PROYECTO FIN DE GRADO

TÍTULO:

Aplicación para predecir afluencia de gente en las calles de Madrid

AUTOR/A: Cristina García Calvo

TITULACIÓN: Grado en Ingeniería Telemática

TUTOR/A: Javier Martín Rueda

DEPARTAMENTO: Ingeniería Telemática y Electrónica

VºBº TUTOR/A

Miembros del Tribunal Calificador:

PRESIDENTE/A: Inmaculada Álvarez de Mon Rego

TUTOR/A: Javier Martín Rueda

SECRETARIO/A: Miguel Ángel Valero Duboy

Fecha de lectura: Jueves 13 de Julio de 2023

Calificación:

El Secretario/La Secretaria,

Quiero expresar mi más sincero agradecimiento a mis padres, quienes han sido una parte fundamental de mi vida y de mi carrera académica.

A mi padre, le agradezco su influencia en despertar en mí la curiosidad por la ciencia y tecnología desde una edad temprana. Su amor por el conocimiento ha sido una influencia significativa en mi camino académico.

A mi madre, quiero agradecerle de todo corazón por enseñarme con su ejemplo el valor del esfuerzo. Gracias a ella, he aprendido a no rendirme y a dar siempre lo mejor de mí en cada desafío que se me presenta.

Quiero dedicar un agradecimiento especial a mi pareja, quien ha sido una pieza fundamental en este viaje académico. Tu apoyo incondicional, paciencia y constante motivación han sido una fuente de fuerza en los momentos de desánimo. Gracias por creer en mí y por estar a mi lado en cada paso del camino.

Sin su apoyo, este trabajo fin de grado no habría sido posible. Su influencia positiva en mi vida y su amor incondicional han sido un regalo invaluable.

Resumen

En el presente Trabajo de fin de Grado se realiza el desarrollo de una aplicación basada en técnicas de aprendizaje automático para predecir la afluencia de usuarios en las calles principales de Madrid. Para lograr esto, se tienen en cuenta varios datos como los días festivos, el clima y si hay partido de fútbol, los cuales se utilizan como variables de entrada en los algoritmos de aprendizaje automático.

Se realiza un análisis comparativo de diferentes algoritmos de aprendizaje automático, incluyendo Regresión Lineal, Árboles de Decisión, Bosques Aleatorios y Gradient Boosting. Estos algoritmos se aplican a los datos disponibles para determinar cuál de ellos proporciona las mejores predicciones de la afluencia de usuarios en las calles principales.

Se evalúa el rendimiento de cada uno utilizando varias métricas de evaluación. Entre estas métricas se encuentran el Error Cuadrático Medio (MSE), la Raíz del Error Cuadrático Medio (RMSE), y el Coeficiente de Determinación (R^2).

Además, se lleva a cabo la descarga y transformación automática de los datos necesarios para el análisis. Se implementan técnicas de optimización con el fin de mejorar la eficiencia del programa.

El proyecto también incluye la visualización de datos, lo cual permite presentar los resultados de manera clara y comprensible. Para hacer la aplicación más amigable con los usuarios, se desarrolla una interfaz web utilizando el framework Flask. Esta interfaz permite a los usuarios interactuar con la aplicación de manera intuitiva, brindando la capacidad de realizar consultas, visualizar resultados y obtener predicciones de la afluencia de usuarios en tiempo real.

Abstract

In this Bachelor's Degree Final Project, a machine learning application is developed to predict user influx in the main streets of Madrid. To achieve this, various data such as holidays, weather, and football matches are considered, which are used as input variables in the machine learning algorithms.

A comparative analysis of different machine learning algorithms is performed, including Linear Regression, Decision Trees, Random Forests, and Gradient Boosting. These algorithms are applied to the available data to determine which one provides the best predictions of user influx in the main streets.

The performance of each algorithm is evaluated using several evaluation metrics, including Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and Coefficient of Determination (R^2).

Additionally, automatic data downloading, and transformation are carried out to prepare the data for analysis. Optimization techniques are implemented to improve the program's efficiency.

The project also includes data visualization, which allows presenting the results in a clear and understandable manner. To make the application more user-friendly, a web interface is developed using the Flask framework. This interface enables users to interact with the application intuitively, providing the capability to make queries, visualize results, and obtain real-time predictions of user influx.

Tabla de contenido

Tabla de Figuras

Índice de tablas

1. Introducción

Debido a la situación pandémica que se declaró en 2020, el mundo ha experimentado cambios significativos en comparación con el pasado. Tanto las empresas como los ciudadanos han tenido que adaptarse rápidamente, lo cual ha generado nuevas necesidades y un mayor uso de la tecnología.

A pesar de todos los desafíos que ha planteado, esta situación ha impulsado un gran avance tecnológico, ya que se ha buscado reactivar la economía minimizando los riesgos para la salud en la medida de lo posible. Esto ha llevado a un mayor reconocimiento y aplicación del aprendizaje automático, también conocido como machine learning, en los últimos años. Gracias a esta tecnología, podemos tomar decisiones más inteligentes y mejor fundamentadas.

1.1 Motivación

- Impacto social: Ayudar a las personas a planificar sus actividades diarias de manera más eficiente, evitando aglomeraciones y reduciendo el tiempo de espera en lugares concurridos. Al desarrollar esta aplicación, estaría contribuyendo a mejorar la experiencia de las personas en la ciudad y a promover un entorno más seguro y cómodo.

- Mejora de la planificación urbana: puede ser valiosa para las autoridades locales y los planificadores urbanos. Proporcionar datos precisos sobre los patrones de afluencia en diferentes áreas de la ciudad puede ayudar en la toma de decisiones sobre la distribución de recursos, la gestión del tráfico y la planificación de eventos. La aplicación podría contribuir a una mejor planificación urbana y a la optimización de los servicios públicos en Madrid.

- Innovación y aprendizaje: desarrollar esta aplicación me permitirá mejorar mis habilidades técnicas. Aprenderé sobre técnicas de aprendizaje automático y análisis de datos, que son muy valoradas en el mercado laboral actual. Este proyecto me brinda la oportunidad de innovar, experimentar y aprender en un área de gran demanda y crecimiento.

1.2 Objetivos

El objetivo de este proyecto es desarrollar una aplicación accesible para todos los ciudadanos que sea capaz de predecir la afluencia de usuarios de la vía en las calles principales de la ciudad de Madrid.

Para lograrlo, se utilizarán los datos abiertos proporcionados por la Comunidad de Madrid. Estos datos incluyen información sobre la cantidad de personas que transitan a pie y en bicicleta en diferentes calles, en diferentes días y horarios de la semana [4].

La aplicación mostrará predicciones de afluencia para fechas y horas específicas. Con el fin de lograrlo, los objetivos generales del proyecto son los siguientes:

- Descargar y transformar automáticamente los datos de la página de la Comunidad de Madrid.
- Realizar un análisis comparativo de los distintos algoritmos de Machine learning eligiendo el algoritmo con el porcentaje de error más bajo.
- Visualizar los datos mediante gráficas y mapas, facilitando la comprensión y la interpretación de la información por parte de los usuarios.
- Adquirir los conocimientos para el desarrollo de una aplicación de aprendizaje automático.
- Profundizar en el lenguaje de Python y las bibliotecas pandas, numpy, sklearn y flask.
- Implementar procesamiento paralelo para mejorar la eficiencia y reducir los tiempos de ejecución.
- Registrar los eventos y mensajes de interés mediante la ejecución del programa con logging.

2. Marco tecnológico

En este apartado de explican algunos términos y conceptos requeridos para entender el desarrollo del proyecto.

2.1 Modelo de aprendizaje automático

Un modelo captura las relaciones y patrones importantes presentes en los datos y las utiliza para realizar predicciones y obtener información.

Una vez que el modelo ha sido entrenado, se puede utilizar para realizar predicciones en nuevos datos sin etiquetar. El modelo toma las características de entrada y genera una salida o predicción basada en los patrones aprendidos durante el entrenamiento [1].

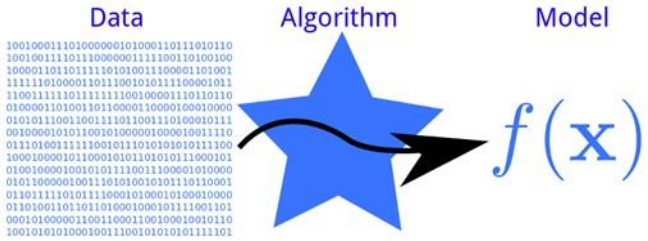

Figura 2.1: Modelo de aprendizaje automático [2]

2.2 Tipos de aprendizaje

Aprendizaje supervisado

En el aprendizaje supervisado, se proporciona al modelo un conjunto de datos etiquetados, donde cada ejemplo está asociado con una etiqueta o clase conocida. El objetivo es que el modelo aprenda a mapear las características de entrada a las etiquetas correspondientes. Esto permite al modelo realizar predicciones o clasificar nuevos datos en función de los patrones aprendidos durante el entrenamiento [3].

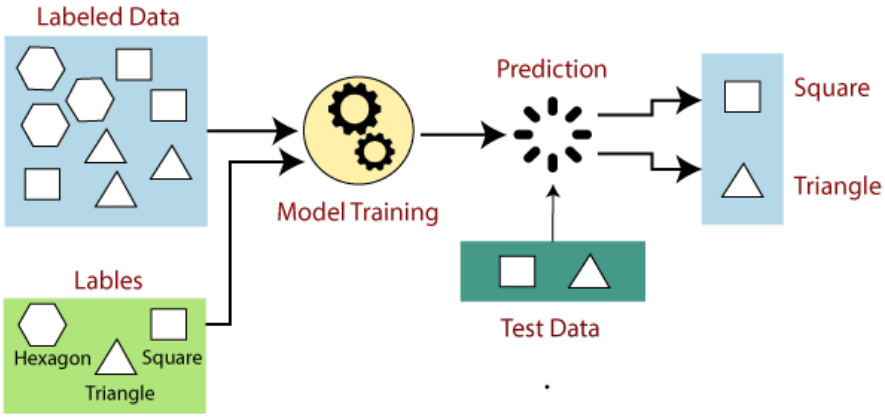

Figura 2.2: Aprendizaje supervisado [4]

Aprendizaje no supervisado

En el aprendizaje no supervisado, no se proporcionan etiquetas en los datos de entrenamiento. El objetivo principal es descubrir patrones, estructuras o relaciones ocultas en los datos. Los algoritmos de aprendizaje no supervisado pueden realizar tareas como la agrupación (clustering), donde se identifican grupos o clústeres de datos similares, o la reducción de dimensionalidad, donde se busca representar los datos en un espacio de menor dimensión [5].

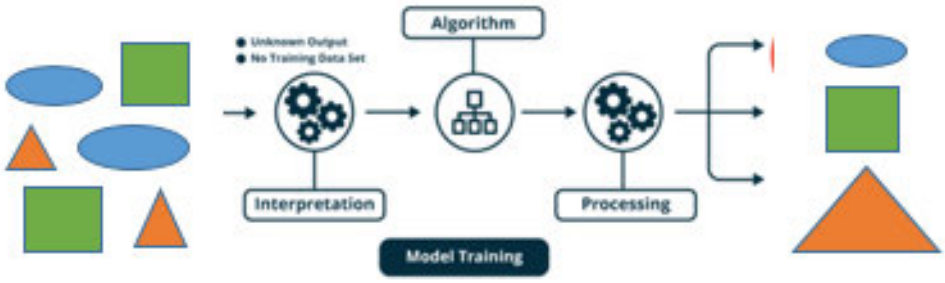

Figura 2.3: Aprendizaje no supervisado [6]

Aprendizaje por refuerzo

En el aprendizaje por refuerzo, el modelo, llamado agente, aprende a través de la interacción con un entorno. El agente toma acciones en un entorno y recibe recompensas o castigos según el resultado de esas acciones. El objetivo del agente es aprender una política de toma de decisiones que maximice las recompensas a largo plazo. El aprendizaje por refuerzo se utiliza en problemas de toma de decisiones secuenciales [7].

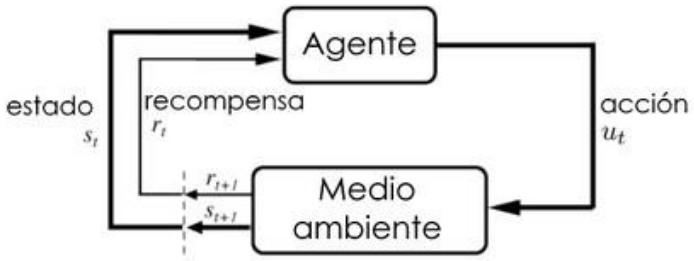

Figura 2.4: Aprendizaje por refuerzo [8]

Aprendizaje semisupervisado

El aprendizaje semisupervisado es una combinación de aprendizaje supervisado y no supervisado. Aquí, se utilizan datos etiquetados y no etiquetados en el entrenamiento. La idea es que los datos no etiquetados proporcionen información adicional para mejorar el rendimiento del modelo, aprovechando tanto la información conocida como la estructura oculta en los datos no etiquetados [9].

Aprendizaje por transferencia

El aprendizaje por transferencia en lugar de entrenar un modelo desde cero, se transfieren los conocimientos. Esto es útil cuando hay pocos datos disponibles en el dominio objetivo o cuando el modelo ya ha sido entrenado en un dominio relacionado [10].

2.3 Algoritmos de aprendizaje supervisado

Regresión lineal

Es un algoritmo utilizado para problemas de regresión, donde se busca predecir un valor numérico continuo. El objetivo es encontrar la mejor línea recta que se ajuste a los datos de entrenamiento [11].

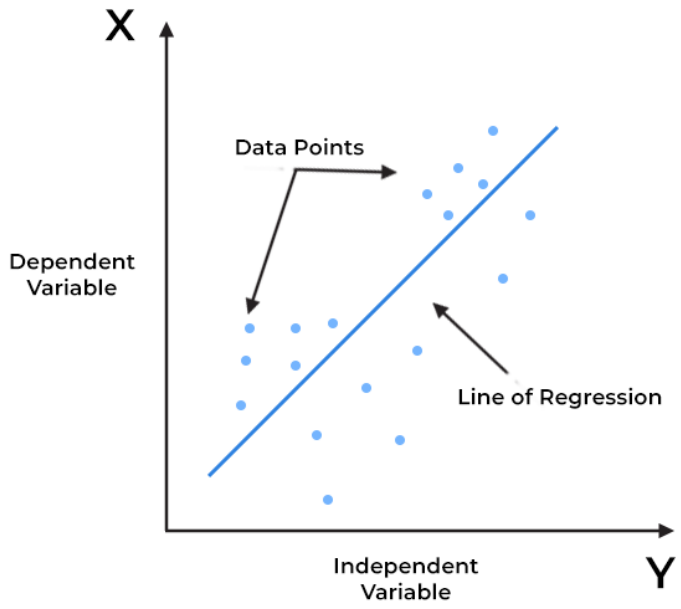

Figura 2.5: Algoritmo regresión lineal [12]

Regresión Polinomial

La regresión polinomial es un caso especial de la regresión lineal. Mientras que la regresión lineal asume una relación lineal entre las variables, la regresión polinomial permite modelar relaciones mediante el ajuste de polinomios de grado mayor [13].

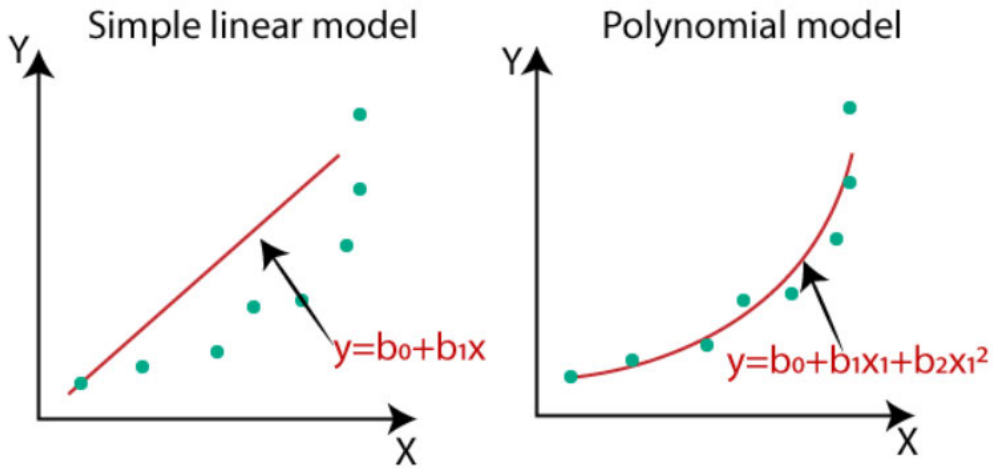

Figura 2.6: Algoritmo de regresión polinomial [14]

Vectores de soporte de Regresión

El algoritmo de vectores de soporte de regresión busca encontrar una curva o hiperplano que se ajuste a los datos de entrenamiento.

El hiperplano obtenido modelará el comportamiento de los datos y estará acompañado de un rango de margen máximo tanto en el lado positivo como en el negativo. Este rango tiene la misma forma que la curva. Los datos que se encuentren fuera de este rango se consideran errores y se mide la distancia entre ellos y los rangos. Esta distancia se llama épsilon y afecta la ecuación final del modelo [15].

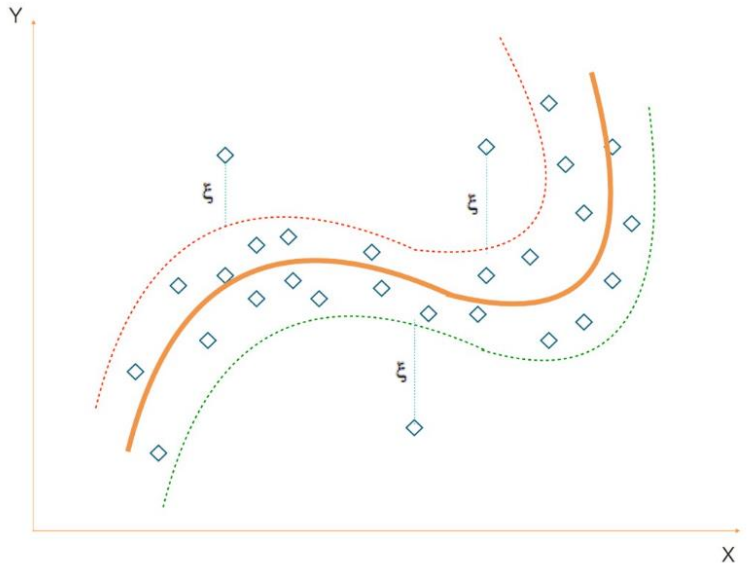

Figura 2.7: Algoritmo de vectores de soporte de regresión [15]

Árboles de decisión

Este algoritmo divide el conjunto de datos en diferentes regiones según las características, formando un árbol de decisión. Es utilizado tanto para clasificación como para regresión, y proporciona una estructura clara y fácil de interpretar [16].

