro Dataset a partir de otras ya definidas. Sin embargo, por los mismos motivos ya explicados en el análisis del sentimiento, decidimos tratar esta etapa de forma independiente.

Por último, veamos cómo queda la distribución de *topics* en nuestros documentos:



MODELOS NO NEURONALES

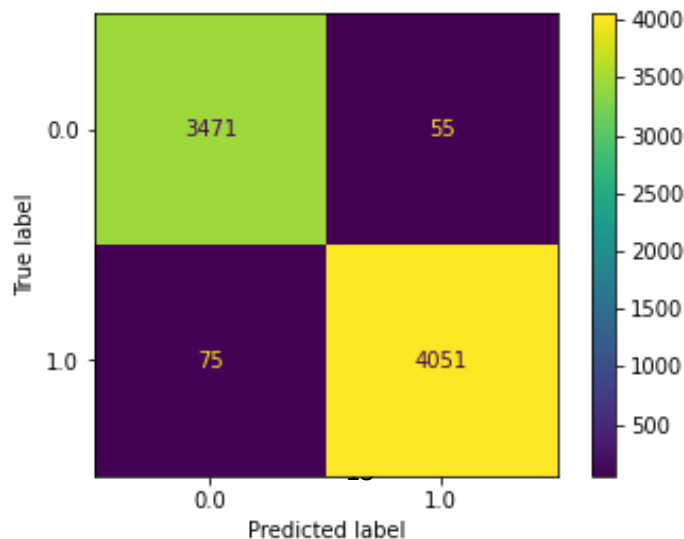
En esta sección aplicaremos modelos no neuronales, entrenándolos sobre nuestros datos y haciendo predicciones. Previamente empleamos algoritmos *TF-IDF Vectorizer* para transformar nuestros documentos, anteriormente procesados, a un formato representativo numérico que sea entendible como *input* por los modelos, quedándonos con los cien términos más relevantes en términos de frecuencia. Junto con estos términos, añadimos como variables de entrada las definidas anteriormente durante el procesamiento de los datos. La explicación teórica del fundamento de cada algoritmo es algo que no trataremos en este documento, que no persigue dicho objetivo.

Comenzamos dividiendo los datos en dos conjuntos diferentes a los que llamamos *train* y *test*, entrenando sobre el *train* los modelos *Support Vector Classifier* (SVC), *Regresión Logística*, *K-nearest-neighbours* (KNN), *Árboles de decisión* (DTC), *Random Forest* y *Naive Bayes* (NB). Este entrenamiento lo hacemos sin hacer un ajuste de parámetros de los modelos para obtener unos resultados orientativos que nos informen de cuáles son los que tienen un mejor rendimiento y un mayor potencial.

Modelos	Accuracy
SVC	0.9956
Random Forest	0.9957
Logistic Regressor	0.9951
KNN	0.9724
NB	0.9744
DTC	0.9904

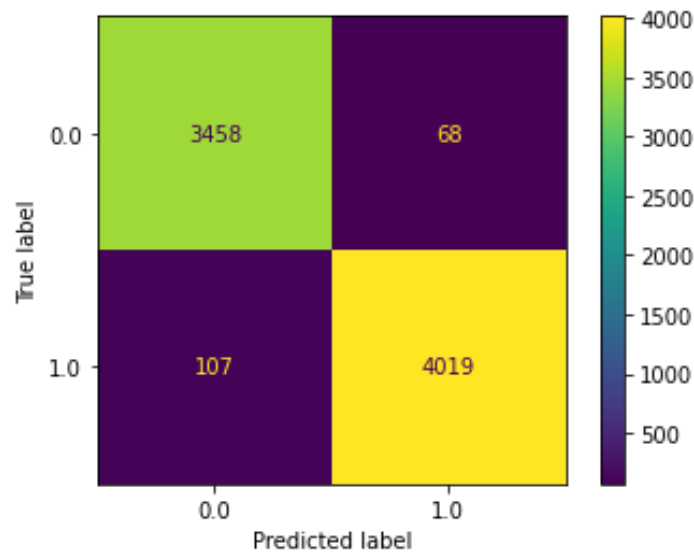
A la vista de estos resultados, decidimos ajustar los hiperparámetros del *Support Vector Classifier* y el *Random Forest* aplicando la técnica *RandomizedSearchCV* que nos proporciona la librería *Sklearn*. Las matrices de confusión tras hacer predicciones sobre el *test* son las siguientes (0 representa las noticias falsas y 1 las confiables):

Mejores parámetros: {'n_estimators': 500, 'max_depth': 20, 'criterion': 'gini'}



Matriz de confusión del Random Forest.

