UNIVERSIDAD DE BURGOS
ESCUELA POLITÉCNICA SUPERIOR
Grado en Ingeniería Informática

TFG del Grado en Ingeniería Informática

Digitalización del Boletín de Turismo del Observatorio de la Provincia de Burgos

Presentado por Christian Andrés Núñez Duque
en Universidad de Burgos — 7 de julio de 2025
Tutor: Bruno Baruque Zanón
Cotutor: Julio César Puche Regaliza

UNIVERSIDAD DE BURGOS
ESCUELA POLITÉCNICA SUPERIOR
Grado en Ingeniería Informática

D. Bruno Baruque Zanón, profesor del departamento de Digitalización, área de Ciencia de la Computación e Inteligencia Artificial y D. Julio César Puche Regaliza, profesor del departamento de Economía Aplicada, área de Métodos Cuantitativos para la Economía y la Empresa.

Exponen:

Que el alumno D. Christian Andrés Núñez Duque, con DNI 71301258Q, ha realizado el Trabajo final de Grado en Ingeniería Informática titulado Digitalización del Boletín de Turismo del Observatorio de la Provincia de Burgos.

Y que dicho trabajo ha sido realizado por el alumno bajo la dirección del que suscribe, en virtud de lo cual se autoriza su presentación y defensa.

En Burgos, 7 de julio de 2025

Vº. Bº. del Tutor:	Vº. Bº. del co-tutor:
D. Bruno Baruque Zanón	D. Julio César Puche Regaliza

Resumen

En este proyecto se ha desarrollado un sistema de extracción de reseñas y recursos junto a su preprocesado para entrenar una red neuronal que sea capaz de clasificar los recursos (POI) por su tipo. Además, se ha creado un cuadro de mando en PowerBI que permite visualizar los resultados obtenidos así como métricas relevantes sobre las reseñas y la distribución de los recursos. El objetivo es facilitar la digitalización del boletín de turismo del Observatorio de la Provincia de Burgos, permitiendo una mejor gestión y análisis de la información disponible.

Descriptores

redes neuronales, procesamiento de lenguaje natural, turismo, BERT, reseñas, recursos turísticos, puntos de interés, clustering, sistema de recomendación, cuadro de mando, digitalización, análisis de datos, inteligencia artificial

Abstract

In this project, a system for extracting reviews and resources (POI) and their preprocessing has been developed to train a neural network capable of classifying resources by type. In addition, a PowerBI dashboard has been created to visualise the results obtained as well as relevant metrics on the reviews and the distribution of resources. The objective is to facilitate the digitalisation of the tourism bulletin of the Observatory of the Province of Burgos, enabling better management and analysis of the available information.

Keywords

neural networks, natural language processing, tourism, BERT, reviews, tourist resources, points of interest, clustering, recommender system, dashboard, digitisation, data analysis, artificial intelligence

Índice general

Índice de figuras

Índice de tablas

1. Introducción

1.1. Marco del trabajo

El turismo es uian actividad que conlleva que las personas se desplacen de su lugar habitual de forma temporal para visitar otros lugares, ya sea por ocio, negocios, cultura u otras razones. Esta actividad puede ser entendida como una forma de consumo ya que los viajeros gastan dinero en bienes y servicios durante su estancia en el destino turístico.[5]

En España, el turismo es una de las principales fuentes de ingresos y empleo. En 2023, la actividad turística supuso un 13,4 % del PIB del país. Esto no es casualidad, sus playas, gastronomía, clima mediterráneo y la riqueza artística y arquitectónica han logrado posicionarlo entre los tres países más visitados en todo el mundo llegando a recibir alrededor de 85 millones de turistas en ese mismo año.[29].

Debido a la importancia del turismo en la economía española, la Sociedad Mercantil Estatal para la Gestión de la Innovación y las Tecnologías Turísticas (SEGITTUR) lleva varios años trabajando para contribuir al desarrollo, modernización y mantenimento de la industria turística española a través de la innovación tecnológica. Trata de mejorar la competividad, calidad y sostenibilidad en los ámbitos medioambientales, económicos y sociales relacionados con el turismo.[26]

Con el objetivo de mejorar la competividad de los destinos turísticos y la calidad de vida de los viajeros, la Secretaría de Estado de Turismo (SETUR) promueve el programa Destino Turístico Inteligente (DTI) el cual es gestionado por SEGITTUR. Un DTI es un destino turístico innovador con una accesibilidad global y que cuenta con una infraestructura tecnológica

avanzada. Esta infraestructura garantiza el desarrollo sostenible del destino y trata de mejorar la experiencia de los viajeros y su calidad de vida.[7]

1.2. Contenido del trabajo

El objetivo principal del trabajo es la creacón y publicación de un cuadro de mando con el objetivo de que la provincia de Burgos alcance el distintivo de Destino Turístico Inteligente. Para ello, se ha desarrollado un sistema de extracción automatizado de reseñas de puntos de interés en la provincia de Burgos. Con esa información se crea el cuadro de mando que permite analizar y monitorizar distintas métricas relevantes como la distribución de los puntos de interés por categorias. Además, se ha desarrollado una red neuronal entrenada con un conjunto de datos formado por reseñas extraídas de la misma forma con el objetivo de clasificar por categorías las reseñas proporcionadas externamente.

1.3. Estructura de la memoria

La memoria del proyecto se estructura en siete capítulos:

- **Capítulo 1: Introducción.** Presenta el marco del trabajo, el contenido y la estructura de la memoria.
- **Capítulo 2: Objetivos del proyecto.** Define los objetivos generales y específicos del proyecto.
- **Capítulo 3: Conceptos teóricos.** : Explica los conceptos teóricos necesarios para entender el proyecto.
- **Capítulo 4: Técnicas y herramientas.** Describe las técnicas y herramientas utilizadas en el desarrollo del proyecto.
- **Capítulo 5: Aspectos relevantes del desarrollo del proyecto.** Detalla el desarrollo del proyecto, incluyendo la extracción de reseñas, la creación del cuadro de mando y el entrenamiento de la red neuronal.
- **Capítulo 6: Trabajos relacionados.** Presenta una revisión de trabajos relacionados con el proyecto, incluyendo estudios previos y proyectos similares.
- **Capítulo 7: Conclusiones y líneas de trabajo futuras.** Resume las conclusiones del proyecto y propone posibles trabajos futuros.

Además, se incluten varios anexos que contienen información adicional relevante para el proyecto:

- **Anexo A: Plan de proyecto software.** Contiene el plan de proyecto, incluyendo la planificación temporal y el desarrollo del mismo y la viabilidad económica y legal.
- **Anexo B: Especificación de requisitos.** Incluye los requisitos funcionales y no funcionales del proyecto, así como los casos de uso.
- **Anexo C: Especificación de diseño.** Explica el diseño del sistema, incluyendo la arquitectura o las interfaces entre otras cosas.
- **Anexo D: Documentación técnica de programación.** Describe la parte técnica del proyecto como las explicaciones sobre el repositorio del código fuente.
- **Anexo E: Documentación de usuario.** Incluye de forma detallada todo lo que el usuario necesita saber y poseer para poder ejecutar y/o utilizar el proyecto principalmente en forma de manual de usuario.
- **Anexo F: Anexo de sostenibilización curricular.** Presenta una reflexión sobre los aspectos de sostenibilidad abordados en el proyecto.

1.4. Materiales entregados

Los materiales entregados al finalizar el proyecto son:

- Varios scripts de Python utilizados para la extracción de reseñas.
- Un conjunto de datos con las reseñas extraídas utilizadas para entrenar la red neuronal.
- Un modelo de red neuronal entrenado para clasificar reseñas por categorías.
- Un cuadro de mando interactivo creado con Power BI publicado en Power Pages.
- Una memoria del proyecto junto a sus anexos en formato PDF creado con LaTeX.
- Un repositorio de GitHub con el código fuente del proyecto y la documentación.[17]

2. Objetivos del proyecto

Este proyecto tiene como finalidad desarrollar una solución tecnológica que permita analizar la reputación online de distintos destinos turísticos mediante un cuadro de mando desde el que se puedan visualizar los datos obtenidos relativos a las reseñas de forma sencilla y comprensible. Para ello, se desarrollará un sistema de análisis de reputación online que permita obtener información relevante sobre la percepción de los usuarios sobre un destino turístico. Realizar esta tarea conlleva los siguientes pasos:

- Automatizar la obtención de datos.
- Clasificar e interpretar los datos.
- Sistema de análisis de sentimientos.
- Crear un cuadro de mando.
- Publicar los resultados.

2.1. Automatizar la obtención de datos

Para que el sistema de análisis de reputación online sea eficiente y aplicable a múltiples destinos turísticos, es fundamental automatizar el proceso de recopilación de datos. Se desarrollará un sistema de extracción de datos automatizado que permita obtener información relevante de distintas fuentes, como reseñas de usuarios, puntuaciones, etc. Estos datos están publicados en Internet de forma altruista por diferentes personas que buscan aconsejar a otros usuarios. Esto permitirá que el sistema escale sin intervención manual constante, garantizando una actualización en tiempo real y una mayor precisión en la información analizada.

2.2. Clasificar e interpretar los datos

Hacer una clasficación e interpretación de los datos obtenidos previamente: recursos, reseñas, etc. El objetivo de este proceso es obtener información relevante. Para ello, se clasificarán las reseñas de los usuarios en positivas, negativas o neutras. Además, se analizarán las palabras más frecuentes para identificar temas recurrentes en las reseñas. Esta clasificación será clave para el siguiente paso del análisis de sentimientos.