El árbol está formado por distintos nodos:

- Nodo de decisión: evalúa una característica específica del conjunto de datos y realiza una división basada en esa característica.
- Nodo terminal: no tiene ramas salientes y representa una conclusión o resultado final.

Para predecir un valor partimos del nodo raíz, que sería el primero del árbol, y hacemos diferentes comparaciones siguiendo las ramas y saltando de un nodo a otro hasta llegar a la predicción final.

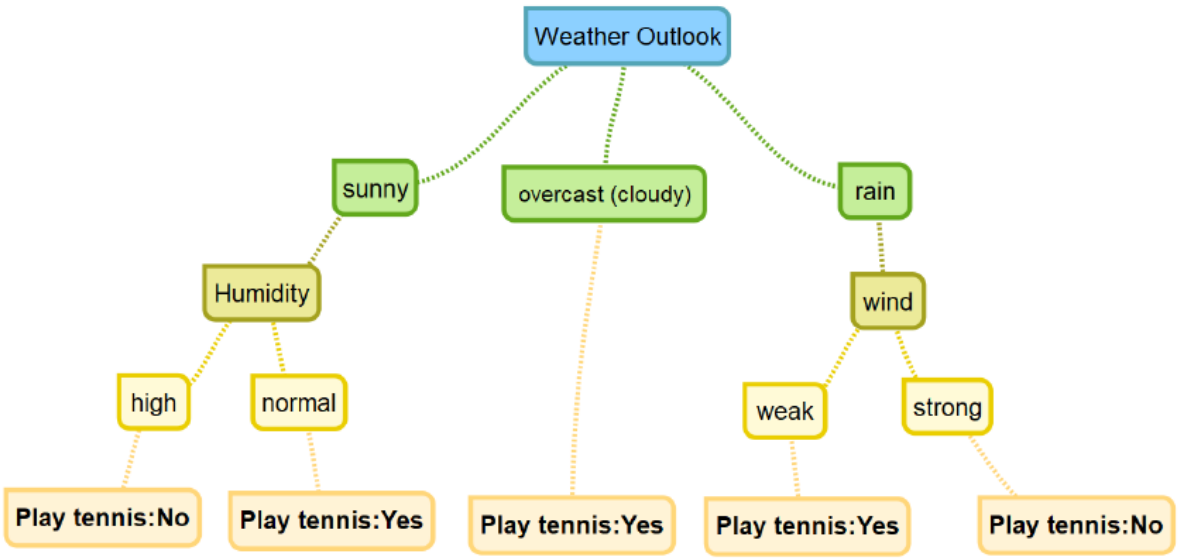

Figure 1: Play Tennis Decision tree

Figura 2.8: Algoritmo árbol de decisión [17]

Bosques aleatorios

Es un método de ensamble que utiliza múltiples árboles de decisión. Cada árbol es entrenado con una muestra aleatoria del conjunto de datos, y las predicciones finales se obtienen mediante un proceso de votación o promediando las predicciones individuales de los árboles [18].

Cada árbol se entrena en un subconjunto de datos obtenido mediante muestreo aleatorio con reemplazo, lo que implica que cada árbol se entrena en una muestra única de datos. Por lo general, se utiliza la raíz cuadrada (√m) del número total de características como tamaño del subconjunto de características en cada nodo. Como resultado, cada árbol en el bosque presenta pequeñas diferencias en términos de los datos en los que fue entrenado y las características que utilizó. Esto lleva a que cada árbol genere predicciones ligeramente diferentes para el mismo conjunto de datos de entrada. En el caso de problemas de clasificación, se selecciona la clase más frecuente entre los árboles, mientras que en problemas de regresión se realiza un promedio de las predicciones numéricas de los árboles.

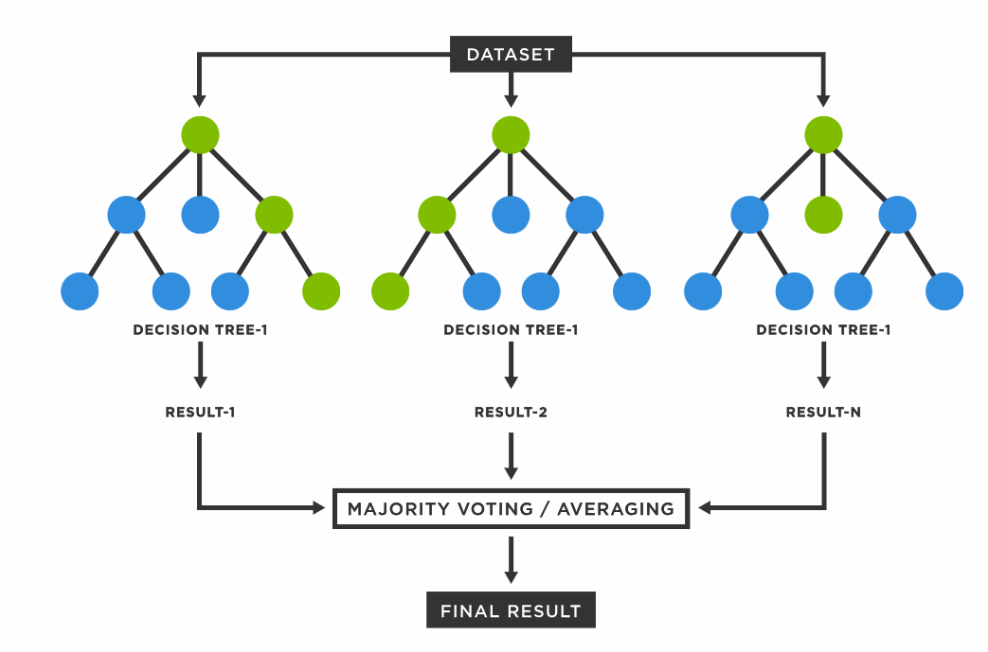

Figura 2.9: algoritmo bosques aleatorios [19]

Gradient Boosting

Es un algoritmo de ensamble que construye múltiples árboles de decisión de manera secuencial, enfocándose en corregir los errores del modelo anterior. Utiliza el principio de gradiente descendente para ajustar los errores residuales y mejorar gradualmente la precisión del modelo [20].

Genera un primer árbol y luego realiza la validación o evaluación de su desempeño. Las muestras que fueron clasificadas incorrectamente por este primer árbol se utilizan para entrenar el siguiente árbol en el proceso de construcción. De esta manera, cada árbol subsiguiente se enfoca en corregir las predicciones erróneas del árbol anterior, lo que contribuye a mejorar la precisión global del modelo.

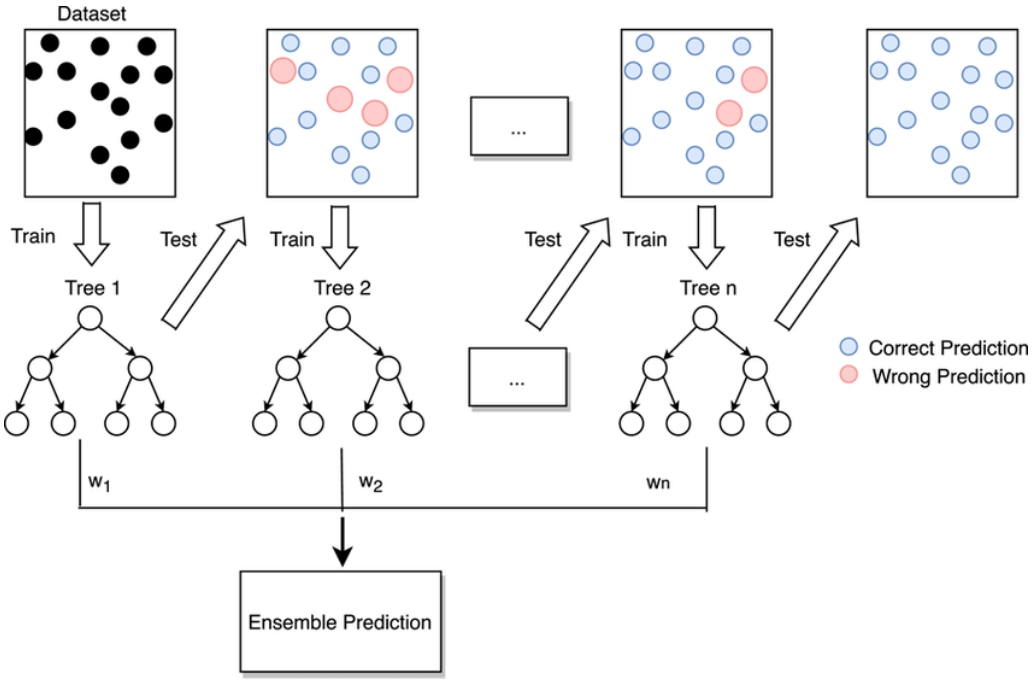

Figura 2.10: Algoritmo gradient boosting [21]

2.4 Preprocesamiento de datos

El preprocesamiento de datos desempeña un papel fundamental en el modelo de aprendizaje automático al preparar y transformar los datos para su óptimo funcionamiento. Permite abordar datos faltantes o incorrectos, estandarizar y normalizar los datos, codificar variables categóricas, tratar el desequilibrio de clases y reducir la dimensionalidad. Estas transformaciones mejoran la calidad de los datos, evitan sesgos, facilitan la interpretación y optimizan el rendimiento del modelo, lo que se traduce en resultados más precisos y confiables en el proceso de aprendizaje automático [22].

Gestión de valores nulos

La gestión de valores nulos consiste en el proceso de identificar, tratar y manejar los valores faltantes en un conjunto de datos. Algunas posibles soluciones a este problema incluyen: [23]

- Eliminación de muestras o de características que incluyan valores nulos
- Reemplazo por un valor
- Asignación de una categoría exclusiva
- Predicción de los valores nulos

El escalado de características

El escalado de características es un proceso de preprocesamiento de datos numéricos que se utiliza para asegurar que todas las características o variables tengan un rango similar o estén en la misma escala. El objetivo principal del escalado de características es evitar que una característica domine o tenga un impacto desproporcionado en el análisis y así mejorar la interpretación de los resultados [24].

Codificación One-hot

La codificación one-hot es una técnica utilizada para representar variables categóricas como variables binarias en el análisis de datos sin establecer relaciones de orden o magnitud entre las categorías [25].

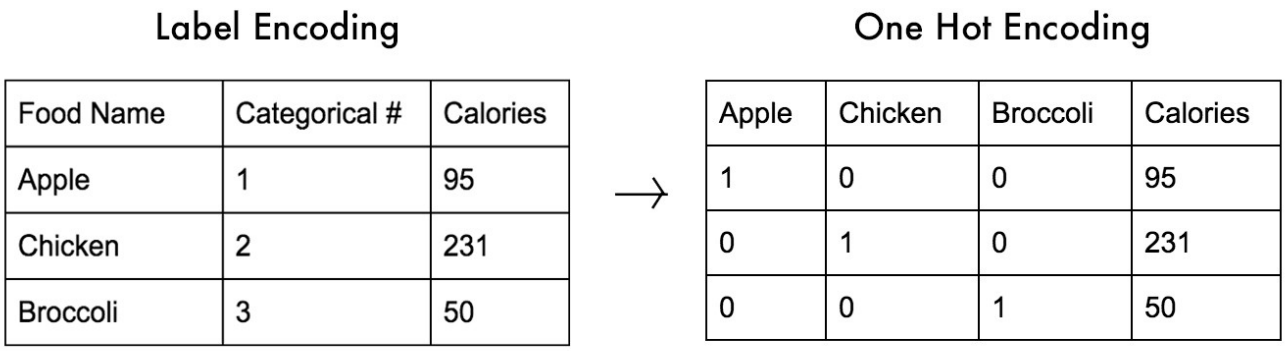

Figura 2.11: Ejemplo codificacion One-hot [26]

División de datos en conjuntos de entrenamiento, validación y prueba

Los datos se dividen en tres conjuntos: entrenamiento, validación y prueba. El conjunto de entrenamiento se utiliza para entrenar el modelo, el conjunto de validación se utiliza para ajustar los hiperparámetros del modelo y el conjunto de prueba se utiliza para evaluar la precisión final del modelo.

2.5 Técnicas para la evaluación de los resultados

Existen varias técnicas para evaluar los resultados del modelo, algunas de las usadas en este proyecto son:

Error cuadrático medio (MSE)

El MSE, la ecuación (1), es una métrica que calcula el promedio de los errores al cuadrado entre las predicciones del modelo y los valores reales. Cuanto menor sea el valor del MSE, mejor será el ajuste del modelo a los datos [27].

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(yi - \hat{y}i)^2 \quad (1)$$

- N es el número total de observaciones en el conjunto de datos.
- y_i es el valor real de la i-ésima observación.
- $\hat{y}_i$ es la predicción del modelo para la i-ésima observación.

Raíz del error cuadrático medio (RMSE)

El RMSE, la ecuación (2), se calcula como la raíz cuadrada del MSE, lo que proporciona una medida del error promedio en la misma unidad que la variable objetivo. Cuanto menor sea el valor del RMSE, mejor será el ajuste del modelo a los datos [27].

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(yi - \hat{y}i)^2} = \sqrt{MSE} \quad (2)$$

Error absoluto medio (MAE)

El MAE, la ecuación (3), se calcula como la diferencia absoluta entre cada valor real y su correspondiente predicción, y luego se promedian todas estas diferencias. Cuanto menor sea el valor del MAE, mejor será el ajuste del modelo a los datos [27].

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|yi - \hat{y}i| \quad (3)$$

Coeficiente de determinación (R^2)

El coeficiente de determinación, ecuación (4), es una métrica que indica la proporción de la varianza de la variable objetivo que es explicada por el modelo. Toma valores entre 0 y 1, donde 0 indica que el modelo no explica la varianza y 1 indica un ajuste perfecto [27].

$$R^2 = 1 - \frac{MSE(modelo)}{MSE(línea\ base)} \tag{4}$$

Siendo el MSE (línea base) la ecuación (5)

$$MSE(línea\ base) = \frac{1}{N}\sum_{i=1}^{N}(yi - \bar{\bar{y}})^2 \tag{5}$$

- ȳ es la media de los valores objetivo en el conjunto de datos.

2.6 Patrón arquitectónico MVC

El patrón arquitectónico Modelo-Vista-Controlador (MVC) en diseñar y organizar las aplicaciones de software separando la lógica de negocio, la presentación de la interfaz de usuario y la gestión de eventos. El usuario interactúa con la interfaz de usuario y la vista envía eventos o solicitudes al controlador.

Proporciona una estructura clara y modular, facilitando el desarrollo, el mantenimiento y la evolución de la aplicación [28].

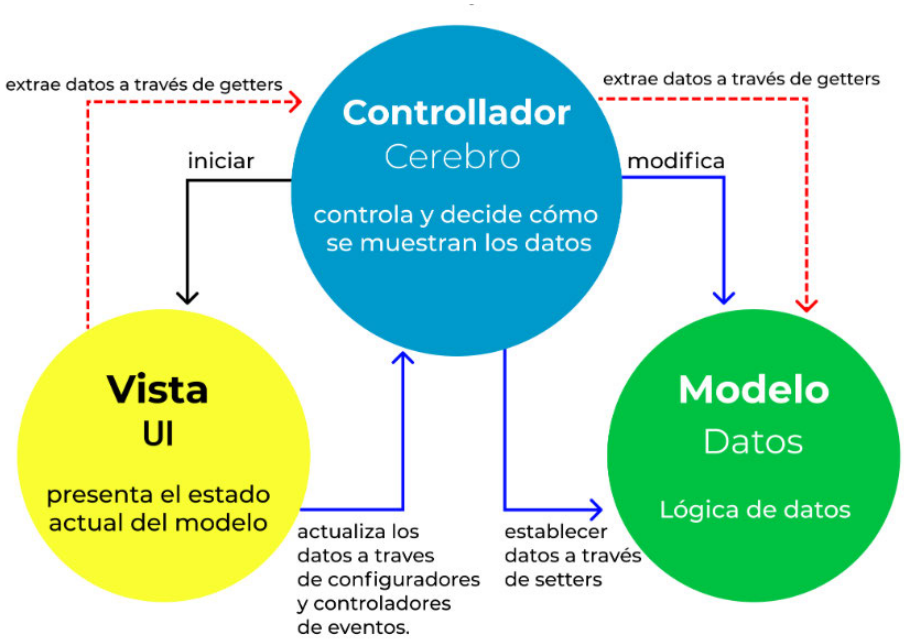

Figura 2.12: Patrón MVC [28]

2.7 Herramientas de desarrollo

Scikit-learn

Scikit-learn es una biblioteca popular y de código abierto en Python que se utiliza para realizar tareas de aprendizaje automático. Proporciona una amplia gama de algoritmos, herramientas de preprocesamiento de datos y utilidades para tareas relacionadas con el aprendizaje automático, como la evaluación de modelos y la selección de características.

Scikit-learn se basa en las bibliotecas NumPy y SciPy y ofrece una interfaz sencilla para aplicar algoritmos de aprendizaje automático a conjuntos de datos [29].

Folium

Folium es una biblioteca de Python que se utiliza para la visualización interactiva de datos geoespaciales. Está basada en la popular biblioteca JavaScript Leaflet.js y proporciona una interfaz fácil de usar para crear mapas interactivos en Python [30].

Programación por hilos

La programación por hilos es una técnica que consiste en la ejecución de varias operaciones dentro de un programa. Cada flujo de ejecución es denominado hilo.

