Universidad Europea

UNIVERSIDAD EUROPEA DE MADRID

ESCUELA DE ARQUITECTURA, INGENIERÍA Y DISEÑO

GRADO EN INGENIERÍA MATEMÁTICA APLICADA AL ANÁLISIS DE DATOS

PROYECTO FIN DE GRADO

Aplicación de modelos NLP a la prensa digital española

Pablo de Juan de Uribe

Dirigido por

Alvaro Sánchez Pérez

CURSO 2021-2022

TÍTULO: Aplicación de modelos NLP a la prensa digital española

AUTOR: Pablo de Juan de Uribe

TITULACIÓN: Grado en ingeniería matemática aplicada al análisis de datos

DIRECTOR/ES DEL PROYECTO: Alvaro Sánchez Pérez

FECHA: Julio de 2022

RESUMEN

El presente proyecto se basa en desarrollar un algoritmo fragmentado en tres módulos distintos que faciliten a cualquier persona la lectura en prensa digital, suministrando una manera veloz, eficaz y concisa de obtener y procesar la información más relevante. La implementación de estos módulos ha dado como resultado: categorizar el género al que pertenece una noticia, el sentimiento que pretende transmitir y por último resumir los aspectos clave. Para el desarrollo de cada módulo NLP se han trabajado técnicas matemáticas y modelos analíticos distintos. El proyecto incluye además la creación y gestión de un servidor en la nube que recopilará, procesará y gestionará los datos y modelos generados, y que se utilizaría para su posterior consumo.

Para ello se ha analizado datos reales de la prensa española, en específico se ha estudiado detenidamente el periodo de 2012 a enero de 2022 de ocho periódicos españoles, estos datos han permitido trabajar las cuatro fases de vida de un proyecto Big Data. Finalmente se han realizado gráficos y diversas visualizaciones para mostrar de manera visual los resultados al aplicar el modelo del presente proyecto.

Palabras clave: Análisis de Prensa, Big Data, AWS, Modelos NLP, Python, Ciclo completo del dato.

ABSTRACT

The present project is based on developing an algorithm fragmented into three different modules that make it easier for anyone to read the digital press, providing a fast, efficient and concise way to obtain and process the most relevant information. The implementation of these modules has resulted in: categorizing the genre to which a news item belongs, the feeling it intends to convey and finally summarizing the key aspects. For the development of each NLP module, different mathematical techniques and analytical models have been used. In addition to the creation of a cloud server to manage the data and models generated.

For this purpose, real data from the Spanish press has been analyzed, specifically the period from 2012 to January 2022 of eight Spanish newspapers has been studied in detail, these data have allowed to work the four phases of life of a Big Data project. Finally, graphs and various visualizations have been made to show in a visual way the results of applying the model of this project.

Keywords: Press Analysis, Big Data, AWS, NLP Models, Python, Full Data Cycle.

AGRADECIMIENTOS

A mi familia, en especial a mis padres por los valores de autoexigencia, resiliencia y perseverancia que han hecho posible llegar hasta aquí, a mi pareja y amigos, por el apoyo incondicional para alcanzar esta meta. A mis profesores y tutor de TFG, por estos años de enseñanza, los conocimientos impartidos, la sabiduría transmitida, y la ayuda proporcionada.

Cita - frase célebre / Dedicatoria

“Perseverance is the hard work you do day after day, after you get tired of the hard work you already did.”

TABLA RESUMEN

	DATOS
Nombre y apellidos:	Pablo de Juan de Uribe
Título del proyecto:	Proyecto para sintetizar la información transmitida en las noticias de la prensa española digital
Directores del proyecto:	Alvaro Sánchez Pérez
El proyecto ha implementado un producto: (esta entrada se puede marcar junto a la siguiente)	NO
El proyecto ha consistido en el desarrollo de una investigación o innovación: (esta entrada se puede marcar junto a la anterior)	SI
Objetivo general del proyecto:	Sintetizar la información y elementos relevantes de las noticias de prensa digital española

Índice

Índice de Figuras

Las figuras e ilustraciones utilizadas a lo largo del presente documento son las siguientes:

Índice de Tablas

Las diferentes tablas utilizadas a lo largo del documento se recopilan a continuación:

Capítulo 1. RESUMEN DEL PROYECTO

1. Contexto y justificación

Dado el desarrollo actual de la sociedad de la información, el aumento de datos y el crecimiento de las plataformas que ofrecen diferentes modalidades de periodismo a través de internet, es de suma importancia implementar un proyecto que mejore y reduzca el texto mostrado por las noticias. Agilizando así la lectura y el tiempo implementado en buscar un periódico cuya noticia contenga la información que el lector espera hallar.

2. Planteamiento del problema

La principal problemática que se ha podido observar a la hora de realizar este proyecto se resume en la siguiente idea, el texto es un formato sumamente lento a la hora de comprender una noticia, ya que, cualquier otro medio de transmisión de información es más veloz para el ser humano a la hora de procesar y retener información.

Para reducir esta problemática se han llevado a cabo tres modelos del ámbito NLP que, al combinarlos ofrecen una mejor alternativa para el lector, ya que, estos modelos se encargarán de categorizar y resumir las noticias. En específico estos modelos nos brindarán: la categoría a la que pertenece (sanidad, deporte, política...), el sentimiento que pretende transmitir (positivo, negativo o neutral) y por último un resumen de la noticia.

3. Objetivos del proyecto

El objetivo general del presente trabajo es analizar diferentes periódicos españoles y recopilar un gran número de datos e información comenzando con información de periódicos del año 2012 y finalizando el análisis en enero del 2022. Teniendo como resultado final un modelo de datos que evite la lectura completa de toda la noticia original y agilice el proceso de captación de información mediante la categorización y resumen de las noticias. Todo este proceso será llevado a cabo en un servidor construido en la nube, para sobrellevar las dificultades de procesamiento de altos volúmenes de información, almacenamiento, y escalabilidad.

4. Resultados obtenidos

Como resultados del presente trabajo, se ha podido destacar lo siguiente:

1.- Se ha construido un servidor Linux en la nube de AWS. Para construir dicho servidor se ha llevado a cabo un análisis extenso sobre los recursos, y alternativas disponibles, ya que, se ha tenido que configurar hasta el mínimo detalle para su funcionamiento, como puede ser la cantidad de memoria que debe tener este servidor, la capacidad de computación, o el almacenamiento necesario para desarrollar en su completitud el proyecto.

2.- Una vez construido de principio a fin el servidor, se ha procedido a extraer diez años de noticias pertenecientes a ocho periódicos distintos españoles, ya que no había conjuntos de datos sobre las noticias preexistentes. Estos datos que han sido generados, han sido obtenidos por medio de un programa de WebScrapping construido con Python. Se ha elegido Python como lenguaje de programación y desarrollo debido a que abarca y gestiona de manera eficaz un alto volumen de datos.

3.- A continuación se ha procedido a construir tres modelos de NLP, estos modelos han sido alimentados por los datos extraídos de la prensa, gracias al programa que se ha desarrollado para el WebScrapping. Como resultado de la creación de los tres modelos NLP se ha podido obtener un algoritmo final que tenga como efecto categorizar las noticias de los periódicos, identificar el sentimiento que se pretende transmitir y resumir la noticia.

4.- Por último, se han realizado gráficas que permiten una visualización del resultado de la aplicación de los modelos creados sobre los datos recopilados. En las gráficas se ha podido destacar datos significativos de las noticias recopiladas, como puede ser: la media de cuantos caracteres tiene los periódicos, la distribución de categorías a lo largo del tiempo, o las palabras más significativas para determinar el sentimiento del entre otros.

5. Estructura de la memoria

A continuación se detalla y desglosa la organización de esta memoria, mencionando sus apartados y una explicación sobre su contenido.

- **Capítulo 1**: Se expone un breve resumen del proyecto realizado, haciendo énfasis en el contexto y justificación del proyecto, el planteamiento del problema, los objetivos pautados, y los resultados obtenidos
- **Capítulo 2**: Se expone detalladamente el contexto en el que se desarrolla el presente proyecto, y la importancia de llevarlo a cabo. Se hace especial énfasis en el estado del arte de la cuestión, para comprender las soluciones existentes, la diferenciación de nuestro proyecto, y la manera de plantearlo.
- **Capítulo 3**: Se detallan tanto el objetivo general que busca cumplir el proyecto, como los objetivos específicos que harán posible la culminación de este.
- **Capítulo 4**: En este capítulo se describe muy detalladamente todo el proceso seguido, los pasos realizados, y las justificaciones necesarias para explicar cada acción tomada hasta alcanzar el objetivo del proyecto.
- **Capítulo 5**: Se exponen los resultados obtenidos al realizar el proyecto.
- **Capítulo 6**: Se desarrollan las conclusiones del presente trabajo
- **Capítulo 7**: Se detallan las futuras líneas de trabajo para continuar expandiendo y mejorando el proyecto desarrollado.
- **Capítulo 8**: Se exponen las referencias utilizadas para documentarse a la hora de desarrollar el proyecto.

Capítulo 2. ANTECEDENTES / ESTADO DEL ARTE

2.1 Estado del arte

La relevancia de la prensa digital.

Para poder comprender el presente proyecto es de suma importancia contextualizar la prensa digital.

En España, las primeras experiencias de prensa digital nacen entre 1944 y 1999. Uno de los primeros proveedores de acceso a internet es Servicom. El primer periódico en España que realizó una versión online fue el periódico catalán Aquí en 1995, le siguió El Periódico de Catalunya utilizando formatos de HTML y PDF. A medida que los años pasaban, los medios digitales españoles empezaron a utilizar versiones digitales, siendo La Estrella Digital el primer periódico en trabajar 100% en digital sin edición impresa. [11]

En específico, hubo dos sucesos que impactaron de manera internacional las impresiones digitales de periódico. Por un lado el 11 de septiembre de 2001 con el atentado de las torres gemelas ocasionó que el periódico español El mundo colapsara por el gran número de visitas que recibió. Unos años más adelante con el atentado del 11 de marzo de 2004 marcó una evolución en el mundo digital de periódicos españoles ya que el servidor de publicidad fue desconectado durante varios días.

Desde entonces la tecnología y la necesidad de transmitir noticias a nivel mundial ha ido avanzando, de tal manera que a día de hoy no hay un periódico nacional sino que, son varias las Comunidades Autónomas que tienen su propio periódico como por ejemplo Madrid, Cataluña, Andalucía, Comunidad Valenciana, Castilla y León, Cantabria, Canarias... entre otras, sumando un total de 90 periódicos digitales. [12]

En esta era digital, interconectada globalmente, y con la capacidad de generar datos sobre cualquier área de nuestra vida, se ha alcanzado el punto en el que generamos 2.5 quintillones de bytes al día, y tan solo en los últimos dos años, hemos producido el 90% de los datos mundiales. [1] Es por ello por lo que, el resumen automático de texto puede ser una solución para el gran número de lectores que consumen prensa digital, optimizando una cantidad considerable de tiempo a la hora de leer el periódico y permitiendo mantenerse actualizado sobre el estado del mundo al comprimir los artículos de noticias, la documentación técnica o los libros a un formato mucho más digerible.

.

Como se observa en el gráfico [Ilustración 1] el acceso a internet como medio de penetración a la hora de consumir información en España tiene el puesto número uno, por lo que trabajar el periódico en formato digital como proyecto ha sido de relevancia no solo académica sino que, es una solución que se puede aplicar fuera de un ámbito estudiantil.

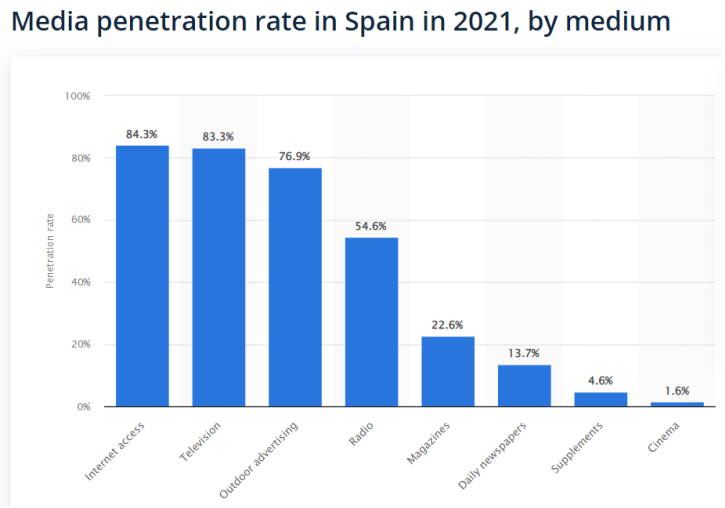


Ilustración 1. Índice de penetración de los medios de comunicación en España en 2021

Dentro del internet como proveedor de noticias digitales el periódico digital es el medio más frecuentado por los españoles, donde nueve de las diez fuentes más utilizadas para el consumo de noticias en España son periódicos. Ello quiere decir que, en España a la hora de elegir un medio que proporcione noticias en formato digital se elige el periódico frente a, por ejemplo, la televisión online o la radio digital. Véase el gráfico [Ilustración 2]

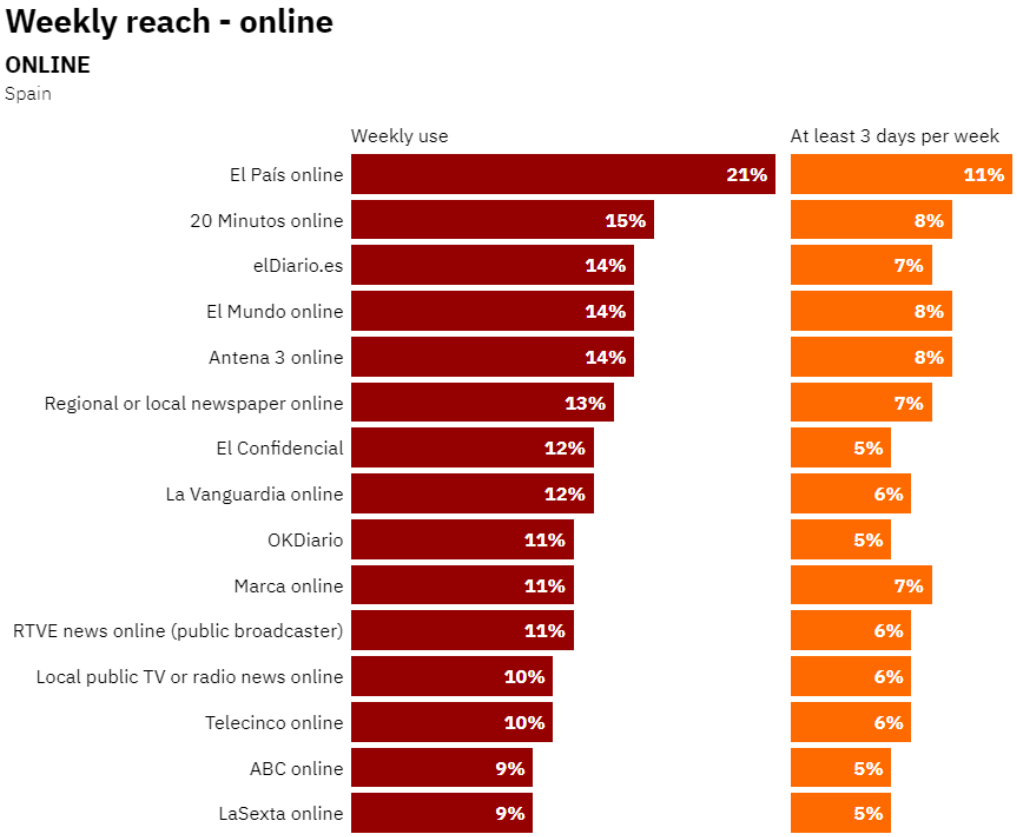


Ilustración 2. Fuentes digitales más utilizadas para consumir noticias

Tecnología Actual

El procesamiento del lenguaje natural (NLP) es la rama de la informática -y más concretamente, de la inteligencia artificial o IA que se ocupa de dotar a los ordenadores de la capacidad de entender textos y palabras habladas del mismo modo que los seres humanos.

Esta rama de estudio combina la lingüística junto con modelos estadísticos, modelos de aprendizaje automático o Machine Learning, y modelos de análisis profundo(Deep Learning) para trabajar datos en formato de texto o voz y comprender el significado implícito en ellos. Algunos de los ejemplos comunes en esta área en desarrollo son la creación de chatbots o asistentes digitales como Alexa de Amazon, o Google Home, sistemas de traducción automática en tiempo real, o la generación de texto mediante voz. [19]

Debido a que el modelo final, se compone principalmente de tres algoritmos capaces de resolver tres tareas NLP diferentes, vamos a exponer a continuación cuales son las técnicas y tecnologías actuales más relevantes a la hora de trabajar dichos problemas.

1- En primer lugar, para sintetizar la información del artículo, se ha realizado una tarea denominada resumen de texto, la cual se define como la capacidad de escribir una versión más corta y condensada de un párrafo, un artículo o un libro conservando la mayor parte del significado del texto original. [3] Esta tarea de NLP se divide en dos posibles enfoques, el primero de ellos es conocido como resumen abstractivo, y se centra en la generación de frases nuevas tras comprender el significado del texto procesado. El segundo enfoque se denomina resumen extractivo.

El enfoque abstractivo es lo más similar a como los humanos generamos resúmenes, ya que la idea principal consiste en comprender los elementos lingüísticos del texto, y en base a ellos generar nuevas frases que expliquen la información consumida. El desarrollo de este enfoque conlleva un alto coste computacional y suele trabajarse mediante modelos de Deep análisis muy complejos. El enfoque Extractivo, se centra en extraer las frases más relevantes del texto parseado y concatenarlas para generar el resumen final, esto se consigue aplicando diferentes cálculos y modelos matemáticos como puede ser la generación de documento de frecuencia de palabras TF-IDF para posteriormente identificar los elementos más relevantes. [10]

2- En segundo lugar la tarea de categorizar texto, se considera una tarea de predicción donde se procura determinar a qué subgrupo pertenece un elemento especifico tras analizar la información disponible sobre él, estos subgrupos pueden ser categorías de periódicos, tipos de película, u otras diversas aplicaciones. Debido a este enfoque en la predicción, estos problemas se solían plantear y tratar con modelos de regresiones logísticas, máquinas de soporte vectoriales, u otros modelos lineales que dependían de una previa generación de vectores de contenido “feature vectors” sobre los que aprender. Sin embargo en los últimos 7 años se está expandiendo y trabajando mucho en la aplicación de Deep análisis y redes neuronales para poder trabajar estos problemas y obtener una comprensión mejor. [14]

3- Por último hemos de cubrir la tarea de análisis de sentimiento. La capacidad de determinar el sentimiento o polaridad de un texto se entiende como “ poder identificar y categorizar computacionalmente las opiniones expresadas en un texto”, especialmente para determinar si la actitud del escritor hacia un determinado tema es positiva o negativa. Las aplicaciones de estos sistemas son de gran utilidad por diversas razones, como puede ser el conocer la opinión o reputación de una compañía, objeto o producto. Por este motivo, el análisis de sentimiento es una tarea altamente trabajada en el área NLP. Dentro del estudio de estas tareas encontramos que la tendencia actual se centra de nuevo en aplicaciones de Deep análisis y combinaciones de redes neuronales para obtener los mejores resultados posibles, actualmente el mejor algoritmo desarrollado combina una red neuronal LSTM y una red neuronal recurrente RNN para obtener un entendimiento muy efectivo sobre el sentimiento presente en el texto analizado.