Mejores parámetros: {'C': 1000, 'gamma': 0.001, 'kernel': 'rbf'}



Matriz de confusión del Support Vector Classifier.

Las métricas que emplearemos para analizar los resultados son la precisión, el recall y el f-1 score.

- La precisión es el cociente $\frac{tp}{tp+fp}$, donde tp es el número de positivos verdaderos y fp el número de falsos positivos. La precisión, intuitivamente, es la habilidad del clasificador de no predecir una noticia falsa como confiable.
- El recall es el cociente $\frac{tp}{tp+fn}$, donde tp es el número de positivos verdaderos y fn el número de falsos negativos. El recall, intuitivamente, es la habilidad del clasificador de encontrar todas las noticias confiables
- El f-1 score puede interpretarse una media armónica entre la precisión y el recall, siendo 1 el mejor valor posible y 0 el peor.
-

	Precision	Recall	f-1 score
0	0.98	0.98	0.98
1	0.99	0.98	0.98

Métricas obtenidas por el Random Forest Classifier.

	Precision	Recall	f-1 score
0	0.98	0.98	0.98
1	0.98	0.98	0.98

Métricas obtenidas por el Support Vector Classifier.

MODELOS NEURONALES

Aplicamos también redes neuronales, que suelen mostrar un rendimiento muy bueno tanto en problemas de regresión como de clasificación. En este caso generaremos tres modelos, una red densa, una red recursiva de tipo LSTM (*Long Short-Term Memory*) y una red híbrida formada tanto por capas convolucionales como por capas bidireccionales.

En los tres casos hemos realizado ajustes de hiperparámetros mediante *RandomizedSearchCV* y hemos aplicado medidas para evitar el overfitting de las redes. Entre estas medidas destacamos la utilización de capas *Dropout*, de capas *Pooling*, de *EarlyStopping* y de regularizadores. De nuevo, no vamos a explicar en el fundamento teórico de cómo funcionan las diferentes redes neuronales o de las medidas que luchan contra el overfitting al no ser este el objetivo del documento.

- **Red neuronal Densa Multicapa:**

La red neuronal densa multicapa, generalización del perceptrón, permite detectar relaciones no lineales entre las variables, que es la principal limitación del perceptrón. Dicho de otro modo, permite resolver problemas que no son linealmente separables. Esto la convierte en una solución adecuada a la hora de detectar asociación de patrones ocultos y, por tanto, adecuada para este estudio.

Tras haber ajustado los parámetros de nuestra red (nº de neuronas, nº de capas, medidas contra el overfitting...) la red queda con la siguiente arquitectura:

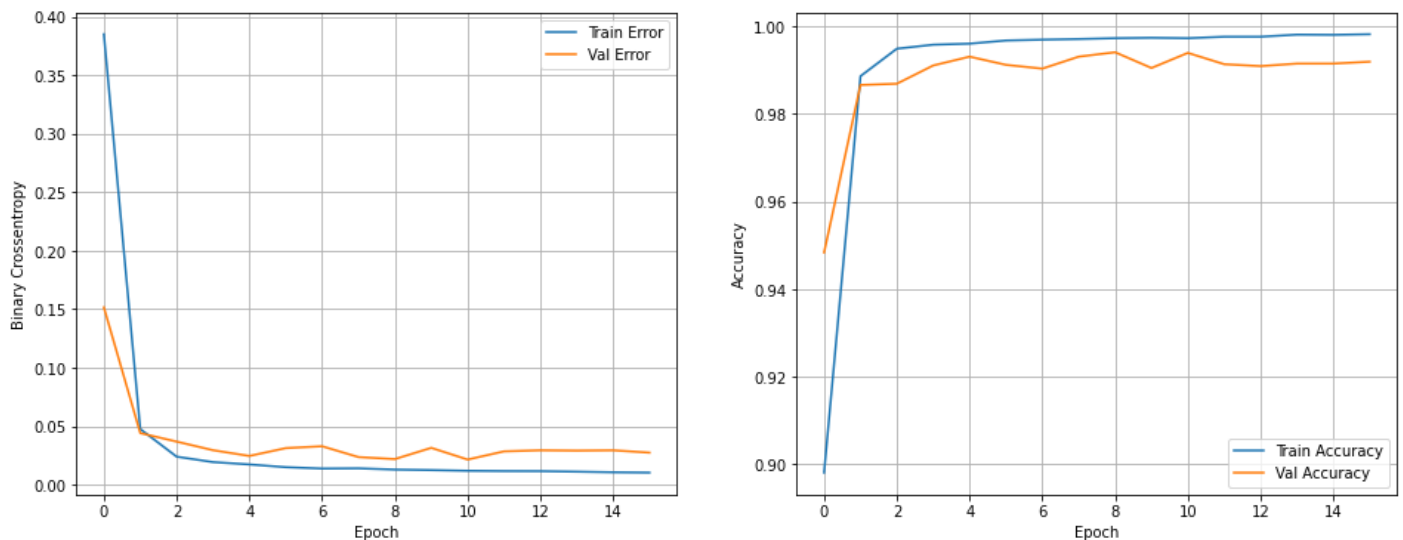
Model: "Dense_Tuning_Model"