Cuando se utiliza la programación por hilos, un programa puede ejecutar múltiples hilos simultáneamente, lo que permite realizar tareas concurrentes y aprovechar mejor los recursos del sistema. Cada hilo puede realizar diferentes tareas de forma independiente y simultánea, lo que puede mejorar el rendimiento y la capacidad de respuesta de una aplicación [31].

Concurrent futures

Concurrent.futures es un módulo de la biblioteca estándar de Python. Proporciona una interfaz de alto nivel para la ejecución concurrente de tareas en paralelo de forma asíncrona.

El módulo proporciona dos clases:

- Los “ejecutores” (executors): se utilizan para gestionar grupos de trabajadores
- Los "futuros" (futures): se utilizan para gestionar los resultados calculados por los trabajadores

La aplicación crea una instancia del ejecutor y luego envía las tareas que se deben ejecutar. Cada tarea devuelve un objeto Future cuando se inicia, lo cual representa el resultado pendiente de la tarea. Cuando la aplicación necesita el resultado de una tarea en particular, utiliza el objeto Future correspondiente para esperar y obtener el resultado [32].

Flask

Flask es un framework ligero y flexible escrito en Python utilizado para crear páginas webs. Se basa en el concepto de "microframework", lo que significa que proporciona solo las herramientas básicas para construir una aplicación web [33].

3. Especificaciones y restricciones de diseño

Las especificaciones y restricciones de diseño para la aplicación son las siguientes:

- Desarrollo de una aplicación que prediga la cantidad de los usuarios de la vía creando un modelo de aprendizaje automático.

- La aplicación descargará los datos abiertos de la página de la Comunidad de Madrid [34], relativa a la cantidad de peatones y bicicletas correspondientes a diferentes calles en diferentes días y horas de semana.

- Se tendrán en cuenta los datos climatológicos y los partidos de futbol para la creación del modelo y posterior predicción.

- Se realizará un procesamiento y transformación de los datos descargados para su posterior análisis. Esto implica aplicar diferentes técnicas y operaciones para limpiar, filtrar, normalizar o enriquecer los datos, según sea necesario.

- La interfaz de usuario mostrará los resultados en un mapa interactivo donde aparecerá:

    - Nombre de la calle
    - Día y hora de la predicción
    - Predicción de gente

- La aplicación ofrecerá una funcionalidad de actualización del modelo para mejorar continuamente la precisión de las predicciones. Esto implicará la incorporación de nuevos datos y el reentrenamiento del modelo cuando se solicite.

- La aplicación desarrollada ofrecerá una funcionalidad de visualización de datos para facilitar la comprensión y análisis de la predicción de afluencia de gente en las calles de Madrid.

- La aplicación se desarrollará en Python y se utilizará un controlador de versiones para gestionar el código fuente.

- Se emplearán bibliotecas adecuadas para el procesamiento de datos, creación de modelos automáticos y creación de mapas.

- La aplicación estará diseñada para ser independiente del sistema operativo en el que se ejecute.

4. Descripción de la solución propuesta

4.1 Fases de la creación del modelo de aprendizaje automático

Importación de datos

En esta fase, se importan los datos necesarios para entrenar y evaluar el modelo. Esto puede incluir conjuntos de datos históricos que contengan información sobre la afluencia de gente en diferentes calles de Madrid en distintos momentos. Es importante asegurarse de que los datos se importen de manera adecuada y estén en un formato compatible con el modelo.

Limpieza de datos

En esta fase, se realiza un proceso de limpieza de los datos importados para asegurarse de que estén libres de errores, duplicados o valores atípicos que puedan afectar negativamente el rendimiento del modelo. Esto implica la eliminación o corrección de datos incorrectos, la gestión de valores faltantes y la normalización de la estructura de los datos para facilitar su procesamiento.

Procesamiento de los datos

A continuación, se realizan análisis estadísticos y transformaciones adicionales para mejorar la calidad de los datos y prepararlos para el modelado. Esto incluye técnicas como la normalización de los datos numéricos para que estén en la misma escala, la codificación de variables categóricas, la selección de características relevantes y la división del conjunto de datos en conjuntos de entrenamiento y prueba.

Creación del modelo

En esta fase, se procede a la creación del modelo utilizando diferentes algoritmos de regresión. Se consideran los siguientes algoritmos: Regresión lineal, árboles de decisión, bosques aleatorios y Gradient Boosting. Se realiza un ajuste de hiperparámetros para encontrar la mejor configuración.

Evaluación y selección del modelo

Se entrena y evalúa el rendimiento de cada modelo utilizando las métricas de evaluación: el error cuadrático medio (MSE), la raíz del error cuadrático medio (RMSE) y el coeficiente de determinación (R^2). Luego, se selecciona el algoritmo que haya demostrado el mejor rendimiento en función de estas métricas. Esto asegura que el modelo seleccionado sea

capaz de proporcionar las predicciones más precisas y confiables de la afluencia de gente en cada calle de Madrid.

Visualización de resultados

En esta fase, se utiliza la visualización de datos para comprender mejor las características de los datos y los resultados del modelo seleccionado. Se generan gráficos por cada calle de Madrid que muestran:

- Eficiencia del modelo: Se crean gráficos que representan la eficiencia del modelo en cada calle de Madrid, mostrando el Coeficiente de Determinación (R^2) obtenido en los datos de entrenamiento, de prueba y la puntuación general.

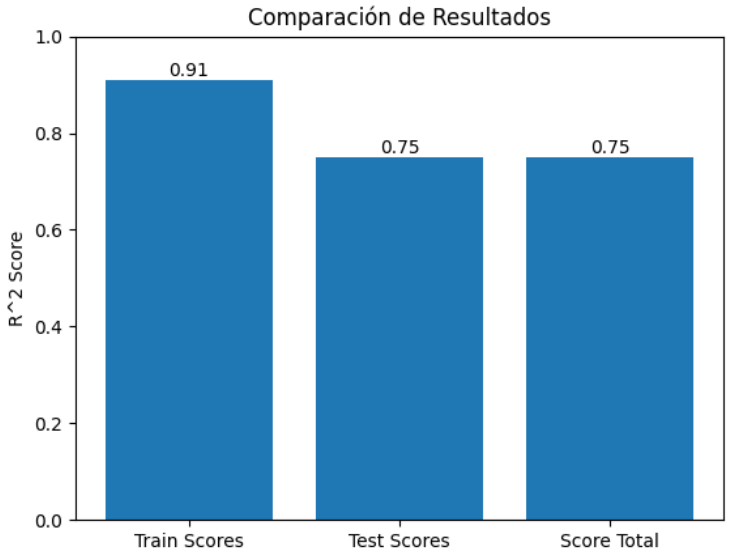

Figura 4.1: Ejemplo de visualización del coeficiente de determinación

- Importancia relativa de los parámetros del modelo: Se generan gráficos que muestran la importancia relativa de los parámetros del modelo en cada calle de Madrid. Esto permite identificar qué variables o características tienen un mayor impacto en las predicciones de afluencia de gente en cada calle. Estos gráficos pueden ayudar a comprender qué factores son más relevantes para cada ubicación específica.

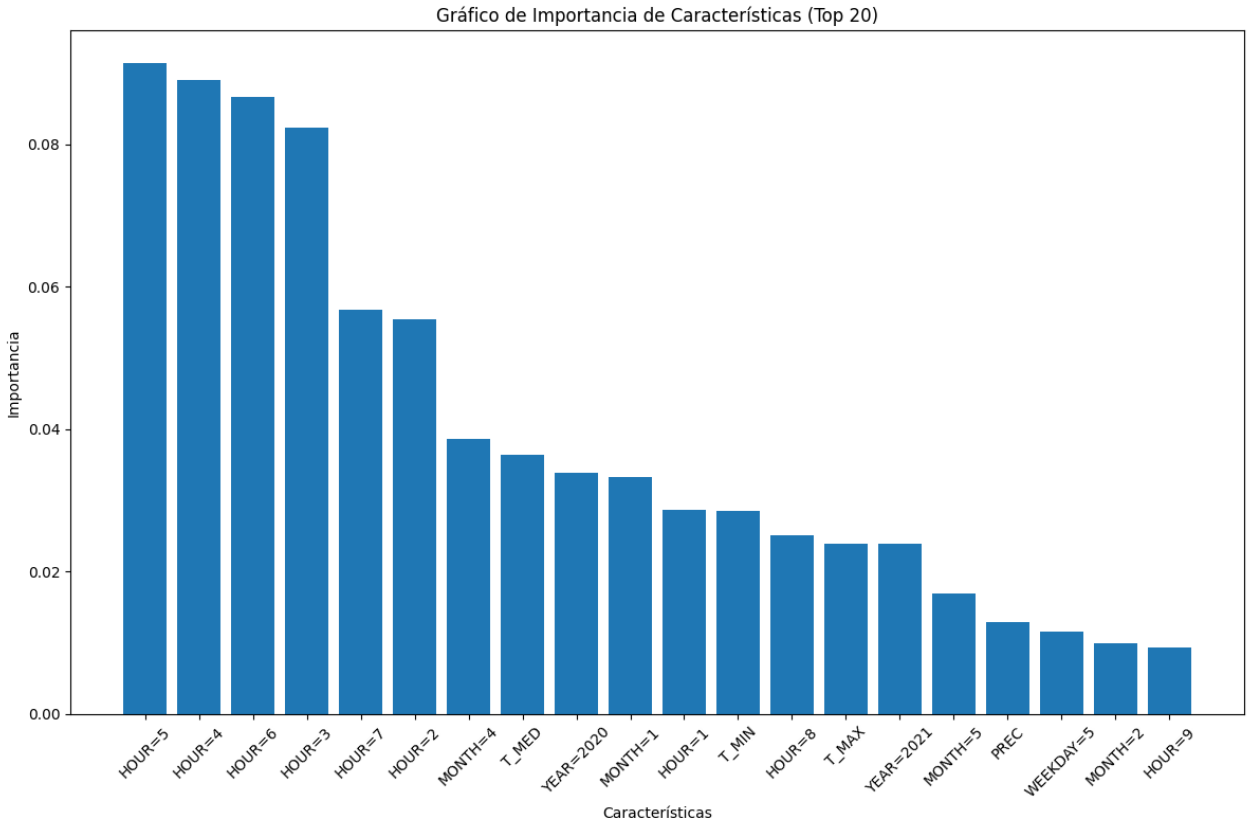

Figura 4.2: Ejemplo de visualización de importancia de características

- Ejemplo de predicción por cada calle de Madrid: Se realiza un ejemplo de predicción para cada calle de Madrid proporcionada en los datos, utilizando el modelo entrenado para predecir la afluencia de personas en cada calle específica. Los resultados de estas predicciones se representan en gráficos que comparan la cantidad de usuarios reales con los valores predichos por el modelo. Estos gráficos incluyen 10 muestras aleatorias del conjunto de datos, lo que permite una comparación detallada de las predicciones. Cada barra en el gráfico representa una columna del conjunto de datos, lo cual implica que muestra un día y hora aleatorios de una calle específica. El objetivo principal del gráfico no es destacar un día o una hora en particular, sino comparar los datos predichos con los datos reales. Esto nos permite evaluar la precisión de las predicciones realizadas por el modelo.

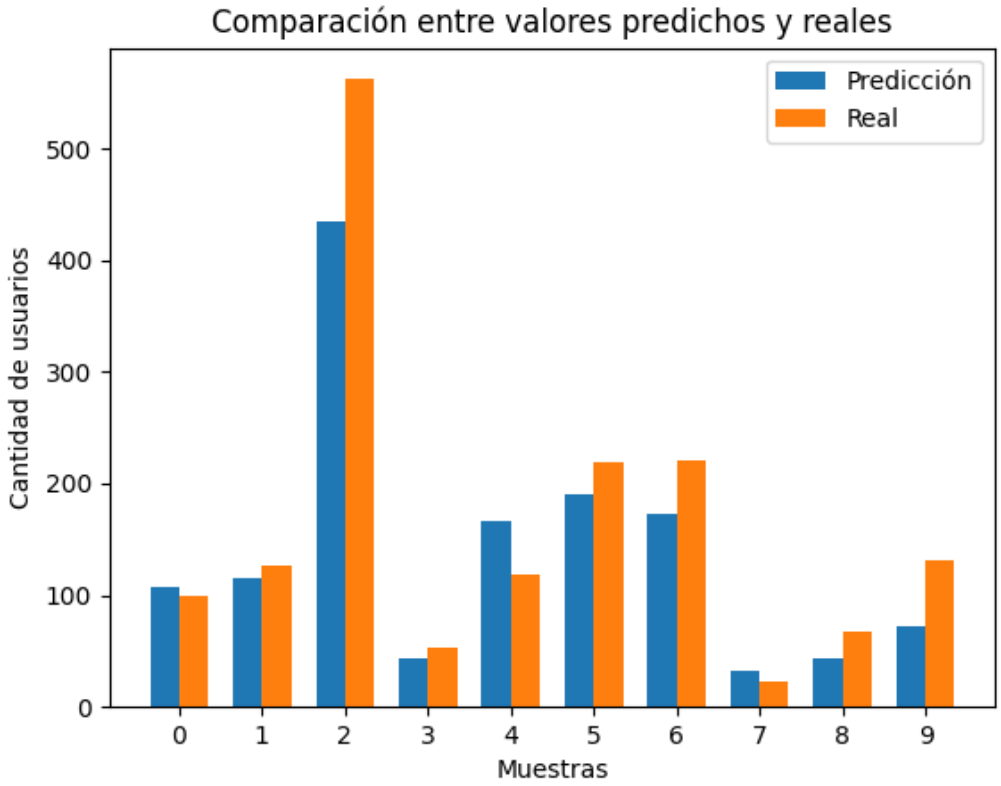

Figura 4.3: Ejemplo de predicción de valores de 10 muestras del modelo

4.2 Conjunto de datos

Definición del conjunto de datos

En la realización del proyecto se han necesitado de distintas fuentes de datos:

- Aforos de peatones y bicicletas: este conjunto de datos es la principal fuente para el entrenamiento del modelo, sobre el que se planteó el proyecto inicialmente. Estos datos proporcionan información sobre los usuarios de la vía pública en las calles de Madrid.

- Calendario: Se ha utilizado un calendario para identificar y tener en cuenta los días festivos y eventos especiales. Esta información es relevante, ya que la presencia de usuarios en la vía puede variar significativamente en estos días, lo que afecta a la predicción del modelo.

- Datos climatológicos: se han considerado los datos de temperatura y precipitaciones. Las variaciones en el clima pueden influir en la cantidad de usuarios en las calles, ya que las personas pueden optar por actividades al aire libre según las condiciones climáticas.

- Partidos de fútbol: Se ha tenido en cuenta la información sobre los partidos de fútbol, ya que estos eventos pueden tener un impacto significativo en el flujo de peatones. La celebración de un partido puede generar un aumento o disminución en el número de personas en áreas específicas.

Estas fuentes de datos se han utilizado de manera conjunta para enriquecer el modelo de predicción y mejorar su capacidad para estimar el número de usuarios de la vía pública. Al considerar factores como los días festivos, las condiciones climáticas y los partidos de fútbol, se puede lograr una mayor precisión en las predicciones del modelo.

Aforos de peatones y bicicletas

Para el entrenamiento del modelo se utiliza el conjunto de datos "Aforos de peatones y bicicletas" proporcionado por los datos abiertos de la comunidad de Madrid [35].

Este conjunto proporciona información detallada sobre la cantidad de peatones y bicicletas en diferentes calles de la ciudad de Madrid. Estos datos se recopilan en puntos de medición específicos ubicados en diversas calles, y se realizan mediciones diarias cada hora.

Los datos están organizados en un archivo csv estructurado con los siguientes campos:

- FECHA: Fecha en la que se recogieron los datos.
- HORA: Hora en la que se realizó la recogida de datos.
- IDENTIFICADOR: Identificador del aforo, indicando el número de estación permanente de aforo tanto para peatones como para bicicletas.
- BICICLETAS/PEATONES: Número de peatones o bicicletas contadas en un intervalo de quince minutos.
- NÚMERO_DISTRITO: Número del distrito donde se encuentra la estación de aforo.
- DISTRITO: Denominación del distrito donde se encuentra la estación de aforo.

- NOMBRE_VIAL: Calle donde se encuentra la estación de aforo.
- NÚMERO: Número de la calle donde se encuentra la estación de aforo.
- CÓDIGO_POSTAL: Código postal donde se encuentra la estación de aforo.
- OBSERVACIONES_DIRECCION: Información adicional relacionada con la dirección.
- LATITUD: Coordenada de latitud de la ubicación de la estación de aforo.
- LONGITUD: Coordenada de longitud de la ubicación de la estación de aforo.

Estos datos son recopilados y proporcionados por la Dirección General de Planificación e Infraestructuras de Movilidad de la Comunidad de Madrid, quienes son los responsables de mantener y actualizar esta información.

Calendario laboral

El conjunto de datos "Calendario laboral" recoge información sobre los días laborables y festivos en la ciudad de Madrid. Está gestionado por la Dirección General de Función Pública, específicamente la Subdirección General de Relaciones Laborales [34].

Los datos en el conjunto incluyen:

- Fecha: en formato dd/mm/aaaa, el día de la semana correspondiente a esa fecha
- Dia_semana: Día de la semana correspondiente a la Fecha.
- Laborable/festivo/domingo festivo: etiqueta que indica si es un día laborable para el día del año y el día de la semana señalados.
- Tipo de festivo: indica si se trata de festivo nacional, de la Comunidad de Madrid o local (ciudad de Madrid).
- Festividad: nombre de la festividad según la legislación del país

Los datos están disponibles de forma anual. La estructura del fichero de datos se describe en formato csv.

Temperaturas

Los datos relacionados con las temperaturas pasadas se obtienen de AEMET Open Data, que es una plataforma que proporciona acceso a datos climáticos históricos. AEMET es la Agencia Estatal de Meteorología en España y su plataforma Open Data permite acceder de manera abierta y gratuita a una amplia variedad de datos meteorológicos [36].