2.3. Sistema de análisis de sentimientos

El análisis de sentimientos permite interpretar de forma automática la opinión de los usuarios sobre un destino turístico. Se desarrollará un sistema de análisis de sentimientos que permita obtener información sobre la percepción de los usuarios. Se utilizarán reseñas de usuarios para extraer dicha información sobre los destinos turísticos. El proceso ETL (Extract, Transform, Load) es fundamental para preparar las reseñas de usuarios para el análisis de sentimientos. Para que el sistema de análisis de sentimientos funcione correctamente, los datos deben estar estructurados, como puntuaciones y palabras clave, que faciliten el análisis. Sin embargo, muchas reseñas son textos no estructurados. En estos casos, el sub-sistema de Text Mining se encarga de procesar y extraer información útil del texto, como palabras clave y sentimientos, utilizando técnicas de procesamiento de lenguaje natural. Esto convierte los datos no organizados en información estructurada que luego se puede analizar con el objetivo de evaluar la percepción de los usuarios sobre los destinos turísticos.

2.4. Crear un cuadro de mando

Crear un cuadro de mando interactivo destinado al análisis de la reputación online, aplicable a distintos destinos turísticos. Este cuadro de mando contendrá información visual y estructurada sobre la percepción de los usuarios sobre dichos destinos turísticos. Esta tarea se realizará con PowerBI, una herramienta de análisis de datos que permite crear gráficos e informes visuales. PowerBI permite crear cuadros de mando interactivos y visuales para visualizar los datos de forma clara y concisa. Además, permite usar DAX, un lenguaje de fórmulas que permite realizar cálculos para obtener medidas útiles en el análisis de datos.

2.5. Publicar los resultados

Finalmente, para que los resultados sean accesibles y comprensibles para todos, se publicarán en una página web creada con PowerPages en la que se alojará el cuadro de mando creado con PowerBI previamente. Esto se hace con el fin de que los datos sean accesibles y comprensibles fácilmente por los usuarios. Dicha plataforma permite la creación de páginas web de forma sencilla y rápida sin necesidad de escribir código. Se puede alojar el cuadro de mando de PowerBI simplemente copiando su URL o su código en HTML. Dado que el objetivo es únicamente mostrar información no es necesario implementar una sistema de cuentas. En relación con esto, es suficiente con crear una cuenta de administrador que configure las consultas necesarias.

3. Conceptos teóricos

Para llevar a cabo la realización y comprensión de este proyecto, es necesario conocer una serie de conceptos teóricos que se desarrollan en este capítulo. Estos conceptos son fundamentales para entender el contexto y las herramientas utilizadas en el proyecto. En este capítulo se explican conceptos relacionados con APIs, procesamiento de lenguaje natural (NLP), redes neuronales y sus mecanismos de entrenamiento, así como el manejo de datos y formatos de almacenamiento.

3.1. Application Programming Interface (API)

Una API es un conjunto de definiciones y protocolos utilizados con el objetivo de comunicar dos aplicaciones. [20] Existen diferentes tipos de APIs, pero para este proyecto me centraré en las APIs privadas de Google Places, Overpass y de Apify. A través de una API se pueden realizar peticiones a un servidor para obtener información o realizar acciones. Esto se suele realizar a través del protocolo HTTP y desde Python se puede hacer con la biblioteca requests.

Las peticiones se realizan a través de URLs y pueden incluir parámetros que especifican la información que se desea obtener o la acción que se desea realizar. Cabe destacar que normalmente las APIs tienen límites de uso, es decir, un número máximo de peticiones que se pueden realizar en un periodo de tiempo determinado. Existen varios tipos de peticiones, las más comunes son GET, POST, PUT y DELETE pero para obtener información basta con las GET. Al ser una API privada es necesario obtener acceso a ella mediante una clave o key de API que se obtiene al registrarse en el servicio.

La obtención de esta clave varía dependiendo de la empresa y el servicio, para acceder a la API de Google Places es necesario registrar una dirección de facturación en una cuenta de Google Cloud y para la API de Apify basta con registrarse en su página web.

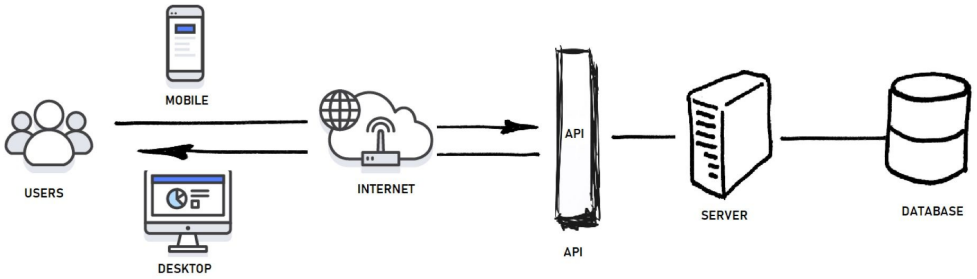

Figura 3.1: Arquitectura de un sistema con una API

Mapas y Geolocalización

Los mapas y la geolocalización son herramientas fundamentales para la visualización y análisis de datos geoespaciales. Permiten representar información geográfica de forma visual y realizar análisis espaciales. En este proyecto se utilizan mapas de uso libre para obtener la ubicación de los establecimientos y las reseñas obtenidas a través de las APIs.

Como se ha mencionado anteriormente, a través de las APIs se pueden obtener datos de forma sencilla. De esta forma se pueden obtener datos de geolocalización como la latitud y longitud de un establecimiento a través de OpenStreetMaps de forma gratuita.

3.2. Procesamiento de Lenguaje Natural (NLP)

El procesamiento de lenguaje natural es una rama de la inteligencia artificial que hace de puente entre la informática y la lingüística. Su objetivo es que las máquinas puedan comprender el lenguaje humano y procesarlo tal y como lo haría un ser humano. [6] Hoy en día se encuentra presente en muchos ámbitos de nuestra vida cotidiana, como por ejemplo en los asistentes virtuales o los traductores automáticos. Respecto a las reseñas, el NLP permite analizar el texto de las reseñas para extraer información relevante, como la opinión del usuario o la polaridad (positiva o negativa) y

estos aspectos pueden ser útiles para mejorar la experiencia del usuario o para realizar análisis de sentimiento.

Para llevar a cabo el procesamiento de lenguaje natural, se utilizan diferentes técnicas y herramientas que permiten analizar y comprender el texto. Las redes neuronales son una de las herramientas más utilizadas en el procesamiento de lenguaje natural, ya que permiten aprender patrones y relaciones complejas en los datos.

3.3. Perceptrón multicapa (MLP)

Una red neuronal es un modelo de aprendizaje automático inspirado en el funcionamiento del cerebro. [14] El perceptrón multicapa (MLP) es un tipo de red neuronal feedforward formado por varias capas. Se encuentra dentro del grupo de las redes neuronales artificiales (ANN). Se compone de una capa de entrada, varias ocultas y una de salida.

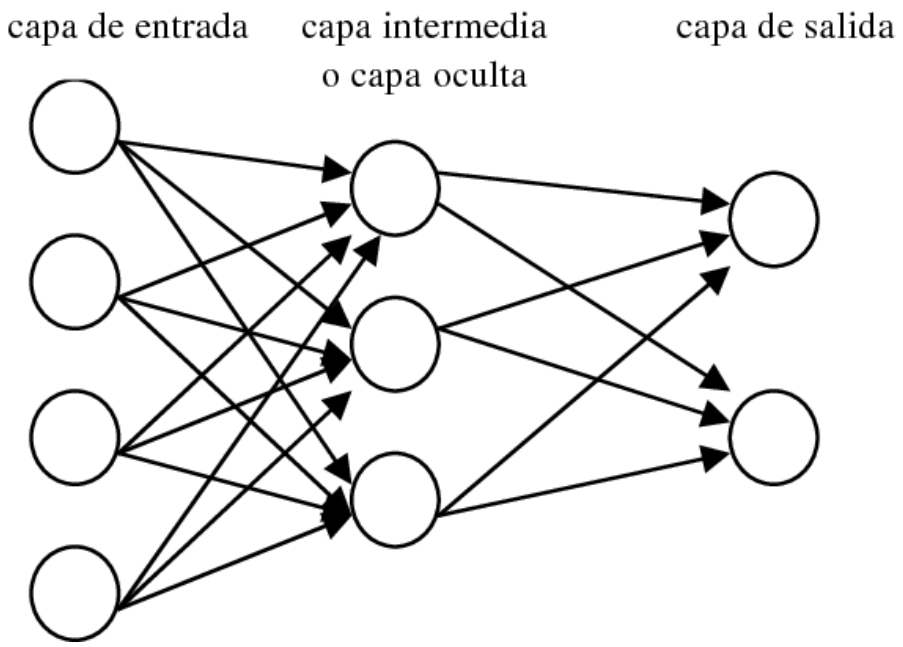

Figura 3.2: Diagrama del perceptrón multicapa

Cada capa está formada por varias neuronas con unos pesos ponderados. A través de las funciones de activación, se realiza una combinación lineal de las entradas y los pesos para obtener la salida de cada neurona. El entrenamiento de la red neuronal sirve para ir modificando y ajustando los

valores de los pesos para que la salida de la red neuronal converja con la esperada.