Soluciones Similares

A pesar de que actualmente no existe una solución que combine sintetización de texto, catalogación y extracción de sentimiento bajo un producto final, existen algunas alternativas conocidas que desarrollan muy bien una de las áreas que este proyecto busca trabajar

1. **Google News**

Google News es un agregador de noticias automatizado que proporciona a sus usuarios noticias relevantes actualizadas cada 15 minutos en base a ciertos parámetros de selección previamente configurados como puede ser la temática sobre la que leer. Este servicio, que actualmente se posiciona como el mayor agregador de noticias mundial, rastrea constantemente la web en busca de noticias escritas y las indexa de manera similar que el buscador Google search engine, funcionalmente analizando el texto encontrado, extrayendo los elementos, y eliminando las partes consideradas irrelevantes. Esta extracción selectiva se considera un algoritmo de reducción de texto ya que fundamentalmente clasifica y devuelve las frases que considera más relevantes y con esa información determina las noticias a mostrar a sus usuarios e incluso ofrece vistas previas antes de entrar en las noticias para obtener una idea general del artículo.[2]

2. **SMMRY BOT**

Las empresas tecnológicas no son las únicas que trabajan la extracción de información de artículos de texto digitales, varias redes sociales como Twitter o Reddit están dedicando muchos recursos a la difusión de noticias en sus redes, incluso creando secciones enteras dedicadas a las noticias, y tomando ese rol de medio de comunicación alternativo a los periódicos tradicionales. Estas plataformas que provienen de empresas puramente tecnológicas están además apostando por la aplicación de modelos de datos a todos los niveles y esto ha desembocado en la creación de diferentes bots que resumen texto en sus secciones de noticias. SMMRY es un bot que navega la plataforma de Reddit en busca de artículos de noticias, y adjunta a ellos un resumen con la certeza de ser relevante y poseer menos del 50% de caracteres que la noticia original. Estas nuevas tecnologías y formatos de proporcionar información relevante, resumida y fácil de consumir están atrayendo mucho flujo de usuarios de internet a su sección de noticias, reforzando así la relevancia de esta clase de proyectos de Inteligencia Artificial. [13]

2.2 Contexto y justificación

El formato de la prensa digital actual resulta efectivo a la hora de transmitir información, sin embargo no es eficiente ni veloz, el tiempo y esfuerzo requeridos para comprender todo lo que los periódicos nos cuentan en sus artículos es mucho mayor que el necesario para consumir esa misma información en otros formatos como vídeos o audios, los cuales son un 95% más efectivos a la hora de transmitir información.[6] Por este motivo plataformas como Twitter se están popularizando a la hora de transmitir noticias en la población, gracias a sus formatos cortos y sencillos, por lo que los periódicos deberían buscar adoptar esta tendencia y no perder su liderazgo a la hora de transmitir noticias.

Actualmente no existe un servicio que proporcione versiones simplificadas de las noticias ni formatos más sencillos para consumir el texto, sin embargo la tecnología para realizar estas tareas está desarrollada, y es un campo de estudio muy activo y con

muchos recursos en la comunidad científica. Es por ello que, el desarrollo de este proyecto, proporciona al lector una noticia resumida, explicando la temática, y su sentimiento, puede ser un gran valor añadido y método eficaz de consumir texto, el cual también podría ser adaptado por los periódicos como antecedente antes de adentrarnos en una noticia, o como complemento explicativo para ofrecer ambas versiones al lector y que elija este como estar informado y a que granularidad de detalle.

.

2.3 Planteamiento del problema

Con la finalidad de evitar emplear tiempo en buscar matices clave a la hora de leer una noticia o eludir leer largos párrafos de texto que buscan mantener al lector dentro de una noticia, ha surgido la necesidad de crear desde cero una herramienta sencilla que proporcione un resumen, categoría e informe del sentimiento que se pretende producir a la hora de leer una noticia.

Por ello, se propone una solución que proporcione un resumen del texto a consumir, explique la temática/categoría a la que pertenece, e indique el sentimiento positivo, negativo o neutral expresado en las noticias españolas. Además esta solución puede ser un gran complemento añadido para incorporar por parte de los diferentes periódicos españoles, y con la suficiente precisión puede llegar a ser un sustitutivo a las noticias originales analizadas.

Capítulo 3. OBJETIVOS

3.1 Objetivos generales

El objetivo general del presente trabajo consiste en generar tres modelos NLP que agilicen la lectura de los periódicos digitales, suministrando una manera veloz, eficaz y concisa de obtener y procesar la información más relevante. Para obtener este resultado ha sido necesario la recopilación de diez años de noticias de prensa española, mediante la creación de un programa (webscarping) utilizado para la extracción de datos y generación de un conjunto de datos limpios y útiles(dataset) ya que, previo a este proyecto no existían datasets relevantes sobre la prensa española digital.

3.2 Objetivos específicos

- Seleccionar los periódicos más relevantes y leídos por la población española para realizar el proyecto con datos reales

- Analizar las diferentes páginas web de los periódicos seleccionados, al igual que estudiar las fuentes de datos disponibles, para que, mediante técnicas de webscraping, se pueda recopilar todas las noticias de los últimos diez años.

- Tratar, almacenar, y analizar los datos recopilados, para generar y trabajar sobre un dataset limpio y de calidad.

- Desarrollar un modelo que identifique el contexto de la noticia del periódico y genere a su vez una categoría específica para cada noticia. En específico el modelo descrito podrá clasificar la noticia dependiendo del género periodístico que exprese por ejemplo (sanidad, deporte, política, historia…etc.)

- Desarrollar un modelo que proporcione, antes de comenzar a leer una noticia en formato digital, la comprensión del sentimiento que el autor pretende transmitir. Se ha dividido la comprensión del sentimiento en tres subcategorías: la primera un sentimiento positivo, la segunda un sentimiento neutro y por último uno negativo.

- Desarrollar un modelo capaz de sintetizar el texto de una noticia, extrayendo los componentes principales y relevantes, y que transmita la misma información que la noticia original, con menos del 50% de caracteres.

- Desarrollar el proyecto en un entorno en la nube, como AWS para gestionar los problemas reales a los que se enfrentan esta clase de proyectos como, grandes volúmenes de datos, capacidad de procesamiento, o escalabilidad.

- Llevar a cabo todas las fases de un ciclo de vida de proyecto BIG DATA, las cuales son:
  - Recogida y filtrado de datos.
  - Extracción de datos.
  - Validación y limpieza de los datos.
  - Análisis de los datos.
  - Visualización de los datos.

- Evaluar los modelos generados y representar visualmente los resultados obtenidos.

- Realizar un análisis descriptivo sobre el dataset una vez obtenidos todos los datos, y aplicado todos los modelos sobre ellos para comprender mejor las noticias españolas.

- Demostrar la adquisición de conocimiento de diferentes asignaturas cursadas durante la carrera y empleadas en conjunto para desarrollar el proyecto.

3.3 Beneficios del proyecto

Debido a que el formato de texto es uno de los medios más lentos para el ser humano de procesar información y a la vez un medio digital utilizado con alta frecuencia, este proyecto permitirá beneficiar al lector reduciendo el tiempo necesario para obtener la información relevante de una noticia, en lugar de perderse en el lenguaje superfluo y las ideas repetitivas. Agilizando de esta manera, el consumo de noticias escritas y mejorando la capacidad del lector mantenerse informado de una manera más efectiva.

Otro beneficio importante a la hora de realizar este proyecto ha sido visualizar de todas las fases necesarias para que un proyecto de big data tome vida, solventar los desafíos que se ha encontrado a la hora de realizar este proyecto implementando las bases y principios que los analistas necesitan a la hora de desarrollar un proyecto de principio a fin, demostrándose la aplicabilidad de este resultado fuera del ámbito estudiantil.

Capítulo 4. DESARROLLO DEL PROYECTO

4.1 Planificación del proyecto

Para la planificación del proyecto, se ha utilizado una metodología de trabajo Agile siguiendo un orden de tareas en formato de cascada. El proyecto se ha ido realizando con el desarrollo de bloques de trabajo (sprints) que han tenido una duración de una a dos semanas. Estos bloques han sido utilizados para alcanzar metas establecidas y culminar bloques del proyecto de manera semanal. Debido a situaciones externas e incidentes por falta de docentes disponibles para tutorizar el proyecto, la planificación del mismo ha consistido en semanas bastante intensas con tareas sumamente definidas. Véase a continuación el siguiente gráfico [Ilustración 3, 4 y 5].

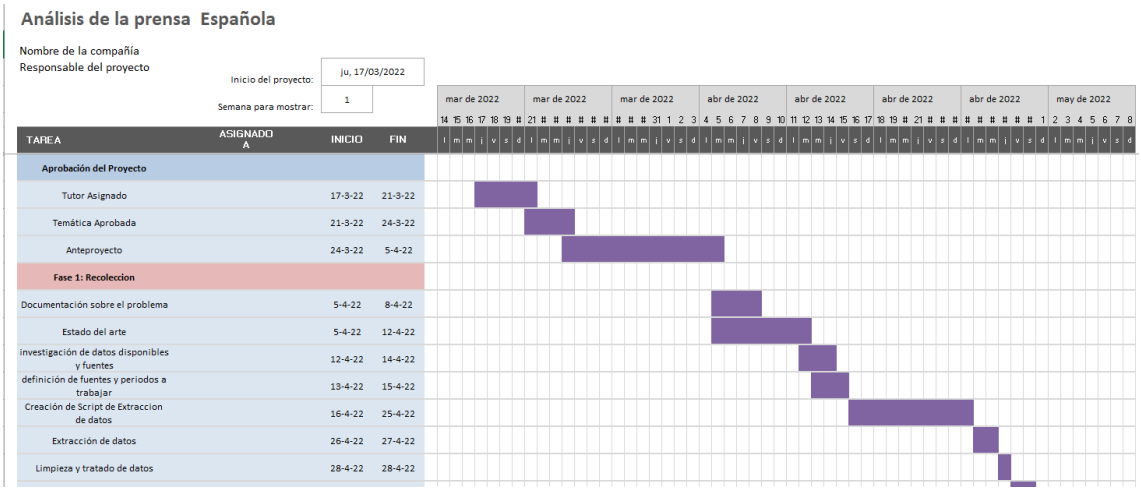


Ilustración 3. Diagrama Gantt sobre planificación del proyecto (1/3)

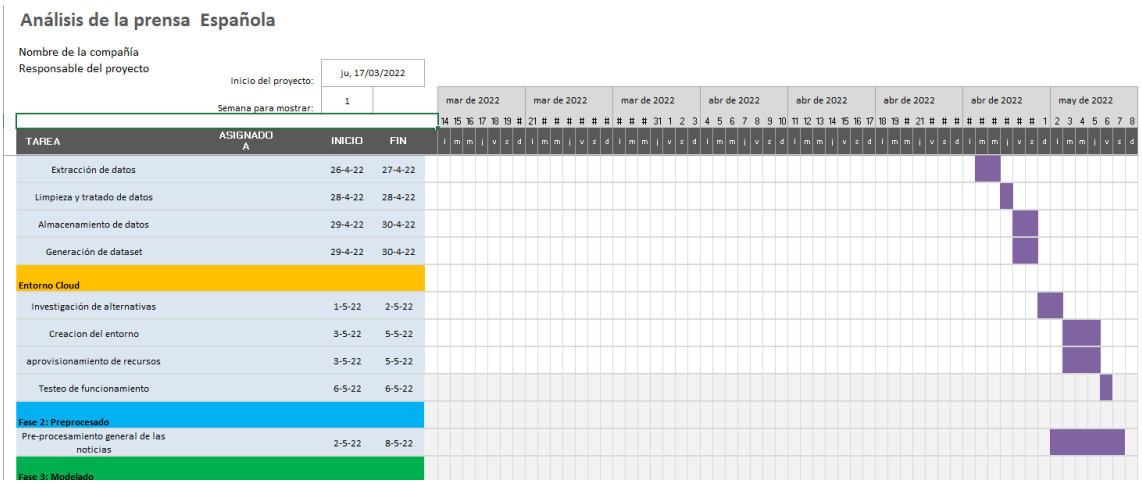


Ilustración 4. Diagrama Gantt sobre planificación del proyecto (2/3)

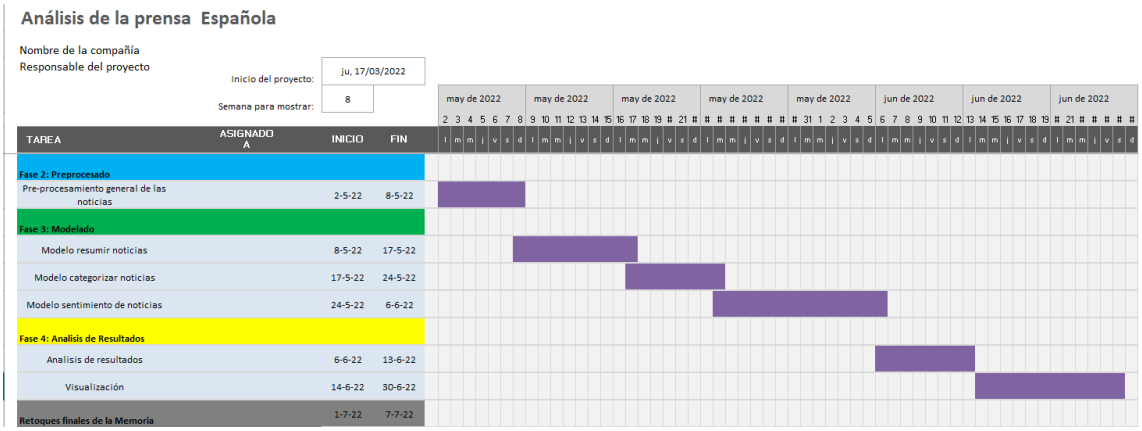


Ilustración 5. Diagrama Gantt sobre planificación del proyecto (3/3)

Desglose de tareas

Fase 0: Aprobación del proyecto (17 de marzo- 5 abril)

- Tutor Asignado: Se define el docente que tutelará el proyecto
- Temática Aprobada: Tras una serie de reuniones entre docente y estudiante, se define el proyecto a realizar, el alcance, y otros aspectos primordiales.
- Anteproyecto: Se genera un primer documento en el que se expone la motivación y justificación del proyecto, los límites y objetivos establecidos, y el plan de acción desglosado en tareas para realizar el proyecto.

Fase 1: Recolección de información

- Documentación sobre el problema: Se realiza un estudio sobre el problema a resolver con el presente proyecto(agilizar el consumo de noticias de texto), los retos y dificultades característicos de este problema NLP, y la importancia y relevancia de abordar este proyecto.
- Estado del arte: Se realiza un estudio sobre el estado del arte actual en la materia y las soluciones similares existentes para así poner en perspectiva la aplicación entre los programas/servidores/herramientas que dan una solución parcial al problema planteado en el proyecto y lo desarrollado y diferenciador en este trabajo
- Investigación de datos disponibles y fuentes: Se ha investigado los diferentes conjuntos de datos comúnmente utilizados para proyectos similares al actual, y se ha valorado las diferentes fuentes y páginas de los periódicos para comprender los datos disponibles
- Definición de fuentes y periodos a trabajar: Tras concluir que no existe un dataset relevante para llevar a cabo el proyecto, nos proponemos a construir nuestros datos, extrayendo la información de los periódicos españoles digitales, definiendo las fuentes más relevantes y el periodo de extracción.
- Creación de Script de Extracción de datos: Se desarrolla un programa que navega los portales de los periódicos españoles extrayendo todas las noticias pertenecientes al periodo de 2012 hasta enero de 2022, teniendo en cuenta los diferentes diseños web y formatos utilizados en cada periódico.
- Extracción de datos : Una vez construido el script se ejecuta para que vaya extrayendo todas las noticias sobre las que se trabajara en el proyecto.

- Limpieza y tratado de datos: Se evalúan los datos recolectados y se realizan diferentes limpiezas y correcciones para finalizar con un dataset limpio, útil y relevante.
- Almacenamiento de datos: Debido al volumen de todos los datos recolectados, se definen los protocolos de almacenamiento para poder trabajar y operar con toda la información.
- Generación de Dataset: Se genera un dataset que será la fuente de información para todos los siguientes pasos del proyecto

Fase 1.5 Entorno de desarrollo en la nube (AWS)

- Investigación de alternativas: Debido a la cantidad de datos generados, trabajar en un entorno local no resulta viable por motivos de almacenamiento de grandes volúmenes de datos, y capacidad de computación necesaria para trabajar el proyecto en un tiempo razonable cumpliendo fechas de entrega.
- Creación del entorno: Tras valorar las alternativas disponibles, construimos en la plataforma de Amazon Web Services un servidor Ubuntu para desarrollar la extracción y el análisis de los datos.
- Aprovisionamiento de recursos: Definimos las características del servidor, el almacenamiento, la memoria, los núcleos, e instalamos las diferentes herramientas que se necesitarán para el proyecto en nuestro nuevo entorno en la nube.
- Testeo de funcionamiento: Se realizan una serie de pruebas para asegurarnos que el entorno está disponible, es accesible, y que efectivamente es más potente que nuestra maquina local.

Fase 2 Preprocesado

- Preprocesamiento general de las noticias: Se realiza un preprocesamiento general sobre el dataset, ya que debido a la diferente naturaleza de los tres modelos a desarrollar, cada modelo conlleva su propio preprocesado específico.

Fase 3: Modelado

- Modelo resumir noticias: Se realiza un estudio previo sobre el resumen de noticias en tareas NLP, se preprocesan los datos generados, se desarrolla el modelo, se entrena, y finalmente se evalúa. Una vez finalizada esta tarea se aplica el modelo a todas las noticias existentes, generando así un resumen de todas ellas para analizar posteriormente.
- Modelo Categorizar Noticias: Se realiza un estudio previo sobre la categorización de noticias en tareas NLP, se preprocesan los datos generados, se desarrolla el modelo, se entrena, y finalmente se evalúa. Una vez finalizada esta tarea se aplica el modelo a todas las noticias existentes, denominando así una categoría a todas las noticias, para analizar posteriormente.
- Modelo sentimiento de noticias: Se realiza un estudio previo sobre el análisis de sentimiento en texto, se preprocesan los datos generados, se desarrolla el modelo, se entrena, y finalmente se evalúa. Una vez finalizada esta tarea se aplica el modelo a todas las noticias existentes, denominando así todos los textos como positivos, negativos, o neutrales, para analizar posteriormente.

Fase 4: Análisis de Resultados

- Análisis de Resultados y Visualización: Se analizan los textos tras haber aplicado todos los modelos, y se generan diferentes visualizaciones para mostrar las diferentes características de las noticias españolas, y las diferencias entre los principales periódicos que las publican.

Retoques finales de la Memoria:

- Se realizan los retoques finales sobre la memoria desarrollada para explicar el trabajo realizado.

4.2 Descripción detallada de la solución, metodologías y herramientas empleadas

El desarrollo de este proyecto se ha realizado teniendo en cuenta tres grandes bloques, El primero de ellos la obtención y tratamiento de los datos, el segundo bloque ha sido el modelado de los datos y desarrollo de algoritmos, y el tercer bloque ha sido la implementación visual y análisis de los resultado.

Sin embargo, antes de comenzar a explicar en detalle el proceso seguido y el desarrollo de la solución, hemos de mencionar el entorno en el que esta solución ha sido posible. En el inicio del proyecto, la extracción de datos se desarrolló en un entorno local, con los recursos y los límites que este ofrecía, sin embargo en cuanto la comenzó la extracción de datos, y los registros de noticias superaban la cifra de un millón de registros, fue evidente que gestionar esta cantidad de datos en un tiempo razonable iba a requerir escalar el proyecto y conseguir más recursos, por ello tomamos la decisión de migrar el proyecto a la nube.

Tras comparar los servicios de Azure, Google Cloud Plataform, y Amazon Web Services, decidimos optar por gestionar un servidor en AWS, gracias a la sencillez de la plataforma, la experiencia previa en esta, y la existencia de una capa gratuita con suficientes recursos para desarrollar todo el proyecto.

4.2.1 Servidor en la nube

Para crear nuestro servidor en la plataforma de AWS comenzamos por crear una instancia EC2 (Elastic Computing) de tipo t2 micro, la cual forma parte de la capa gratuita de Amazon, por lo que mientras no sobrecarguemos el servidor, no debería incurrirnos ningún coste por operar en ella. A la hora de aprovisionar nuestro servidor en Amazon vamos a elegir un sistema operativo Linux, y basarnos en Ubuntu 20.04 gracias a la estabilidad de esta versión y la flexibilidad que Linux nos ofrece para aprovisionar nuestro servidor como necesitemos. Tras determinarla distribución sobre la que crearemos el servidor, configuraremos los recursos necesarios para el funcionamiento del servidor, siendo estos 25GB de almacenamiento en EBS (Elastic Block Storage), 750 horas mensuales de cómputo, y un millon de peticiones al mes, abordando así todo lo necesario para el desarrollo del proyecto. Una vez creada la instancia, definimos los puertos e ip de conexión del servidor, un fichero .pem con las claves de acceso, y nos conectamos a través de la consola de Windows a través del protocolo SSH para asegurar nuestra conexión y evitar accesos no deseados, obteniendo así un sistema robusto.