Los datos de la climatología diaria incluyen los siguientes atributos:

- Fecha: La fecha en que se tomó la medición (en formato AAAA-MM-DD)
- Indicativo: Un número de identificación para la estación meteorológica
- Nombre: El nombre de la ubicación donde se encuentra la estación meteorológica
- Provincia: La provincia donde se encuentra la estación meteorológica
- Altitud: La altitud de la estación meteorológica (en metros)
- Tmed: La temperatura promedio del día (en grados Celsius)

- Prec: La cantidad de precipitación que cayó durante el día (en milímetros)
- Tmin: La temperatura mínima registrada durante el día (en grados Celsius)
- Horatmin: La hora en que se registró la temperatura mínima (en formato HH:MM)
- Tmax: La temperatura máxima registrada durante el día (en grados Celsius)
- Horatmax: La hora en que se registró la temperatura máxima (en formato HH:MM)
- Dir: La dirección del viento (en grados)
- Velmedia: La velocidad media del viento (en metros por segundo)
- Racha: La velocidad máxima del viento registrada durante el día (en metros por segundo)
- Horaracha: La hora en que se registró la velocidad máxima del viento (en formato HH:MM)
- PresMax: La presión atmosférica máxima registrada durante el día (en hectopascales)
- HoraPresMax: La hora en que se registró la presión atmosférica máxima (en formato HH)
- PresMin: La presión atmosférica mínima registrada durante el día (en hectopascales)
- HoraPresMin: La hora en que se registró la presión atmosférica mínima (en formato HH)

Para realizar predicciones del pronóstico de los próximos 5 días, se utiliza la web OpenWeatherMap, que es una plataforma que ofrece una API gratuita y de pago para acceder a datos meteorológicos. Para poder utilizarla en este proyecto, se ha realizado el registro en su sitio web y se ha obtenido una API key, la cual se utiliza para realizar solicitudes y recibir los datos meteorológicos necesarios [37].

La respuesta se obtiene en formato json, donde los principales atributos de la predicción son:

- "dt": La fecha y hora de la medición en formato UNIX timestamp.
- "main":
  - "temp": La temperatura actual en grados Kelvin.
  - "feels_like": La sensación térmica en grados Kelvin.
  - "temp_min": La temperatura mínima registrada en grados Kelvin.
  - "temp_max": La temperatura máxima registrada en grados Kelvin.
  - "pressure": La presión atmosférica en hectopascales.
- "weather":
  - "id": Un identificador numérico para el tipo de clima.
  - "main": La categoría principal del clima (por ejemplo, nubes, lluvia, etc.).
  - "description": Una descripción más detallada del clima.
  - "icon": Un código que representa el ícono asociado al clima.
- "clouds":
  - "all": El porcentaje de cobertura de nubes.
- "wind":
  - "speed": La velocidad del viento en metros por segundo.
  - "deg": La dirección del viento en grados.

    - "gust": La velocidad de ráfaga máxima del viento en metros por segundo.
- "visibility": La visibilidad en metros.
- "pop": La probabilidad de precipitación.
- "sys":
    - "pod": Indica si es de día (d) o de noche (n).
- "dt_txt": La fecha y hora de la medición en formato de texto legible.

Dado que el pronóstico proporcionado por OpenWeatherMap solo abarca los próximos 5 días, para realizar predicciones a más largo plazo se utilizan los datos medios de temperaturas de cada mes [38].

Promedio	ene.	feb.	mar.	abr.	may.	jun.	jul.	ago.	sept.	oct.	nov.	dic.
Máxima	10 °C	12 °C	16 °C	18 °C	23 °C	29 °C	33 °C	32 °C	27 °C	20 °C	14 °C	11 °C
Temp.	5 °C	6 °C	10 °C	12 °C	16 °C	22 °C	26 °C	25 °C	21 °C	15 °C	9 °C	6 °C
Mínima	1 °C	1 °C	4 °C	6 °C	10 °C	14 °C	17 °C	17 °C	13 °C	9 °C	4 °C	1 °C

Figura 4.4: Datos medios de temperatura [38]

Partidos de fútbol

Para el conjunto de datos de los partidos de fútbol no se encontró una base de datos confiable para los partidos de fútbol, por lo que se ha creado un archivo CSV con los partidos más relevantes de este año [39].

4.3 Tratamiento de datos

Selección de características

Se seleccionan las columnas más significativas de cada conjunto de datos para utilizarlas como entradas en el entrenamiento del modelo.

Las características seleccionadas son las siguientes:

- 'FESTIVO': Esta característica indica si el día corresponde a un día festivo o no. Es una variable binaria donde 1 representa un día festivo y 0 representa un día no festivo. Esta columna permite al modelo capturar posibles variaciones en el comportamiento o patrones durante los días festivos.

- 'YEAR': Esta característica representa el año en el que se registra la medición. Es una variable numérica que indica el año en el que se recopilaron los datos.

- 'MONTH': Esta característica indica el mes correspondiente a la medición. Es una variable numérica que representa el mes del año.

- 'DAY': Esta característica representa el día del mes en el que se registra la medición. Es una variable numérica que indica el día específico en el mes.

- 'HOUR': Esta característica indica la hora del día en la que se registra la medición. Es una variable numérica que representa la hora en formato de 24 horas.

- 'WEEKDAY': Esta característica indica el día de la semana correspondiente a la medición. Puede ser una variable numérica que representa el día de la semana, donde 0 representa el domingo, 1 el lunes y así sucesivamente.

- 'EN_FUTBOL': Esta característica indica si hay un evento de fútbol programado para el día y hora de la medición. Es una variable binaria que representa la presencia o ausencia de un evento de fútbol.

- "T_MED", "T_MIN", "T_MAX": Estas características representan las temperaturas promedio, mínima y máxima registradas en el momento de la medición. Son variables numéricas.

- 'PREC': Esta característica representa la cantidad de precipitación registrada en el momento de la medición.

Gestión de valores nulos

Dado que la proporción de valores nulos es relativamente pequeña en relación con el tamaño total del conjunto de datos, se ha optado por eliminar las filas que contienen valores nulos utilizando el método dropna de Pandas. Al aplicar dropna al DataFrame df, se descartan todas las filas que contienen al menos un valor nulo en alguna de las columnas.

De esta manera, se asegura que el conjunto de datos utilizado para el entrenamiento del modelo esté compuesto únicamente por filas completas, sin valores faltantes en ninguna de las columnas.

Transformación logarítmica

Para mejorar el rendimiento del modelo, se ha aplicado una transformación logarítmica a los parámetros de entrada. Esta transformación tiene como objetivo reducir la asimetría de los datos, especialmente si presentan un sesgo hacia un extremo. Al aplicar la función logarítmica, se logra disminuir el impacto de los valores atípicos, lo que ayuda al modelo a tener una mejor comprensión de los datos y a realizar predicciones más precisas [40].

Preprocesamiento de datos

Para el preprocesamiento de datos en este proyecto, se utilizan dos técnicas específicas.

- Para las variables categóricas, que incluyen 'FESTIVO', 'YEAR', 'MONTH', 'DAY', 'HOUR', 'WEEKDAY' y 'EN_FUTBOL', se utiliza OneHotEncoder. Esta técnica convierte estas variables en representaciones numéricas utilizando el enfoque de "one-hot encoding".

- Para las variables numéricas, que incluyen "T_MED", "T_MIN", "T_MAX" y "PREC", se utiliza MinMaxScaler. Esta técnica se encarga de escalar y normalizar los valores numéricos dentro de un rango específico, lo que ayuda a evitar problemas relacionados con la magnitud de las variables.

4.4 División del conjunto de entrenamiento y prueba

La división del conjunto de datos nos sirve para evaluar el rendimiento y la capacidad predictiva de un modelo entrenado en datos no vistos previamente.

La función train_test_split de sklearn nos permite hacer la división del conjunto de datos en dos bloques: de entrenamiento y prueba de un modelo. La división se realiza de manera aleatoria para evitar cualquier sesgo en la selección de los datos de entrenamiento y prueba.

El parámetro train_size especifica la proporción del conjunto de datos que se asignará. En este proyecto se ha establecido en 0.7, lo que significa que el 70% de los datos se utilizarán para entrenar el modelo y el 30% al conjunto de prueba.

Cabe destacar que en este proyecto específico no se ha realizado una división adicional del conjunto de datos para incluir un conjunto de validación separado. La división se ha llevado a cabo únicamente en conjuntos de entrenamiento y prueba.

Sin embargo, la evaluación del modelo se realiza utilizando métricas de evaluación como MSE, RMSE y R^2 en el conjunto de prueba. Estas métricas permiten validar y evaluar la precisión y el rendimiento del modelo en datos no vistos previamente.

Aunque no se cuente con un conjunto de validación específico, utilizar métricas de evaluación en el conjunto de prueba proporciona una estimación del rendimiento del modelo en datos no utilizados durante el entrenamiento.

4.5 Modelo de Machine learning

En el proyecto, se implementa un enfoque de modelado individual para cada calle de Madrid. Esto implica crear un modelo separado para cada calle, lo que permite capturar las características y patrones específicos de cada ubicación.

Tubería de datos

Una tubería de datos, también conocida como flujo de trabajo de datos, es una construcción lógica que representa un proceso dividido en fases.

Las tuberías permiten encapsular y organizar de manera eficiente las diferentes etapas de procesamiento de datos, lo que facilita su reutilización y reproducción en diferentes conjuntos de datos. Además, ayudan a evitar fugas de información entre los conjuntos de entrenamiento y prueba al aplicar cada transformación de manera consistente en ambos conjuntos [41].

Para la realización de la tubería en nuestro modelo utilizamos la función make_pipeline proporcionada por sklearn. La tubería contiene los siguientes pasos:

- Preprocesamiento de datos: Se utiliza la función make_column_transformer de sklearn para realizar las transformaciones señaladas en el punto 4.3.4.

- Modelo de Regresión: Se utiliza el TransformedTargetRegressor como el estimador principal del pipeline. Este estimador permite aplicar una transformación logarítmica a la variable objetivo antes de entrenar el modelo de regresión.

Algoritmos

En el proyecto se han creado los modelos con los algoritmos presentados anteriormente en el punto dos. Para su realización se han predefinido unos hiperparámetros que, después de la creación de los modelos se han ajustado mediante el método grid search proporcionado por sklearn.

Los hiperparámetros son variables de configuración que se utilizan para administrar el entrenamiento de los modelos de machine learning [42].

Algunos ejemplos de hiperparámetros son [43]:

- Observaciones mínimas para división: define el número mínimo de observaciones que debe tener un nodo para poder ser dividido. Cuanto mayor el valor, menos flexible es el modelo.

- Observaciones mínimas de nodo terminal: define el número mínimo de observaciones que deben tener los nodos terminales. Su efecto es muy similar al de observaciones mínimas para división.

- Profundidad máxima del árbol: define la profundidad máxima del árbol, entendiendo por profundidad máxima el número de divisiones de la rama más larga (en sentido descendente) del árbol.

- Número máximo de nodos terminales: define el número máximo de nodos terminales que puede tener el árbol. Una vez alcanzado el límite, se detienen las divisiones. Su efecto es similar al de controlar la profundidad máxima del árbol.

- Reducción mínima de error: define la reducción mínima de error que tiene que conseguir una división para que se lleve a cabo.

Ajuste de hiperparámetros

Para el ajuste de los hiperparámetros se ha realizado mediante el método grid search proporcionado por sklearn.

El grid search funciona por medio de la comparación realizando una búsqueda de todas las combinaciones de posibles hiperparámetros, entrenando y evaluando cada combinación. Al final, elige la combinación que proporciona el mejor rendimiento.

Persistencia de Modelos y Variables

Al crear el modelo, se utiliza la biblioteca Joblib para guardar el modelo y sus variables asociadas. La biblioteca Joblib es una herramienta de Python que permite almacenar y cargar objetos de Python de manera eficiente.

Al guardar el modelo y las variables se facilita su reutilización, reproducción y evaluación. Esto evita tener que recrear los modelos desde cero cada vez que se ejecuta la aplicación, ahorrando tiempo y recursos computacionales.

4.6 Análisis de resultados

Técnicas

Los modelos entrenados se evalúan con las técnicas el error absoluto medio (MAE), la raíz del error cuadrático medio (RMSE) y el coeficiente de determinación (R^2).

El MAE representa la cantidad promedio de personas por hora en la que las predicciones difieren de los valores reales.

El RMSE representa la raíz cuadrada de la media de los errores al cuadrado entre las predicciones y los valores reales.

Para realizarlo se utilizan las funciones mean_absolute_error, mean_squared_error, r2_score de la biblioteca de sklearn. Estas funciones toman como argumentos las etiquetas reales y las predicciones del modelo y devuelven el valor de la métrica correspondiente.

Evaluación del modelo tras el ajuste de hiperparámetros

Tabla 4.1: Evaluación de los modelos

	Regresión Lineal			Árbol de Decisión			Bosques aleatorios			Gradient Boosting		
	MAE	RMSE	R^2	MAE	RMSE	R^2	MAE	RMSE	R^2	MAE	RMSE	R^2
Entrenamiento	934.44	1432.37	0.39	624.32	1310.75	0.48	723.26	1046.51	0.55	771.78	1005.01	0.65
Prueba	941.88	1430.54	0.38	634.21	1308.95	0.47	788.31	1222.22	0.50	776.06	1220.86	0.58
Desempeño general	941.88	1430.54	0.38	634.21	1308.95	0.47	788.31	1222.22	0.50	776.06	1220.86	0.58

Para un mejor rendimiento del modelo se busca un MAE y RMSE bajos y un valor de R^2 cercano a uno. En base a los resultados obtenidos, podemos observar que el modelo de Gradient Boosting muestra el mejor rendimiento en términos de MAE y RMSE tanto en el conjunto de entrenamiento como en el conjunto de prueba. Además, tiene un coeficiente de determinación R^2 más alto en comparación con los otros modelos.

Dado que la creación de los modelos es un proceso lento, al haber comprobado que el modelo de Gradient Boosting muestra el mejor desempeño se procederá a modificar y hacer refinamientos en ese modelo.

Evaluación del modelo tras añadir el pronóstico del clima

Tabla 4.2: Evaluación con el pronóstico del clima

	Gradient Boosting		
	MAE	RMSE	R^2
Entrenamiento	251.23	503.86	0.90
Prueba	480.21	929.39	0.62
Desempeño general	480.21	929.39	0.62

Se puede observar que después de añadir el pronóstico del clima, el modelo muestra una mejora significativa en términos de MAE y RMSE en el conjunto de entrenamiento. Sin embargo, en el conjunto de prueba, aunque se logra una mejora en comparación con los resultados iniciales, los errores aún son más altos que en el conjunto de entrenamiento. El R^2 muestra una mejora en el conjunto de entrenamiento, lo que indica que el modelo explica una mayor proporción de la varianza en los datos de entrenamiento. En el conjunto de prueba, el R^2 también muestra una mejora en comparación con los resultados iniciales, pero aún hay margen para mejorar.

Evaluación del modelo tras los partidos de futbol

Tabla 4.3: Evaluación con partidos de fútbol

	Gradient Boosting		
	MAE	RMSE	R^2
Entrenamiento	280.01	522.83	0.93
Prueba	461.82	862.16	0.80
Desempeño general	461.82	862.16	0.80

Comparando estos resultados con los resultados previos, podemos observar que ha habido un aumento en el MAE y RMSE tanto en el conjunto de entrenamiento como en el conjunto de prueba. Sin embargo, el R^2 ha mejorado tanto en el conjunto de entrenamiento como en el conjunto de prueba.

Estos resultados indican que el modelo ha experimentado un deterioro en su capacidad de predicción después de los partidos de fútbol. El MAE y RMSE han aumentado, lo que significa que los errores de predicción han aumentado en promedio.

Sin embargo, el coeficiente de determinación ha mejorado, lo que indica que el modelo está explicando una mayor proporción de la varianza en los datos. Esto podría indicar que los partidos de fútbol introducen ruido o complejidad adicional al modelo, lo que resulta en un mayor error de predicción.

Evaluación del modelo tras añadir los datos de 2019

Tabla 4.4: Evaluación con los datos de 2019

	Gradient Boosting		
	MAE	RMSE	R^2
Entrenamiento	594.14	1519.64	0.60
Prueba	787.16	1724.85	0.49
Desempeño general	787.16	1724.85	0.49

Comparando estos resultados con los resultados anteriores, podemos observar que la adición de los datos de 2019 ha llevado a un deterioro en el desempeño del modelo. Tanto el MAE como el RMSE han aumentado significativamente en el conjunto de entrenamiento y prueba, lo que indica un mayor error promedio de predicción. Además, el R^2 ha disminuido tanto en el conjunto de entrenamiento como en el conjunto de prueba.

En base a estos resultados, eliminamos los datos de 2019 del modelo y reevaluamos cómo se comporta con los datos previos (pronóstico del clima y partidos de fútbol) para determinar si el modelo mejora en su capacidad de predicción

Eliminación de los datos de 2019

Tabla 4.5: Evaluación sin los datos de 2019

	Gradient Boosting		
	MAE	RMSE	R^2
Entrenamiento	286.39	543.35	0.92
Prueba	460.55	856.35	0.81
Desempeño general	460.55	856.18	0.81

Al eliminar los datos de 2019, el modelo ha logrado una reducción en los errores promedio de predicción (MAE y RMSE) tanto en el conjunto de entrenamiento como en el conjunto de prueba. Además, el coeficiente de determinación R^2 ha aumentado en ambos conjuntos, lo que indica que el modelo está explicando una mayor proporción de la varianza en los datos.

Puesto que el modelo se entrena con los datos de los peatones relativos a 2020 y 2021, añadir los datos de 2019, donde aún no estaban presentes los confinamientos ni las restricciones relacionadas con la pandemia, podría introducir un sesgo o ruido no representativo en el modelo y afectar negativamente su capacidad de predicción en el contexto.

Por lo tanto, al excluir los datos de 2019 se obtienen resultados más precisos y relevantes en relación con la situación que se vivió durante 2020 y 2021.

4.7 Definición del sistema propuesto para la aplicación

La aplicación web permite visualizar un mapa interactivo de calles de Madrid y realizar predicciones de peatones en función de la fecha y hora seleccionadas. También proporciona análisis de datos y opciones para actualizar los modelos utilizados en las predicciones. Para la mejora de la eficiencia de la aplicación, se utiliza programación por hilos y concurrent features. Los registros de la aplicación se almacenan en un archivo accesible llamado "log_file.log".