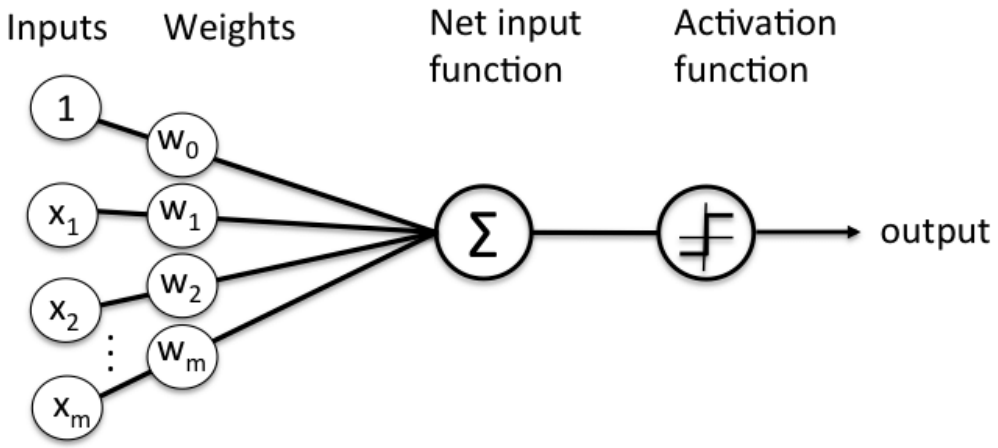

Figura 3.3: Diagrama de una neurona

Representación de datos

Tokenización

Al tratar con textos, es necesario representarlos de una forma que las máquinas puedan entender a través de un preprocesado. Un texto es una secuencia de caracteres, y para que las máquinas puedan procesarlo, es necesario convertirlo en una representación numérica. Para lograrlo se realiza un proceso conocido como tokenización[25] en el que se divide el texto en unidades más pequeñas, llamadas tokens. Existen varios tipos de tokenización dependiendo de la unidad final que se quiera obtener. Un ejemplo de tokenización por palabras y por frases con la frase '*¿Cómo estás?*' sería:

- **Tokenización por palabras:** ['¿', 'Cómo', 'estás', '?']
- **Tokenización por caracteres:** ['¿', 'C', 'o', 'm', 'o', ' ', 'e', 's', 't', 'á', 's', '?']

Vectorización

Para que el modelo pueda trabajar con las cadenas obtenidas tras las tokenización, es necesario convertirlas en vectores numéricos. Para ello se

realiza un proceso de vectorización en el que se asigna un número a cada token, creando así un vocabulario. Este vocabulario es un diccionario que asocia cada token con un número único. Una vez se tiene el vocabulario, se puede representar cada token como un vector numérico. En mi caso, he utilizado BERT Embeddings. Este método crea una representación vectorial basandose en el contexto en el que aparece cada token en el texto. Para ello, utiliza un modelo preentrenado de BERT (Bidirectional Encoder Representations from Transformers).

Codificación y preprocesado de datos

Existen conjuntos de datos etiquetados y no etiquetados. Esto depende de si se dispone de información adicional sobre los datos que los divida en categorías o clases. Posteriormente se mencionará la relación entre estos conceptos y los tipos de aprendizaje. De igual forma que se debe vectorizar el texto, es necesario codificar las etiquetas de los datos para que el modelo pueda trabajar con ellas. Se realiza un proceso de codificación o conversión de las etiquetas a números enteros a través de un encoder que representan cada una de las clases o etiquetas del conjunto de datos previo al entrenamiento del modelo.

A continuación es necesario dividir el conjunto de datos en tres subconjuntos, uno de entrenamiento, otro de validación y otro de test. El entrenamiento se utiliza para entrenar el modelo y el conjunto de validación para ajustar los hiperparámetros. El de test se usa evaluar el rendimiento del modelo una vez entrenado.

También se eliminan las stopwords o palabras vacías. Estas son las palabras que no aportan información relevante al texto y se repiten en la mayoría de veces como artículos, preposiciones o conjunciones.

Problemas de generalización

Hay dos casos que pueden llevar a confusión, sobreentrenamiento (overfitting) y subentrenamiento (underfitting). El sobreentrenamiento ocurre cuando el modelo se ajusta demasiado a los datos del conjunto de entrenamiento logrando memorizarlos y no generaliza bien a los datos de validación, mientras que el subentrenamiento ocurre cuando el modelo no se ajusta lo suficiente a los datos de entrenamiento y no aprende lo suficiente.

Regularización

La regularización es una técnica utilizada para evitar el sobreentrenamiento y mejorar la generalización de las redes neuronales. Existen varias técnicas de regularización[18], pero las utilizadas en el proyecto son:

- **Dropout:** Esta técnica consiste en desactivar aleatoriamente un porcentaje de neuronas durante el entrenamiento. Esto evita que el las neuronas memoricen las entradas. El porcentaje se define directamente en el modelo.
- **Early Stopping:** Esta técnica consiste en detener el entrenamiento cuando la métrica de validación deja de mejorar. En ese caso se guarda el modelo del epoch o ciclo anterior. Para ello, se monitoriza el rendimiento en cada epoch del modelo. Se suele implementar mediante una función llamada callback que es la que se encarga de la monitorización y de detener el entrenamiento cuando sea necesario y conveniente.
- **Data Augmentation:** Esta técnica consiste en aumentar el conjunto de datos de entrenamiento mediante la creación de nuevas muestras a partir de las existentes. Por ejemplo, en el caso del proyecto, se puede cambiar palabras por sinónimas.

. También es importante tener en cuenta que la regularización puede aumentar el tiempo de entrenamiento del modelo, ya que se añaden pasos adicionales al proceso de entrenamiento.

Métricas de evaluación

Hay varias métricas[3] que se puede utilizar para evaluar el rendimiento del modelo y tras interpretarlas llegar a la conclusión de si se da uno de estos dos casos. En un problema de clasificación binario (2 clases) se pueden obtener valores que representan la predicción sobre los datos.

- **Verdaderos Positivos (TP):** Instancias en las que la clase positiva es correctamente predicha por el modelo.
- **Verdaderos Negativos (TN):** Instancias en las que la clase negativa es correctamente predicha por el modelo.
- **Falsos Positivos (FP):** Instancias en las que el modelo predice incorrectamente la clase positiva.

- **Falsos Negativos (FN):** Instancias en las que el modelo predice incorrectamente la clase negativa.

Accuracy

La accuracy (no confundir con la precisión) representa el porcentaje de aciertos positivos y negativos sobre el total. Su fórmula es:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{3.1}$$

Esta métrica es útil cuando las clases están equilibradas, pero puede ser engañosa si hay un desbalance entre las clases. Esto se debe a que un modelo puede tener una alta accuracy simplemente prediciendo la clase mayoritaria, sin aprender realmente de los datos. Esta es una de las métricas que pueden dar sospechas de que el modelo está sobreentrenado o subentrenado, ya que si la accuracy es muy alta en el conjunto de entrenamiento pero baja en el conjunto de validación, es probable que el modelo esté sobreentrenado. Si la accuracy es baja en ambos conjuntos, es probable que el modelo esté subentrenado.

Existe otra variación de la accuracy llamada top k accuracy donde k es el número de clases y que considera correcta una predicción si la clase real está entre las k clases con mayor probabilidad de ser la correcta.

Precisión

La precisión sirve para conocer que porcentaje de valores predichos como positivos son realmente positivos. Su fórmula es:

$$\text{Precisión} = \frac{TP}{TP + FP} \tag{3.2}$$

Esta métrica es útil cuando se quiere minimizar el número de falsos positivos, es decir, cuando se quiere evitar clasificar incorrectamente un caso como positivo.

Recall

El recall representa el porcentaje de verdaderos positivos que han sido identificados correctamente por el modelo. Su fórmula es:

$$\text{Recall} = \frac{TP}{TP + FN} \tag{3.3}$$

F1 Score

Esta métrica es una combinación de la precisión y el recall con el objetivo de encontrar un equilibrio entre ambas y obtener un valor más objetivo. Su fórmula es:

$$\text{F1-Score} = 2 \cdot \frac{\text{Precisión} \cdot \text{Recall}}{\text{Precisión} + \text{Recall}} \tag{3.4}$$

Es una métrica muy útil en problemas de clasificación desbalanceados donde una de las clases es mucho más frecuente que la otra, ya que considera tanto los falsos positivos como los falsos negativos y permite ver si a pesar de una alta accuracy se están identificando correctamente las clases minoritarias.

Loss

Es una métrica que mide como de incorrectas son las predicciones del modelo respecto a los valores reales. [13] Al entrenar un modelo se trata de minimizar esta métrica lo más posible. De la misma forma que la accuracy, si el valor de la loss es muy bajo en el conjunto de entrenamiento pero alto en el conjunto de validación, es probable que el modelo esté sobreentrenado.

Los optimizadores son algoritmos que se utilizan para minimizar la función de pérdida del modelo durante el entrenamiento. Existen varios tipos de optimizadores, pero el utilizado en el proyecto es Adam (Adaptive Moment Estimation), que utiliza una media móvil exponencial del gradiente para ajustar las tasas de aprendizaje.

Matriz de confusión

La matriz de confusión es una tabla que muestra el número de verdaderos positivos, verdaderos negativos, falsos positivos y falsos negativos. Esto se complica cuando se trata de un problema de clasificación multiclase[21], ya que en este caso la matriz de confusión tendrá una fila y una columna por cada clase, es decir que de tener 10 clases tendremos una matriz de tamaño 10x10. En estos casos hay que tener en cuenta como se calculan los valores iniciales:

- **Verdaderos Positivos (TP):** El número de veces que la clase fue correctamente predicha. Corresponde al valor en la diagonal de la matriz para la clase considerada.
- **Verdaderos Negativos (TN):** La suma de los valores de la fila a calcular excepto los de las predicciones de la clase que estamos calculando.