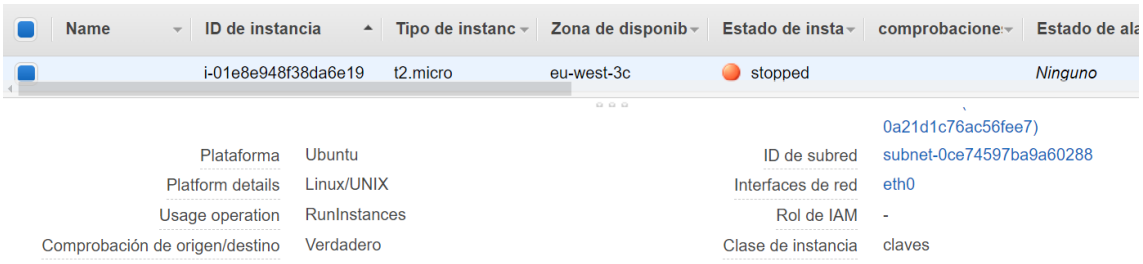


Ilustración 6 creación de la instancia del servidor en la nube

Dentro del servidor hemos de instalar las herramientas necesarias para el proyecto, pero debido a que utilizamos AWS principalmente por su capacidad de almacenamiento y computación, nos limitamos a instalar Anaconda, Python versión 3, y jupyter notebook como IDE de desarrollo. Una vez realizado estas instalaciones, desde la línea de comandos inicializamos el entorno de jupyter mediante el comando

nohub jupyter-notebook --ip 0.0.0.0 --no-browser --allow-root

Este comando nos permite cerrar la sesión de la terminal sin la consecuencia de desconexión del entorno ya que tras perder la conexión con el cliente, el servidor almacenara todo en el fichero nohub y no forzara una desconexión. Tan solo queda abrir el navegador web de nuestro ordenador, escribir la ip del servidor, introducir la contraseña definida, y ya tenemos un entorno Python listo para trabajar.

```
(pyenv) ubuntu@ip-172-31-33-237:~$ jupyter notebook --generate-config
Writing default config to: /home/ubuntu/.jupyter/jupyter_notebook_config.py
(pyenv) ubuntu@ip-172-31-33-237:~$ jupyter notebook password
Enter password:
Verify password:
[NotebookPasswordApp] Wrote hashed password to /home/ubuntu/.jupyter/jupyter_notebook_config.json
(pyenv) ubuntu@ip-172-31-33-237:~$ jupyter-lab --ip 0.0.0.0 --no-browser --allow-root
[I 2022-05-08 13:09:00.899 ServerApp] jupyterlab | extension was successfully linked.
[W 2022-05-08 13:09:00.903 NotebookApp] 'password' has moved from NotebookApp to ServerApp. This config will be passed t
o ServerApp. Be sure to update your config before our next release.
[I 2022-05-08 13:09:00.910 ServerApp] nbclassic | extension was successfully linked.
[I 2022-05-08 13:09:00.912 ServerApp] Writing Jupyter server cookie secret to /home/ubuntu/.local/share/jupyter/runtime/
jupyter_cookie_secret
[I 2022-05-08 13:09:01.229 ServerApp] notebook_shim | extension was successfully linked.
[I 2022-05-08 13:09:01.261 ServerApp] notebook_shim | extension was successfully loaded.
[I 2022-05-08 13:09:01.262 LabApp] JupyterLab extension loaded from /home/ubuntu/pyenv/lib/python3.8/site-packages/jupyt
erlab
[I 2022-05-08 13:09:01.262 LabApp] JupyterLab application directory is /home/ubuntu/pyenv/share/jupyter/lab
[I 2022-05-08 13:09:01.266 ServerApp] jupyterlab | extension was successfully loaded.
[I 2022-05-08 13:09:01.270 ServerApp] nbclassic | extension was successfully loaded.
[I 2022-05-08 13:09:01.271 ServerApp] Serving notebooks from local directory: /home/ubuntu
[I 2022-05-08 13:09:01.271 ServerApp] Jupyter Server 1.17.0 is running at:
[I 2022-05-08 13:09:01.271 ServerApp] http://ip-172-31-33-237:8888/lab
[I 2022-05-08 13:09:01.271 ServerApp]  or http://127.0.0.1:8888/lab
[I 2022-05-08 13:09:01.271 ServerApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirm
ation).
```

Ilustración 7. Comandos ejecutados para inicializar Jupyter

Finalmente observamos que somos capaces de conectarnos a nuestro servidor, mediante el uso de cualquier navegador web, la dirección IP de nuestro servidor, y la contraseña definida para gestionar los accesos.

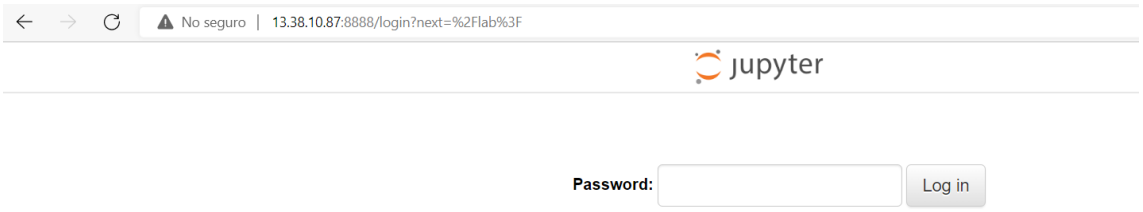


Ilustración 8. Entorno Jupyter correctamente funcionando y accesible a través del navegador web

4.2.2 Extracción de los datos

La clave para un buen modelo, es la calidad de sus datos.(M. West)[20]

A la hora de desarrollar cualquier proyecto de Big Data, el elemento más importante e imprescindible son los datos, sin embargo no cualquier dato sirve, sin la existencia de datos limpios, de calidad, y significativos, cualquier otra acción que trate de obtener un buen modelo o resultado será en vano.

Durante la realización del estado del arte y la investigación sobre los datasets existentes sobre análisis de noticias, rápidamente nos dimos cuenta de dos grandes problemas, el primero que los datasets más utilizados para el desarrollo de tareas NLP son datasets en inglés, lo cual no ayuda a desarrollar un buen modelo sobre prensa española, y en segundo lugar, los datasets existentes en español no son significativos para nuestro proyecto. Por este motivo se toma la decisión de extraer datos de 8 periódicos españoles diferentes para tener variedad de fuentes y evitar comenzar a introducir sesgo en el proceso, de modo que se estudia cuáles son los periódicos digitales más relevantes y que valga la pena analizar.

	Newspaper	Location
1	El País	Madrid
2	Marca	Madrid
3	El Mundo	Madrid
4	AS	Madrid
5	ABC	Madrid
6	La Vanguardia	Barcelona
7	20 Minutos	Madrid
8	El Mundo Deportivo	Barcelona
9	Sport	Barcelona
10	El Economista	Madrid
11	El Periodico de Catalunya	Barcelona
12	Expansión	Madrid
13	Diário de Avisos	Santa Cruz de Tenerife
14	Público	Madrid

Ilustración 9. Periodicos Digitales españoles más leídos en la península Ibérica 2021

Tras investigar el número de lectores, el ranking de relevancia de los procuradores de prensa digital, y la existencia de un histórico de noticias en las páginas web de los periódicos, decidimos extraer las noticias de los siguientes periódicos: El Mundo, El

País, OK Diario, ABC, 20 minutos, El Confidencial, El expansión, y El Diario Público. Una vez elegidas las fuentes, se determina un marco temporal de 10 años, desde 1/01/2012 hasta 1/01/2022.

Para desarrollar el script que mediante técnicas de webscrapping extraiga una a una las noticias existentes, primero indagamos en la estructura que presenta cada una de las webs a trabajar. En esta fase nos damos cuentas que periódicos como la Vanguardia publican en sus archivos históricos fotos de los periodicos que se publicaron en una fecha indicada, por lo que al no existir texto sobre la noticia sino una foto, hemos de descartar esta fuente que en principio sería muy relevante. Otro problema a la hora de analizar las estructuras de las diferentes webs de los periodicos surge al darnos cuenta de que la estructura HTML en la que se almacenan las noticias cambió drásticamente en los periodicos entre 2015 y 2016, por lo que un script general no va a ser capaz de extraer correctamente todos los datos, de modo que se estructura en dos versiones diferentes que tienen en cuenta el marco temporal a trabajar.

```
<ul class="resultados-buscador" id="buscador_expansion">
  <li> == $0
    <h2>
      <a href="https://www.expansion.com/2015/01/01/economia/1420105213.html" title=
      "De Guindos asegura que la recuperación es real, pero la crisis no ha acabad
      o">1. De Guindos asegura que la recuperación es real, pero la crisis no ha
      acabado</a>
    </h2>
    <p>…</p>
  </li>
```

Ilustración 10. Estructura HTML El Expansion 01/01/2015

```
<div id="destino">
  <div class="listado">
    <article class="noticia">
      <header>
        <h1>
          <a title="El EI planeaba perpetrar un atentado suicida en Múnich" href="https://
          www.expansion.com/sociedad/2016/01/01/5685dc69ca4741ad648b4628.html">El EI
          planeaba perpetrar un atentado suicida en Múnich</a> == $0
        </h1>
      </header>
      <div class="entradilla">…</div>
    </article>
```

Ilustración 11. Estructura HTML El Expansion 01/01/2016

Finalmente otro problema a destacar es la organización de directorios de cada periódico y las urls generadas para acceder al histórico. Cada periódico gestiona sus urls de maneras muy diferentes por lo que hay que tener en cuenta las diferentes peculiaridades para recopilar satisfactoriamente los datos. Algunos ejemplos de esto son los siguientes

Periódico	Matiz
El Mundo	La url agrega la fecha en formato YYY/MM/DD, La estructura de etiquetas HTML varía entre años
El País	La url agrega la fecha en formato YYY-MM-DD
OK Diario	La primera noticia del día está almacenada en un elemento HTML diferente al resto de noticias
ABC	Incluye el **nombre** del mes en español en su URL, el número de día lo agrega como "diaX" siendo x el valor del día, y agrega el número de la página donde se encuentra la noticia en la edición impresa original EJ: https://www.abc.es/temas/2010/enero/dia1/pagina-3.html
El confidencial	Tiene dos ediciones una de mañana y otra de noche, con noticias diferentes, por lo que hay que considerar agregar el parámetro edición a la url

Una vez estudiadas las diferentes estructuras y elementos HTML de cada periódico a extraer, programamos un Web Scrapper en Python 3 con la ayuda de la librería beautiful soup. La estructura general de funcionamiento del algoritmo es la siguiente:

Algoritmo para extraer noticias de periódicos

1. Defino la fecha de inicio (2012, 1, 1) y fecha de fin (2022, 1, 1) a extraer.
2. Inicializo una lista donde almacenar todas las noticias existentes en la página que navegue, esta lista contendrá los links a el texto de la noticia de modo que texto será extraído más adelante, el periódico del cual se extrae, la fecha, y el titular que aparece.
3. Genero un bucle que navegue desde la fecha de inicio hasta la fecha de fin incrementando el día de uno en uno, este bucle también considera el número de días que el mes actual tiene (28 vs 30 vs 31), y el año en el que se encuentra para evitar errores al navegar años bisiestos.
4. Para cada fecha individual, navego la web del periódico establecido modificando la manera de insertar la fecha como parámetro, navego la estructura HTML teniendo en cuenta los diferentes elementos y tags en los que se almacenan las noticias, y almaceno en una lista todas las noticias del día analizado.
5. Devuelvo la lista final con todas las noticias de todos los días recorridos.

Generamos una lista para cada uno de los periodicos previamente definidos

La lista final devuelta contiene la siguiente estructura

Titulo	Link	Periódico	Fecha
"Titular de la noticia"	Link extraído de la página web, que lleva al texto de la noticia	Nombre del Periódico	Dia/Mes/Año
¡Alarma! Empleado desmotivado	https://www.expansion.com/2012/06/11/empleo/desarrollo-de-carrera/1339435015.html	Expansión	11/06/2012

Tras ejecutar el programa para todos los periódicos y recopilar el listado completo de urls, podemos resumir las noticias obtenidas en la siguiente tabla de resultados

Periódico	N.º de noticias	Tiempo de ejecución	Fecha de inicio
El Mundo	112989	3040.9 Seg	01/01/2012
El País	99441	6824.0 Seg	01/01/2012
ABC	528856	7637.2 Seg	01/01/2012
20 Minutos	1713281	2121 Seg	01/01/2012
El Confidencial	691600	4389 Seg	01/01/2012
El Expansion	313077	2273.2 Seg	01/01/2012
Diario Público	284316	7942.8 Seg	01/01/2012
OK Diario	111994	2475.1 Seg	01/09/2015

4.2.3 Limpieza de los datos

Tras extraer todas las URL con las que trabajaremos a lo largo del proyecto, vamos a realizar una serie de arreglos y mejoras, para garantizar la calidad de los datos obtenidos.

La **primera mejora** consiste en asegurar que todas las urls obtenidas son válidas y accesibles. Tras analizar los listados de links generados, nos damos cuenta de que en ocasiones los periodicos eliminan de los links la cabecera, es decir, https://www.periodico.es/ , ya que el usuario ya se encuentra dentro de este dominio, y tan solo le ha de redirigir a el articulo elegido. Como este no es nuestro caso a la hora de operar los links, hacemos uso de expresiones regulares de manera que verificamos

que todas las URLs contengan la cabecera mencionada anteriormente, y para aquellas que no la tengan, se les adjunta. Un caso especial ocurre en las noticias del Diario Publico, en las que se puede dar el caso en el que las noticias extraídas contengan una doble barra // en las diferentes partes de la url, lo que interfiere a la hora de navegar la url, por ello de nuevo utilizando expresiones regulares, sustituimos el elemento // por una barra simple / , pero siempre dejando intacto la primera doble barra existente, la cual forma parte de la cabecera agregada (https:***//***).

La **segunda mejora** se centra en eliminar url's duplicadas, esto aplica tanto a noticias que aparecen dos veces, como a hipervínculos de navegación que los periódicos insertan a lo largo de su web, los cuales no hacen referencia a noticias. Para ello realizamos un conteo de registros únicos y el número de veces que aparece cada link en un periódico particular, evitando el caso en el que dos periódicos publiquen la misma noticia, y eliminamos los links que no aportan información útil.

```
https://www.elmundo.esdeportes.html                                  1052
https://www.elmundo.es/television/programacion-tv/                    872
                                                                      ...
https://www.elmundo.es/america/2013/04/19/noticias/1366359464.html      1
```

Ilustración 12. Ejemplo de URL no útiles a eliminar en la segunda corrección

La **tercera mejora** consiste en eliminar los links que hagan referencia a noticias fuera del alcance del proyecto, o que carezcan de información en formato de texto. Esto puede ser desde noticias que al abrirlas contienen un video pero no texto al respecto, campañas publicitarias, o suplementos asociados al periódico original como puede ser la revista Yo dona en el caso de El Mundo. Para esta mejora aplicamos expresiones regulares que evalúan las URLs y en caso de pertenecer a las categorías mencionadas anteriormente, se eliminan del listado final.

La **cuarta mejora** continua la línea de las mejoras anteriores, eliminando URL inservibles para asegurar que todos los datos recopilados son útiles. En esta última mejora, observamos que ciertas filas procuran url's incorrectas, y la característica común de ellas es que no finalizan con ".html", por ello a través de expresiones regulares, eliminamos del listado aquellas url que no finalicen con ".html".

La **quinta mejora** consiste en estudiar la distribución de las noticias obtenidas en base a los meses y años recopilados, para garantizar que hemos obtenido datos de todas las fechas, y no solo muchos datos en unas pocas fechas concretas. Para ello dibujamos gráficos de líneas para observar picos que presenten una posible falta de datos en el proceso de recolección, estos gráficos han de interpretarse con cautela, y con criterio para evitar llegar a conclusiones erróneas sobre la apariencia de la distribución.

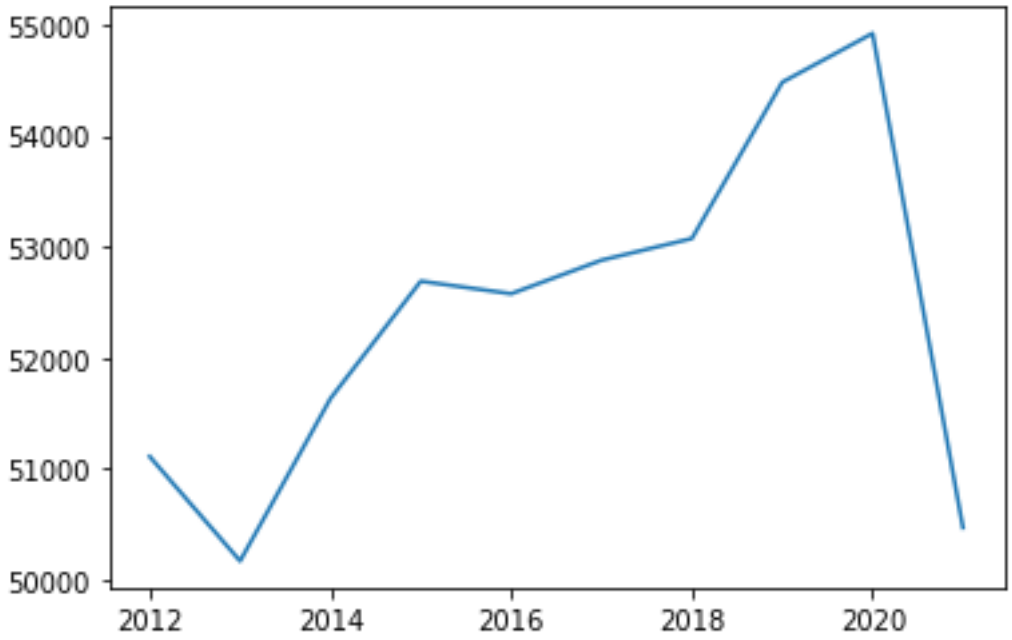


Ilustración 13. Grafico generado sobre la distribución de noticias en el ABC

En esté grafico podría parecer que en 2013 existe una falta de información o que algo ha sucedido en el proceso de recolección que habría que arreglar, sin embargo observando detenidamente el eje de ordenadas (eje vertical) observamos concurren entre 50 y 55 mil noticias, lo que realmente no es una falta de información en un año particular sino la mera distribución y cantidad de noticias generadas por el periódico en el periodo.

A continuación observamos los valores actuales sobre la distribución de las noticias a lo largo de los años para comprender mejor el gráfico anterior, y razonar si existe una diferencia significativa para necesitar una nueva recolección sobre un periódico o fecha particular.

También organizamos la distribución en base a los años, y los meses como segundo filtro, para garantizar que tenemos información sobre todos los meses de todos los años.

	Fecha	Count
0	2012	51111
1	2013	50171
2	2014	51633
3	2015	52689
4	2016	52576
5	2017	52878
6	2018	53073
7	2019	54479
8	2020	54919
9	2021	50471

Ilustración 14. Tabla generada sobre la distribución de noticias en el ABC por año

	Año	Mes	Count
0	2012	1	4358
1	2012	2	4066
2	2012	3	4346
3	2012	4	4214
4	2012	5	4358
..	...	...	...
115	2021	8	4676
116	2021	9	4509
117	2021	10	4654
118	2021	11	4450
119	2021	12	205

Ilustración 15. Tabla generada sobre la distribución de noticias en el ABC por año y mes

Finalmente para aquellas noticias en las que encontramos falta de información o una mala recolección de datos, volvemos a realizar todo el proceso de recolección y corrección en los pasos anteriores, arreglando los elementos que provocaron un fallo originalmente, hasta concluir con un dataset de distribución balanceada.

4.2.4 Extracción de texto

Una vez generado el dataset con todas las urls de noticias publicadas en los últimos 10 años, nos hace falta programar una función que sea capaz de abrir uno a uno los links de cada periódico, parsear el texto de la noticia y devolverlo para ser almacenado y trabajado posteriormente. Para ello una vez más indagamos en las diferentes estructuras que los periódicos utilizan para escribir y publicar artículos digitales, y analizamos el código HTML junto a sus elementos, para obtener una comprensión sobre donde se almacena el texto correspondiente a la notica. A continuación programamos en Python una función que recibe como entrada un link, lo navega gracias a la librería Beautiful Soup, y va buscando dentro de los tags html los párrafos que construyen el texto gracias al análisis previo realizado. En caso de no conseguir detectar el texto de la noticia, se establecen dos casos extremos para garantizar la extracción correcta de la noticia, el primero es devolver todos los párrafos (elemento) presentes en la página navegada, y el segundo es devolver un mensaje base “Texto no Extraído” para evitar que la función falle y perder las noticias correctamente extraídas.

A la hora de desarrollar el modelo de predicción de categoría, se construyeron dos posibles funciones capaces de devolver la categoría de un texto, una de ellas se desarrolló utilizando un enfoque de modelo supervisado, lo que requiere el conocimiento y catalogación previa de los datos para “aprender” cuando acierta o falla una categorización y mejorar. Para alimentar este modelo mencionado nos hizo falta extraer la categoría presente en los artículos ya extraídos para utilizarla como etiqueta (label) en el modelo. Esta categoría se extrae de la noticia mediante una función que parsea las noticias una a una y busca en los elementos HTML predefinidos la categoría de la noticia seleccionada. En caso de no encontrar la categoría presente en la noticia, utilizamos expresiones regulares para devolver el nombre de la categoría que previamente nos viene definida en el propio enlace de la noticia.