El sistema propuesto se basa en una arquitectura Cliente-Servidor, donde el servidor Flask desempeña el papel del componente de servidor. Además, se utiliza el patrón Modelo-Vista-Controlador (MVC) para estructurar el código dentro del servidor Flask.

Cliente-servidor

La arquitectura del sistema sigue una estructura cliente-servidor, donde la interfaz de usuario interactúa con el backend a través de solicitudes HTTP. El backend se encarga de procesar las solicitudes, interactuar con los datos y los modelos, y enviar las respuestas correspondientes a la interfaz de usuario.

Patrón Modelo-Vista-Controlador (MVC)

- El componente Modelo está representado por el módulo PredecirMadrid, que se encarga de interactuar con los módulos subyacentes al modelo y realizar las predicciones correspondientes.

- El componente Vista está representado por las rutas y vistas en Flask, que definen las diferentes páginas y la interacción con el usuario. Aquí se establecen las respuestas a las solicitudes del cliente y se generan las vistas adecuadas para mostrar la información.

- El componente Controlador es manejado por el propio servidor Flask, que se encarga de recibir las solicitudes del cliente, invocar las acciones correspondientes en el modelo y actualizar las vistas en función de los resultados.

Lenguaje de programación y control de versiones

- La aplicación se desarrolla utilizando Python.
- Se utiliza el sistema de control de versiones Git para el manejo del código fuente.

Bibliotecas

- Se utiliza Pandas, NumPy y datetime para el procesamiento de datos.
- Se emplea Folium bibliotecas para la creación y visualización de mapas.
- Scikit-learn para realizar las tareas correspondientes con el modelo.
- Joblib para el almacenamiento de los modelos y las variables.
- Logging para la generación de los logs.
- Threading y Concurrent para un procesamiento eficiente de los datos.
- Flask para la creación del servidor web.
- Matplotlib para la representación de los datos.

Organización de la aplicación

1. Parte de los datos: en esta sección, se importan y unifican todos los datos necesarios para la creación del modelo. La clase "DatosConjuntos" se encarga de cargar los datos desde diferentes fuentes y unificarlos, como se puede observar en la siguiente Figura 4.5.

2. Parte del modelo: en esta sección, se encuentran las diferentes clases relacionadas con la creación y el manejo del modelo. La clase "PredecirMadrid" actúa como una interfaz para unificar y gestionar estas clases, como se puede observar en la siguiente Figura 4.6.

Diagrama de clases

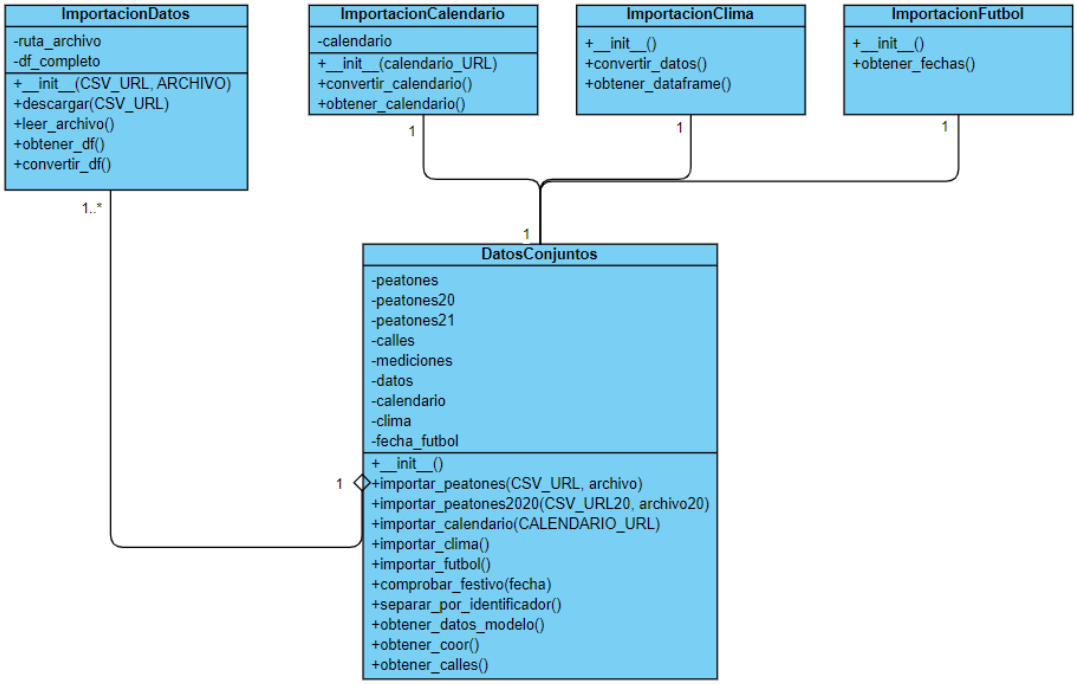

Figura 4.5: Diagrama de clases importación de datos

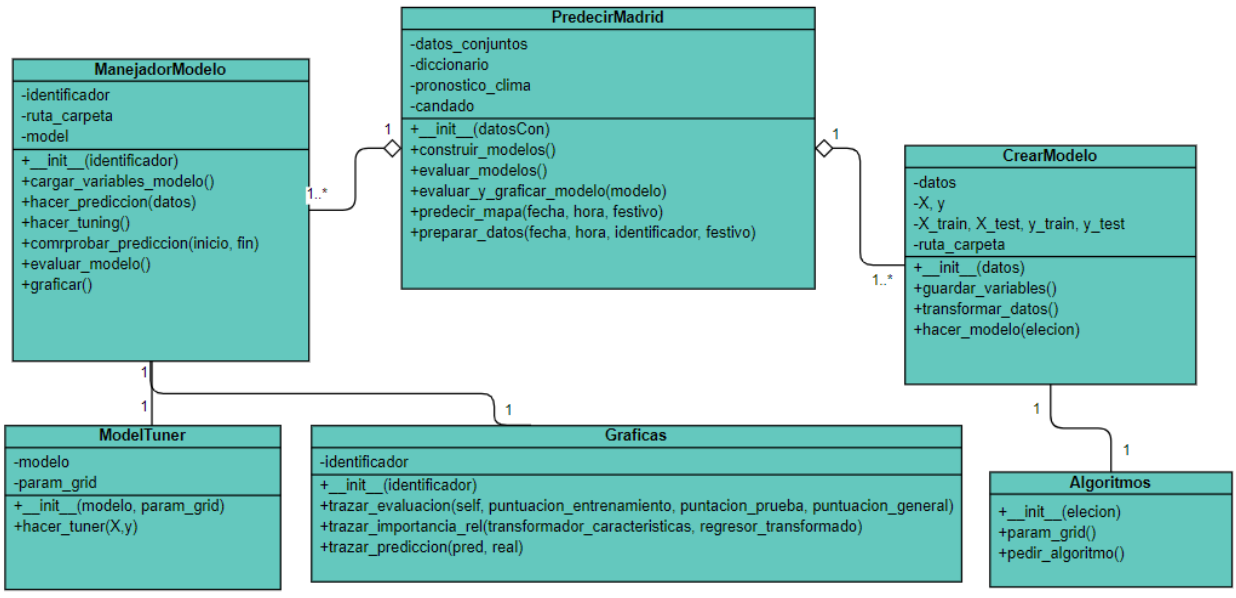

Figura 4.6: Diagrama de clases módulos del modelo

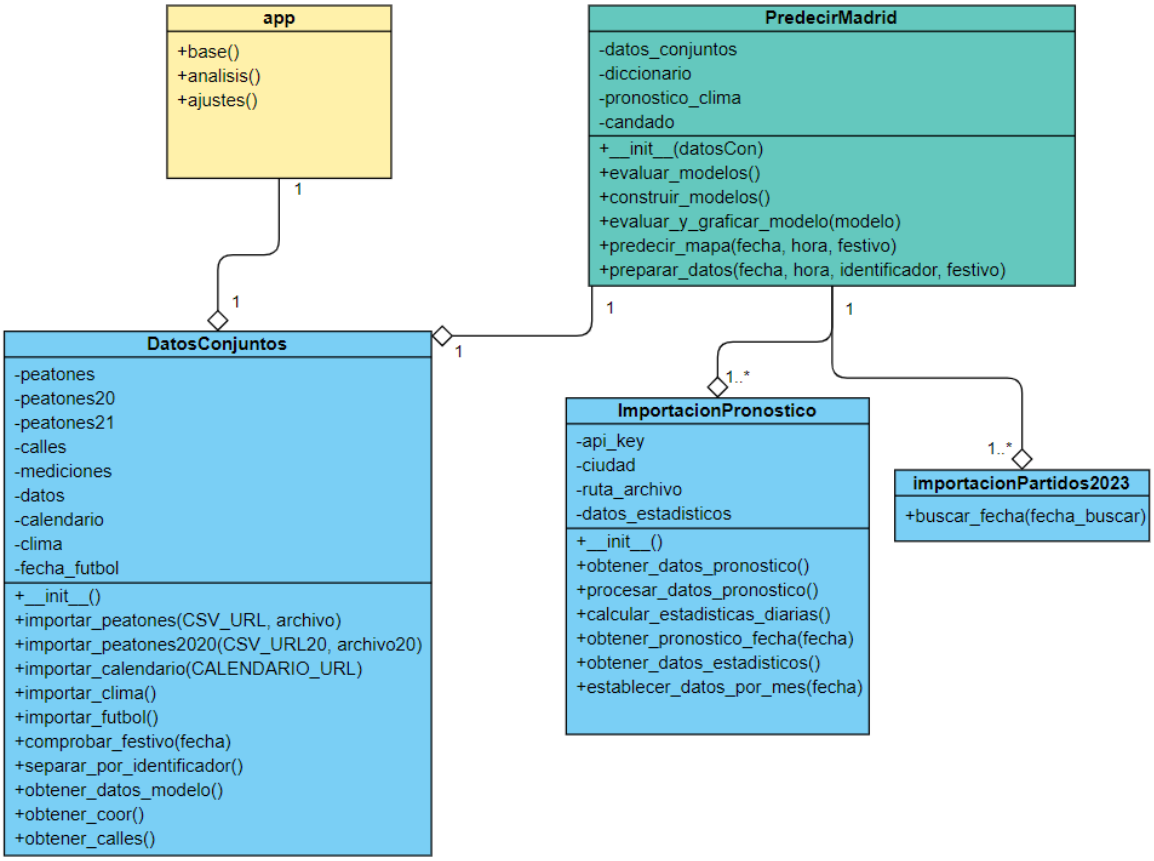

Figura 4.7: Diagrama de clases general

Diagramas de secuencia

Para explicar el funcionamiento de la aplicación se ha considerado que el diagrama de secuencia de la realización de una predicción y el diagrama de secuencia de la creación de los modelos son los más relevantes.

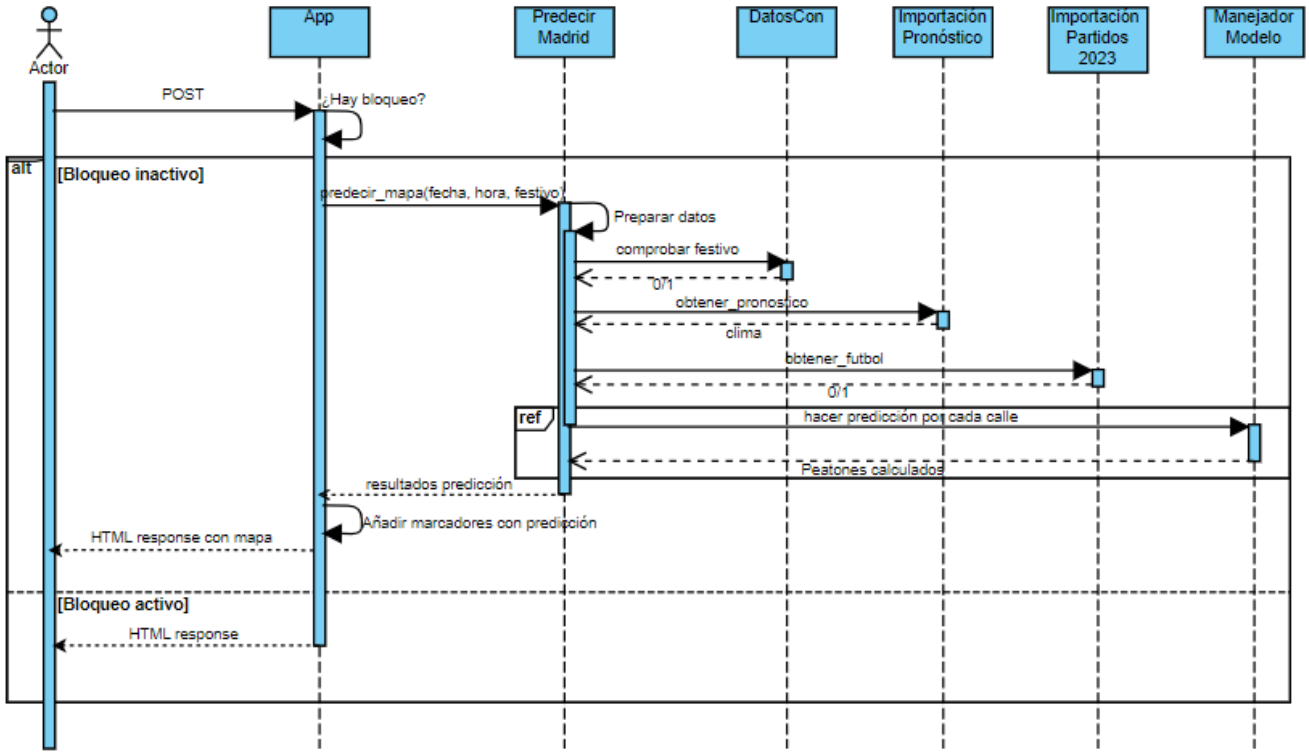

Figura 4.8: Diagrama de secuencia realizar predicción

1. El usuario accede a la página principal de la aplicación web, donde se muestra el mapa de Folium centrado en Madrid.

2. El usuario completa el formulario con la fecha y hora para la predicción y envía la solicitud.

3. El servidor recibe la solicitud POST y llama a la función base().

4. La función base() verifica si el bloqueo está activo.

5. Si el bloqueo está activo, se envía una respuesta al usuario indicando que no se pueden hacer predicciones mientras se crean los modelos y que debe esperar.

6. Si el bloqueo no está activo, se continúa con la ejecución.

7. Se obtienen la fecha y la hora proporcionadas por el usuario en el formulario.

8. Se realiza la petición para la predicción del mapa mediante el método predecir_mapa(fecha, hora) con la fecha y hora proporcionadas por el usuario en el formulario.

9. Se preparan los datos antes de realizar la predicción:

    a. Se llama a la clase DatosConjuntos para comprobar si el día elegido es festivo.

    b. Se llama a la clase ImportaciónPronóstico para obtener el pronóstico de ese día o los valores comunes climatológicos para ese mes si la predicción es más lejana de 5 días.

    c. Se llama a la clase ImportacionPartidos2023 para comprobar si hay un partido el día seleccionado.

10. Se manda la petición ya con los datos preparados a la clase ManejadorModelo donde se realiza la predicción por cada calle.

11. Si se produce un error durante la predicción, se captura la excepción y se registra un mensaje de error.

12. Se crea un marcador en el mapa en la ubicación, el nombre de la ubicación y el número de peatones predicho por cada calle.

13. Se renderiza la plantilla "mapa.html" y se le pasan los datos del mapa con la fecha y hora de la predicción.

14. La plantilla se muestra al usuario, y se muestra el mapa con los marcadores y la información de la predicción.

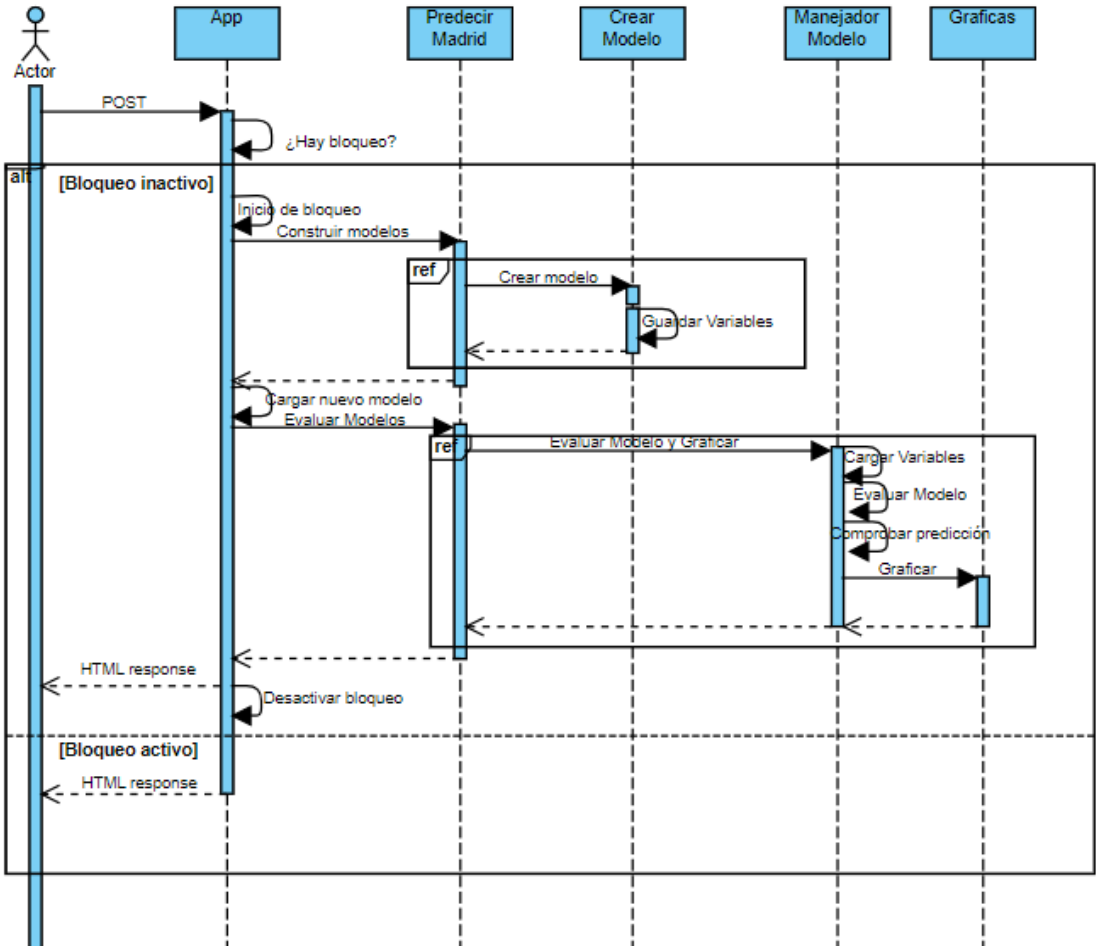

Figura 4.9: Diagrama de secuencia crear modelos

1. El usuario realiza una solicitud POST a la ruta '/ajustes'.
2. Se verifica si el candado está activo.
3. Si el candado está activo se devuelve como respuesta el mensaje "Ya se están creando los modelos, espere".
4. Se adquiere el candado para bloquear las solicitudes.
5. Se llama al método construir_modelos para crear los modelos.
    a. La clase de PredecirMadrid realiza una iteración por cada calle solicitando la creación de los modelos a la clase CrearModelo.

b. La clase CrearModelo guarda el modelo y sus variables de forma que no se tengan que crear cada vez que se realice una predicción.