Layer (type)	Output Shape	Param #
input_layer (Dense)	(None, 64)	7424
first_dropout (Dropout)	(None, 64)	0
Dense1 (Dense)	(None, 64)	4160
second_dropout (Dropout)	(None, 64)	0
output_layer (Dense)	(None, 1)	65

=====
Total params: 11,649
Trainable params: 11,649
Non-trainable params: 0

Arquitectura de la red neuronal densa multicapa

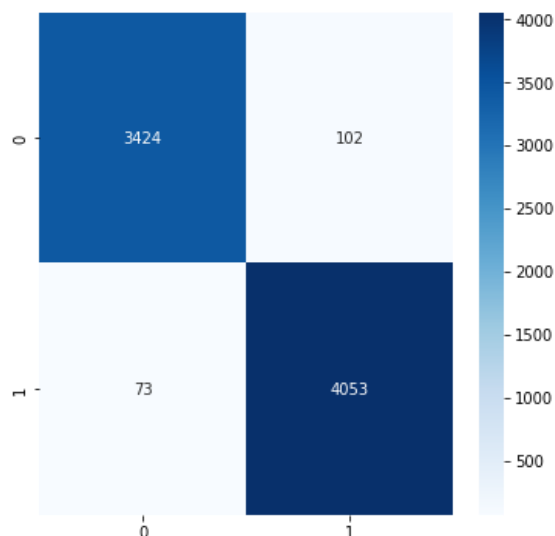
Mostramos también la evolución de la función de error y de la precisión a lo largo del entrenamiento, pudiéndose observar cómo ambas funciones alcanzan un valor aproximadamente constante. Además, el hecho de que estas funciones calculadas sobre los datos del *train* alcancen valores y tendencias muy similares las obtenidas sobre el set de validación nos asegura la ausencia de *overfitting*.



Función de error y Accuracy de la red densa multicapa a lo largo de los epochs

Finalmente obtenemos la matriz de confusión y las métricas ya definidas anteriormente al predecir nuestro modelo sobre los datos de test:

	precision	recall	f1-score	support
0.0	0.98	0.97	0.98	3526
1.0	0.98	0.98	0.98	4126
accuracy			0.98	7652
macro avg	0.98	0.98	0.98	7652
weighted avg	0.98	0.98	0.98	7652



Métricas y matriz de confusión de la red densa multicapa

- **Red Neuronal Recurrente:**

Las redes neuronales recurrentes no tienen una estructura de capas, sino que permiten conexiones arbitrarias entre las neuronas, incluso pudiendo crear ciclos. Con esto se consigue crear la temporalidad, permitiendo que la red tenga memoria.