4.2.5 Multi Procesamiento

Un recurso clave para poder procesar todos los datos, aplicarlos modelos a todas las noticias, y hacer un uso eficiente de todos los recursos a nuestra disposición ha sido el multi procesamiento. El multi procesamiento se define como “la ejecución de dos o más programas o secuencias de instrucciones simultáneamente por un ordenador con más de un procesador” [7]

Anteriormente ya hemos introducido la necesidad de migrar el proyecto a un servidor en la nube para poder beneficiarnos de todos los recursos a nuestra disposición en este entorno, sin embargo por mucho escalado horizontal de recursos que hagamos y por mucha capacidad de procesamiento que induzcamos en nuestro servidor, todavía existe un peligroso cuello de botella que limita nuestra capacidad de procesar los datos y aplicar los modelos a desarrollar sobre todos los datos disponibles.

Al ejecutar una tarea sobre una noticia especifica, el tiempo de ejecución dependerá de dos factores, el primero de ellos, la velocidad para producir un resultado, está ligada directamente a la cantidad de recursos disponibles, y en segundo lugar el número de vece que ha de ejecutar la tarea, la cual está ligada al número de datos a procesar.

Debido a que nuestro proyecto trabaja con más de 3,8 millones de registros, la ejecución de tareas una a una por cada registro resulta un impedimento para finalizar en tiempo y forma el proyecto.

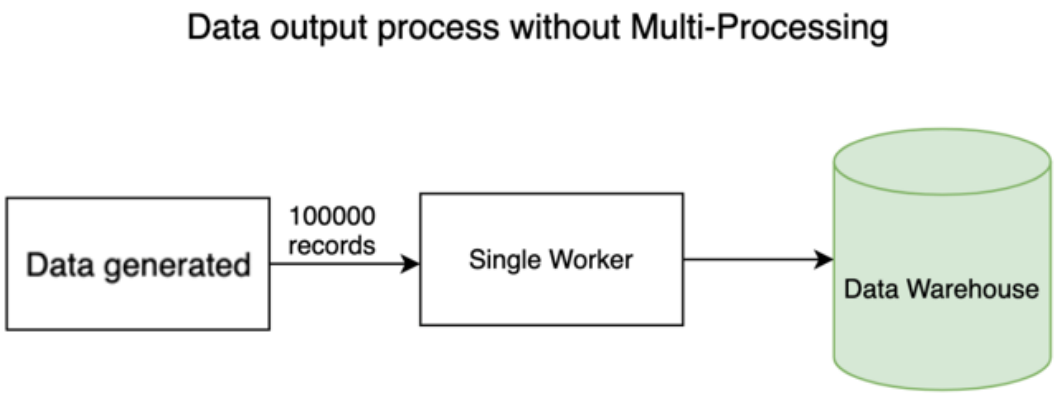


Ilustración 16. Ejecución de tareas sin multiprocesamiento [8]

Para solventar este cuello de botella, hemos aplicado los conceptos del multiprocesamiento a nuestras funciones en Python, de manera que se dividide la carga de trabajo en diferentes segmentos de igual tamaño, y se ejecuta simultáneamente todos los procesos para aprovechar al máximo todos los recursos de los que disponemos, y para reducir drásticamente los tiempos de ejecución necesarios a todos los niveles del proyecto. Desde la extracción de texto al parsear las noticias una a una, hasta la aplicación de los modelos finales sobre el dataset conjunto de noticias correspondientes a los 8 periodicos

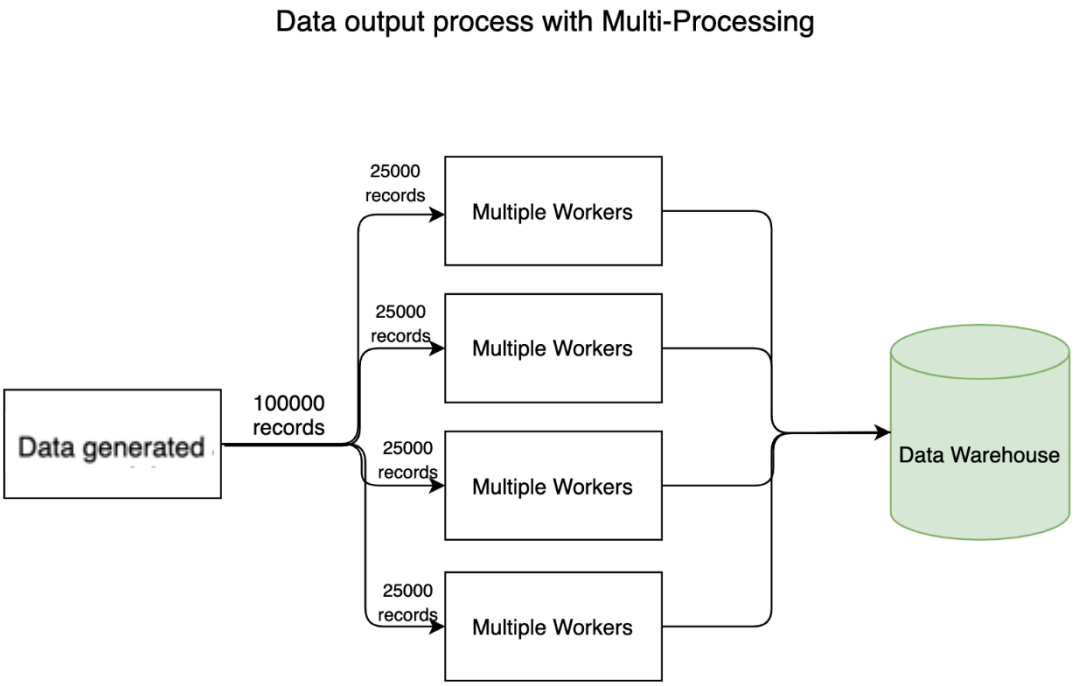


Ilustración 17. Ejecución de tareas con multiprocesamiento [8]

4.2.6 Modelo de Extracción de Categoría

Una vez generado el dataset de noticias limpias y con texto disponible, nos disponemos a generar el primer algoritmo de nuestro proyecto. El propósito de este primer algoritmo será de devolver la categoría a la que pertenece una noticia una vez consumida y analizada.

Recordando el estudio sobre el estado del arte realizado anteriormente, observamos que existen dos corrientes a la hora de enfocar un problema de detección de categoría, o "Topic Modeling". Debido a que estas corrientes proponen dos enfoques muy diferentes, vamos a realizad varios modelos pertenecientes a ambas ramas, para quedarnos con aquel que ofrezca mejores resultados y capacidad de comprender las noticias.

La detección de categoría es una técnica para extraer temas ocultos de grandes volúmenes de texto, para la extracción de estos temas nuestro primer modelo va a consistir en un algoritmo de aprendizaje automático no supervisado conocido como LDA (Latent Dirichlet Allocation).

LDA fue desarrollado por primera vez por David Blei, Andrew Ng, y Michael Jordan, quienes juntos expusieron su modelo en Blei et al. (2003). LDA es un modelo probabilístico generativo similar a Naive Bayes, cuyo funcionamiento en rasgos generales se puede explicar cómo representar los temas aparentes en el texto mediante las probabilidades de sus palabras, permitiendo así descubrir temas presentes u ocultos,

ya que agrupa las palabras en función de su coocurrencia en un documento o en nuestro caso artículo periodístico respectivo.[9]

El primer paso para implementar este modelo en nuestro proyecto, es el procesamiento y limpieza de los datos, a pesar de que en los apartados anteriores ya trabajamos este aspecto de nuestro dataset, ahora hemos de trabajar los aspectos que se aplican al problema que estamos enfocando, la detección de categoría. En primer lugar observamos la distribución de categorías por periódico, y observamos que existen demasiadas categorías, y que muchas de ellas no comprenden ni siquiera el 1% de los artículos totales del periódico.

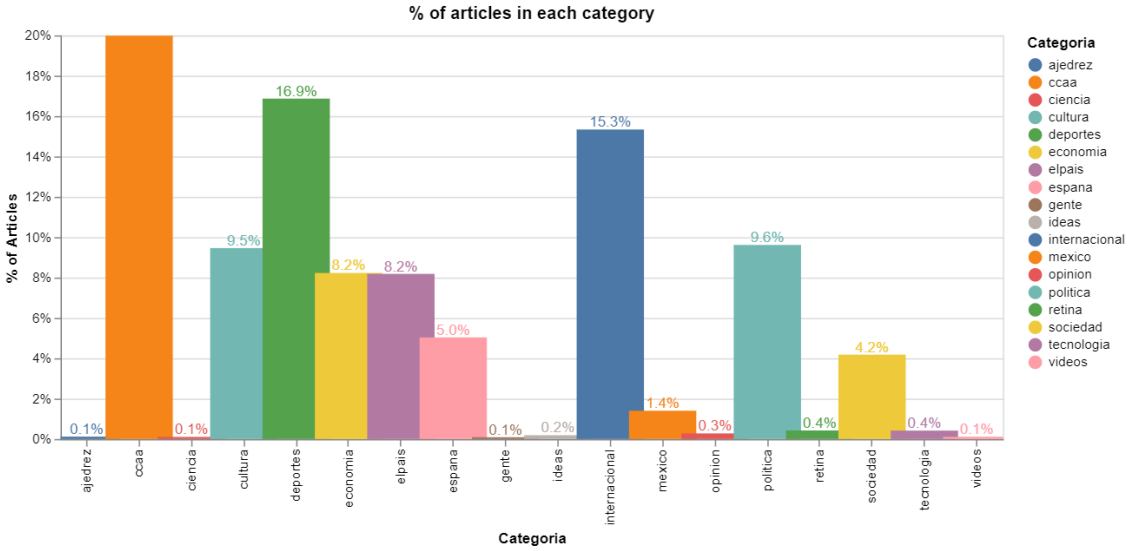


Ilustración 16 Distribución de categorías más relevantes en el periódico El País

Índice	Categoría	Recuento	Porcentaje Total
5	catalunya	1	0.000015
26	loteria-de-navidad	2	0.000030
30	navidad	2	0.000030
23	icon-design	3	0.000045
27	loteria-del-nino	3	0.000045
38	ser	4	0.000060
19	estilo	5	0.000075
9	comunicacion	5	0.000075
20	extra	6	0.000090

34	podcasts	6	0.000090
4	buenavida	8	0.000120
16	eps	12	0.000180
33	planeta-futuro	14	0.000210
32	pandora-papers	15	0.000225
37	revista-de-verano	16	0.000240
22	icon	16	0.000240
28	masterdeperiodismo	18	0.000270
12	diario	24	0.000359
3	babelia	26	0.000389
1	america	33	0.000494
17	escaparate	39	0.000584
14	educacion	45	0.000674
8	clima-y-medio-ambiente	49	0.000734
41	television	73	0.001093
21	gente	88	0.001318
7	ciencia	101	0.001512
42	videos	111	0.001662
0	ajedrez	116	0.001737
24	ideas	185	0.002770
31	opinion	274	0.004103
40	tecnologia	412	0.006169
36	retina	418	0.006259
29	mexico	1386	0.020753
39	sociedad	4135	0.061914
18	espana	4979	0.074552

15	elpais	8095	0.121208
13	economia	8139	0.121867
10	cultura	9358	0.140119
35	politica	9511	0.142410
25	internacional	15169	0.227128
11	deportes	16683	0.249798
6	ccaa	19768	0.295990

Como observamos en la tabla y gráfica anteriores, existen muchas categorías poco relevantes en los periódicos, y tratar de comprender y explicar todas estas categorías cuando contienen tan pocos artículos conlleva a un modelo muy poco preciso. Por este motivo vamos a entrenar el modelo de categorías utilizando solo las categorías más relevantes. Estas categorías se eligen teniendo en cuenta, el número de artículos que contienen, la relevancia de ellas dentro de un periódico especifico, y la existencia de esta categoría en varios periódicos.

Tras realizar este análisis concluimos con que el modelo se entrenará y categorizará en base a las siguientes categorías: CCAA, cultura, deportes, economía, empresas, España, internacional, noticias, política, y sociedad.

A continuación comenzamos el preprocesado del texto de las noticias pertenecientes a las categorías relevantes.

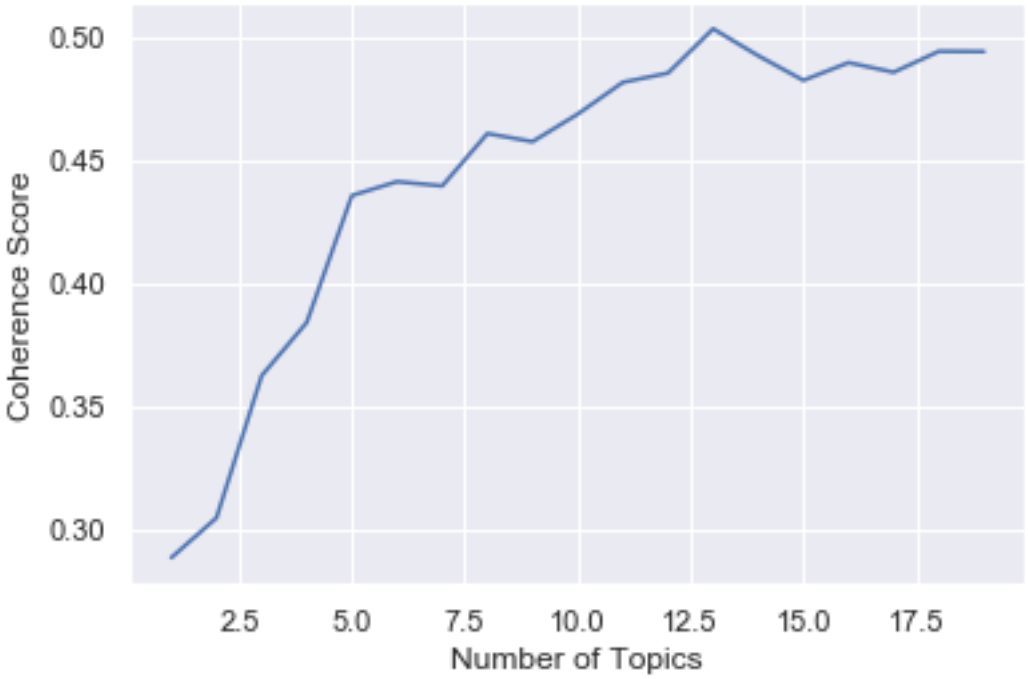
1. Nuestro primer paso consiste en romper el texto de cada noticia en frases, y estas frases convertirlas en tokens
2. A continuación, utilizaremos la librería SpaCY para poder asignar a cada token su análisis morfológico, identificando verbos, adjetivos, pronombres y otros elementos sintácticos.
3. Una vez asignados los elementos sintácticos, procedemos a normalizar el texto de manera que trabajemos con las mismas estructuras a lo largo de los artículos, y que las palabras escritas de manera diferente pero proporcionen la misma información, sean identificadas de la misma manera. Para conseguir esto convertimos todo a minúscula, rompemos las frases en palabras(tokenizar), y lematizamos. Lematizar consiste en reducir todas las palabras a su raíz original, de modo que las palabras “diré” y “dijéramos” ambas serán convertidas en “decir” y así contabilizar adecuadamente el uso y relevancia de cada palabra.

4. Eliminamos de cada frase los siguientes elementos adverbios, pronombres, conjunciones, elementos de puntuación, espacios, números y símbolos.

```
Out[8]: 0       [popular, anunciar, cnmv, resultado, oferta, r...
        1       [google, apple, sacar, provecho, quiebra, koda...
        2       [dejar, situación, económico, despacho, líder,...
        3       [gobierno, asegurar, imponer, reforma, recorte...
        4       [banco, británico, barclays, mantener, convers...
                                       ...
        7895    [concejal, delegado, presidencia, modernizació...
        7896    [flexión, ejercicio, básico, hora, entrenamien...
        7897    [deporte, hábito, saludable, debeer, incorpora...
        7898    [servir, proteger, paso, tiempo, convertir, fi...
        7899    [decena, persona, manifestar, domingo, brusela...
```

Ilustración 17. Visualización del pre procesado sobre un primer conjunto de noticias

5. Analizamos cuantas veces aparece cada palabra dentro del texto y generamos un diccionario que contenga un listado de todas las palabras y el número de veces que aparecen para cada texto.
6. Los tokens que aparecen en más del 50% del diccionario total se eliminan debido a que no aportan información relevante sobre ninguna categoría particular.
7. Estudiamos la distribución de palabras en cada categoría , así como la distribución de categorías en cada noticia para comprender que palabras definen una categoría aplicando el concepto de LDA explicado anteriormente.
8. Basándonos en los algoritmos de John Mclevey Se calcula una puntuación que mide el grado de similitud semántica entre las palabras con mayor relevancia sobre cada tema. Para ello utilizamos los algoritmos C_umass y C_v, estos algoritmos devuelven La puntuación de coherencia para C_v va de 0 (incoherencia completa) a 1 (coherencia completa), y valores negativos para C_umass, por lo que valores superiores a 0.5 ya presentan significación estadística.[38]



9.

Ilustración 20 Grafico Generado con el modelo C_V

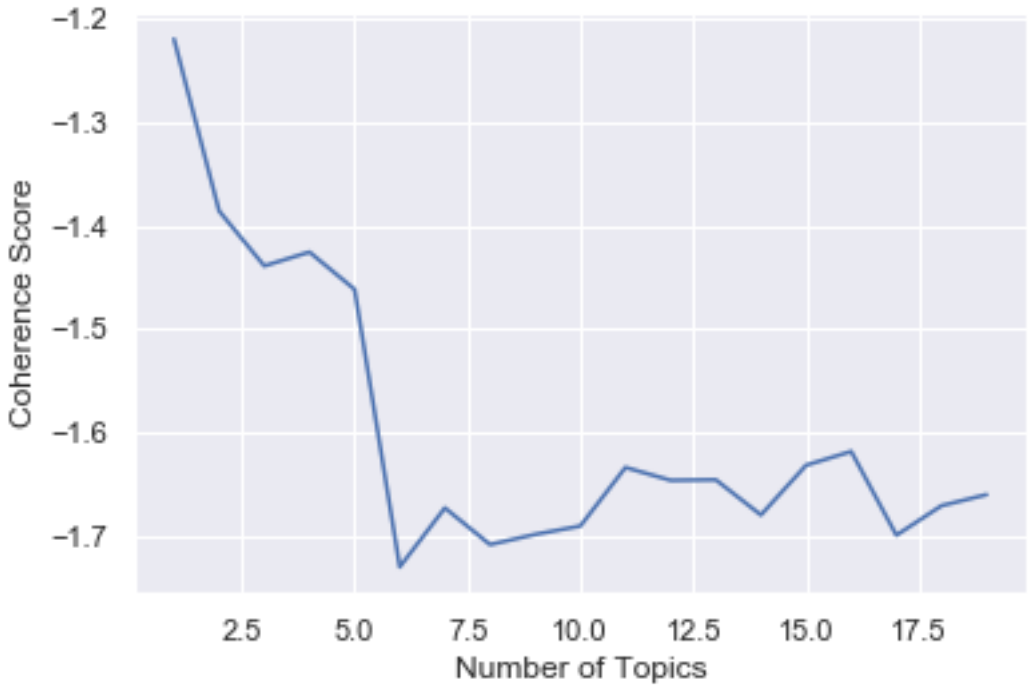


Ilustración 21.8 Grafico Generado con el modelo C_umass

Observando los resultados de ambos modelos, en particular los picos de C_ummas en los valores 6 y 17, y los picos de C_v en los valores 13 y 18, podemos concluir con

certeza que existen alrededor de 17 temas relevantes distintos en nuestro listado de noticias original.