6. Se llama a la clase ManejadorModelo por cada calle para evaluar los modelos construidos:
    a. Carga los modelos y sus variables que han sido creados.
    b. Evalúa el modelo calculando su MSE, RMSE y el R^2.
    c. Comprueba la predicción para 10 muestras
    d. Llama a la clase Graficas que se encarga de graficar la evaluación, la importancia relativa y un ejemplo de muestra de predicción.
7. Se obtiene la fecha actual como la fecha de la última actualización.
8. Si se produce una excepción durante el proceso, se registra un mensaje de error y se establece la fecha de la última actualización como "Error al actualizar el modelo".
9. Se libera el candado para permitir nuevas solicitudes.
10. Se renderiza el template 'ajustes.html' con la fecha de la última actualización.

Interfaz de usuario

A continuación, se muestra cómo se visualiza la aplicación:

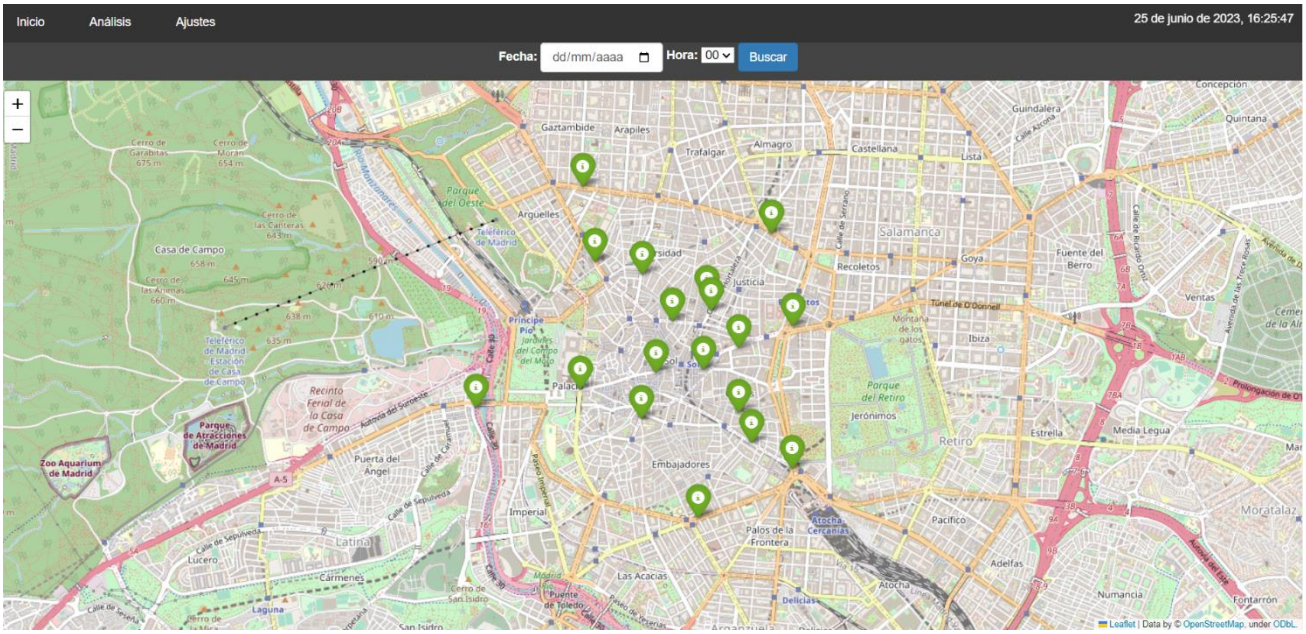

Figura 4.10: Inicio app

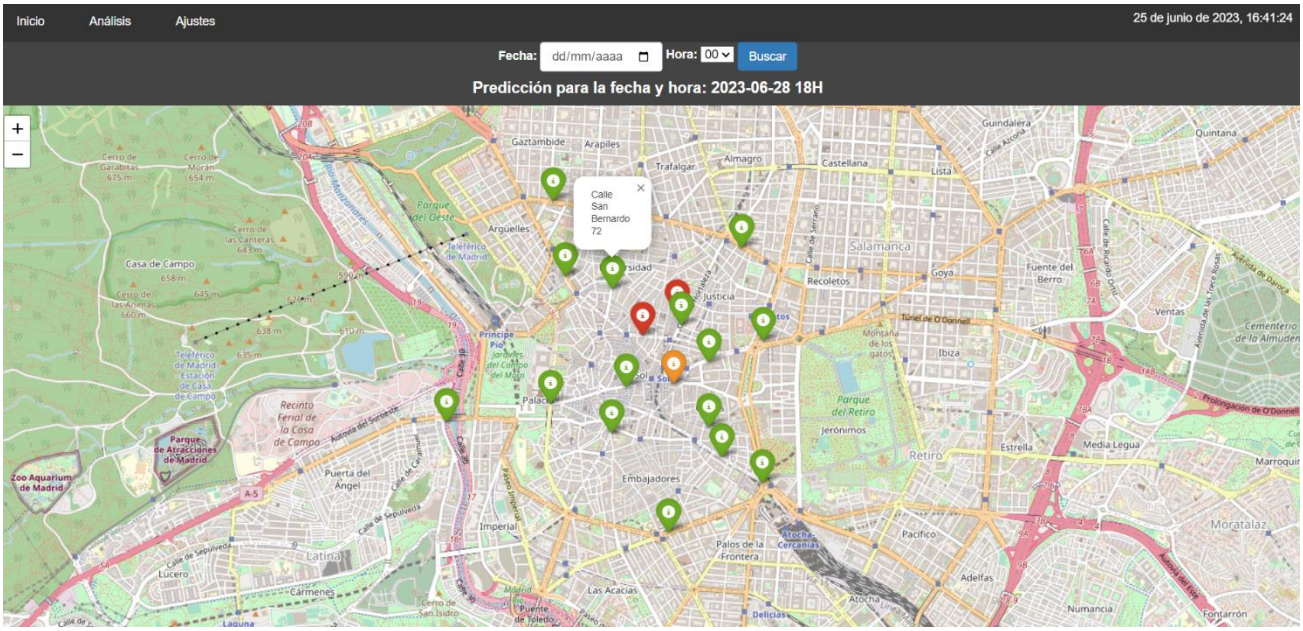

Figura 4.11: Ejemplo de predicción 1

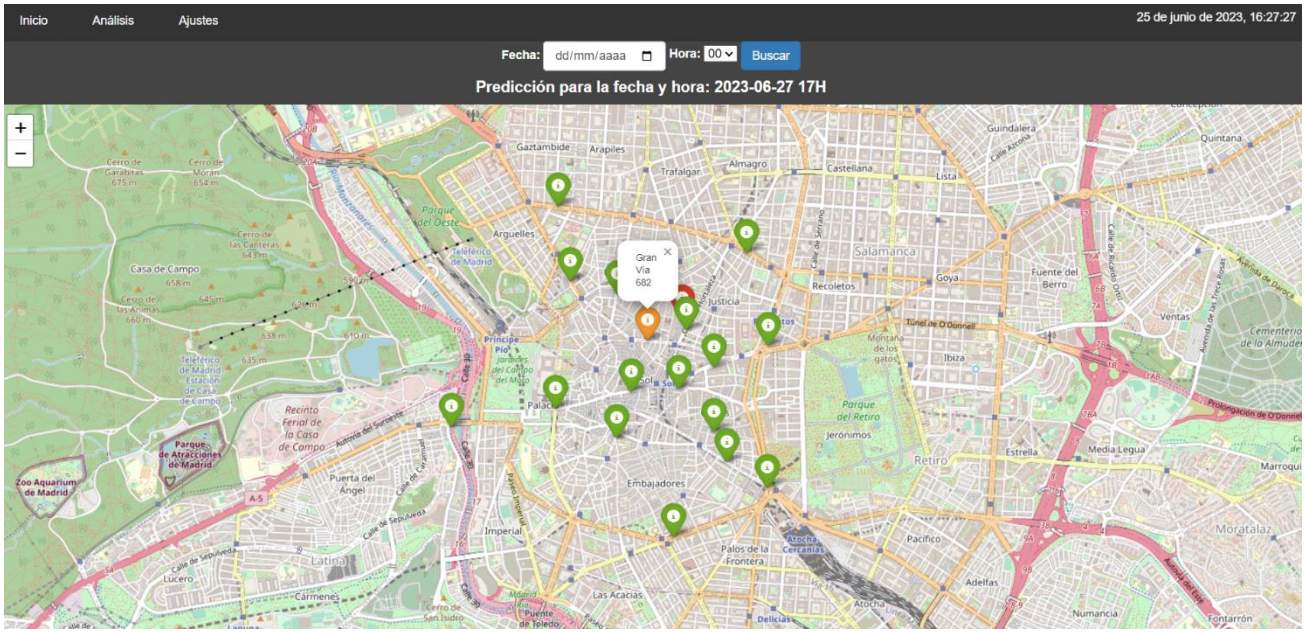

Figura 4.12: Ejemplo de predicción 2

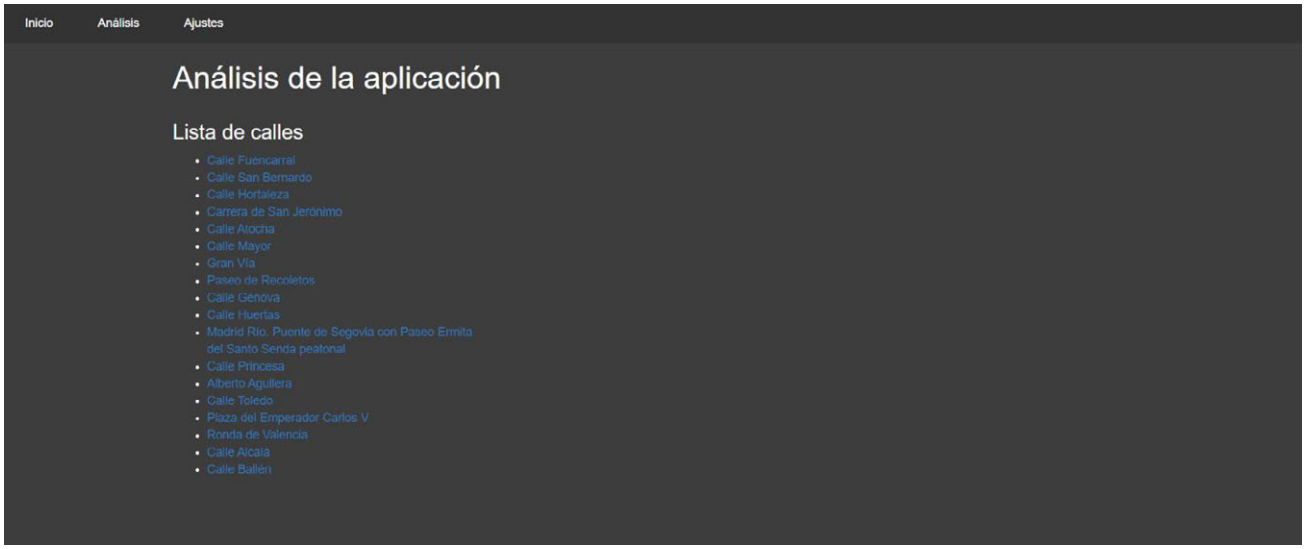

Figura 4.13: Visualización del análisis de la aplicación

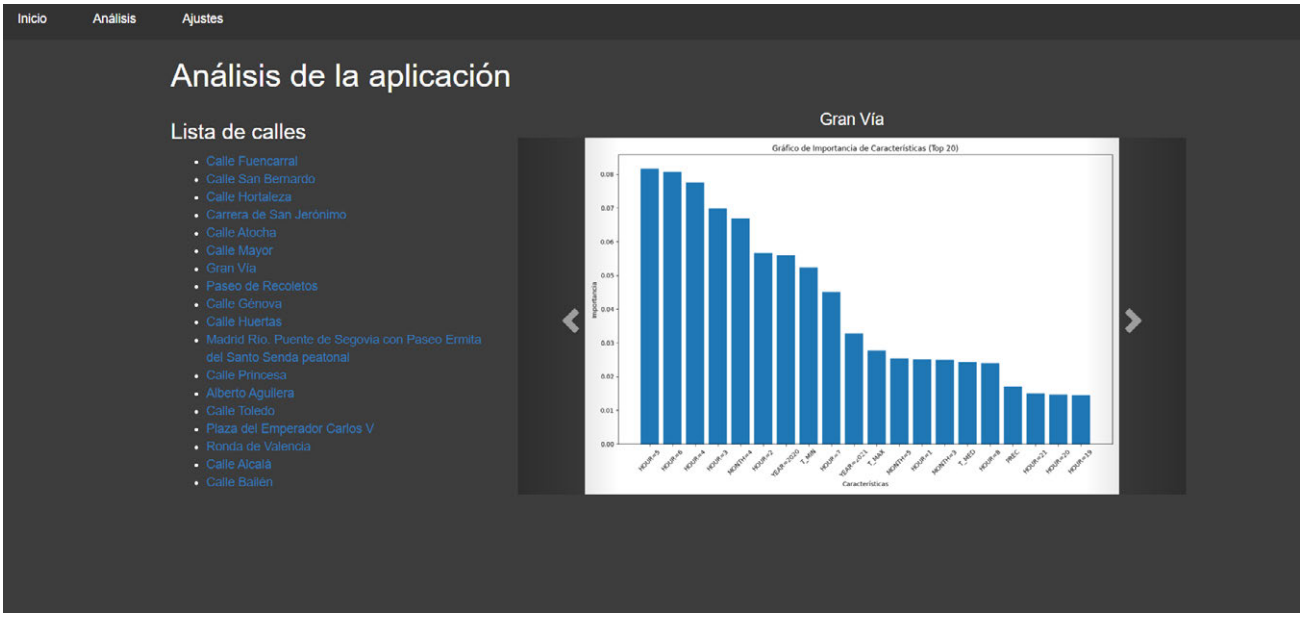

Figura 4.14: Ejemplo de visualización importancia relativa

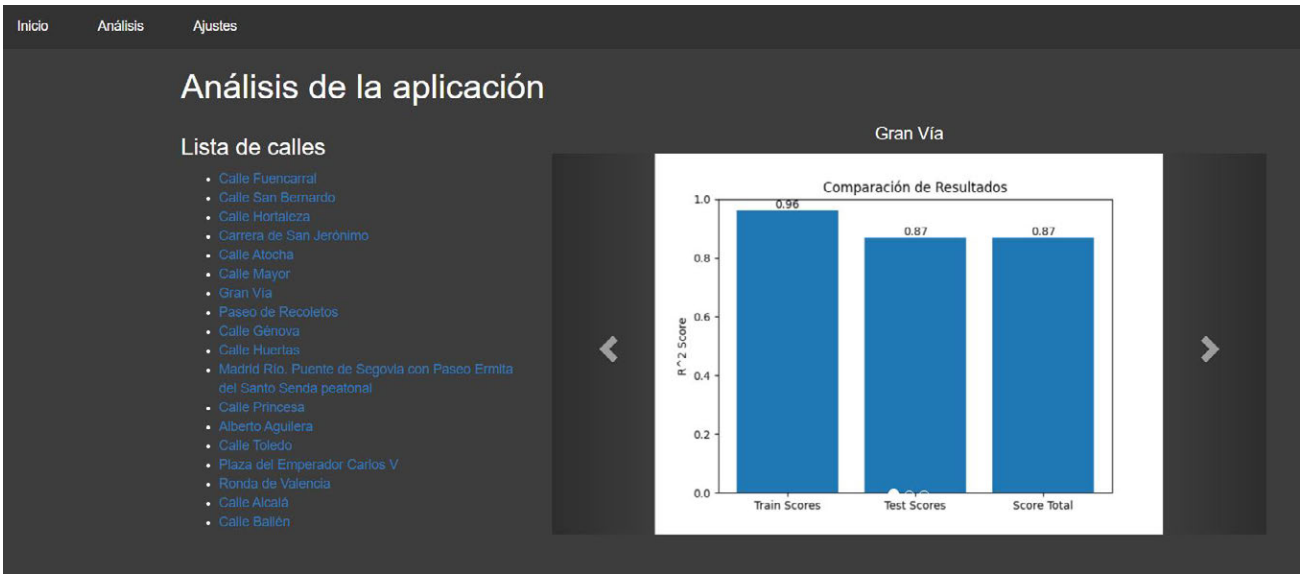

Figura 4.15: Ejemplo de visualización del coeficiente de determinación

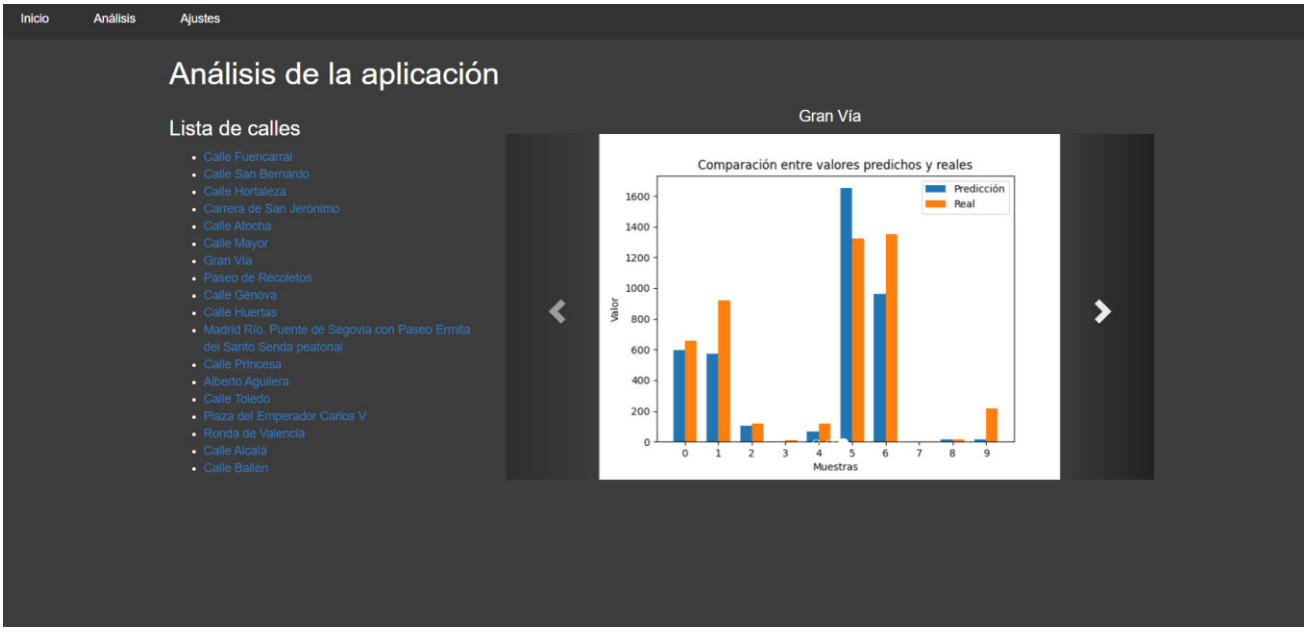

Figura 4.16: Ejemplo visualización de muestra de predicciones

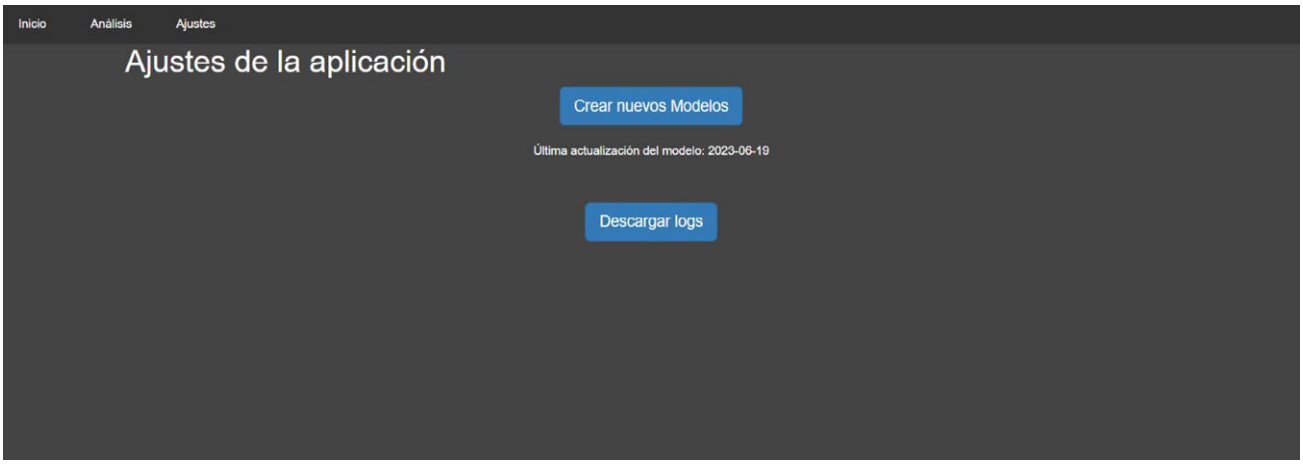

Figura 4.17: Ajustes de la aplicación

5. Resultados

Comparando los resultados de todos los modelos podemos observar que hay bastante disparidad entre uno y otro.

El mejor resultado es el modelo de la calle Gran Vía, con un coeficiente de relación de 0.87, es decir que el 87% de la variabilidad de los datos se puede explicar. Este resultado indica una correlación positiva fuerte, lo que sugiere que el modelo tiene un buen rendimiento en la predicción de la cantidad de peatones en la calle Gran Vía.

El peor resultado corresponde con la calle Huertas, con un coeficiente de relación de 0.45. Este resultado indica una correlación más débil, lo que sugiere que el modelo tiene un rendimiento más bajo en la predicción de la cantidad de peatones en la calle Huertas.

Un coeficiente de determinación de 0.45 sugiere que los datos utilizados para evaluar el modelo en la calle Huertas no muestran una relación tan fuerte entre las variables de entrada y la cantidad de usuarios en la vía. Esto puede indicar que hay otras variables o factores específicos de la calle Huertas que influyen en la afluencia de usuarios y que no se están teniendo en cuenta en el modelo.

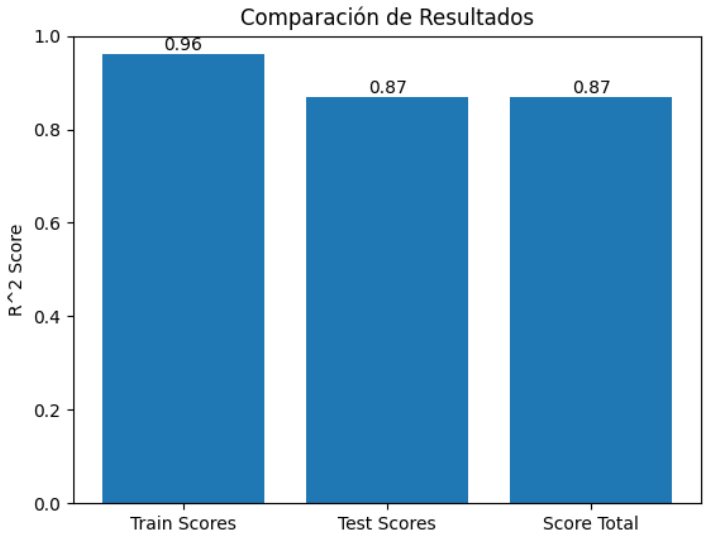

Figura 5.1: Coeficiente de determinación calle Gran Vía

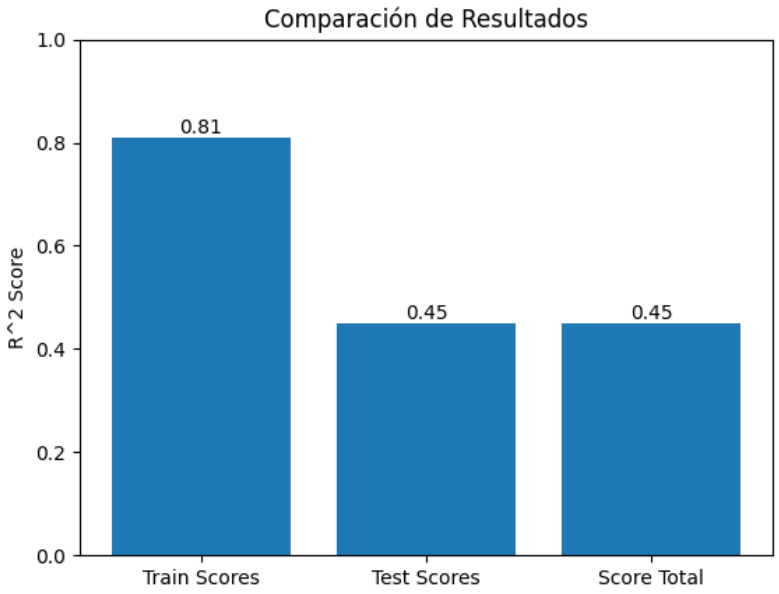

Figura 5.2: Coeficiente de determinación calle Huertas

5.1 Interpretación del comportamiento de los modelos