- **Falsos Positivos (FP):** La suma de los valores de la columna de la clase que estamos calculando excepto el valor de la clase positiva.
- **Falsos Negativos (FN):** La suma de los valores de la fila de la clase que estamos calculando excepto el valor de la clase positiva.

	Pred. A	Pred. B	Pred. C
Real A	40	2	3
Real B	4	35	1
Real C	5	3	38

Figura 3.4: Matriz de confusión para un problema de clasificación de 3 clases

Los cálculos para obtener las métricas anteriores para la clase A serían:

- **TP:** M[Real A][Pred A] = 40
- **TN:** M[Real A][Pred B] + M[Real A][Pred C] = 2 + 3 = 5
- **FP:** M[Real B][Pred A] + M[Real C][Pred A] = 4 + 5 = 15
- **FN:** M[Real B][Pred B] + M[Real B][Pred C] + M[Real C][Pred B] + M[Real C][Pred C] = 35 + 1 + 3 + 38 = 77

Funciones de activación

Las funciones de activación son las encargadas de transformar la señal recibida en la de salida que se transmite a la siguiente capa. Estas funciones son fundamentales para que la red neuronal pueda aprender y generalizar patrones en los datos. Las funciones de activación introducen no linealidades en el modelo, lo que permite a la red aprender relaciones complejas entre las entradas y las salidas. Existen varias funciones de activación[1], pero en este proyecto se utiliza Softmax. Softmax es una función transforma la señal de entrada en una distribución de probabilidad sobre las clases. Es útil para problemas de clasificación multiclase. Su fórmula es:

$$\mathrm{f(x)} = \frac{e^{x_i}}{\sum_{j=1}^{K} e^{x_j}} \tag{3.5}$$

Aprendizaje

Existen varios tipos de aprendizaje: supervisado, no supervisado y por refuerzo. Para este apartado me centraré en el supervisado que es el utilizado.

El aprendizaje supervisado es el que cuenta con un conjunto de datos etiquetados, es decir, un conjunto de datos en el que cada entrada tiene una etiqueta o clase asociada. El modelo aprende a partir de estos datos y se ajusta para predecir las etiquetas de nuevas entradas. Existen dos tipos de aprendizaje supervisado, la clasificación y la regresión. La clasificación es el proceso de asignar una etiqueta o clase a una entrada, mientras que la regresión es el proceso de predecir un valor numérico continuo a partir de una entrada. Un ejemplo de clasificación sería predecir si una reseña es positiva o negativa o determinar de que tipo de establecimiento es, mientras que un ejemplo de regresión sería predecir la puntuación de una reseña.[8]

BERT(Bidirectional Encoder Representations from Transformers)

BERT es un modelo de lenguaje de código abierto cread por Google en 2018. Este modelo se basa en una arquitectura de transformadores. Este modelo es capaz de entender el contexto de una palabra en una frase basandose en las palabras que la rodean. Por esta razón se dice que es bidireccional.[24] En la actualidad, hay muchos modelos preentrenados de BERT, incluso en varios idiomas, que se pueden utilizar para tareas de procesamiento de lenguaje natural como clasificación de texto, análisis de sentimiento o respuesta a preguntas.

Los transformadores estan compuestos de dos bloques: el codificador (encoder) y el decodificador (decoder). El codificador se encarga de procesar la entrada y de tokenizar el texto, mientras que el decodificador se encarga de generar la salida.[22]

Entrenamiento y parámetros

Para entrenar una red neuronal, hay que definir varios parámetroos previamente a su entrenamiento.

- **Tasa de aprendizaje (learning rate):** Define cuánto se ajustan los pesos de la red neuronal en cada actualización durante el entrenamiento. Si es demasiado alta, el modelo puede no llegar a una solución óptima; si es demasiado baja, el proceso de entrenamiento será lento.
- **Cantidad de épocas (epochs):** Indica cuántas veces el modelo recorre el conjunto completo de datos de entrenamiento. Un número

excesivo puede causar sobreajuste, mientras que uno insuficiente puede resultar en un modelo poco entrenado.

- **Tamaño del batch (batch size):** Especifica cuántos ejemplos se procesan antes de actualizar los pesos. Un batch grande puede acelerar el entrenamiento pero dificultar la generalización, mientras que uno pequeño puede hacer el proceso más inestable.
- **Configuración de capas y neuronas:** Se refiere a la cantidad de capas ocultas y el número de neuronas en cada una. Demasiadas capas o neuronas pueden provocar sobreajuste; muy pocas pueden limitar la capacidad de aprendizaje del modelo.

También es importante definir las funciones de activación y pérdida así como el optimizador, el callback o los conjuntos de validación y entrenamiento.

3.4. Manejo de datos

Azure SQL Database

Azure SQL Database es un motor de base de datos en la nube de Microsoft Azure. Esta plataforma proporciona una base de datos SQL segura, escalable y de alta disponibilidad.[19] Algunas de sus consultas mas comunes son SELECT para obtener datos, INSERT para insertar nuevos datos, UPDATE para actualizar datos y DELETE para eliminar datos. A través de estas consultas es muy sencillo manejar y filtrar los datos almacenados en la base de datos que pueden ser los datos obtenidos en otras partes del proyecto en otros formatos como CSV o JSON.

Se suele interactuar con Azure SQL Database a través de una conexión ODBC o JDBC, que permite establecer una conexión entre la aplicación y la base de datos. Esto se puede hacer utilizando la biblioteca pyodbc en Python e instalando un driver (ODBC Driver 17 for SQL Server) de la web de Microsoft. También se puede interactuar con la base de datos a través de SQL Server Management Studio (SSMS), que es una herramienta de administración de bases de datos similar a MySQL Workbench proporcionada por Microsoft.

Para conectarse a la base de datos, es necesario proporcionar la cadena de conexión que incluye el nombre del servidor, el nombre de la base de datos, el usuario y la contraseña. Es necesario que la IP del usuario que se

quiera conectar este permitida en las reglas de firewall del servidor de la base de datos.

4. Técnicas y herramientas

4.1. Control de versiones

Git es un sistema de control de versiones que permite llevar un registro de los cambios realizados en un proyecto durante su desarrollo. Permite trabajar de forma colaborativa, visualizar fácilmente el historial de cambios y revertir a versiones anteriores si es necesario. Además, me permite trabajar con varios ordenadores y mantener el proyecto sincronizado entre ellos.

Figura 4.1: Logotipo de Git

4.2. Alojamiento de código

GitHub es una plataforma de alojamiento de código que utiliza Git como sistema de control de versiones. Permite almacenar el código fuente de un proyecto, colaborar con otros desarrolladores y gestionar el historial de cambios. Además, GitHub ofrece características como la gestión de incidencias, revisiones de código y documentación del proyecto. Adicionalmente permite la creacion de tableros de proyectos para organizar tareas y hacer seguimiento del progreso del proyecto. Por último y como se mencionó anteriormente, permite la integración con Visual Studio Code para gestionar el repositorio directamente desde el IDE.

Figura 4.2: Logotipo de GitHub

4.3. Entornos de desarrollo integrado (IDE)

Visual Studio Code

Principalmente se ha utilizado Visual Studio Code como IDE para la creación de los scripts. Visual Studio Code permite la ejecución de dichos scripts, así como su depuración. Por último y como se mencionó anteriormente, permite integrar una cuenta de GitHub para gestionar el repositorio del proyecto directamente desde el IDE. Además, se puede instalar varias extensiones de gran ayuda para el desarrollo del proyecto.

Figura 4.3: Logotipo de VisualStudioCode

Extensiones

- **Python**: Proporciona soporte para el lenguaje Python, incluyendo resaltado de sintaxis, autocompletado y depuración.
- **LaTeX Workshop**: Permite trabajar con LaTeX, facilitando la edición y compilación de documentos LaTeX.
- **GitHub**: Permite integrar una cuenta de GitHub para gestionar el repositorio del proyecto directamente desde el IDE.
- **CSVLint**: Permite validar y formatear archivos CSV, y permite ver visualmente los diferentes campos del fichero.

Google Colab

Adicionalmente, se ha utilizado Google Colab para el desarrollo y ejecución e la red neuronal. Google Colab es un entorno de desarrollo basado en la nube y accesible con una cuenta de Google que permite ejecutar código Python en notebooks Jupyter. La principal ventaja de utilizar Google Colab es que de forma gratuita permite utilizar GPUs para acelerar el entrenamiento de modelos de aprendizaje automático reduciendo el tiepmo de ejecución.

Figura 4.4: Logotipo de Google Colab

4.4. Lenguajes de programación

Python

Python es un lenguaje de programación de alto nivel, interpretado y multipropósito. Es comunmente utilizado en el ámbito del análisis de datos, la inteligencia artificial y el desarrollo web. En este proyecto, se ha utilizado Python para desarrollar los scripts de extracción de datos y la red neuronal para la clasificación de los puntos de interés.

Figura 4.5: Logotipo de Python

LaTeX

LaTeX es un sistema de preparación de documentos que permite crear documentos de alta calidad tipográfica, especialmente en el ámbito académico

y científico. También permite trabajar con LaTeX tras instalar Strawberry, una extensión que facilita la edición y compilación de documentos LaTeX. Para la redacción de la memoria del proyecto, se ha utilizado Visual Studio Code con la extensión LaTeX Workshop.

Figura 4.6: Logotipo de LaTeX

Markdown

Markdown es un lenguaje de marcado ligero que permite escribir texto con formato de manera sencilla. En este proyecto se ha utilizado para la redacción del fichero README.md del repositorio.