A continuación devolvemos el listado de palabras que nuestro modelo considera que define cada categoría. Estos resultados constituyen un listado de 17 “categorías” enumeradas del 1-17, y las palabras más relevantes junto a su probabilidad de aparecer en la categoría seleccionada. Para interpretar estos resultados hemos de entender que cada uno de los 17 listados de palabras corresponde a una “categoría” entendida por el algoritmo, pero necesitaríamos a un experto sobre la prensa para poder correctamente definir que listado de palabras corresponde a que categorías existentes en la prensa actual.

```
  '0.065*"euros" + 0.059*"millón" + 0.025*"mes" + 0.016*"gasto" + 0.013*"dato" + 0.012*"empleo" + 0.012*"comunidad" + 0.012*"im
puesto" + 0.011*"vivienda" + 0.010*"cifra"'),
 (1,
  '0.032*"policía" + 0.023*"mujer" + 0.018*"persona" + 0.018*"civil" + 0.016*"guardia" + 0.016*"víctima" + 0.015*"agente" + 0.0
13*"hombre" + 0.012*"violencia" + 0.011*"menor"'),
 (2,
  '0.050*"españa" + 0.042*"país" + 0.029*"español" + 0.027*"europeo" + 0.023*"gobierno" + 0.018*"europa" + 0.017*"ministro" +
0.017*"europea" + 0.014*"unión" + 0.012*"unido"'),
 (3,
  '0.015*"público" + 0.014*"trabajo" + 0.012*"empresa" + 0.012*"servicio" + 0.012*"proyecto" + 0.011*"social" + 0.011*"gobiern
o" + 0.010*"trabajador" + 0.008*"plan" + 0.008*"medida"'),
 (4,
  '0.029*"madrid" + 0.026*"josé" + 0.023*"juan" + 0.020*"premio" + 0.019*"carlos" + 0.017*"maría" + 0.016*"españa" + 0.015*"re
y" + 0.015*"garcía" + 0.014*"director"'),
 (5,
  '0.037*"ciudad" + 0.029*"zona" + 0.020*"ayuntamiento" + 0.020*"hora" + 0.016*"obra" + 0.015*"agua" + 0.015*"metro" + 0.012*"c
entro" + 0.012*"municipal" + 0.012*"calle"'),
 (6,
  '0.013*"sistema" + 0.012*"forma" + 0.011*"poder" + 0.010*"marca" + 0.010*"permitir" + 0.009*"tiempo" + 0.009*"tipo" + 0.009
*"usuario" + 0.009*"dato" + 0.009*"modelo"'),
 (7,
  '0.029*"gobierno" + 0.027*"pp" + 0.025*"partido" + 0.018*"psoe" + 0.016*"sánchez" + 0.013*"socialista" + 0.011*"político" +
0.011*"presidente" + 0.009*"elección" + 0.009*"congreso"'),
 (8,
  '0.028*"país" + 0.026*"presidente" + 0.016*"político" + 0.015*"gobierno" + 0.012*"unidos" + 0.010*"elección" + 0.009*"milita
r" + 0.009*"partido" + 0.009*"estadounidense" + 0.008*"líder"'),
 (9,
  '0.014*"vida" + 0.011*"ver" + 0.011*"querer" + 0.010*"mundo" + 0.010*"tiempo" + 0.009*"pasar" + 0.009*"cosa" + 0.008*"histori
a" + 0.008*"tener" + 0.008*"dejar"'),
 (10,
  '0.030*"caso" + 0.023*"salud" + 0.021*"persona" + 0.020*"sanitario" + 0.018*"pandemia" + 0.018*"coronavirus" + 0.017*"comunid
ad" + 0.016*"semana" + 0.015*"sanidad" + 0.014*"hospital"'),
 (11,
  '0.079*"barcelona" + 0.072*"catalán" + 0.057*"cataluña" + 0.047*"generalitat" + 0.041*"valido" + 0.022*"político" + 0.019*"es
paña" + 0.015*"derecho" + 0.014*"manifestación" + 0.012*"constitución"'),
 (12,
  '0.043*"banco" + 0.023*"mercado" + 0.019*"punto" + 0.016*"entidad" + 0.015*"precio" + 0.015*"financiero" + 0.013*"deuda" + 0.
013*"millón" + 0.013*"economía" + 0.012*"valor"'),
 (13,
  '0.030*"partido" + 0.029*"equipo" + 0.020*"madrid" + 0.018*"real" + 0.017*"jugador" + 0.016*"minuto" + 0.015*"jugar" + 0.014
*"juego" + 0.013*"gol" + 0.013*"punto"'),
 (14,
  '0.055*"millón" + 0.038*"empresa" + 0.032*"euros" + 0.032*"compañía" + 0.017*"grupo" + 0.017*"negocio" + 0.016*"venta" + 0.01
4*"mercado" + 0.012*"operación" + 0.012*"producto"'),
 (15,
  '0.196*"electrónico" + 0.191*"correo" + 0.074*"noticia" + 0.071*"nombre" + 0.068*"enviar" + 0.066*"guardar" + 0.062*"actualiz
ado" + 0.023*"abc" + 0.012*"españa" + 0.011*"madrid"'),
 (16,
```

Ilustración 22 Resultado del modelo LDA

Aprendizaje Supervisado

Nuestro segundo modelo para categorizar una noticia se corresponde a un enfoque de aprendizaje supervisado. Debido a la dificultad técnica de interpretar adecuadamente los resultados del modelo anterior sin la ayuda de un experto, nos proponemos enfrentar el problema esta vez con el enfoque de aprendizaje supervisado. Para ello utilizaremos las categorías previamente extraídas en la sección de Extracción de texto, y las utilizaremos para enseñar a nuestro algoritmo a entender que textos pertenecen a cada categoría.

Como mencionamos anteriormente, debido al amplio número de categorías existentes en el dataset, hemos reducido el número de muestras a utilizar, a tan solo aquellas pertenecientes a las categorías más presentes y relevantes en los periódicos, siendo estas CCAA, cultura, deportes, economía, empresas, España, internacional, noticias, política, y sociedad.

Preparación de los datos

A la hora de enfrentar un problema de Machine Learning mediante el uso de aprendizaje supervisado, es crucial asegurarse que el dataset utilizado contenga datos balanceados. Esto significa que ha de existir una proporción similar para todas las categorías sobre las que aprender, para evitar que el modelo aprenda erróneamente. Esto es sencillo de entender mediante el siguiente ejemplo. Imaginemos que nuestro dataset se compone de tres categorías, y la distribución de datos equivale a la siguiente.

categoría 1 =90% , categoría 2= 7% , categoría 3 =3%. Si nuestro modelo no aprende nada y tan solo predice categoría 1 para todos los casos, la precisión de el modelo seria del 90% lo que aparentaría ser una puntuación muy buena, sin embargo debido a la mala distribución de los datos originales, este algoritmo fallara con certeza a la hora de implementarlo en un entorno real.

Tras analizar la distribución original de nuestros datos, equivalentes a las 8 categorías seleccionadas, observamos una clara falta de balanceo entre las diferentes categorías más representativas.

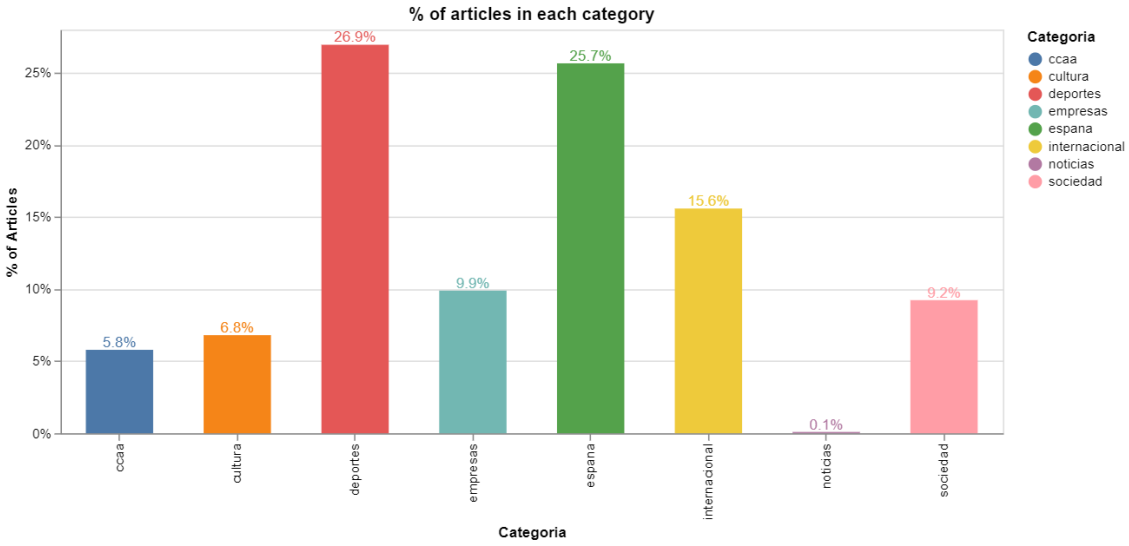


Ilustración 23 Distribución de las categorías en % para todos los periódicos combinados

Para solucionar esta disparidad entre los datos, vamos a aplicar técnicas de remuestreo para ofrecer a todas las categorías el mismo peso. Comenzamos por realizar un "downsampling" sobre las categorías mayoritarias, "deportes" y "España", esto implica reducir el número de registros existentes en esta categoría para reducir el peso que tienen en la proporción final.

A continuación aplicamos el proceso contrario “Upsampling” sobre las clases minoritarias, lo cual implica realizar un remuestreo con reemplazo para aumentar el número de casos existentes en las categorías, y de este modo aumentar la proporción de noticias de la categoría imputada.

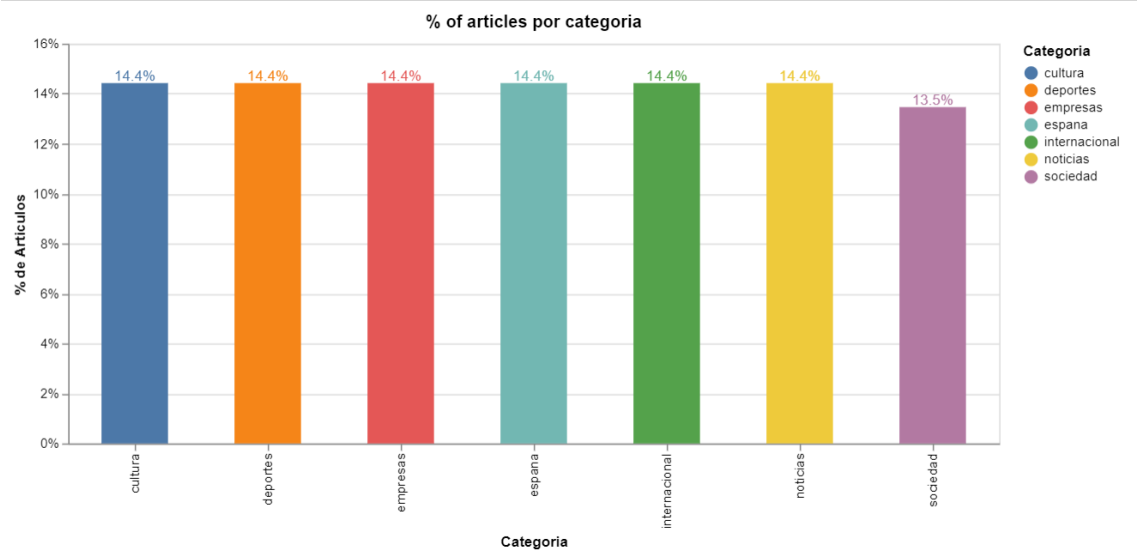


Ilustración 24. categorías Balanceadas mediante upsample y downsample

Tras balancear la distribución de las categorías, analizamos también la longitud de los artículos por cada categoría para observar si también tiene un impacto relevante a la hora de determinar la categoría. Dibujando los diagramas de cajas observamos que no existe grandes diferencias entre las categorías excepto para los artículos de empresa, cuya media se encuentra por debajo del resto de categorías.

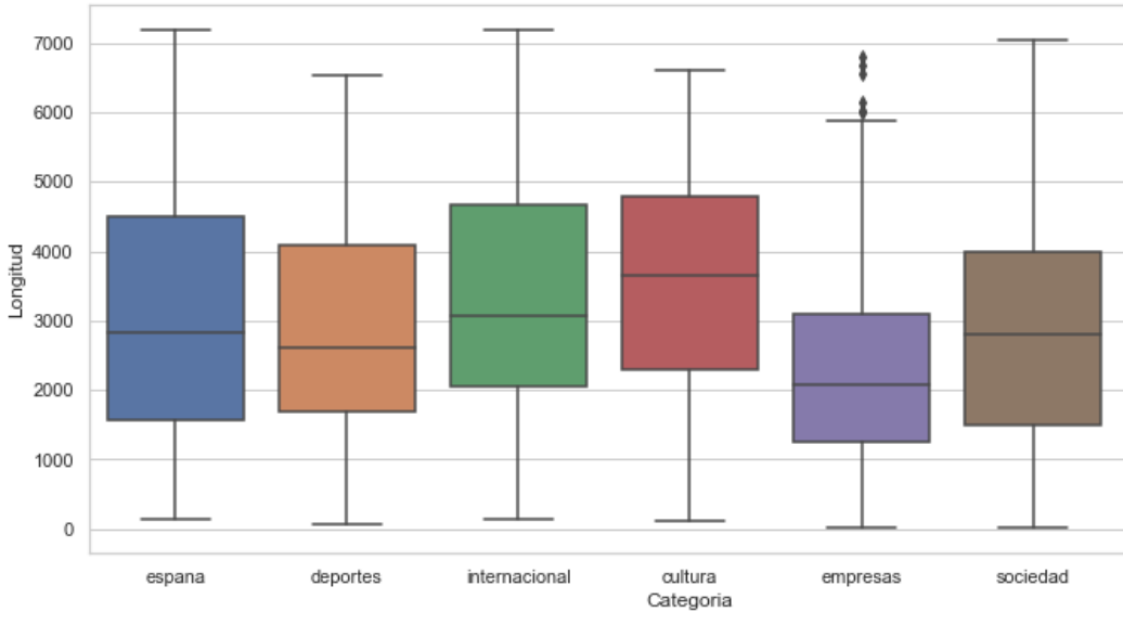


Ilustración 25. Diagrama de Cajas, longitud de artículos por categoría

A continuación trabajamos sobre el texto presente en las noticias, aplicando una limpieza común en proyectos sobre texto, para garantizar la calidad del texto extraído, y la comprensión de nuestro modelo sobre dicho texto. Para ello realizamos las siguientes operaciones:

- Limpieza de caracteres especiales: los caracteres especiales, como las comillas dobles, saltos de línea "\n", o caracteres como la arroba han sido eliminados del texto debido a que no aportan información relevante sobre la noticia.
- Se ha convertido todo el texto a minúsculas para no diferenciar entre las palabras que comienzan con la primera letra en mayúscula.
- Signos de puntuación: se han eliminado caracteres como "?", "!", ";".
- Lematizamos el texto. Lematizar consiste en reducir todas las palabras a su raíz original, de modo que las palabras “diré” y “dijéramos” ambas serán convertidas en “decir” y así contabilizar adecuadamente el uso y relevancia de cada palabra.
- Palabras de parada (stopwords): palabras como "qué" o "el" serán comunes a todos los textos, por lo que pueden representar ruido y reducir la comprensión de nuestro modelo final, por lo que se han eliminado.

Una vez finalizada la limpieza de texto, y de manera similar a la alternativa No supervisada, a la hora de definir las características de nuestro conjunto de datos, vamos a utilizar el conteo de palabras en un texto mediante TF e IDF.

TF-IDF es una puntuación que representa la importancia relativa de un término en el documento o en nuestro caso una noticia específica, en relación con todo el conjunto de noticias existentes (corpus) . TF significa Term Frequency (frecuencia de términos), e IDF significa Inverse Document Frequency (frecuencia inversa de documentos):[10]

$$TFIDF(t,d) = TF(t,d) \times \log\left(\frac{N}{DF(t)}\right)$$

Being:

- t: term (i.e. a word in a document)
- d: document
- $TF(t)$: term frequency (i.e. how many times the term t appears in the document d)
- N: number of documents in the corpus
- $DF(t)$: number of documents in the corpus containing the term t

Ilustración 26 Formula de cálculo para la Frecuencia de términos (TF)

El valor del TF-IDF aumenta proporcionalmente al número de veces que aparece una palabra en el documento y se compensa con el número de documentos del corpus que la contienen, lo que ayuda a ajustar el hecho de que algunas palabras aparecen con más frecuencia por la mera manera de redactar y transmitir las ideas del artículo.

También tiene en cuenta el hecho de que algunos documentos pueden ser más grandes que otros al normalizar el término TF (expresando en su lugar las frecuencias

relativas de los términos). Una gran ventaja de utilizar TF-IDF es La sencillez del modelo, en conjunción con los buenos resultados a la hora de aplicarlo en problemas de este dominio, motivo por el cual lo aplicamos tanto para aprendizaje supervisado como no supervisado. Además de ser un modelo sencillo, TF-IDF es rápido a la hora de obtener resultados, lo que ya hemos demostrado que en nuestro caso es un elemento vital. Finalmente un aspecto con gran importancia a la hora de elegir este modelo ha sido la capacidad de modificar los parámetros como número de palabras a estudiar, frecuencia mínima o máxima de palabras existentes, y otros componentes que nos permitirán personalizar el modelo para nuestro conjunto de datos.

A la hora de definir los parámetros del modelo, se ha ido probando a modificar distintos valores hasta alcanzar un resultado óptimo, por ello los parámetros más relevantes que se han elegido en el algoritmo final son:

- Valor de Frecuencia de documento =15, esto indica que si una palabra aparece en menos de 15 noticias de su categoría, no se tiene en cuenta como elemento relevante.
- Numero de documentos= 100%, esto implica que se tienen en cuenta todas las noticias de una categoría a la hora de analizar una categoría especifica, en el caso de poseer información sobre los artículos más leídos de una categoría, se podría modificar este valor para considerar los más relevantes.
- Características =450, esto indica el número de palabras que tiene en cuenta para definir una categoría, este parámetro ha de ser modificado con cautela, porque incluir un numero demasiado alto llevaría a un sobreajuste sobre las palabras presentes en NUESTRO dataset, y no aprender adecuadamente valores fuera de estos.

Una ventaja adicional sobre la elección de TF-IDF procede de la naturaleza del propio modelo, donde cuanto mayor sea el número de artículos que tengamos para generar tokens y comprender las palabras relevantes, más probable será que las palabras que aparezcan en una noticia nunca vista estén incluidas en nuestro listado de tokens. Debido a la gran cantidad de datos extraídos al analizar 10 años de noticias, esto es una ventaja para nuestro modelo.

Una vez generado el vector TF-IDF y contabilizadas las palabras únicas y sus ocurrencias, tan solo nos falta determinar qué modelo de Machine Learning utilizar para predecir la categoría. Llegados a este punto, nos encontramos ante la decisión de utilizar modelos convencionales, o análisis profundo (“Deep Learning”) para determinar la categoría del artículo. Debido al alto tiempo de entrenamiento de los modelos de análisis profundo, y a los buenos resultados conseguidos mediante modelos convencionales, no desarrollaremos Deep learning en este primer modelo.

Para determinar qué modelo utilizar a la hora de Categorizar, hemos entrenado y evaluado los siguientes modelos con nuestros datos.

1. Una regresión logística multinomial
2. Random Forest
3. SVM (Maquina de Soporte Vectorial

4. K Vecinos Cercanos

Para cada uno de estos modelos, se han importado las librerías de Python que los definen, se han modificado los hiper parámetros para obtener el mejor resultado posible, se han entrenado y evaluado con un proceso de validación cruzada, y se han evaluado los resultados del modelo junto con una matriz de confusión para comprender el desempeño en cada categoría individual.

A la hora de evaluar el desempeño de los modelos uno con otros, hemos utilizado la precisión (Accuracy) como métrica de rendimiento, debido a que otras medidas como el Recall y la puntuación F1 (F1- Score) tienen en cuenta y contabilizan individualmente los falsos positivos y falsos negativos, mientras que en nuestro caso no distinguimos entre ellos debido a que solo nos interesa si categoriza bien o no.

Tras completar los pasos anteriores de todos los modelos obtenemos la siguiente tabla de resultados

	Model	Training Set Accuracy	Test Set Accuracy
2	Random Forest	0.876868	0.868805
0	KNN	0.876868	0.842566
1	Logistic Regression	0.821741	0.819242
3	SVM	0.774343	0.720117

Ilustración 27. Tabla de resultados para modelos de predicción de categoría

En ella observamos que el modelo con mejor desempeño tanto en los datos de entrenamiento, como en los de testeo es el **modelo Random Forest**. Este Es el modelo final que utilizaremos para determinar la categoría a la que pertenece un articulo.

Para acabar de comprender el desempeño de nuestro modelo sobre cada categoría observamos su matriz de confusión y su desempeño en la clasificación, donde podemos destacar que el modelo se equivoca muy poco a la hora de clasificar la categoría, destacando que no falla en las categorías de España, cultura, o noticias, y encuentra mayor dificultad a la hora de categorizar las noticias de sociedad, confundiéndolas con las noticias de tipo empresas.