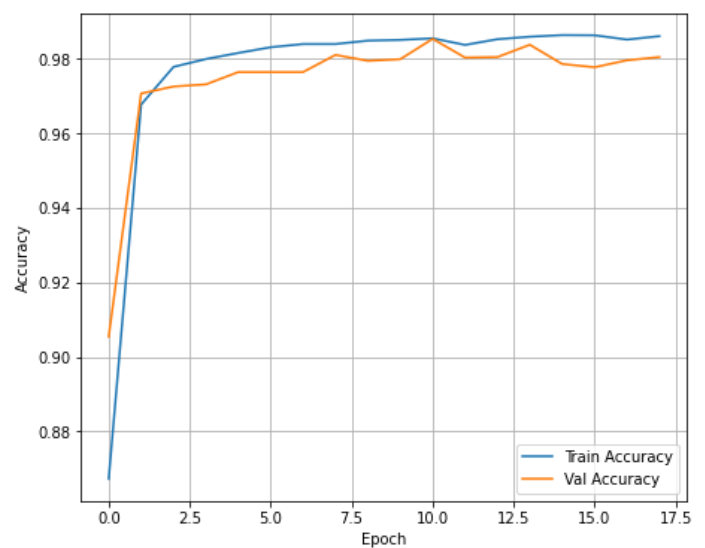
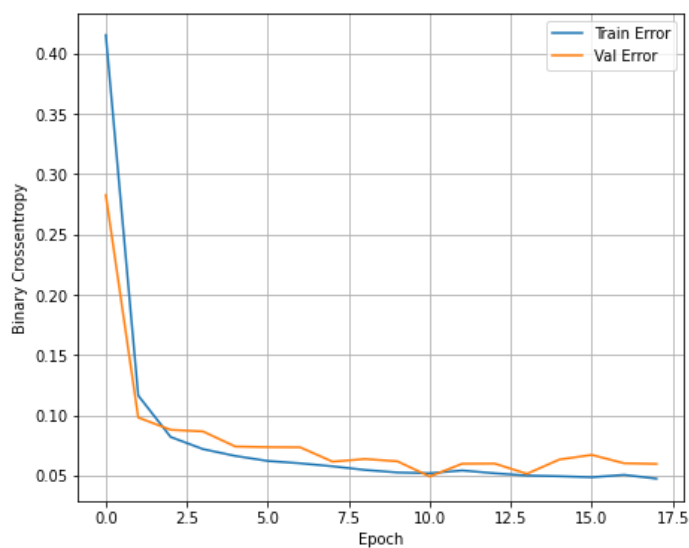
Los datos introducidos en el momento t de la entrada, son transformados y van circulando por la red en los instantes de tiempo o espacio siguientes $t+1$, $t+1$, etc... Esta característica convierte a estas redes en candidatas a la hora de aplicarlas a nuestro estudio, puesto que estamos tratando con secuencias de palabras cuyo orden es importante a la hora de transmitir el mensaje y este orden en consideración en los modelos de redes recurrentes.

De nuevo, mostramos la arquitectura tras haber ajustado los hiperparámetros, la evolución del overfitting y los resultados de las predicciones.

Model: "Modelo1_LSTM"

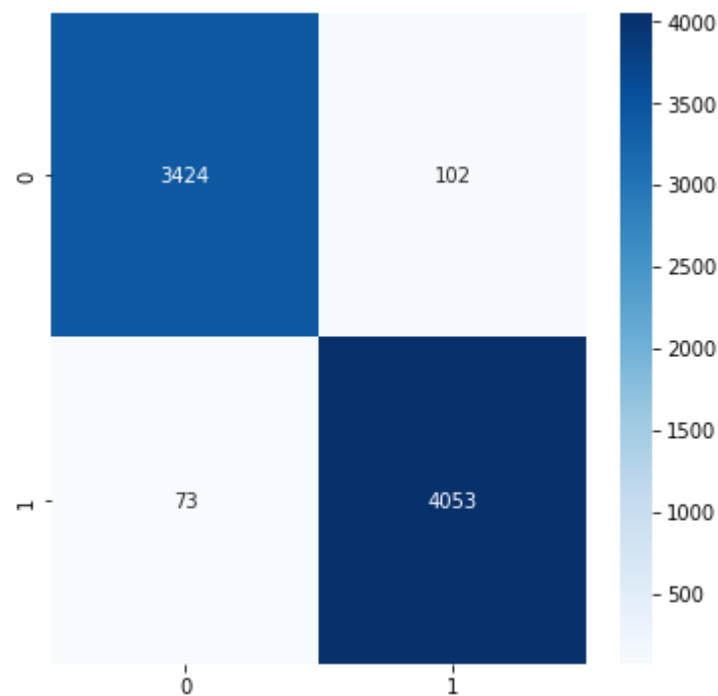
Layer (type)	Output Shape	Param #
LSTM1 (LSTM)	(None, 1, 100)	86400
LSTM2 (LSTM)	(None, 100)	80400
Output_Layer (Dense)	(None, 1)	101
=====		
Total params: 166,901		
Trainable params: 166,901		
Non-trainable params: 0		