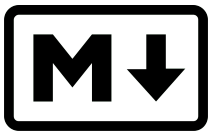

Figura 4.7: Logotipo de Markdown

SQL

Para la gestión de la base de datos se ha usado SQL como lenguaje de consulta. Permite almacenar, modificar y extraer información de manera eficiente. Adicionalmente, se ha utilizado SSMS como herramienta para diseñar y gestionar la base de datos de manera más amigable.

Figura 4.8: Logotipo de SQL

Overpass QL

Overpass Query Language (Overpass QL) es un lenguaje de consulta utilizado para extraer datos de OpenStreetMap. Permite realizar consultas

complejas sobre los datos geoespaciales de OSM, filtrando por diferentes criterios como tipo de elemento, ubicación, etiquetas, etc.

Figura 4.9: Logotipo de Overpass Turbo

DAX

DAX (Data Analysis Expressions) es un lenguaje de fórmulas utilizado en Power BI. Permite realizar cálculos y consultas sobre los datos importados en Power BI, facilitando la creación de medidas y columnas calculadas.

Figura 4.10: Logotipo de DAX

4.5. Librerías

Pandas

Pandas es una biblioteca de Python que permite manipular estructuras de datos como dataframes para tratar con la información. Además, ayuda a analizar datos de manera eficiente, facilitando tareas como la limpieza, transformación y visualización de datos.

Time

La libería Time de Python proporciona funciones para trabajar con el tiempo y las fechas. Su principal uso y objetivo es medir el tiempo transcurrido durante la ejecución de un programa, lo que permite optimizar el rendimiento.

Requests

Requests es una biblioteca de Python que permite realizar solicitudes de manera sencilla. Permite enviar peticiones a servidores web y recibir respuestas. Se ha utilizado para realizar peticiones a la API de Google Places.

Multiproccessing

Multiproccessing es una biblioteca de Python que permite la ejecución de tareas en paralelo utilizando múltiples procesos. Para el proyecto se ha utilizado Value, una clase de la biblioteca Multiproccessing, que permite compartir datos entre procesos.

JSON

JSON es una biblioteca de Python que permite trabajar con datos en formato JSON (JavaScript Object Notation). Para el proyecto se ha utilizado para almacenar los datos extraídos Overpass Turbo y extraer los placeIds de la API de Google Places.

CSV

CSV es una biblioteca de Python que permite trabajar con archivos CSV (Comma-Separated Values). CSV es un formato de archivo utilizado para almacenar información. Cada línea del archivo representa una fila y los campo está separado por comas. En este proyecto se ha utilizado para almacenar los placeIds de los puntos de interés extraídos de la API de Google Places.

Concurrent.futures

Concurrent.futures es una biblioteca de Python que permite ejecutar tareas de forma concurrente utilizando hilos o procesos. Permite paralelizar la ejecución de tareas, lo que puede mejorar el rendimiento en operaciones que requieren mucho tiempo de procesamiento como la extracción de los placeIds de los puntos de interés a partir de la API de Google Places.

Apify_client

Apify_client es una biblioteca que permite la creación de un agente de Apify, una plataforma que permite la extraccuón de reviews a partir de la

URL o de un placeId de Google Maps. Permite interactuar con la API de Apify para realizar tareas como la ejecución de agentes, la gestión de datos y la monitorización de tareas.

dotenv

Dotenv es una biblioteca de Python que permite cargar variables de entorno desde un archivo .env en el proyecto. Se utiliza para gestionar los datos sensibles como las claves de las APIs de Google Places y Apify, evitando que se expongan directamente en el código fuente.

Gender_guesser

Gender_guesser es una biblioteca de Python que permite predecir el género de una persona a partir de su nombre. Necesita recibir un nombre como parámetro y devuelve el género asociado a ese nombre.

TensorFlow

TensorFlow es una biblioteca de código abierto que permite la creación y entrenamiento de modelos de aprendizaje automático. En este proyecto se ha utilizado para crear y entrenar la red neuronal que clasifica los puntos de interés en función de su tipo.[28]

Sklearn

Sklearn es una biblioteca de Python que proporciona herramientas para el aprendizaje automático. Principalmente se ha utilizado para crear los conjuntos de entrenamiento y prueba, así como para obtener métricas interesantes en el entrenamiento del modelo como el F1-score.

pyodbc

Pyodbc es una biblioteca de Python que permite conectarse a bases de datos utilizando ODBC (Open Database Connectivity). En este proyecto se ha utilizado para conectarse a la base de datos de Azure SQL Database y realizar consultas sobre ella desde Python. Es necesario instalar un driver ODBC para SQL Server, como el .ºDBC Driver 17 for SQL Server.ª cual se puede encontrar en la web de Microsoft.

pysentimiento

Pysentimiento es una biblioteca de Python que permite realizar análisis de sentimiento en texto en español. A través de esta biblioteca se puede obtener una función de polaridad para un texto que indica si el texto es positivo, negativo o neutral.

langdetect

Langdetect es una biblioteca de Python que permite detectar el idioma de un texto. Utiliza un modelo de aprendizaje automático para identificar el idioma del texto y devuelve el código ISO del idioma detectado. Algunos ejemplos de códigos ISO son .^{es}"para español, .^{en}"para inglés o "fr"para francés.

deep_translator

Deep_translator es una biblioteca de Python que permite traducir texto entre diferentes idiomas utilizando diferentes servicios de traducción. En este proyecto se ha utilizado para traducir los nombres de los puntos de interés a español para el entrenamiento de la red neuronal. Previamente se ha utilizado langdetect para detectar el idioma del nombre del punto de interés y así traducirlo al español si es necesario.

re

Re es una biblioteca de Python que permite trabajar con expresiones regulares. Se ha utilizado en el preprocesamiento de las reseñas para eliminar caracteres o secuencias no deseadas.

4.6. Base de datos

Como gestor de la base de datos se ha utilizado Azure SQL Database. Permite almacenar, modificar y extraer información de manera eficiente en la nube a través de Azure. Además, se ha utilizado SQL Server Management Studio (SSMS) para realizar consultas sobre la base de datos de forma visual.

Figura 4.11: Logotipo de SSMS

4.7. Análisis de datos

Se ha considerado el uso de PowerBI y Google Data Studio como herramientas para el análisis de datos y la creación de cuadros de mando interactivos. Dado que PowerPages permite la integración del dashboard con PowerBI, se ha optado por esta herramienta para la creación del cuadro de mando del proyecto.

PowerBI es una herramienta de análisis de datos que permite crear informes y cuadros de mando interactivos. Permite importar datos de diversas fuentes, transformarlos y visualizarlos de manera clara y concisa. El hecho de que permita la importación de datos desde una base de datos SQL de Azure facilita la decisión de utilizar esta herramienta para la creación del cuadro de mando del proyecto. Además es una herramienta muy fácil e intutitiva de utilizar.

Figura 4.12: Logotipo de PowerBI

4.8. Publicación de resultados

PowerPages es una plataforma de Microsoft que permite crear sitios web y aplicaciones web de manera rápida y sencilla. Además, ofrece la posibilidad de integrar el cuadro de mando creado en PowerBI para que los usuarios puedan ver e interactuar con los datos de manera visual.

Figura 4.13: Logotipo de PowerPages

4.9. Defectos de código

Docker

Docker es una plataforma que permite crear, desplegar y ejecutar aplicaciones en contenedores. Esto facilita la ejecución de aplicaciones en diferentes entornos o máquinas ya que no es necesario instalar toads las dependencias manualmente. En este proyecto se ha utilizado Docker para ejecutar SonarQube, facilitando su instalación y configuración.

Figura 4.14: Logotipo de Docker

SonarQube

Para la detección de defectos de código se ha usado SonarQube. SonarQube es una plataforma de código abierto que permite analizar el código fuente de un proyecto para detectar errores, fallos de seguridad yO malas prácticas entre otras cosas. Gracias a su análisis ha sido posible detectar posibles defectos de seguridad.

Figura 4.15: Logotipo de SonarQube

5. Aspectos relevantes del desarrollo del proyecto

Se presentan a continuación los aspectos más relevantes del desarrollo del proyecto de forma ordenada según las diferentes fases del proyecto.

5.1. Extracción de datos

Propuestas de extracción de datos

A continuación se presentan las diferentes propuestas de extracción de datos que se han considerado para la obtención de los datos necesarios para el desarrollo del proyecto:

- **TripAdvisor API:** La API de TripAdvisor contiene millones de reviews de usuarios sobre destinos turísticos de todo el mundo. Debido a sus límites de reviews[15] (5 por lugar, es decir, 5 reviews por solicitud) no es factible. Ofrecen 5000 peticiones gratuitas al mes. [16]

Solicitudes	Costo por solicitud
0 - 5,000	€0.00
5,001 - 20,000	€0.00876
20,001 - 100,000	€0.00815
100,001 - 500,000	€0.00762
500,000+	€0.00718

Tabla 5.1: Costos por solicitud en la API de TripAdvisor

- **API de Google Places (textSearch) y Google Maps Reviews Scraper de Apify:** Utilizando textSearch se puede obtener un identificador para cada POI de forma gratuita e ilimitada tras crear una cuenta de prueba e introducir una tarjeta de crédito. Para ello, es necesario tener un punto de referencia (coordenadas) y un radio de búsqueda. Los puntos de referencia se pueden obtener mediante una consulta en Overpass Query Language desde su interfaz web para extraer las coordenadas de OSM.

  Al contrario que en el caso anterior, ahora no se emplean los puntos límites de los municipios, solo un punto céntrico de cada uno de ellos. El problema en este caso es que no se obtienen todos los POI de un municipio, sino solo los que se encuentran dentro del radio de búsqueda.