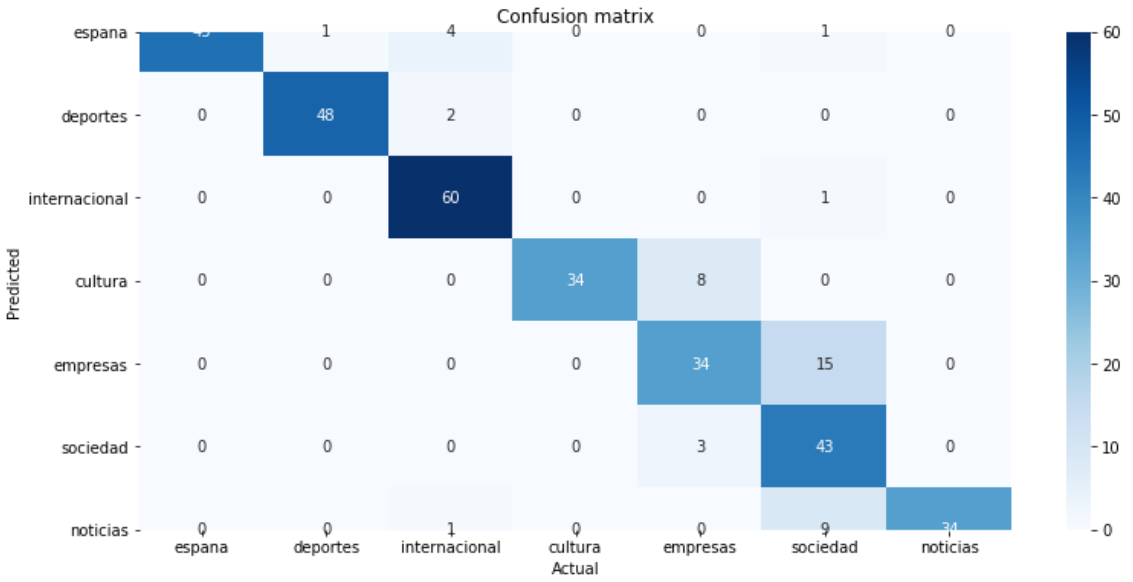


Ilustración 28. Matriz de confusión del desempeño del modelo Random Forest

4.2.7 Modelo de Resumen de Texto

A continuación procedemos a realizar un modelo capaz de consumir el texto presente en una noticia, y devolvernos una versión simplificada del mismo, en un número menor de líneas y presentando la información más relevante presentada en el texto original.

Para ello una vez más nos encontramos ante la opción de realizar un modelo de análisis profundo (Deep Learning) para aprovechar la alta capacidad de comprensión de estos, y la existencia de varios millones de datos presentes en nuestro dataset que permitirían alimentar estos modelos. O por el contrario, podemos optar por un método más sencillo con un coste de tiempo menor, a costa de la precisión obtenida al final. Debido a la importancia del elemento resumen en nuestro proyecto, a pesar del coste de tiempo que supone entrenar modelos de Deep Learning, hemos optado por desarrollar una red neuronal Convolucional para obtener el mejor resultado posible.

Al igual que con cualquiera de los modelos anteriores, nuestro primer paso para conseguir un modelo de Machine Learning consiste en analizar los datos y asegurar que son óptimos para nuestro algoritmo. Comenzamos por revisar que no existan datos faltantes en el Texto, ni que existan noticias duplicadas. Acto seguido realizamos las siguientes transformaciones sobre todos los Textos de todas las noticias almacenadas:

- Convertimos todo el texto a minúsculas para evaluar de la misma manera las palabras que tienen la primera letra en mayúscula, y las que no.
- Eliminamos cualquier texto perteneciente a etiquetas html
- Eliminamos los caracteres extraños encontrados a lo largo de las noticias, y que no proporcionan información aparente para nuestro modelo, esto pueden ser saltos de línea, paréntesis, o comillas

- Continuamos normalizando el texto mediante la eliminación de signos de puntuación y de exclamación
- Eliminamos las palabras que no aportan valor al texto (stopwords)
- Finalmente rompemos el texto original en frases.

Cabe mencionar que en este modelo no queremos lematizar las palabras como en el modelo de Categoría, ya que nuestro interés es que cada frase y palabra aporte su significado y forma verbal original, para explicar el mensaje que conllevaba.

Tras finalizar la limpieza de los datos, hemos de desarrollar el modelo para poder generar un resumen, este modelo se ha planteado de manera recibirá como entrada una o varias frases, y devolverá como salida una predicción compuesta por la secuencia de frases más relevantes. Para ello vamos a modelar el proceso con una estructura codificador-decodificador, donde la entrada y la salida poseen diferentes longitudes.

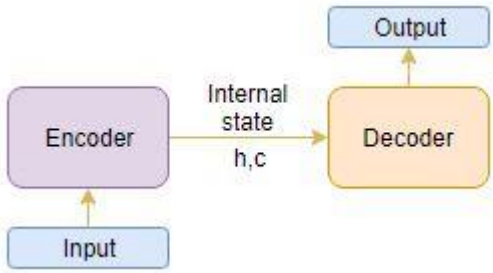


Ilustración 9. Esquema de Codificador- Decodificador

Para nuestro codificador y decodificador vamos a construir una Red Neuronal LSTM (Long Short Term Memory), ya que este tipo de red neuronal posee la ventaja de ser capaz de capturar las dependencias a largo plazo, y también evitan la fuga de gradiente que ocurre en otro tipo de redes neuronales. Es decir, evita que al tener varias capas en la red neuronal, las capas inferiores actualicen la importancia de las palabras ya procesadas a una velocidad pequeña, aumentando de esta manera el tiempo de entrenamiento de la red.

Un modelo de memoria a largo plazo como codificador (LSTM) funciona de manera que lee toda la secuencia de entrada como primer proceso, a continuación, en cada paso de tiempo, se introduce una palabra de la secuencia de texto seleccionada en el codificador. Después, se procesa la información en cada paso de tiempo y se captura la información contextual presente en la secuencia de entrada.

El Decodificador se compone por otra red neuronal LSTM diferente, que también ingesta la secuencia original decidida palabra por palabra. Una vez seleccionada la primera palabra de la secuencia trata de predecir la misma línea de texto desplazada un paso de tiempo, de modo que intenta predecir la siguiente palabra de la frase, dada la palabra anterior.

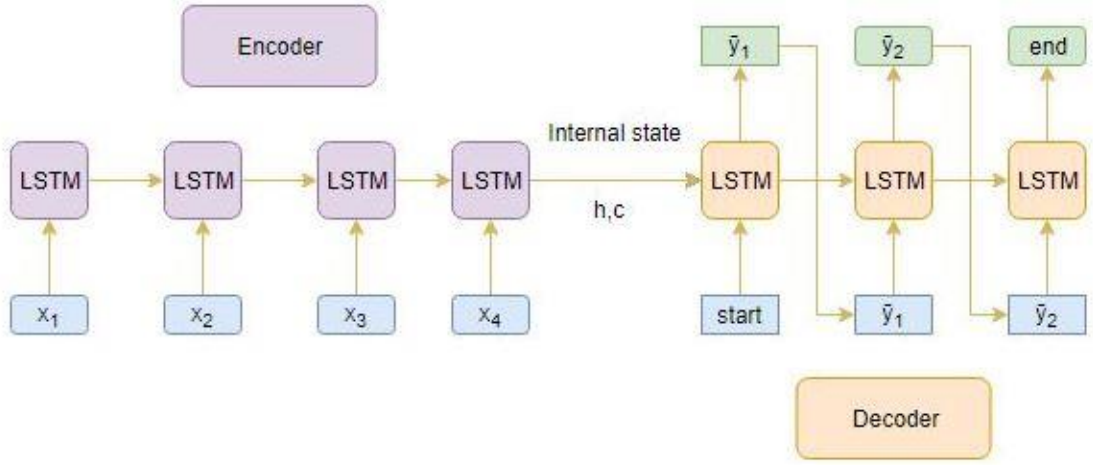


Ilustración 30. Esquema orientativo Encoder-Decoder LSTM

Una vez definido que la red neuronal que queremos utilizar será de tipo LSTM, importamos las librerías necesarias para construirla, y definimos los parámetros del modelo más importantes, estos son los siguientes:

- return_sequences=True: Se devuelve en todo momento el estado de las capas ocultas de la red con la información acerca del número de elementos analizados, el número de saltos de tiempo realizados, y el tamaño final.
- Batch size: Indica el número de datos incluidos en cada capa para analizar
- Epoch size: Indica el número de veces que se va a entrenar cada capa con tamaño batch.

Tras definir nuestra Red neuronal, la entrenamos con los datos recolectados y procesados sobre las noticias de los periódicos, y observamos la evolución del error mediante la función “categorical_crossentropy” esta función mide la diferencia de probabilidad entre la palabra predicha y la palabra que viene a continuación, de manera que el modelo devuelve la palabra más probable para continuar la frase, y la función contrasta el valor de la probabilidad, cuanto más cercana a 1 mejor. Finalmente, se devuelve la frase con mayor probabilidad de contener la información relevante, considerando todas las frases previas analizadas.

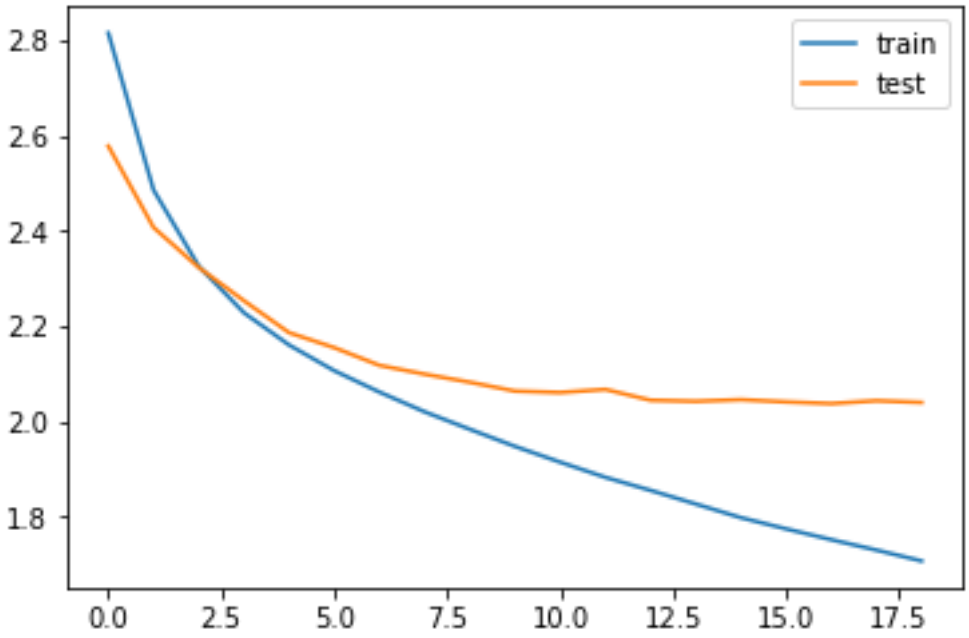


Ilustración 31. Evolución del error de la Red Neuronal

Tras medir la evolución del error a lo largo del entrenamiento y los ciclos Epoch, decidimos realizar 15 ciclos de entrenamiento, ya que el error no se reduce significativamente pasado este valor.

Al concluir nuestra Red neuronal y desarrollar la capacidad de generar resúmenes en base a texto, hemos de validar como de buenos son los resúmenes generados. Para ello vamos a utilizar unas métricas conocidas como ROUGE.

ROUGE es un conjunto de métricas para evaluar el resumen automático de textos y las traducciones automáticas . Existen diversas variantes de la métrica ROUGE; sin embargo, la idea básica que subyace es la de asignar una puntuación numérica a un resumen que nos indique lo bueno que es en comparación con uno o varios resúmenes de referencia.[16]

ROUGE compara los n-gramas (conjunto de n palabras consecutivas) del resumen generado con los n-gramas de los textos originales. Por ejemplo, Rouge-2 cojera pares de 2 palabras consecutivas en las frases del resumen y los contrasta contra los pares de palabras presentes en el texto original, calculando el número de pares presentes en el resumen que también aparecen en el texto para decidir un porcentaje de acierto. Para calcular Rouge utilizamos tres métricas principales,

- Recall: cuenta el número de n-gramas que aparecen en el resumen del modelo y en el texto original y luego divide este número por el número total de n-gramas en el texto original, esto busca determinar si el resumen contiene toda la información del texto original.
- Precisión: : de manera similar, cuenta el número de n-gramas que aparecen en el resumen del modelo y en el texto original y luego divide este número por el

número total de n-gramas en el resumen, esto premia que tengamos pocos n-gramas y que sean precisos, para evitar simplemente generar frases en el resumen para cubrir todos los casos del texto original.

- Puntuación F1: Combina las dos métricas anteriores para otorgar igual relevancia a la hora de determinar le modelo a seleccionar, se calcula mediante la formula F1 Score = 2*(Recall * Precisión) / (Recall + Precisión)

Tras aplicar ROGUE sobre todos nuestros resúmenes, y obtener la media ponderada de todos los resultados ROGUE llegamos a la siguiente tabla de información

```
[{'rouge-1': {'r': 0.7368421052631579,
   'p': 0.7777777777777778,
   'f': 0.7567567517604091},
  'rouge-2': {'r': 0.5, 'p': 0.5294117647058824, 'f': 0.514285709289796},
  'rouge-l': {'r': 0.7368421052631579,
   'p': 0.7777777777777778,
   'f': 0.7567567517604091}}]
```

Ilustración 32. Evaluación de los resúmenes generados mediante ROUGE

Y podemos concluir que la obtención de una puntuación de F igual a 0.75 para n-gramas de tamaño uno y n gramas de tamaño extenso (frase más larga del resumen contrastada con el texto original), los resúmenes son significativos.

4.2.8 Modelo de Análisis de Sentimiento

Llegados al modelo de Análisis de Sentimiento, hemos de desarrollar un proceso que reciba como entrada un texto compuesto por diferentes frases, evalue todas ellas como un conjunto único, y nos devuelva la información sobre como el mensaje está redactado y los datos que expresa, pudiendo estos ser Positivos, Negativos o neutrales.

Para desarrollar esta clase de modelos, existen dos maneras de plantear la cuestión y enfocar el desarrollo de la solución para obtener el sentimiento. La primera corriente se enfoca en analizar las palabras individuales que componen el texto, y clasificarlas una a una según su mensaje sea positivo(ej.: Delicioso, espectacular, etc.), negativo(Repugnante, horrible, etc.) o neutral (edificio, plato, etc.). El desarrollo de estas soluciones requiere la posesión previa de un conjunto de datos que proporcione al modelo la comprensión sobre el significado y el tipo de mensaje anteriormente descrito, Debido a que no contamos con este conjunto de datos para las palabras españolas presentes en nuestras noticias, no trabajaremos esta corriente.

En su lugar el enfoque que utilizamos para desarrollar el sentimiento expresado en las noticias, consiste en evaluar todas las frases en conjunto, y determinar si el texto expresa un mensaje positivo, negativo, o neutral. Para ello trabajaremos en primer lugar los datos presentes en el dataset “SFU corpus”[17], el cual consiste en un conjunto de datos con varios miles de reseñas de diferentes productos (libros, películas, servicios, etc.) y una valoración positiva o negativa.

Tras entrenar nuestro modelo con dichos datos y obtener la capacidad de descifrar el sentimiento de un texto extenso, compuesto por varias frases, aplicaremos el modelo sobre los datos de nuestras noticias para determinar que intención se expresa en estas.

Indagando en el proceso seguido para la construcción del modelo final de sentimiento, el primer paso consiste en analizar los datos del dataset de reseñas, y limpiar el texto presente en el mediante el preprocesado que hemos estado siguiendo para todos los modelos anteriores, y que corresponde a las buenas prácticas del desarrollo de un modelo de Machine Learning.

Comenzamos por separar nuestro conjunto de datos en dos grupos. El primer grupo será el grupo de entrenamiento, sobre el cual aplicaremos las transformaciones necesarias para aprender a descifrar el sentimiento, y sobre el cual nuestro modelo final Aprende. El segundo grupo será el grupo de evaluación, y serán algunas de las reseñas del dataset original sobre las cuales no realizaremos ninguna transformación para asegurarnos que el modelo final es capaz de entender los datos presentes en el texto de la manera que vienen de su origen, lo cual será lo más similar a la información que encontraremos en nuestras noticias.

Tras separar los datos de los grupos con una distribución 80%-20% respectiva a los grupos anteriores, aplicamos los siguientes pasos para limpiar el texto:

- Tokenizamos el texto en frases
- Convertimos todo el texto a minúsculas para evaluar de la misma manera las palabras que tienen la primera letra en mayúscula, y las que no.
- Eliminamos caracteres extraños como saltos de línea, paréntesis, o comillas
- Eliminamos signos de puntuación y de exclamación
- Eliminamos las palabras que no aportan valor al texto (stopwords)

Tras limpiar el texto con los pasos anteriores, generamos dos listas para contabilizar las palabras de la reseña, cada lista cogerá todas las reseñas de un tipo (positivas o negativas) y contabilizará cuantas veces aparecen cada palabra en las reseñas para entender que palabras se repiten y son significativas en las reseñas de tipo positivo y negativo respectivamente.

Para evitar que nuestro modelo sobre aprenda al analizar todas las palabras de nuestro dataset, vamos a definir un vocabulario compuesto por el 50% de las palabras y frases más significativas de carácter positivo y el 50% de las palabras y frases más significativas de carácter negativo.

A continuación transformamos nuestras palabras en vectores de valores, de modo que las palabras utilizadas de manera similar, adquieran una representación similar y de este modo comprender mejor su significado. Esta transformación se realiza mediante el método de “Word Embedding” que “consiste en representar las palabras individuales como vectores de valor real en un espacio vectorial donde cada palabra está representada por un vector de valores compuesto de varias dimensiones. Estos valores

se contrastan con las diferentes dimensiones de las palabras dispersas, para explicarlas como la codificación de un solo punto".[18]

Una vez obtenida la representación numérica de los valores de las palabras de los textos originales, diseñamos la red neuronal que aprenderá sobre todos los vectores generados. Para ello definimos una red neuronal Convolucional con 10 bloques de filtros para procesar las palabras en paralelo y agilizar el proceso de entrenamiento, de manera similar al multiprocesamiento. Además configuramos La capa de salida con una función de activación "sigmoide" para obtener un valor entre 0 y 1 para la predicción sobre el sentimiento negativo o positivo de la reseña.

Tras entrenar la red neuronal medimos el porcentaje de acierto mediante la métrica de precisión ("Accuracy"), debido a la gran cantidad de parámetros modificables en esta red neuronal, y la búsqueda de optimizar nuestra red para conseguir la mejor precisión posible, entrenamos la red varias veces con diferentes parámetros y los almacenamos en una tabla para evaluar que configuración resulta la más optima, como observamos en la ilustración 33

Epoch	Batch	Accuracy
8	8	81.058019
4	8	80.887371
12	5	80.716723
4	4	80.546075
6	3	80.546075
...	...	...
3	10	55.802047
2	1	43.515357
7	7	43.515357
1	2	43.344709
5	4	43.344709

Ilustración 33. Desempeño de Redes Neuronales modificando los parámetros.

Tras medir el desempeño y comprender los parámetros de la Red Neuronal, nos decantamos por esta red neuronal Recurrente, con activación Sigmoide para devolver la representación de noticias positivas como 0 y negativas como 1, cantidad bloques entre los que dividir los datos procesados en cada ciclo(batch) igual a 8, y numero de ciclos a entrenar (epoch) igual a 8, de modo que alcanzamos una precisión del 81%

4.2.9 Visualizaciones de los datos

El objetivo de la sección de visualización consiste en representar los resultados obtenidos una vez aplicados todos los modelos a las noticias, de modo que para cualquier noticia seleccionada tenemos los siguientes campos para trabajar de manera visual.

Título de la noticia	Link	Periódico	Fecha	Año	Mes	Categoría	Texto	Resumen	Sentimiento

Para ello vamos a utilizar una combinación de gráficos generados Tableau, con funciones animadas y visuales, y visualizaciones generadas mediante código en Python para aquellas funcionalidades que Tableau no nos permita. A la hora de realizar los gráficos también hemos de tener en cuenta dos grandes limitaciones, la primera es que el gran volumen de datos que poseemos suele producir el fallo de las aplicaciones de visualización tras generar gráficos con todos ellos. Sumado a esto hemos de considerar como puede observarse en el apartado de extracción de noticias, que no todos los periódicos tienen publicadas la misma cantidad de noticias. Por ejemplo el periódico El Expansion publicaba 12 noticias al día en su página web, pero en 2015 reducen este número a 7 noticias al día, esto provoca grandes desajustes a la hora de contrastar unos periódicos contra otros ya que el periódico 20 minutos posee más de 1 millon de noticias por sí solo, lo que compone un tercio del dataset.

Para solucionar estas dos problemáticas que nos surgen hemos optado por realizar un muestreo de datos estratificado, lo cual garantiza que las distribuciones presentes en el dataset original completo se mantienen, y a su vez poseemos menos datos para poder analizar si arriesgarnos a una caída del sistema.