Arquitectura de la red neuronal recurrente



Función de error y Accuracy de la red recurrente a lo largo de los epochs.

	precision	recall	f1-score	support
0.0	0.98	0.97	0.98	3526
1.0	0.98	0.98	0.98	4126
accuracy			0.98	7652
macro avg	0.98	0.98	0.98	7652
weighted avg	0.98	0.98	0.98	7652



Métricas y matriz de confusión de la red densa recurrente

- **Red neuronal híbrida (bidireccional y convolucional)**

En este modelo empleamos dos tipos de capas. Por un lado, una capa neuronal recurrente bidireccional. La información en las redes bidireccionales puede propagarse en los dos sentidos, mientras que una red neuronal recurrente solo puede propagarse hacia adelante. Esto significa que cuando se procesan datos de series temporales, el elemento anterior puede afectar a este último, y este último también puede afectar al anterior. Por otro lado, empleamos una capa convolucional. Las redes convolucionales derivan del perceptrón multicapa y se diferencian de este en el momento en el que se les aplica el producto escalar matricial a diferentes conjuntos de datos cercanos (convoluciones). La matriz constante que multiplica a las diferentes matrices de datos se denomina kernel.

De esta forma, la red convolucional aplica el kernel a una serie de ventanas de n términos, desplazándose progresivamente a través del texto y devolviendo valores escalares. Finalmente, estos valores escalares son el *input* de nuestra red bidireccional, que recordemos que tienen en consideración valores pasados y futuros en la predicción de cada elemento t .

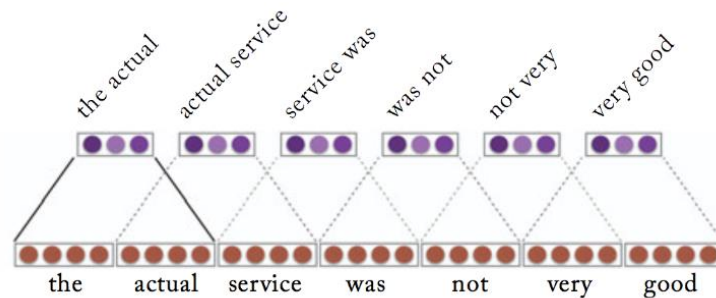
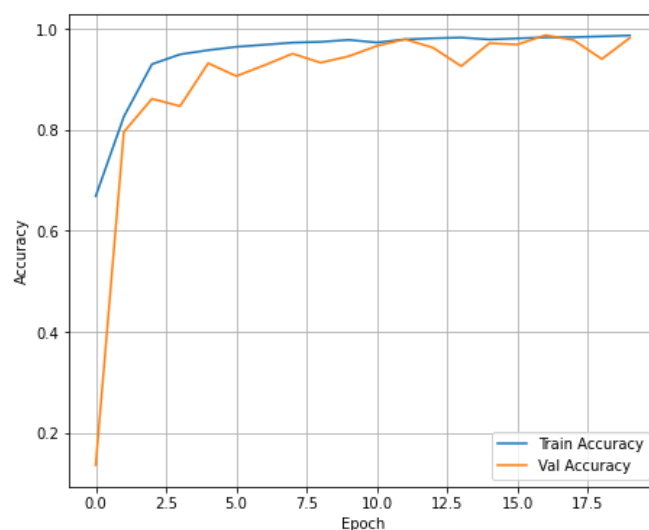
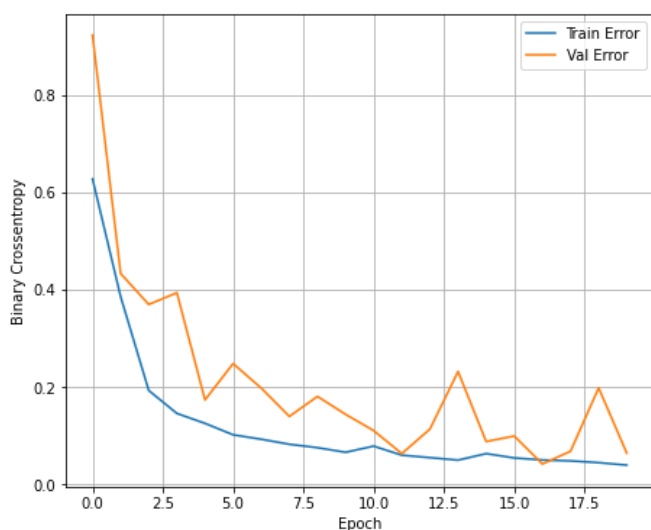