- **API de Google Places (nearbySearch) y ficheros binarios de OpenStreetMaps(OSM):** OSM proporciona ficheros binarios (.osm.pbf) que se pueden convertir a GeoJSON para obtener información relevante sobre los municipios como las coordenadas de sus límites. Con la API de Google Places se pueden extraer reviews utilizando nearbySearch que permite obtener las reviews de los POI más relevantes que se encuentren cerca de un punto dado por sus coordenadas.

  Sin embargo, esta API solo permite obtener 5 reviews por lugar y 60 lugares por cada búsqueda alrededor de un punto. Esto implica que si se quiere obtener información de un municipio con más de 60 POI, se deben realizar múltiples búsquedas y muchas peticiones.

- **Consultas de Overpass QL, API de Google Places (nearbySearch) y Google Maps Reviews Scraper de Apify:** A través de la interfaz web de Overpass QL al igual que en el caso anterior, se pueden obtener los límites de los municipios mediante consultas en Overpass QL. Esta herramienta permite extraer reviews de Google Maps de forma ilimitada y gratuita.

  Con estos identificadores se puede automatizar la extracción de reviews a través de Apify que ofrece 14285 reviews gratuitas por cada cuenta gratuita creada. A partir de esas reviews, se debe pagar una cuota mensual teniendo en cuenta que cada 1000 reviews equivalen a 0,35$. Con la cuenta Business esto cambia ya que el precio de una review es de 0,00035$ mientras que con las anteriores es de 0,0006.$ También ofrecen un descuento del 50 % para cuentas educativas[10]. También ofrecen un descuento de 10 % si se paga anualmente.

Se muestra una tabla con los precios de las cuentas de Apify[11] en dólares américanos:

Tipo de cuenta	Número de reviews	Precio/mes al 50 %
Free	14285	0$
Starter	111428	19,5$
Scale	568571	99,5$
Business	2854285	499,5$
Enterprise	Ilimitado	Hablar con Apify

Tabla 5.2: Costos por solicitud en Apify

Carga de datos a la base de datos

Debido a las limitaciones de las propuestas anteriores, únicamente para cargar la base de datos con los recursos a mostrar en el cuadro de mando se ha optado por utilizar un scraper de código abierto disponible en GitHub[4] que permite extraer las reseñas y los recursos de forma gratuita y más masiva. Dado que el proyecto tiene un enfoque educativo, el desarrollador de dicho scraper me ha ofrecido la versión de pago de forma gratuita.

5.2. Clasificación de los datos

Con el objetivo de crear un conjunto de datos para entrenar la red neuronal, se han extraído reseñas de puntos de interés con categorías similares para después clasificarlas en grupos según su categoría. Para realizar esta tarea se ha ejecutado el pipeline de extracción de datos descrito anteriormente. Una vez obtenidas las reseñas se ha montado manualmente un conjunto de datos etiquetado y multiclase con las reseñas obtenidas.

Exploratory Data Analysis (EDA)

Previamente al entrenamiento de la red neuronal, se ha realizado un análisis exploratorio de los datos (EDA) para ver el conjunto de datos de forma visual. Esto permite ver la distribución de reseñas por recurso o la longitud de las reseñas.

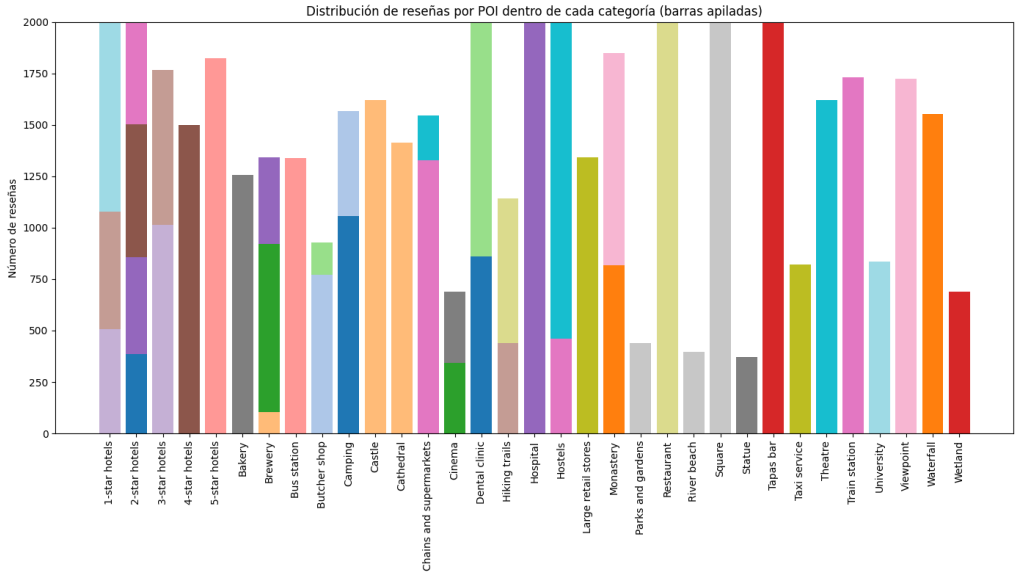

Figura 5.1: Distribución de reseñas por recurso

5.3. Aprendizaje automático

Para el entrenamiento de la red neuronal se ha utilizado el conjunto de datos etiquetado y multiclase obtenido en la fase anterior. Debido al tiempo de ejecución del entrenamiento de la red neuronal se ha decidido ejecutar el código en Google Colab, un entorno en línea ofrecido por Google de forma gratuita que permite ejecutar código Python en la nube y utilizar GPUs para acelerar el entrenamiento de modelos de aprendizaje automático. El código se ha desarrollado en Python utilizando las librerías TensorFlow y Keras para el entrenamiento de la red neuronal.

Primero se carga el dataset en un DataFrame de Pandas y a continuación se realiza un preprocesamiento con el objetivo de eliminar filas con valores nulos o vacías que no aportan información relevante al modelo. A continuación, se realiza una codificación a números enteros de las etiquetas de las reseñas con el LabelEncoder de la librería Scikit-learn. Después se cargan el modelo y tokenizer BERT ya preentrenados para utilizarlos en el preprocesamiento de las reseñas. (dccuchile/bert-base-spanish-wwm-cased)

Se divide el dataset en un conjunto de entrenamiento, validación y test utilizando la función train_test_split de Scikit-learn. A continuación, se realiza un proceso de data augmentation para aumentar el número de reseñas en el conjunto de entrenamiento. Este proceso está basado en cambiar

sinónimos de palabras en las reseñas para generar nuevas reseñas que aporten más información al modelo evitando usar stopwords.

Se crea un vectorizador de texto TF-IDF para transformar las reseñas en vectores numéricos que puedan ser utilizados por el modelo de red neuronal. Para obtener los embeddings de los subconjuntos de datos, se utiliza el modelo BERT preentrenado y el vectorizador de texto TF-IDF para cada subconjunto, concatenando ambos vectores al final del proceso en cada caso.

Cabe destacar que los siguientes hiperparámetros se han obtenido a través de un proceso de random search con el objetivo de encontrar la mejor combinación de hiperparámetros para el modelo.

Para acabar se crea el modelo de la red neuronal. Las capas del modelo son las siguientes:

- Capa densa con 512 unidades y función de activación ReLU.
- Capa de dropout con una tasa de 0,3 para evitar el sobreajuste del modelo.
- Capa densa con 64 unidades y activación ReLU.
- Capa de dropout con una tasa de 0,2 para evitar el sobreajuste del modelo.
- Capa de salida en forma de capa densa con una activación softmax para la clasificación multiclase.

El modelo se compila con el optimizador Adam con una tasa de aprendizaje de 0.00017954 y la función de pérdida categorical_crossentropy. Finalmente, se entrena el modelo con 15 epochs y un tamaño de lote de 32 aplicando los callbacks definidos anteriormente.

Previamente al entrenamiento del modelo se define una función de callback para que el modelo se detenga con Early Stopping si la validation_loss no disminuye. De la misma forma, se define otro callback para disminuir la tasa de aprendizaje en caso de ser necesario.

Finalmente se entrena el modelo con el conjunto de entrenamiento y validación con 25 epochs y un tamaño de lote de 32. Tras su finalización, se obtiene el informe de clasificación del modelo y la matriz de confusión y se guarda el modelo entrenado para su uso posterior.