Explorando la Categoría generada mediante el primer modelo NLP, encontramos que con el paso de los meses, el tipo de noticia más leído se encuentra en la categoría de

deportes, España, y “noticias”. Este gráfico puede animarse para ir visualizando el cambio de volumen por categoría a través de Tableau.

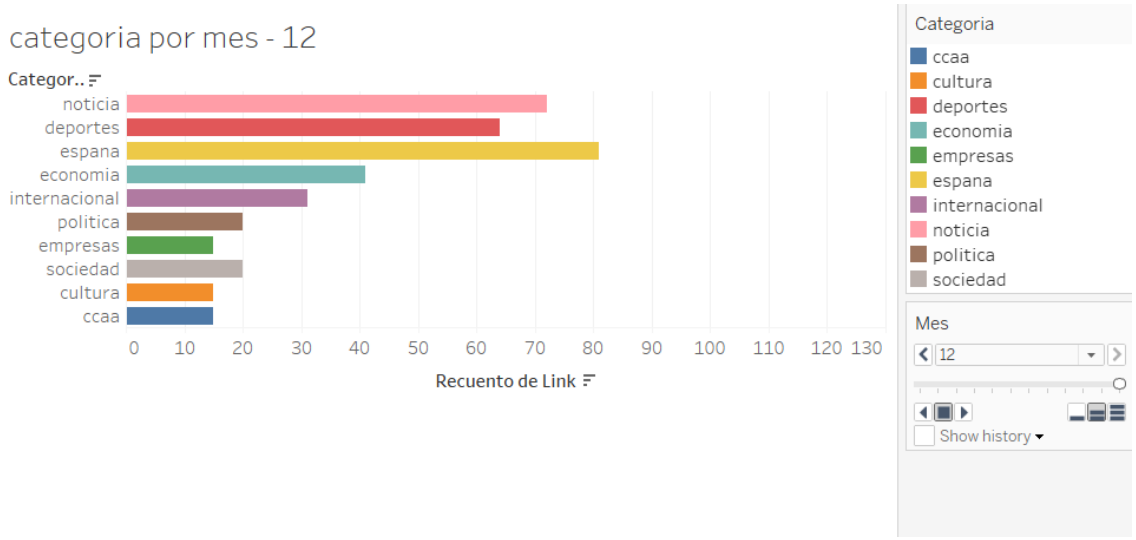


Ilustración 34. Categoría más escrita mes a mes

También observamos que existe una diferencia entre el número de caracteres presentes en una noticia, es decir, la longitud de su texto, y la categoría a la que pertenece, siendo la categoría más extensa la categoría de Política con cuatro mil caracteres de longitud media, y la sección de empresas la más corta, con dos mil trescientos caracteres de media.

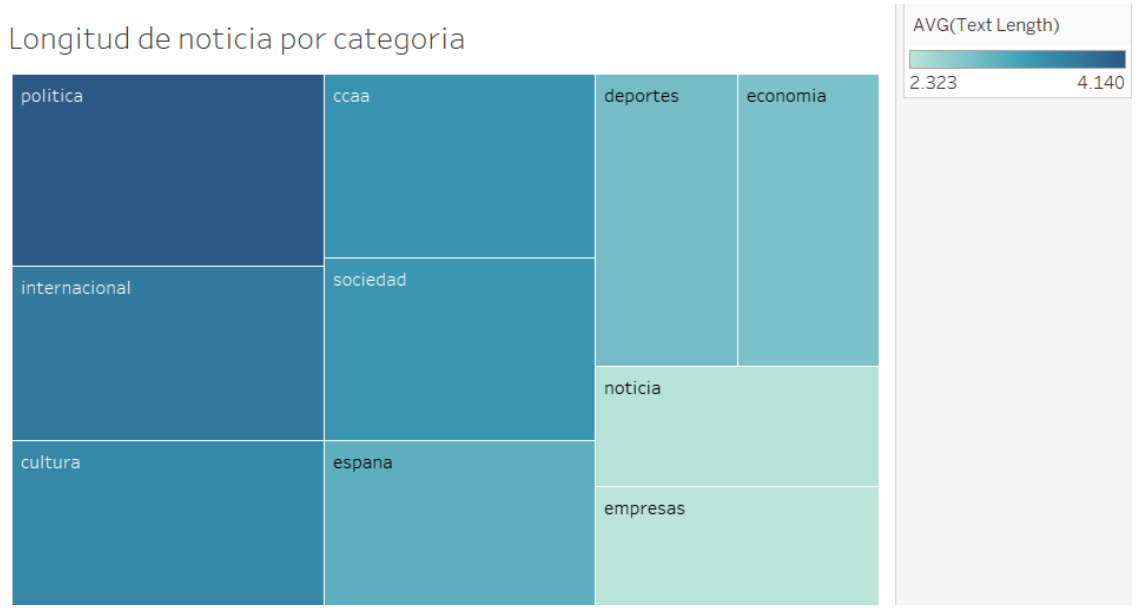


Ilustración 35 Longitud Media por categoría

Finalmente en el apartado de categoría podemos observar la distribución de noticias a lo largo de los 10 años analizados, donde una vez más la sección de España y deportes lidera el ranking sobre las categorías con más noticias escritas.

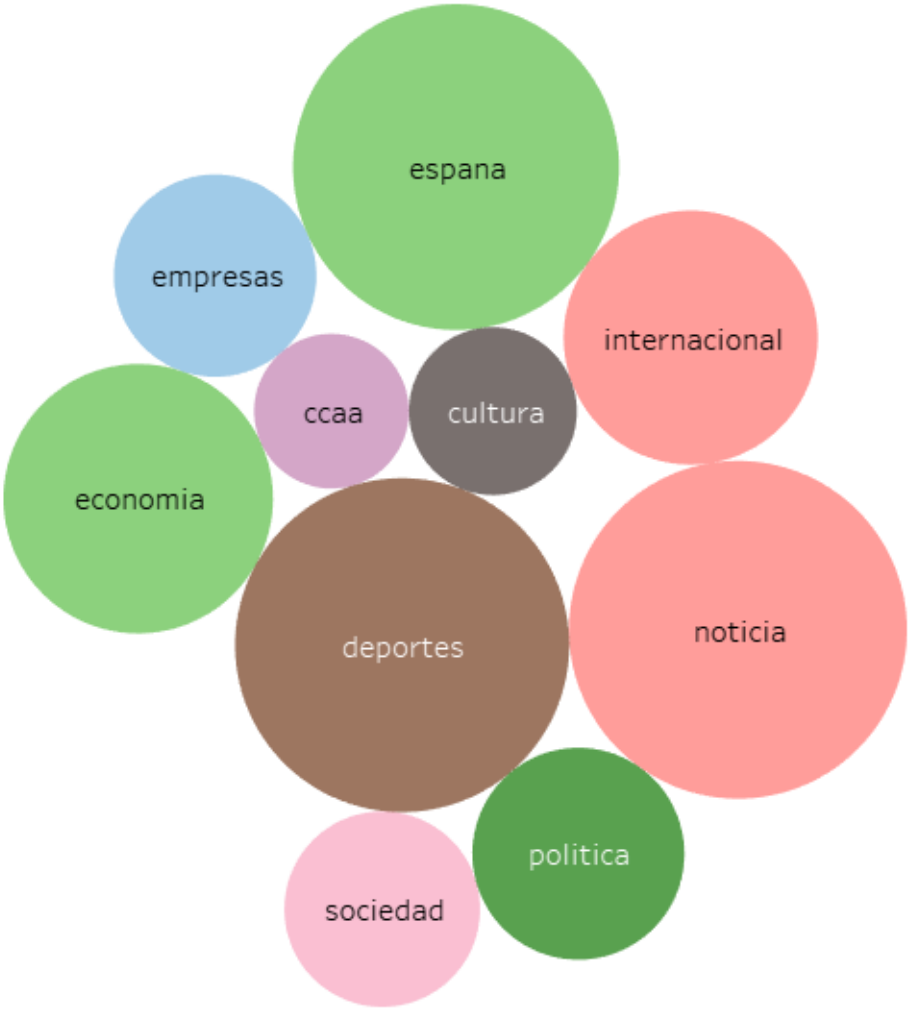


Ilustración 36 Categorías con más noticias en 10 años

Avanzando a los resultados proporcionados por el modelo de sentimiento, podemos observar en el siguiente gráfico que las noticias con lenguaje positivo predominan los artículos periodísticos analizados, seguidas de las noticias neutrales, y muy pocas veces teniendo noticias negativas. De nuevo hemos de mencionar que esos resultados han de interpretarse como el resultado de un muestreo estratificado, y tan solo representarán a la población final y serán fidedignos de la realidad actual en caso de que la muestra tomada represente correctamente a la población.

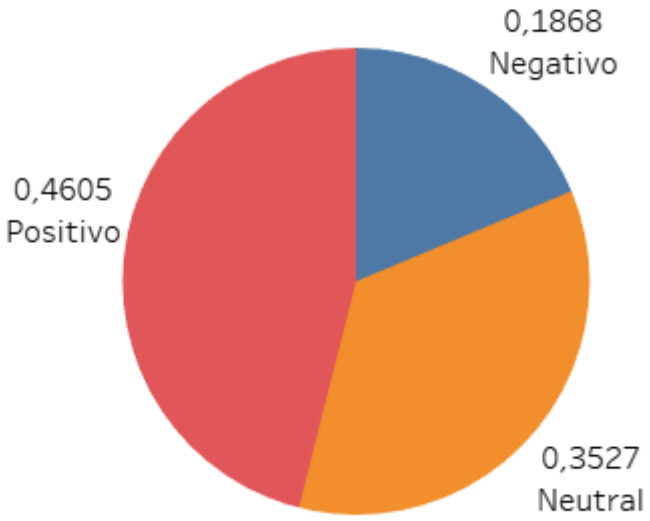


Ilustración 37. Distribución de Sentimiento

A continuación observamos la proporción de noticias positivas y negativas dependiendo de cada periódico. Gracias a este grafico podemos observar que los periódicos el mundo y el país poseen un balance bastante similar en cuanto a la proporción de noticias positivas en relación a las negativas. También cabe destacar que el confidencial posee en proporción más noticias negativas que positivas al compararlo con el resto de los periódicos, y en contra partida, el periódico 20 minutos posee la proporción más elevada de noticias positivas respecto a noticias negativas.

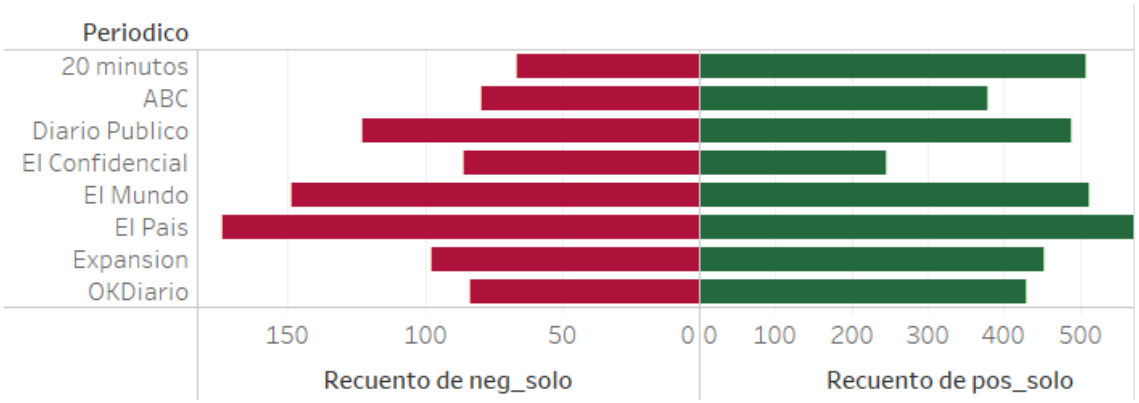


Ilustración 38. Comparación noticias positivas vs negativas por periódico

Finalmente para observar algunos de los parámetros más relevantes a la hora de categorizar un texto, hemos extraído las palabras más significativas para cada tipo de sentimiento, Una vez obtenidas, hemos cruzado las listas entre sí, de manera que las palabras presentes en todas las listas sean descartadas, para finalmente obtener las palabras relevantes presentes únicamente en un tipo de noticia. Para visualizar esta lista de palabras final hemos optado por el formato de Nube de Palabras, de manera que hemos exportado el listado de palabras de las noticias negativas, y las hemos cargado en una nube de palabras con el siguiente resultado

Ilustración 39. Wordcloud Palabras más relevantes de textos Negativos

Finalmente, respecto a los resúmenes generados por los modelos, podemos destacar tres KPI's que no han sido ya explorados ni en los gráficos del modelo NPL, ni en la exploración visual anterior.

En primer lugar, al estudiar la correlación de las palabras encontradas en el resumen contra las diferentes columnas, observamos que existe una muy fuerte correlación entre las palabras presentes en el título y las palabras que aparecen en el resumen, esta correlación de 0.87 indica que los títulos de la prensa española informan adecuadamente del contenido de la noticia.

En segundo lugar, al analizar la longitud de los resúmenes dependiendo del tipo de polaridad, observamos que no existe gran diferencia entre los resúmenes positivos y negativos, sin embargo, los resúmenes pertenecientes a noticias de sentimiento neutrales son ligeramente más cortos

```
Positivo
73.12506871907641
Neutral
59.14730961481917
Negativo
70.99535423925668
```

Ilustración 40 Diferencia de longitudes entre noticias Positivas-Negativas_Neutrales

Finalmente, al contrastar el listado de palabras relevantes en los resúmenes, con la categoría a la que pertenecen, y filtrando según los diferentes tipos de periódicos, hemos obtenido una lista vacía que confirma que los periódicos redactan de manera similar las principales categorías estudiadas.

4.3 Recursos requeridos

Para llevar a cabo el proyecto descrito se han necesitado los siguientes recursos y herramientas

- Ordenador portátil para el desarrollo de código, búsqueda de información, conexión a servicios web, y documentación del proyecto
- Servicios de Google: Google drive para compartir información y documentos con el tutor del proyecto, Google Search Engine para búsqueda de información en la Web, y Google Collab para desarrollo temporal de código
- Servicios de Microsoft Office: Excel para el desarrollo de diagramas Gantt y planificación, así como formato para visualizar los archivos CSV generados con datos. Word para el desarrollo de la documentación del proyecto, tanto para el anteproyecto como la memoria final. Outlook para la comunicación eficaz con el tutor del proyecto
- GitHub: Como repositorio final donde alojar el código desarrollado a lo largo del proyecto.
- Amazon Web Services: Como plataforma de alojamiento del servidor en la nube, y almacenamiento de datos
- Jupyter Notebook: como IDE para la programación del código Python
- Python 3: como lenguaje de código para programar y trabajar todos los datos a lo largo del proyecto
- Conexión a Internet: Para acceder a todos los recursos mencionados anteriormente, en especial al servidor web en la nube.
- Tableau: Para el desarrollo del análisis visual
- Respecto a las librerías utilizadas para la implementación del proyecto en Python, cabe destacar las siguientes:

Librería	Descripción breve
1. **NLTK**	Librería clave para el preprocesado de los datos
a. **Corpus-StopWords**	Contiene una lista de palabras que no aportan información
b. **Tokenize-Sent tokenize,**	Permite romper cadenas de texto en unidades más simples (palabras)
c. **Tokenize-Word tokenize**	Permite romper cadenas de texto en unidades más simples (palabras)
d. **Stem-WordNetLematizer**	Permite aplicar técnicas de lematización para el preprocesado
e. **Clasify**	Permite clasificar noticias una vez entrenado un modelo para ello
f. **NaiveBayesClassifier**	Permite utilizar un modelo Naive Bayes sin necesidad de programarlo desde 0
g. **PorterStemmer**	Permite aplicar técnicas de Stemming para el preprocesado
2. **Numpy**	Librería básica para operar
3. **Pandas**	Librería básica para operar
4. **UnicodeData**	Permite codificar los datos a UTF 8
5. **OS**	Permite hacer llamadas a directorios y acceder a carpetas del pc
6. **Random**	Permite redistribuir elementos de manera aleatoria
7. **Time**	Cronometra el tiempo de ejecución
8. **JobLib**	Guarda un modelo y sus parámetros
9. **MatplotLib**	Permite mostrar elementos de manera más visual
a. **rcParams**	Permite mostrar los parámetros elegidos en un modelo

10. Seaborn	Librería para graficar y mostrar elementos de manera más visual
11. Textblob	Permite tratar los datos en formato Textblob
12. Plotly	Librería para graficar y mostrar elementos de manera más visual
13. Cufflinks	Conecta Plotly con Pandas
14. Re	Librería para el preprocesado de texto (tildes)
15. String	Librería para el preprocesado de texto (tildes)
16. Wordcloud	Permite mostrar elementos de manera más visual
17. Sklearn	Biblioteca de aprendizaje automático, clave para generar el modelo
a. TfidfVectorizer	Permite contar el Term Frequency
b. CountVectorizer	Permite realizar operaciones con Vectores
c. GridSearchCV	Permite trabajar con vectores en cuadriculas
d. train_test_split	Facilita la separación de datos en sets de entrenamiento y testeo
e. metrics	Permite evaluar el algoritmo según diferentes categorías
f. classification_report	Permite visualizar los resultados de un modelo
g. cross_val_score	Permite implementar validación cruzada para evitar el sobre entrenamiento
h. roc_auc_score	Permite medir las predicciones
i. roc_curve	Permite medir el área ROC bajo la curva

j. **confusion_matrix**	Permite generar matrices de confusión
k. **accuracy_score**	Permite visualizar la métrica Accuracy
18. warnings	Permite trabajar con los errores de Python y mostrarlos
19. TensorFlow/Keras	Biblioteca de Redes Neuronales
20. Embedding	Permite embeber en redes neuronales
a. **pad_sequences**	Permite trabajar con redes neuronales
b. **Sequential**	Permite trabajar con redes neuronales secuenciales
c. **one_hot**	Permite el one hot encoding en redes neuronales
d. **LSTM**	Permite trabajar con redes neuronales LSTM
e. **Bidirectional**	Permite trabajar con redes neuronales Bidireccionales
f. **Dense**	Permite trabajar con la densidad de capas de redes neuronales
g. **Dropout**	Permite hacer regularización y evitar overfitting en redes neuronales
21. Requests	Para el acceso a las urls de las noticias
22. Beautiful Soup	Para parsear las noticias en formato HTML y convertirlas a texto

4.4 Presupuesto

El Desarrollo del proyecto se ha realizado mayormente mediante el uso de fuentes y herramientas con licencias gratuitas y sin ningún coste adicional por su uso. El mayor coste económico proviene del material informático necesario para desarrollar el proyecto, sin embargo este puede ser altamente reducido mediante el uso de servidores web y el remplazo del ordenador utilizado en este proyecto por uno de menor coste. El desglose total incurre en una cantidad igual a 1.487,50€

Tipo de coste	Valor	Comentarios
Tiempo invertido		
Horas de trabajo en el proyecto	500	Horas estimadas trabajando activamente en el desarrollo del proyecto
Equipo técnico utilizado		
Ordenador portátil MSI GP62M 7RDX(Leopard)	1.486,23 €	Equipo obtenido previo al proyecto, se adjunta el valor actual de mercado
Software utilizado		
Servidor Ubuntu en la Plataforma Amazon Web Services	1.27€	Coste de mantener y desarrollar el proyecto en AWS, a pesar de configurar todo conforme a la capa gratuita, debido al uso intensivo del servidor, se han superado los limites gratuitos e incurrido un coste de un euro y veintisiete céntimos
Jupyter Notebook	0€	Aplicación Gratuita que no requiere licencia
Python 3	0€	Lenguaje Gratuito que no requiere licencia
Tableau Public	0€	Se ha utilizado la versión gratuita del programa, sin costes de licencia necesarios
Estudios e informes		
Estudios e informes	0€	Toda la documentación utilizada proviene de fuentes gratuitas
Materiales empleados		
Materiales empleados	0€	No se requiere materiales adicionales a los indicados

Capítulo 5. RESULTADOS DEL PROYECTO

Para concluir el presente proyecto se dividen los resultados en los siguientes puntos:

1. El resultado más crucial ha consistido cumplir con el objetivo general del proyecto mediante la creación de tres modelos NLP que agilicen la lectura de una noticia por medio de la generación de un resumen del texto, la categorización de las noticias y la identificación del sentimiento expresado por el autor de la noticia. Este resultado se puede observar en la siguiente ilustración conteniendo una muestra de 5 artículos reales procesados por el modelo final.[Ilustración 31]

Out[213]:

	Resumen	Categoria	Tipo_Polaridad
300	Algunos de los principales rivales de Facebook como el gigante Apple o Microsoft y Google ya cuentan con tecnología integrada de reconocimiento de voz en los sistemas operativos de sus diferentes teléfonos inteligentes.\nFacebook ha comprado la empresa emergente especializada en el reconocimiento de voz Wit.ai, en una operación de la que se desconocen sus términos financieros.\n"La misión de Facebook es conectar a todo el mundo y ofrecer una experiencia increíble a más de 1.300 millones de personas en una plataforma que entienda que el lenguaje natural es importante y creemos que podemos ayudar", asegura hoy la empresa emergente en su blog.\n"Nuestra plataforma seguirá estando abierta y será enteramente gratuita para todo el mundo", añadió la empresa emergente fundada hace apenas 18 meses por Alex Lebrun, Willy Blandin y Laurent Landowsk.\nAlgunos de los principales rivales de Facebook como el gigante Apple o Microsoft y Google ya cuentan con tecnología integrada de reconocimiento ...	empresas	Positivo
301	La empresa de medicamentos biológicos Bionaturis Group comenzó en 2014 su estrategia de crecimiento y expansión internacional.\nSu intención era hacerse con nuevos productos de su sector, ayudándoles a acceder a un mercado mayor.\nSegún los fines que se persigan, se tendrá que evaluar qué tipo de organización se va a comprar: si es del mismo sector, si opera en otros mercados o si realiza una actividad complementaria.\nEl objetivo era entrar y dominar un mercado con grandes aspectos en común y esa empresa nos lo permitía.\nAsí, han dejado de existir proyectos de una u otra empresa y se ha creado un programa común", asegura el consejero delegado.	pymes	Positivo
302	El terremoto de 5,2 grados registrado en la localidad de Ossa de Montiel es el más intenso desde el año 2012, cuando el 30 de julio cinco localidades gallegas sintieron un movimiento sísmico de intensidad 5,6 en la escala de Richter.\nEn los últimos años, el terremoto más dañino ha sido el de Lorca (Murcia), cuya sucesión de temblores, uno de 4,5 y otro de 5,1 grados registrados el 11 de mayo de 2011, dejó ocho víctimas mortales.\nFue la sucesión de seísmos más grave que había sufrido España en cincuenta años.\nEl 17 de diciembre de 2009 se registró un terremoto de 6,3 grados en la costa occidental de Andalucía, pero en aquella ocasión no hubo que lamentar daños personales ni materiales.\nEs el terremoto de mayor intensidad que consta en la historia de la Península ibérica.	entorno	Positivo
303	Y es que hay productos tradicionales, como los huevos y las patatas, que por su versatilidad son un acierto seguro.\nDesde los clásicos huevos fritos con patatas al típico pincho de tortilla o a la famosa receta de huevos rotos de Casa Lucio de Madrid... Una pareja bien avenida a la que no hay quien se resista.\nLos huevos con patatas siempre están a la última, una unión maestra presente en bares y restaurantes, en la cocina de cualquier hogar, y que hasta ha llegado a seducir al universo 'gourmet' con exquisitos bocados de autor.\nTambién se pueden elegir los huevos camperos, procedentes asimismo de gallinas criadas al aire libre y que se recogen a diario.\nDe este modo, presentan una yema con un característico color rojizo y tienen una gran consistencia y un aroma intenso.	tendencias	Negativo
304	CaixaBank obtuvo el año pasado un beneficio de 620 millones, lo que supone un aumento del 96,3% frente a las cuentas de 2013, reformuladas tras un cambio en la contabilidad.\nLa sede de CaixaBankEl resultado de CaixaBank se ha quedado ligeramente por debajo de las previsiones de consenso que esperaban unos 685 millones.\nLa aplicación de esta norma resta al beneficio de CaixaBank del pasado año 503 millones.\nAnálisis fundamental PRECIO OBJETIVO: RECOMENDACIÓN: Análisis tecnico CORTO PLAZO: MEDIO PLAZO: LARGO PLAZO: Resultados Hora de presentación: Antes de la apertura (09:00 h.) Conference Call: (12:00 h.) Ficha completa Nota: Datos de cotización y análisis actualizados a día de hoy.\nCaixaBank ha conseguido reducir su ratio de morosidad en dos puntos porcentuales a lo largo del ejercicio, hasta situarla en el 9,7%.	empresas	Negativo

Ilustración 41. Muestra de 5 noticias Resumidas, categorizadas, y con el sentimiento extraído

2. Para desarrollar nuestro modelo final en base a las noticias de la prensa española, debido a la ausencia de un conjunto de datos preexistente, se ha necesitado extraer todas las noticias digitales de los últimos diez años, esto ha sido posible mediante la creación de un Web Scrapper que extraiga las noticias de ocho periódicos durante el periodo de enero de 2012 hasta enero de 2022. Para la correcta extracción de todas las noticias hemos de considerar las diferencias entre las páginas web de cada periódico, la manera en la que almacenan las noticias, y las diferentes etiquetas HTML que estructuran las páginas web. Este objetivo explicado detalladamente en la sección 4.2.2 Extracción de los datos, ha dado lugar a la generación de un dataset limpio y relevante con 3Millones 885 mil noticias en formato de texto.

3. En el desarrollo del primer modelo NLP, el cual determina la categoría de una noticia en base a su texto, hemos comenzado con un modelo LDA de aprendizaje no supervisado, pero debido a la necesidad de un experto para interpretar adecuadamente los resultados finales, hemos replanteado la manera de enfocar la obtención de la categoría. Para el segundo intento, se han implementado modelos de aprendizaje supervisado. Para la correcta realización de un modelo de aprendizaje supervisado, ha sido crucial realizar un análisis de nuestros datos, en específico el paso más importante

ha sido modificar la cantidad de datos pertenecientes a cada categoría mediante técnicas de remuestreo conocidas como Upsampling y Downsampling, para garantizar que todas las categorías están igualmente representadas en los datos. Tras una limpieza general del texto de las noticias, se ha construido un diccionario con la frecuencia de las palabras existentes en el texto mediante TF-IDF, el cual ha sido suministrado a diferentes modelos para conseguir el mejor resultado posible. Tras entrenar y evaluar cuatro algoritmos de Machine Learning diferente, hemos concluido con un Modelo de Random Forest con una precisión del 86,8%

4. Para el segundo modelo desarrollado, el cual produce resúmenes de las noticias, hemos optado por utilizar modelos de análisis profundo (Deep Learning) gracias a que poseemos suficientes datos como para entrenarlos adecuadamente, y a los buenos resultados que estos modelos ofrecen. Tras realizar una limpieza del texto y eliminar los elementos que no aportan información relevante en las noticias, hemos construido una red neuronal convolucional,que utiliza el proceso de codificador-descodificador compuesto por redes LSTM, las cuales reciben como entrada los textos originales, y devuelven como salida del modelo las frases más relevantes del texto. Para evaluar la calidad de los resúmenes generados, se han utilizado un conjunto de métricas conocidas como rouge, las cuales están altamente aceptadas en la comunidad científica para evaluar los problemas de generación te texto automático. El resultado de Rouge concluye que los resúmenes producidos por nuestro modelo son relevantes al 75%

5. En la construcción del tercer y último modelo, el cual determina el Sentimiento del texto de las noticias, se comenzó tratando de evaluar el mensaje positivo, negativo o neutral de todas las palabras que componen un texto, y en base a esta información determinar el sentimiento. Sin embargo, debido a la carencia de un dataset relevante en español con toda la información sobre las palabras, optamos por modificar el planteamiento del problema, y utilizar el texto de miles de reseñas junto con su valoración positiva o negativa final para entender un texto conjunto en lugar de las palabras individuales. Las transformaciones clave para el desarrollo de este modelo fueron en primer lugar una limpieza del texto general, eliminando los componentes que no aportan información, en segundo lugar limitar el vocabulario introducido al modelo para evitar un sobre aprendizaje, y en tercer lugar, la conversión del texto a vectores de valores para que el modelo identifique la relevancia y similitud de las palabras. Tras esto, desarrollamos una red neuronal LSTM sobre la cual fuimos iterando con diferentes parámetros hasta alcanzar la configuración más optima, hasta llegar al modelo final con una precisión del 81%.

6. Finalmente, en el desarrollo de los gráficos para conocer la información producida por nuestros modelos NLP y oculta en las noticias, empleamos la herramienta de Tableau y diferentes diseños y animaciones para observar el número de noticias generadas a lo largo del tiempo por cada periódico, estudiar las palabras más utilizadas según el tipo de categoría o periodo de tiempo, o visualizar la distribución del sentimiento según la categoría seleccionada, entre otros gráficos.

Capítulo 6. CONCLUSIONES

6.1 Conclusiones del trabajo

Con el aumento constante de los datos y noticias generadas, es necesario mejorar la manera en la que consumimos noticias de texto escrito para poder mantenernos informados sin invertir una cantidad de tiempo excesiva. La propuesta de este proyecto consiste en ofrecer toda la información contenida en una noticia por medio de un resumen generado automáticamente, una categorización del tipo de noticia, e informar sobre el sentimiento expresado en el texto.

La tecnología necesaria para implementar esta propuesta, ya está desarrollada, y evoluciona constantemente gracias a los esfuerzos de la comunidad científica en el campo del estudio del lenguaje natural. Sin embargo esta tecnología todavía no está comúnmente implementada en el sector periodístico, lo que proporciona una oportunidad para desarrollar e implementar esta solución como sustitutivo o apoyo a las noticias ya existentes ofrecidas por los periódicos en sus diferentes webs, de manera que se permita al lector decidir el nivel de granularidad de la información que desea consumir, ya sea mediante el texto original para obtener todos los detalles posibles, o la información más relevante mediante la propuesta desarrollada.

6.2 Conclusiones personales

La idea de este proyecto surgió a finales de febrero, tras intentar recabar información lo más rápido posible sobre el conflicto que surgía entre Rusia y Ucrania. A lo largo que trataba de comprender la situación y los factores que habían desencadenado esta guerra, me frustraba al leer artículos extensos de periódicos, los cuales repetían unas pocas ideas constantemente, pero no mostraban de manera clara y sencilla la situación que se estaba desenvolviendo. De este evento nace de la propuesta de aplicar lo aprendido en la carrera para extraer los matices clave de las noticias, sin tener que leerlas por completo.

Este proyecto me ha brindado la oportunidad de poner en práctica todo lo aprendido en la carrera, al comenzar el proyecto desde cero y tener que atravesar todas las fases de los proyectos big data, comenzando por la extracción y generación de los datos, y finalizando y finalizando con un programa que correctamente simplifica la lectura de las noticias.

Asimismo este proyecto ha supuesto un reto personal por dos motivos principales, el primero de ellos proviene de la pequeña base conocida al inicio del proyecto sobre el desarrollo de problemas del ámbito NLP, la cual ha sido trabajada a diario para poder desarrollar unos modelos precisos y relevantes, lo cual ha sido un proceso muy motivador al observar cómo mi capacidad de entender el problema y desarrollar una solución viable aumentaba a medida que trabajaba en el proyecto. El segundo reto personal ha sido el tiempo dedicado al proyecto, partiendo de la base del ajustado calendario para entregar a tiempo el proyecto y desarrollar todas las fases definidas, en conjunto con trabajar una jornada laboral completa y demandante, ha requerido un esfuerzo diario para tras acabar de trabajar, ponerse a desarrollar el TFG. La pasión por

el tema del proyecto, la auto motivación por desarrollar un modelo relevante, y el apoyo de las personas a mi alrededor han sido los elementos que me han permitido conquistar el desarrollo de este Trabajo de Fin de Grado.

Capítulo 7. FUTURAS LÍNEAS DE TRABAJO

Existen diferentes acciones que se podrían realizar sobre el proyecto desarrollado para continuar expandiendo su alcance y aplicación actual a las noticias que consumimos. Debido a restricciones de tiempo de entrega, y a las propias limitaciones establecidas dentro del proyecto, estas mejoras no han sido desarrolladas pero suponen una tarea interesante para continuar el trabajo.

Aplicación del proyecto: El proyecto se definió para trabajar con los datos de las noticias de prensa digital española, sin embargo el trabajo realizado a la hora de trabajar el formato de texto y producir resúmenes u otra información relevante podría ser aplicado a otros formatos de texto, como libros, artículos académicos, o artículos de un blog. Sería muy interesante observar el funcionamiento de los modelos en estos otros formatos de información contenida en texto y desarrollarlo para aceptar diferentes tipos de texto.

Respecto a los datos trabajados, el proyecto centra su marco en las noticias de los periódicos españoles, otra posible línea de trabajo es incorporar noticias de periódicos diferentes provenientes de distintos países como México, Argentina, u otros países de Latinoamérica que también publican en español.

Respecto a los modelos desarrollados, el modelo de categoría requirió eliminar algunas categorías de los periódicos, por lo que la línea de trabajo más evidente sería desarrollar el modelo para aceptar todas las categorías originales. En este mismo aspecto, el modelo no se desarrolló mediante técnicas de Deep Learning, sin embargo el estado del arte actual refleja que el análisis profundo consigue entrenar modelos con precisión mucho mayor al desarrollado para este proyecto, de modo que cambiar el modelo por una Red neuronal o un modelo de Análisis profundo supondría una clara mejora en el desarrollo final.

Finalmente la aplicación del modelo se ha limitado a procesar las noticias de los últimos diez años, pero no se ha desarrollado ninguna aplicación ni solución comercial que utilice el trabajo aquí desarrollado. Por ello se podría también construir una web o una aplicación y ofrecérsela a los diferentes periódicos para utilizar como herramienta de apoyo y aportar valor añadido para los lectores a la hora de presentar las noticias.

Capítulo 8. REFERENCIAS

[1] “how much data is generated every day” Available online at https://www.the-next-tech.com/blockchain-technology/how-much-data-is-produced-every-day-2019/

[2] “Google’s Summarizing News Articles” Available online at https://research.google/pubs/pub48295/

[3] “Introduction to Text Summarization, State of the Art” Available online at

https://medium.com/besedo-engineering/text-summarization-part-1-a-gentle-introduction-to-automatic-text-summarization-31f14b1f9e53

[4] Ms. Anusha Pai, 2014, Text Summarizer Using Abstractive and Extractive Method, INTERNATIONAL JOURNAL OF ENGINEERING RESEARCH & TECHNOLOGY (IJERT) Volume 03, Issue 05 (May 2014), available at https://www.ijert.org/research/text-summarizer-using-abstractive-and-extractive-method-IJERTV3IS050821.pdf

Som Gupta, S. K Gupta, Abstractive summarization: An overview of the state of the art, Expert Systems with Applications , Volume 121, 2019, Pages 49-65, ISSN 0957-4174, https://doi.org/10.1016/j.eswa.2018.12.011. Available at

(https://www.sciencedirect.com/science/article/pii/S0957417418307735)

[5] “A Primer on Neural Network Models for Natural Language Processing”, Yoav Goldberg, 2015, available at https://arxiv.org/pdf/1510.00726.pdf

[6] Christian Tarchi, Sonia Zaccoletti, Lucia Mason, Learning from text, video, or subtitles: A comparative analysis, Computers & Education, Volume 160, 2021, 104034, ISSN 0360-1315, https://doi.org/10.1016/j.compedu.2020.104034. Available at (https://www.sciencedirect.com/science/article/pii/S0360131520302323)

[7] ”Definition of multiprocessing” Available online at

https://www.britannica.com/technology/multiprocessing

[8] “A guide to multi processing” Available online at

https://www.analyticsvidhya.com/blog/2021/04/a-beginners-guide-to-multi-processing-in-python/

[9] Joshua Charles Campbell, Abram Hindle, Eleni Stroulia, Chapter 6 - Latent Dirichlet Allocation: Extracting Topics from Software Engineering Data, Editor(s): Christian Bird, Tim Menzies, Thomas Zimmermann, The Art and Science of Analyzing Software Data, Morgan Kaufmann, 2015, Pages 139-159, ISBN 9780124115194, https://doi.org/10.1016/B978-0-12-411519-4.00006-9. Available at (https://www.sciencedirect.com/science/article/pii/B9780124115194000069)

[10] “TF-IDF explained” Available online at

https://www.capitalone.com/tech/machine-learning/understanding-tf-idf/

[11] “Evolución de la prensa digital”, evocaimagen.com, Available online at

https://www.evocaimagen.com/cuadernos/cuadernos1.pdf

[12] “Periodicos digitales de España” Available online at
https://www.prensaescrita.com/prensadigital.php&sa=D&source=docs&ust=1656863804764779&usg=AOvVaw0MelOp9fKi8XsR-MrIGLp-

[13] Michael Völske, Martin Potthast, Shahbaz Syed, and Benno Stein. 2017. TL;DR: Mining Reddit to Learn Automatic Summarization. In Proceedings of the Workshop on New Frontiers in Summarization, pages 59–63, Copenhagen, Denmark. Association for Computational Linguistics.

[14] Ms. Anusha Pai, 2014, Text Summarizer Using Abstractive and Extractive Method, INTERNATIONAL JOURNAL OF ENGINEERING RESEARCH & TECHNOLOGY (IJERT) Volume 03, Issue 05 (May 2014)

[15] K. Ghag and K. Shah, “SentiTFIDF – Sentiment classification using relative term frequency inverse document frequency,” International Journal of Advanced Computer Science and Applications, vol. 5, no. 2, 2014

[16] “ROUGE, the ultimate performance metric” Available online at

https://towardsdatascience.com/the-ultimate-performance-metric-in-nlp-111df6c64460

[17] “SFU corpus” Available online at

http://www.sfu.ca/~mtaboada/research/SFU_Review_Corpus.html

[18] “Word Embeddings, what they are and how they work” Available online at

https://machinelearningmastery.com/what-are-word-embeddings/

[19] “What Is NLP” Available online at

https://www.ibm.com/cloud/learn/natural-language-processing

[20] West, Matthew. (2010). Developing High Quality Data Models. Available online at

https://www.researchgate.net/publication/286610894_Developing_High_Quality_Data_Models

[21] “State of the art for text summarization” Available online at

https://medium.com/besedo-engineering/text-summarization-part-2-state-of-the-art-ae900e2ac55f

[22]T. T. Dien, B. H. Loc and N. Thai-Nghe, "Article Classification using Natural Language Processing and Machine Learning," 2019 International Conference on Advanced Computing and Applications (ACOMP), 2019, pp. 78-84, doi: 10.1109/ACOMP.2019.00019.

[23] Samuels, Antony and Mcgonical, John, News Sentiment Analysis, Available online at

https://arxiv.org/abs/2007.02238

[24] D.M.E.D.M. Hussein, “A survey on sentiment analysis challenges,” Journal of King Saud University - Engineering Sciences, vol. 30, no. 4, pp. 330–338, 2018.

[25] E. Haddi, X. Liu, and Y. Shi, “The Role of Text Pre-processing in Sentiment Analysis,” Procedia Computer Science, vol. 17, pp. 26–32, 2013.

[26] M. Devika, C. Sunitha, and A. Ganesh, “Sentiment analysis: a comparative study on different approaches,” Procedia Computer Science, vol. 87, pp. 44–49, 2016.

[27] Aggarwal, C. C., and Zhai, C. (2012). A survey of text classification algorithms. In Mining text data. Springer. 163-222.

[28] Hingmire, S.; Chougule, S.; Palshikar, G. K.; and Chakraborti, S. (2013). Document classification by topic labeling. In SIGIR, 877-880.

[29] Bengio, Y.; Ducharme, R.; Vincent, P.; and Jauvin, C. (2003). A Neural Probabilistic Language Model. JMLR 3:1137-1155.

[30] Cai, L., and Hofmann, T. (2003). Text categorization by boosting automatically extracted concepts. In SIGIR, 182-189.

[31] Collobert R. et al. Natural language processing (almost) from scratch Journal of Machine Learning Research. – 2011. – Т. 12. – №. Aug. – С. 2493-2537.

[32] Hinton, G. E., and Salakhutdinov, R. R. (2006). Reducing the dimensionality of data with neural networks. Science 313(5786):504-507.

[33] Sepp Hochreiter and Jurgen Schmidhuber. 1997. ¨ Long short-term memory. Neural computation 9(8):1735–1780.

[34] Diederik Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 .

[35] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781 .

[36] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013b. Distributed representations of words and phrases and their compositionality. In Advances in neural information processing systems. pages 3111–3119.

[37] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In Empirical Methods in Natural Language Processing (EMNLP). pages 1532– 1543. http://www.aclweb.org/anthology/D14-1162.

[38] “Coherence Score” Available at

https://towardsdatascience.com/evaluate-topic-model-in-python-latent-dirichlet-allocation-lda

[39] Periódicos más leídos de España Available online at

https://es.statista.com/estadisticas/476795/periodicos-diarios-mas-leidos-en-espana/?msclkid=0a9b46b7b7f711eca0c42be82537160e

Capítulo 9. ANEXOS

Todo el código desarrollado a lo largo del proyecto, así como las diferentes visualizaciones y los datasets finales creados y utilizados se encuentran alojados en GitHub, en el siguiente enlace Polamen/TFG (github.com)