Ilustración de una red convolucional sobre una frase con $k=2$. Imagen extraída del libro de Yoav Goldberg, 'Neural Network Methods for NLP'

A pesar de que probamos con arquitecturas complejas aplicando estos dos tipos de capa, no dieron resultados significativamente mejores que la arquitectura elegida finalmente. Es por esto por lo que decidimos mantener el modelo más simple. Mostramos la arquitectura y los resultados:

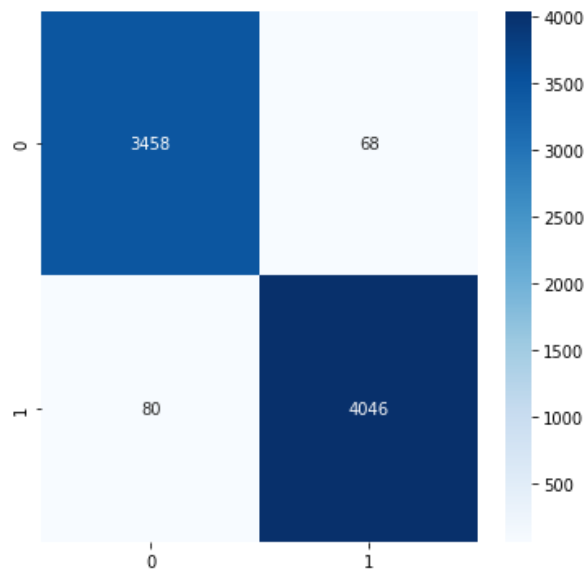
Layer (type)	Output Shape	Param #
CNN1 (Conv1D)	(None, 111, 32)	192
max_pooling1d (MaxPooling1D)	(None, 55, 32)	0
CNN2 (Conv1D)	(None, 51, 32)	5152
max_pooling1d_1 (MaxPooling1D)	(None, 25, 32)	0
Bidirectional (Bidirectional)	(None, 200)	106400
dropout (Dropout)	(None, 200)	0
Output_Layer (Dense)	(None, 1)	201
Total params: 111,945		
Trainable params: 111,945		
Non-trainable params: 0		

Arquitectura de la red neuronal híbrida



Función de error y Accuracy de la red recurrente a lo largo de los epochs.

	precision	recall	f1-score	support
0.0	0.98	0.98	0.98	3526
1.0	0.98	0.98	0.98	4126
accuracy			0.98	7652
macro avg	0.98	0.98	0.98	7652
weighted avg	0.98	0.98	0.98	7652



Métricas y matriz de confusión de la red neuronal híbrida

Conclusión:

Los resultados en secciones anteriores nos demuestran la importancia de los datos de los que se dispone y de su procesamiento. Todos los modelos obtuvieron resultados sorprendentes, siempre por encima del 98% de precisión, probando que la información recibida era fácil de clasificar. Sin embargo, esto genera dudas con respecto a la robustez de los modelos y su capacidad de generalizar. Sería interesante entrenar disponiendo de bases de datos más homogéneas, con más noticias y recogidas por diferentes fuentes

También resulta interesante que las redes neuronales complejas no superaron el rendimiento de las simples. Del mismo modo, las redes neuronales más simple tampoco superan el rendimiento de otros clasificadores no neuronales. Esto nos da a entender que la habilidad de 'recordar' secuencias de datos que tienen las redes neuronales (que puede parecer muy interesante dada la naturaleza del problema) no juega un papel tan importante como el que tienen variables que describen los artículos o la propia frecuencia con la que aparecen ciertos términos

Finalmente, observemos el titular de una noticia calificada como falsa y que era confiable:

"Topless Femen activist tries to snatch Jesus statue from Vatican crib"

Se ve como las palabras que generan emociones influyen al modelo en su predicción. Esto remarca la importancia de aumentar las bases de datos que clasifiquen las noticias como falsas y confiables. El ser humano es, en muchas ocasiones, incapaz de distinguir la veracidad de una noticia. Es por esto por lo que la solución, si existe, se encuentra en patrones ocultos, donde las redes neuronales deberían jugar con ventaja. Se dice que la realidad supera a la ficción y eso es algo que ningún modelo puede aprender a partir del dataset ISOT.