Resultados del entrenamiento

Estos son los resultados obtenidos tras el entrenamiento del modelo:

	precision	recall	f1-score	support
2-star hotels	0.58	0.47	0.52	175
3-star hotels	0.63	0.66	0.65	265
4-star hotels	0.71	0.64	0.67	225
5-star hotels	0.79	0.71	0.75	274
Bakery	0.84	0.87	0.86	188
Brewery	0.83	0.83	0.83	201
Bus station	0.76	0.72	0.74	201
Butcher shop	0.80	0.76	0.78	139
Camping	0.88	0.83	0.86	235
Castle	0.82	0.77	0.79	243
Cathedral	0.84	0.83	0.83	212
Chains and supermarkets	0.67	0.68	0.68	94
Cinema	0.85	0.83	0.84	104
Hiking trails	0.80	0.87	0.83	171
Hospital	0.96	0.98	0.97	338
Hostels	0.80	0.86	0.83	511
Large retail stores	0.86	0.83	0.84	201
Monastery	0.87	0.85	0.86	278
Parks and gardens	0.74	0.65	0.69	66
Restaurant	0.75	0.78	0.76	390
River beach	0.84	0.82	0.83	60
Statue	0.90	0.79	0.84	56
Tapas bar	0.76	0.80	0.78	443
Taxi service	0.84	0.92	0.88	123
Theatre	0.95	0.93	0.94	243
Train station	0.83	0.88	0.86	260
University	0.82	0.89	0.85	125
Viewpoint	0.86	0.87	0.86	259
Waterfall	0.83	0.88	0.85	233
Wetland	0.80	0.77	0.78	103
accuracy			0.81	6416
macro avg	0.81	0.80	0.80	6416
weighted avg	0.81	0.81	0.81	6416

Figura 5.2: Informe de resultados

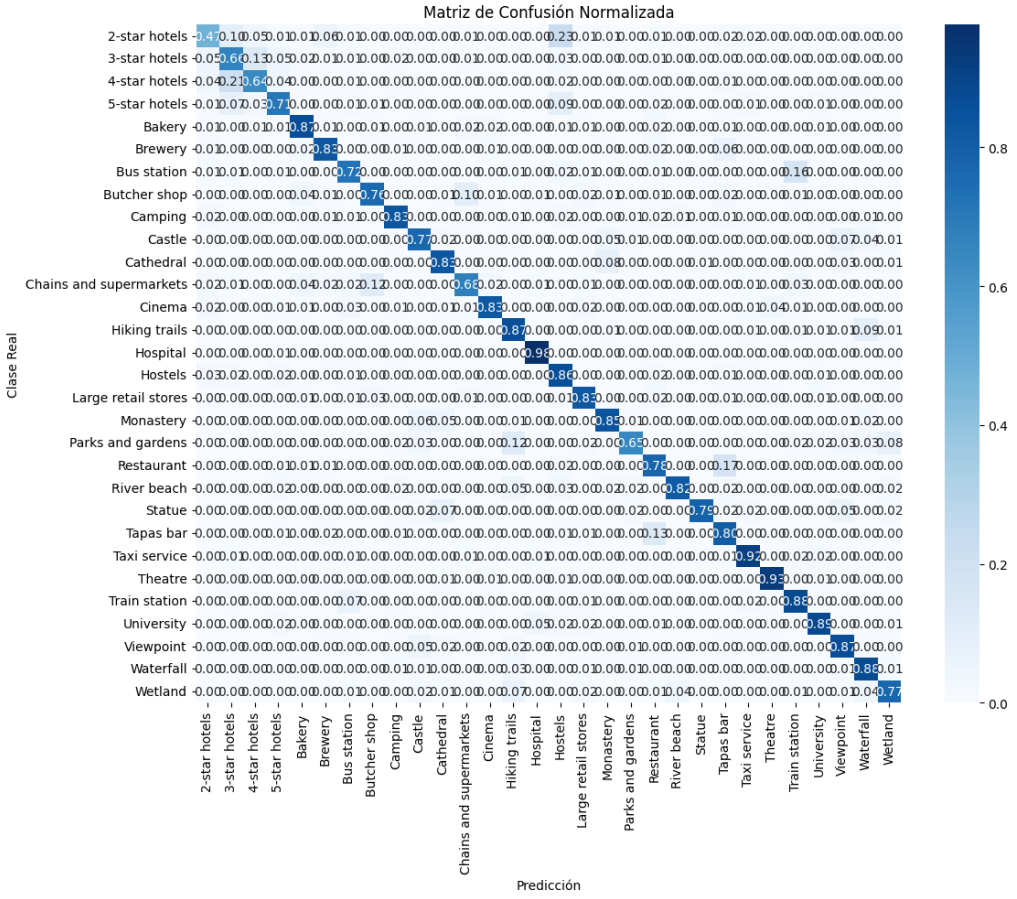

Figura 5.3: Matriz de confusión

Sistema de recomendación

Utilizando clustering K-means se ha creado un sistema de recomendación que permite recomendar puntos de interés a los usuarios en función del texto de las reseñas y la localización del mismo. Estos datos se almacenan en la base de datos para cargarlos sobre el cuadro de mando de Power BI.

5.4. Análisis de reseñas

Modelo de base de datos

Con el objetivo de recoger métricas interesantes sobre las reseñas extraídas en la primera fase, se ha creado un cuadro de mando en Power BI. Este cuadro de mando contiene diferentes visualizaciones que permiten analizar

las reseñas extraídas de los puntos de interés de la provincia de Burgos. Se ha creado un script que se encarga de extraer el nombre de las ciudades o municipios y el nombre de los puntos de interés y guardarlos en la tabla POI. Además, se ha creado otro script que se encarga de extraer información relevante sobre las reseñas: el texto de la reseña, la categoría ofrecida por Apify, la valoración, el nombre del POI y el nombre del usuario entre otras cosas. Previamente a insertar esta información en la tabla reviews, se usa la biblioteca gender_guesser para predecir el génerio del usuario y guardarlo junto al resto de información en la tabla. Finalmente se ha creado la base de datos en la nube de Microsoft Azure y se ha importado la información. Para importar los datos, se ha creado una conexión a la base de datos desde Power BI.

La métrica más interesante analizada en Power BI es el TORI. Su fórmula es la siguiente:

$$TORI(d_n, c_m) = \sum_{ap \in AP_{d_n,c_m}} \frac{\left(TS(ap) + \frac{5}{4} \cdot FP(ap)\right) \cdot N(ap)}{\sum_{ap \in AP_{d_n,c_m}} N(ap)} \tag{5.6}$$

Donde D es el conjunto de destinos, C es el conjunto de categorías, AP_{d_n,c_m} es el conjunto de reseñas de un destino d_n y una categoría c_m, $TS(ap)$ es la puntuación de la reseña ap y $N(ap)$ es el número de reseñas de la reseña ap. El objetivo de esta métrica es comparar de forma objetiva los puntos de interés o destinos turísticos y determinar cuáles son los más relevantes en función de las reseñas obtenidas. Estos calculos se realizan de forma automática en Power BI a través de medidas DAX que permiten realizar cálculos complejos sobre los datos importados.

Además de esto, se han creado visualizaciones para ver la distribución por categorías, fechas y género del usuario de las reseñas obtenidas así como un mapa y una ontología para los tipos de recursos.

5.5. Publicación del cuadro de mando

Finalmente, se ha publicado el cuadro de mando en Power BI para que pueda ser consultado por cualquier persona interesada en el análisis de las reseñas obtenidas. Se ha utilizado PowerPages para crear una página web que permite acceder al cuadro de mando de forma sencilla y rápida.

Figura 5.4: Cuadro de mando creado

5.6. Flujo de trabajo

Se muestra a continuación el flujo de trabajo del proyecto en forma de diagrama junto a las herramientas utilizadas en cada fase del proyecto.

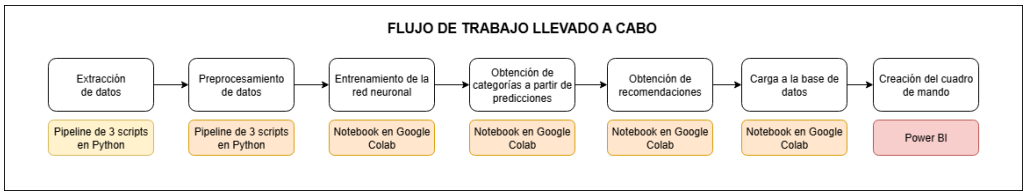

Figura 5.5: Flujo de trabajo

6. Trabajos relacionados

En este capítulo se presentan los trabajos y artículos relacionados con el contenido del proyecto que han sido de mayor relevancia para su desarrollo.

- **Trabajo de Fin de Grado. Inteligencia Artificial aplicada a Destinos Turísticos Inteligentes:** Este proyecto nace de un trabajo de fin de grado realizado por Isabel Marcilla Lombraña para la Universidad de Burgos en 2024. El trabajo se centra en el uso de la inteligencia artificial para mejorar la experiencia del usuario en destinos turísticos inteligentes, y proporciona una base sólida para el desarrollo de este proyecto. Precisamente, el principal objetivo era tratar de llevar el desarrollo del mismo más lejos automatizando la recogida de reseñas que se hacía manualmente. También fue de gran ayuda para determinar algunas categorías a diferenciar en este proyecto por el modelo de clasificación[23]

- **FREE Google Maps Reviews Scraper API - How to Scrape Google Reviews:** Este vídeo tutorial de YouTube explica como utilizar el actor Google Maps Review Scraper de Apify. Al inicio del proyecto cuando estuve buscando formas de extraer las reseñas fue de gran ayuda debido a su bajo coste respecto a otros y a la posibilidad de ejecutarlo directamente desde Python. [2]

- **Documentación de la API de Apify:** La documentación oficial de la API de Apify fue esencial para entender como desarollar el código que interactúa con la API y extrae las reseñas de Google Maps. Proporciona ejemplos claros y una guía completa sobre cómo utilizar sus servicios. [9]

- **Documentación de Google Maps Places API:** Al principio del proyecto estuve probando a extraer las reseñas directamente desde la API de Google Maps. Para ello, me basé en la documentación oficial de Google Maps Places API y estuve probando varias formas y funciones de lograr mi objetivo. A pesar de que finalmente no fue la opción elegida para extraer las reseñas debido a su elevado coste, sirvió para un paso intermedio en el que se extraen los placeIds para utilizarlos posteriormente en la API de Apify. [12]
- **Documentación de Tensorflow Keras:** Durante la creación del modelo de clasificación de reseñas, la documentación de Tensorflow Keras fue de gran ayuda para comprender la sintaxis a utilizar. Además, proporciona ejemplos y guías sobre cómo construir y entrenar modelos de aprendizaje automático, lo que facilitó el proceso de implementación del modelo de clasificación. [27]

7. Conclusiones y Líneas de trabajo futuras

Tras finalizar el desarrollo del proyecto, es importante reflexionar sobre los resultados obtenidos y las lecciones aprendidas. Estas reflexiones no solo ayudan a evaluar el éxito del proyecto, sino que también proporcionan una base para futuras mejoras y desarrollos. En este capítulo se presentan las conclusiones generales del proyecto y se realiza un análsis crítico sobre el mismo para proponer posibles líneas de trabajo futuras.