Las gráficas de importancia de características proporcionan información relevante para comprender el comportamiento de los modelos. Al analizar la Figura 5.3, correspondiente a la calle Princesa, se pueden identificar las características más importantes. En este caso, se observa que las horas comprendidas entre las 2 y las 7 de la mañana tienen un impacto significativo en el modelo. Además, se destaca que los días de la semana relevantes son los correspondientes a los fines de semana.

Esta tendencia puede explicarse por la presencia de numerosos locales de ocio nocturno en la calle Princesa. Durante esas horas y los fines de semana, es probable que haya un mayor flujo de personas y actividad en esa zona, lo cual impacta en la variable que se está analizando.

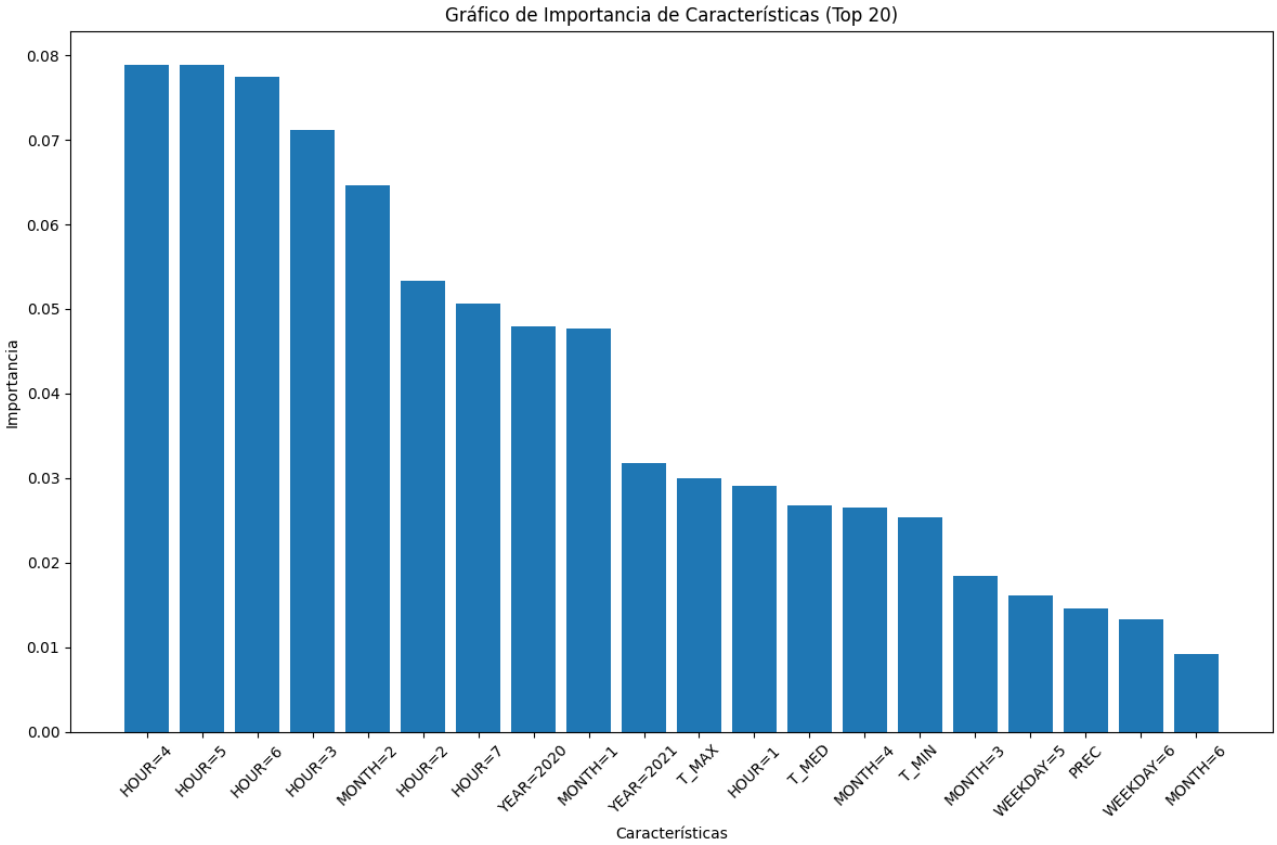

Figura 5.3: Importancia relativa calle Princesa

En la figura 5.4, que se refiere a la importancia relativa de las características de Madrid Río, se evidencia la presencia de variables relacionadas con la temperatura. Esto sugiere que las condiciones climáticas desempeñan un papel importante en la experiencia de las personas que disfrutan del parque y participan en actividades al aire libre en Madrid Río.

Además, se destaca que los fines de semana también se consideran variables relevantes en el modelo. Esto puede indicar que los patrones de uso y la afluencia de visitantes en Madrid Río pueden variar durante los días de descanso en comparación con los días laborables.

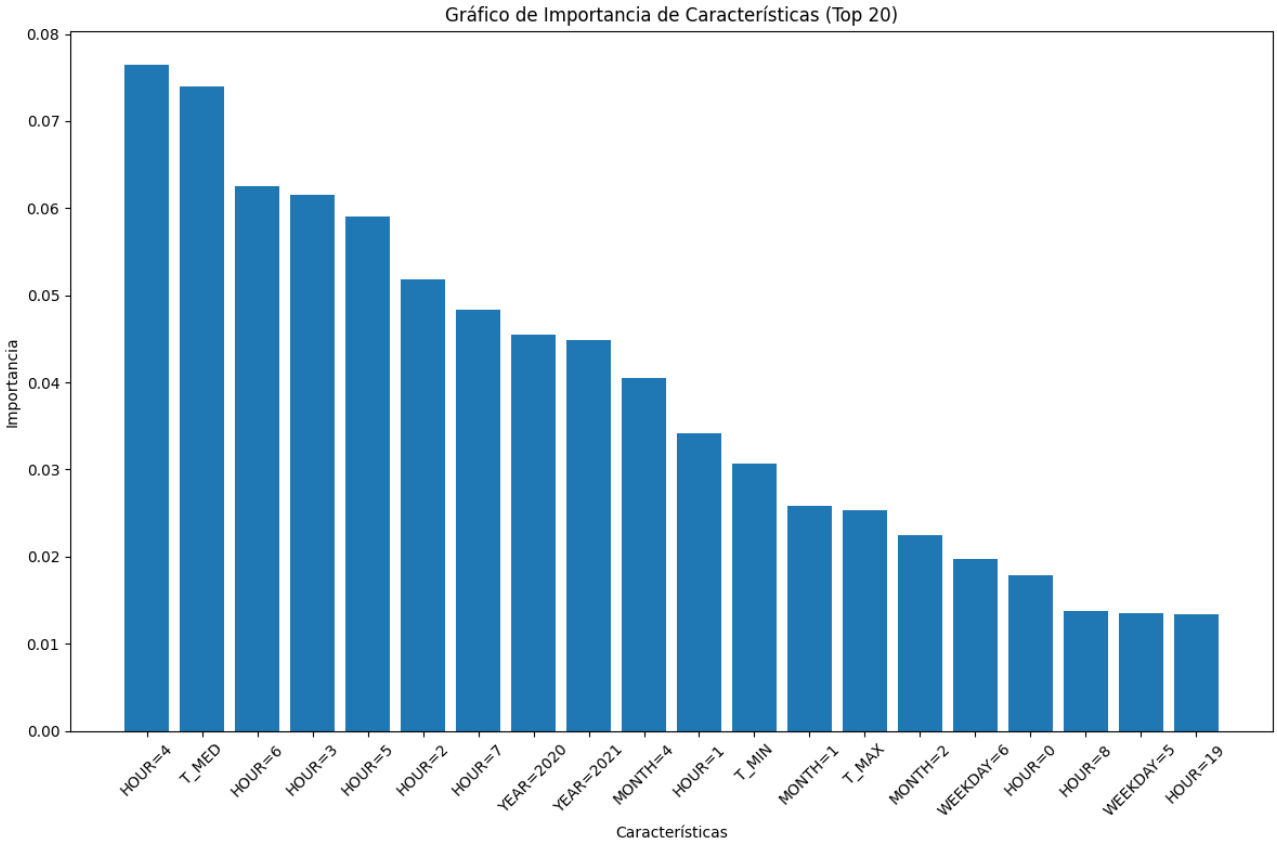

Figura 5.4: Importancia relativa Madrid Río

5.2 Análisis del error obtenido

En el análisis del error obtenido en este proyecto, se identificaron diversos aspectos que pueden influir en la precisión del modelo.

Un aspecto relevante es que el modelo no tiene en cuenta los eventos especiales y puntuales, así como las manifestaciones. Estos eventos pueden tener un impacto significativo en la afluencia de personas en la vía, lo que podría afectar las predicciones realizadas por el modelo. Al no considerar estos factores, es posible que el modelo no capture completamente las variaciones en la cantidad de usuarios durante situaciones particulares.

Otro aspecto que considerar es la no utilización de variables indirectas que podrían estar correlacionadas con la presencia de eventos y manifestaciones. Aunque se exploró esta posibilidad, la falta de información adecuada y la necesidad de validar cuidadosamente estas variables impidieron su inclusión en el modelo final. Esta omisión puede haber contribuido al

error observado al no capturar completamente el impacto real de los eventos especiales en la cantidad de usuarios.

5.3 Problemas encontrados

Durante el desarrollo del proyecto, se identificaron varios desafíos que afectaron su progreso y los resultados obtenidos. A continuación, se presentan los problemas identificados:

- Obtención de datos de diversas fuentes: Surgió el desafío de obtener los datos necesarios de diferentes fuentes. No existía un único conjunto de datos completo que incluyera todas las variables requeridas para el modelo. Como resultado, fue necesario realizar un proceso de Extracción, Transformación y Carga (ETL) para unificar y combinar los datos provenientes de múltiples fuentes en un conjunto de datos coherente y útil.

- Escasez y falta de calidad de datos relacionados con los partidos de fútbol: En particular, se encontraron dificultades para encontrar conjuntos de datos de calidad que incluyeran información relevante sobre los partidos de fútbol.

- Limitaciones computacionales: Durante el ajuste de los hiperparámetros del modelo, se experimentaron limitaciones en términos de recursos computacionales. El proceso de ajuste de hiperparámetros requería un tiempo considerable, lo cual limitaba la capacidad para explorar exhaustivamente todas las combinaciones posibles. Esto podría haber afectado la optimización del modelo y la búsqueda de los mejores parámetros para mejorar su rendimiento.

6. Presupuesto

Desarrollo

El coste asociado para llevar a cabo el desarrollo del proyecto se divide principalmente en recursos computacionales y humanos.

1. Recursos Humanos:

   - Se requeriría el sueldo de un Ingeniero de Aplicaciones durante 1 mes para el desarrollo de la aplicación Flask.

   - Se necesitaría el sueldo de un Ingeniero de Datos durante 3 meses para la creación del modelo.

2. Recursos Tecnológicos:

   Para la implementación del proyecto, se requiere el uso de una instancia EC2 de AWS, la cual proveerá los recursos computacionales necesarios. Las consideraciones y costos asociados al uso de esta instancia son:

   - La ubicación de la instancia sería en región española de AWS.

   - Se seleccionaría una instancia de tamaño mediano, en este caso, una instancia m5.large, cuyo costo es de 0,10€ por hora en la región española.

   - La instancia estaría encendida durante un mínimo de 8 horas al día, de lunes a viernes, durante los 4 meses de desarrollo.

Teniendo en cuenta estas consideraciones, el costo mensual para el uso de la instancia EC2 sería de 16€, lo que se traduce en un costo total de 64€ durante los 4 meses de desarrollo. Es importante destacar que esta misma instancia se utilizará tanto para el desarrollo de la aplicación Flask como para la creación del modelo de datos.

Producción

En producción, el costo de mantener el sistema estará influenciado principalmente por el tráfico de red. A medida que aumenta el tráfico de usuarios y las solicitudes de la aplicación, se necesitarán contratar más instancias para manejar la carga de trabajo de manera eficiente.

Para determinar la cantidad de instancias necesarias, se deben realizar pruebas de carga y análisis de rendimiento. Esto permitirá evaluar el rendimiento del sistema bajo diferentes niveles de tráfico y determinar la capacidad necesaria para manejar la carga prevista.

En el caso específico de mantener la infraestructura 24 horas los 7 días de la semana en la región española de AWS, los cálculos proporcionados son los siguientes:

- Número de instancias: 2 para lograr alta disponibilidad.
- Costo de cada instancia: 0,1€ por hora.
- Horas al mes: 730 horas.

Añadir un balanceador de carga para las 2 instancias no tiene costo adicional, ya que AWS regala una IP pública por instancia.

Sumándolo todo (el precio de las dos instancias) tendríamos un gasto en producción mensual de 146€.

Los cálculos se han realizado con la calculadora de precios de AWS [44].

Es relevante mencionar que, en cuanto a los datos utilizados en el proyecto, al ser de acceso abierto y las herramientas a utilizar contar con licencias gratuitas, no se requerirá de ningún presupuesto adicional para la adquisición de datos.

7. Impacto del proyecto

El proyecto tiene un potencial impacto positivo en términos de Responsabilidad Social y Ambiental. Algunas maneras en las que este proyecto puede contribuir a estos aspectos:

- Fomento de la movilidad sostenible: Al predecir la afluencia de peatones y bicicletas en las calles principales de Madrid, la aplicación puede fomentar la movilidad sostenible al proporcionar información valiosa para que las personas elijan caminar o ir en bicicleta en lugar de utilizar vehículos motorizados. Esto contribuye a reducir la congestión del tráfico, disminuir las emisiones de gases de efecto invernadero y mejorar la calidad del aire en la ciudad.

- Planificación urbana inteligente: Al recopilar datos sobre los usuarios de la vía en diferentes áreas de Madrid, la aplicación puede proporcionar información útil para la planificación urbana y el diseño de infraestructuras. Estos datos pueden ayudar a identificar las zonas con mayor demanda de infraestructuras para peatones y ciclistas, lo que facilita la toma de decisiones informadas en la asignación de recursos y en la mejora de la seguridad vial.

- Promoción de estilos de vida saludables: Al ofrecer información sobre las calles más transitadas por peatones y bicicletas, la aplicación puede fomentar estilos de vida saludables al animar a las personas a caminar o andar en bicicleta en lugar de utilizar medios de transporte menos activos. Esto puede tener beneficios para la salud de los individuos y contribuir a la reducción de enfermedades relacionadas con el sedentarismo.

- Sensibilización y concienciación: Al visualizar los datos de afluencia de peatones y bicicletas en la ciudad, la aplicación puede ayudar a sensibilizar y concienciar a la población sobre la importancia de la movilidad sostenible y los beneficios de desplazarse a pie o en bicicleta. Esto puede promover cambios en los comportamientos y actitudes hacia una mayor adopción de medios de transporte más respetuosos con el medio ambiente.

8. Conclusiones

En conclusión, este Trabajo de fin de Grado ha logrado desarrollar una aplicación basada en técnicas de aprendizaje automático para predecir la afluencia de usuarios en las calles principales de Madrid, logrando satisfacer los objetivos propuestos. Para ello, se crearon modelos de aprendizaje automático específicos para cada calle, considerando las características y patrones individuales de cada ubicación. Durante el análisis comparativo de diferentes algoritmos, se evaluó el rendimiento de Regresión Lineal, Árboles de Decisión, Bosques Aleatorios y Gradient Boosting.

Tras la evaluación, se ha demostrado que el algoritmo de Gradient Boosting es el que mejor se adapta al modelo de predicción de la afluencia de usuarios. Se obtuvieron resultados prometedores, con un coeficiente de determinación (R^2) de 0.87, lo que indica que aproximadamente el 87% de la variabilidad de los datos se puede explicar por el modelo. Esto sugiere que el modelo tiene una buena capacidad para capturar las tendencias y patrones en los datos.

Se logró reducir el MAE (Error Absoluto Medio) de 776.06 a 460.55. Esto implica que, en promedio, las predicciones difieren de los valores reales en aproximadamente 460.55 personas por hora.

Asimismo, el RMSE (Raíz del Error Cuadrático Medio) también se redujo de 1220.86 a 856.18. Esto indica que la dispersión promedio de los errores entre las predicciones y los valores reales se redujo a aproximadamente 856.18 personas al cuadrado por hora.

Los valores de MAE y RMSE son altos debido a la naturaleza impredecible de los datos. Los datos de afluencia de peatones y bicicletas son altamente variables y están influenciados por eventos impredecibles que pueden causar fluctuaciones bruscas en la cantidad de personas. Cuando el modelo no puede capturar correctamente estos eventos impredecibles, puede producir errores con una diferencia alta entre las predicciones y los valores reales, lo que se refleja en valores altos de MAE y RMSE.

Es importante destacar que se han realizado modelos específicos para cada calle de Madrid, lo que ha revelado resultados diferentes para cada uno de ellos. Además, se ha observado que los parámetros más relevantes del modelo también varían según la calle.

Los resultados obtenidos respaldan la validez y utilidad de la aplicación desarrollada. Sin embargo, se puede concluir que la afluencia de usuarios en las calles principales de Madrid es un problema complejo y altamente dependiente de diversos factores, lo que requiere un enfoque específico para cada calle.

Durante el desarrollo del proyecto se implementaron técnicas de optimización para mejorar la eficiencia del programa y se desarrolló una interfaz web utilizando el framework Flask, lo que permitió a los usuarios interactuar de manera intuitiva con la aplicación y obtener predicciones de la afluencia de usuarios en tiempo real.

Líneas futuras

Como líneas futuras de trabajo, se pueden explorar las siguientes mejoras:

1. Utilización de servicios en la nube: Se propone utilizar servicios en la nube, como Amazon SageMaker o Azure Machine Learning, para el desarrollo y despliegue del modelo. Estas plataformas ofrecen recursos especializados, escalabilidad y flexibilidad, lo que facilitará el desarrollo y la gestión del modelo. Además, se pueden aprovechar las capacidades de escalado automático y el uso de recursos especializados en aprendizaje automático que ofrecen estos servicios en la nube.

2. Corrección del sobreajuste (overfitting): es importante abordar el problema del sobreajuste del modelo. El sobreajuste ocurre cuando el modelo se ajusta demasiado a los datos de entrenamiento y no generaliza bien a nuevos datos. Para mitigar el sobreajuste, se pueden aplicar diversas estrategias, como el uso de conjuntos de validación, técnicas de regularización y ajuste de hiperparámetros.

3. Exploración de técnicas avanzadas de aprendizaje automático: Se pueden investigar técnicas más avanzadas, como el aprendizaje profundo (deep learning), para mejorar la precisión y el rendimiento del modelo. La aplicación de modelos de aprendizaje profundo podría permitir descubrir patrones y relaciones más complejas en los datos, lo que podría resultar en predicciones más precisas y mejores resultados.

4. Integración del modelo con el ecosistema Hadoop: Se puede explorar la integración del modelo de predicción con Hadoop, un framework de procesamiento distribuido. Esta integración permitiría aprovechar el procesamiento paralelo y distribuido de datos, así como el almacenamiento escalable que ofrece Hadoop. Esto mejorará la eficiencia y escalabilidad del modelo al manejar grandes volúmenes de datos y permitir un procesamiento más rápido.

9. Lista de referencias bibliográficas

[1] Microsoft, «What is a Machine Learning Model?,» [En línea]. Available: https://learn.microsoft.com/es-es/windows/ai/windows-ml/what-is-a-machine-learning-model.

[2] V. Granville, «Data Science Central,» [En línea]. Available: https://www.datasciencecentral.com/how-to-choose-a-machine-learning-model-some-guidelines/.

[3] IBM, «Supervised Learning,» [En línea]. Available: https://www.ibm.com/es-es/topics/supervised-learning.

[4] «JavaTpoint,» [En línea]. Available: https://www.javatpoint.com/supervised-machine-learning.

[5] AprendeIA, «Aprendizaje no supervisado en Machine Learning,» [En línea]. Available: https://aprendeia.com/aprendizaje-no-supervisado-machine-learning/.

[6] «MyGreatLearning,» [En línea]. Available: https://www.mygreatlearning.com/blog/unsupervised-machine-learning/.

[7] «Aprendizaje por Refuerzo,» [En línea]. Available: https://www.aprendemachinelearning.com/aprendizaje-por-refuerzo/.

[8] «Xataka,» [En línea]. Available: https://www.xataka.com/inteligencia-artificial/conceptos-inteligencia-artificial-que-aprendizaje-refuerzo.

[9] E. Blogthinkbig, «Aprendizaje por Transferencia,» [En línea]. Available: https://datascience.eu/es/aprendizaje-automatico/aprendizaje-por-transferencia/.

[10] DataScience.eu, «Aprendizaje por Transferencia,» [En línea]. Available: https://datascience.eu/es/aprendizaje-automatico/aprendizaje-por-transferencia/.

[11] «Regresión Lineal en Español con Python,» [En línea]. Available: https://www.aprendemachinelearning.com/regresion-lineal-en-espanol-con-python/#more-5722.

[12] «Machine Learning para Todos,» [En línea]. Available: https://machinelearningparatodos.com/problemas-comunes-en-aprendizaje-automatico/.

[13] L. Gonzalez, «Regresión Polinomial – Teoría,» [En línea]. Available: https://aprendeia.com/algoritmo-regresion-polinomial-machine-learning/.

[14] «ML Polynomial Regression,» [En línea]. Available: https://www.javatpoint.com/machine-learning-polynomial-regression.

[15] L. Gonzalez, «Vectores de Soporte Regresión – Teoría,» [En línea]. Available: https://aprendeia.com/algoritmo-maquina-de-vectores-de-soporte-regresion-machine-learning/.

[16] «Qué son los árboles de decisión y para qué sirven?,» [En línea]. Available: https://www.maximaformacion.es/blog-dat/que-son-los-arboles-de-decision-y-para-que-sirven/.

[17] «AnswersDive,» [En línea]. Available: https://www.answersdive.com/ExpertAnswers/decision-tree-one-popular-machine-learning-algorithms-used-along-decision-trees-used-class.

[18] «Introduction to Random Forest in Machine Learning,» [En línea]. Available: https://www.section.io/engineering-education/introduction-to-random-forest-in-machine-learning/.

[19] «TIBCO Reference Center,» [En línea]. Available: [g]https://www.tibco.com/reference-center/what-is-a-random-forest.

[20] «All You Need to Know About Gradient Boosting Algorithm - Part 1: Regression,» [En línea]. Available: https://towardsdatascience.com/all-you-need-to-know-about-gradient-boosting-algorithm-part-1-regression-2520a34a502.

[21] A. Group, «Do you know what is GBM?,» [En línea]. Available: https://aaweg-i.medium.com/do-you-know-what-is-gbm-bcad121a3afc.

[22] «LSI - Universidad de Sevilla,» [En línea]. Available: http://www.lsi.us.es/redmidas/IIreunion/trans/prepro.pdf.

[23] I. Chaos, «Gestión de valores nulos,» [En línea]. Available: https://interactivechaos.com/es/manual/tutorial-de-machine-learning/gestion-de-valores-nulos.

[24] «Escalado de datos,» [En línea]. Available: https://interactivechaos.com/es/manual/tutorial-de-machine-learning/escalado-de-datos.

[25] J. M. Alvarez, «Categorías y la codificación One-Hot,» [En línea]. Available: http://blog.josemarianoalvarez.com/2018/03/15/categorias-y-la-codificacion-one-hot/.

[26] S. Alkan, «Advanced Feature Engineering,» [En línea]. Available: https://www.kaggle.com/code/seneralkan/advanced-feature-engineering.

[27] «Aprendizaje automatico y las Metricas de regresión,» [En línea]. Available: https://sitiobigdata.com/2018/08/27/machine-learning-metricas-regresion-mse/#.

[28] R. D.Hernandez, «El patrón modelo-vista-controlador: Arquitectura y frameworks explicados,» [En línea]. Available: https://www.freecodecamp.org/espanol/news/el-modelo-de-arquitectura-view-controller-pattern/.

[29] «scikit-learn,» [En línea]. Available: https://scikit-learn.org/stable/user_guide.html.

[30] «Folium,» [En línea]. Available: https://python-visualization.github.io/folium/.

[31] A. S. L. y. A. S. Jiménez, «Threading: programación con hilos (I),» [En línea]. Available: https://python-para-impacientes.blogspot.com/2016/12/threading-programacion-con-hilos-i.html#:~:text=En%20programaci%C3%B3n%2C%20la%20t%C3%A9cnica%20que,o%20no%20una%20misma%20tarea..

[32] «Python Module of the Week (PyMOTW-3),» [En línea]. Available: https://rico-schmidt.name/pymotw-3/concurrent.futures/.

[33] «OpenWebinars Blog,» [En línea]. Available: https://openwebinars.net/blog/que-es-flask/.

[34] A. d. Madrid, «Calendario. Datos Abiertos del Ayuntamiento de Madrid,» [En línea]. Available: https://datos.madrid.es/portal/site/egob/menuitem.c05c1f754a33a9fbe4b2e4b284f1a5a0/?vgnextoid=9f710c96da3f9510VgnVCM2000001f4a900aRCRD&vgnextchannel=374512b9ace9f310VgnVCM100000171f5a0aRCRD&vgnextfmt=default.

[35] A. d. Madrid, «Datos Abiertos del Ayuntamiento de Madrid,» [En línea]. Available: https://datos.madrid.es/portal/site/egob/menuitem.c05c1f754a33a9fbe4b2e4b284f1a5a0/?vgnextoid=695cd64d6f9b9610VgnVCM1000001d4a900aRCRD&vgnextchannel=374512b9ace9f310VgnVCM100000171f5a0aRCRD&vgnextfmt=default.

[36] A. E. d. M. (AEMET), «Centro de Descargas de Productos de la AEMET,» [En línea]. Available: https://opendata.aemet.es/centrodedescargas/productosAEMET.

[37] OpenWeather, «OpenWeatherMap API,» [En línea]. Available: https://openweathermap.org/api.

[38] WeatherSpark, «Clima promedio en Madrid, España durante todo el año,» [En línea]. Available:]https://es.weatherspark.com/y/36848/Clima-promedio-en-Madrid-Espa%C3%B1a-durante-todo-el-a%C3%B1o.

[39] U. (. d. A. d. F. Europeas), «Calendario del fútbol europeo 2023: fechas de partidos y sorteo,» [En línea]. Available: https://es.uefa.com/returntoplay/news/027c-16dee77c3246-6622bec638c5-1000--calendario-del-futbol-europeo-2023-fechas-de-partidos-y-sort/.

[40] Brita.mx, «Las seis técnicas principales utilizadas en la ingeniería de Machine Learning,» [En línea]. Available: https://brita.mx/las-seis-tecnicas-principales-utilizadas-en-la-ingenieria-de-machine-learning.

[41] «AprenderBigData.- Pipeline de datos,» [En línea]. Available: https://aprenderbigdata.com/pipeline-de-datos/.

[42] A. W. S. (AWS), «AWS - ¿Qué es la optimización de hiperparámetros?,» [En línea]. Available: https://aws.amazon.com/es/what-is/hyperparameter-tuning/.

[43] «CienciaDeDatos - Árboles de predicción: Bagging, Random Forest y Boosting,» [En línea]. Available: https://www.cienciadedatos.net/documentos/33_arboles_de_prediccion_bagging_random_forest_boosting.

[44] «AWS,» [En línea]. Available: https://calculator.aws/#/addService.

[45] «QuestionPro Blog,» [En línea]. Available: https://www.questionpro.com/blog/es/tipos-de-datos-estadisticos/.

[46] A. d. Madrid, «Datos Abiertos del Ayuntamiento de Madrid,» [En línea].