7.1. Conclusiones

Puedo decir que el proyecto ha tenido un éxito parcial en la consecución de sus objetivos. Esto se debe a que se han logrado implementar todas las funcionalidades requeridas pero no de la forma más óptima. Esto se debe principalmente al elevado costo que supone el desarrollo de un proyecto de estas características. Lograr la automatización de las reseñas fue un gran desafío. Extraer tal cantidad de reseñas desde una API requiere un elevado costo económico y temporal lo que ha limitado la magnitud del proyecto. A pesar de ello, se ha conseguido un producto funcional que cumple con los requisitos establecidos al inicio del proyecto.

A pesar de las limitaciones, puedo decir que he aprendido mucho durante el desarrollo del proyecto. He aprendido a trabajar con APIs y he adquirido experiencia en el manejo de datos. Además, he tenido la oportunidad de aplicar mis conocimientos teóricos en un proyecto práctico de mayor escala a lo que estoy acostumbrado, lo que ha sido muy enriquecedor. He podido aprender a trabajar con bibliotecas como Keras, que es una de las más

utilizadas en el campo del aprendizaje automático o a documentar con LaTeX, que es una herramienta muy útil para la creación de documentos técnicos y científicos. También cabe destacar que he aprendido a organizar el desarrollo de un proyecto software y a gestionar las tareas de desarrollo durante el proyecto.

Creo que en general ha sido una tarea ardua y compleja pero enriquecedora. Algo que no se muestra en el proyecto es el tiempo invertido en la investigación y pruebas de diferentes opciones descartadas hasta llegar a la solución final. Puedo decir con casi total seguridad que ese tiempo ha sido mayor que el tiempo invertido en el desarrollo del proyecto final en sí.

7.2. Líneas de trabajo futuras

A pesar de que el proyecto ha sido un éxito parcial, hay muchas áreas que se pueden mejorar y desarrollar en el futuro. Algunas de las posibles líneas de trabajo futuras incluyen:

- **Mejorar la extracción de datos:** Encontrar una forma de extraer las reseñas de forma más eficiente y completa. Debido al elevado coste económico que conlleva esto ha sido muy difícil lograr extraer las reseñas. Esto podría lograrse mediante alguna API futura. Es importante considerar la legalidad de estas técnicas y el fin del proyecto.
- **Ampliación del conjunto de datos:** Ampliar la base de datos para incluir más subcategorías y obtener un conjunto de datos más diverso. Esto podría mejorar la generalización del modelo y su capacidad para hacer predicciones más precisas. Dependiendo de la ampliación de los datos, incluso podría cambiar el modelo de la base de datos.
- **Mejora del modelo de aprendizaje automático:** Se pueden explorar diferentes arquitecturas de redes neuronales y técnicas de optimización para mejorar la precisión del modelo.
- **Mejora del sistema de recomendación:** Desarrollar una mejora del sistema de recomendación más robusto que sugiera recursos a los usuarios en función de sus preferencias y del análisis de las reseñas. Esto podría mejorar la experiencia del usuario y aumentar la satisfacción de los potenciales viajeros o clientes interesados.

Bibliografía

[1] M. All. Introducción a las funciones de activación en las redes neuronales. https://www.datacamp.com/es/tutorial/introductionto-activation-functions-in-neural-networks?dc_referrer=https%3A%2F%2Fwww.google.com%2F, 2024. [Internet; último acceso 11-Junio-2025].

[2] Apify. Free google maps reviews scraper api - how to scrape google reviews. https://www.youtube.com/watch?v=Qrt6Jm0uOsE&t=1s, 2024. [Internet; último acceso 13-Junio-2025].

[3] R. Díaz Badra. Métricas de clasificación. https://www.themachinelearners.com/metricas-de-clasificacion/, 2025. [Internet; último acceso 10-Junio-2025].

[4] Cheetan11dev. omkarcloud/google-maps-scraper. https://github.com/omkarcloud/google-maps-scraper, 2025. [Internet; último acceso 2-Julio-2025].

[5] D. Cortes. ¿qué es el turismo? https://www.cesuma.mx/blog/que-es-el-turismo.html, 2025. [Internet; último acceso 13-Junio-2025].

[6] Universidad de diseño; innovación y tecnología. ¿qué es el procesamiento del lenguaje natural (nlp)? https://www.xataka.com/basics/api-que-sirve, 2024. [Internet; último acceso 11-Junio-2025].

[7] Secretaría de Estado de Turismo y SEGITTUR. ¿qué es el modelo dti? - dti. https://www.destinosinteligentes.es/que-es-dti/, 2025. [Internet; último acceso 13-Junio-2025].

[8] P. Recuero de los Santos. Tipos de aprendizaje en machine learning: supervisado y no supervisado. https://telefonicatech.com/blog/que-algoritmo-elegir-en-ml-aprendizaje, 2021. [Internet; último acceso 11-Junio-2025].

[9] Apify Developers. Apify api client for python. https://docs.apify.com/api/client/python/docs/overview/introduction, 2025. [Internet; último acceso 14-Mayo-2025].

[10] Apify Developers. Apify for universities. https://apify.com/resources/universities, 2025. [Internet; último acceso 29-Abril-2025].

[11] Apify Developers. Apify pricing. https://apify.com/pricing, 2025. [Internet; último acceso 29-Abril-2025].

[12] Google Developers. Documentación de google maps platform. https://developers.google.com/maps/documentation/places/web-service?hl=es-419, 2025. [Internet; último acceso 10-Mayo-2025].

[13] Google Developers. Regresión lineal: Pérdida. https://developers.google.com/machine-learning/crash-course/linear-regression/loss?hl=es-419, 2025. [Internet; último acceso 11-Junio-2025].

[14] Google Developers. ¿qué es una red neuronal? https://cloud.google.com/discover/what-is-a-neural-network?hl=es-419, (s.f). [Internet; último acceso 2-Julio-2025].

[15] TripAdvisor Developers. Overview. https://tripadvisor-contentapi.readme.io/reference/overview, 2023. [Internet; último acceso 31-Marzo-2025].

[16] TripAdvisor Developers. Tripadvisor. https://www.tripadvisor.com/apidevelopersscheckout, 2025. [Internet; último acceso 29-Abril-2025].

[17] C. A. Núñez Duque. cnunez1/tfg-digitalizacionboletinturismo. https://github.com/cnunez1/TFG-DigitalizacionBoletinTurismo, 2025. [Internet; último acceso 13-Junio-2025].

[18] J. Durán. Técnicas de regularización básicas para redes neuronales. https://medium.com/metadatos/t%C3%A9cnicas-de-

regularizaci%C3%B3n-b%C3%A1sicas-para-redes-neuronales-b48f396924d4, 2019. [Internet; último acceso 11-Junio-2025].

[19] KCP Dynamics. Microsoft azure sql database - microsoft dynamics partners latam | kcp dynamics. https://kcpdynamics.com/microsoft-azure-sql-database/, 2023. [Internet; último acceso 2-Julio-2025].

[20] Y. Fernández. Api: qué es y para qué sirve. https://www.xataka.com/basics/api-que-sirve, 2019. [Internet; último acceso 11-Junio-2025].

[21] A. Jain. Confusion matrix for multiclass classification. https://medium.com/@abhishekjainindore24/confusion-matrix-for-multiclass-classification-91adf26af6de, 2024. [Internet; último acceso 10-Junio-2025].

[22] Z. Kelta. Una introducción al uso de los transformadores y huggingface. https://www.datacamp.com/es/tutorial/an-introduction-to-using-transformers-and-hugging-face, 2024. [Internet; último acceso 2-Julio-2025].

[23] I. Marcilla Lombraña. Inteligencia artificial aplicada a destinos turísticos inteligentes. Trabajo de Fin de Grado no publicado, Universidad de Burgos, 2024. [Internet; último acceso 13-Junio-2025].

[24] J. Canales Luna. ¿qué es el bert? introducción a los modelos bert. https://www.datacamp.com/es/blog/what-is-bert-an-intro-to-bert-models, 2024. [Internet; último acceso 2-Julio-2025].

[25] F. Murzone. Procesamiento de texto para nlp 1: Tokenización. https://medium.com/escueladeinteligenciaartificial/procesamiento-de-texto-para-nlp-1-tokenizaci%C3%B3n-4d533f3f6c9b, 2025. [Internet; último acceso 10-Junio-2025].

[26] SEGITTUR. Ejes de actuación - segittur. https://www.segittur.es/ejes-de-actuacion/, 2025. [Internet; último acceso 13-Junio-2025].

[27] Tensorflow. Module: tf.keras | tensorflow core v2.16.1. https://www.tensorflow.org/api_docs/python/tf/keras/, 2024. [Internet; último acceso 8-Mayo-2025].

[28] Tensorflow. Por qué tensorflow. https://www.tensorflow.org/about?hl=es-419, 2025. [Internet; último acceso 10-Junio-2025].

[29] Wikipedia. Turismo en españa - wikipedia, la enciclopedia libre. https://es.wikipedia.org/wiki/Turismo_en_Espa%C3%B1a, 2025. [Internet; último acceso 13-Junio-2025].