PREDICCIÓN DEL TRÁFICO EN MADRID BASADO EN TÉCNICAS DE APRENDIZAJE

MADRID TRAFFIC PREDICTION BASED ON MACHINE LEARNING TECHNIQUES

Miguel Ángel Portocarrero Sánchez
Ricardo Sebastián Suquillo Muzo

Grado en Ingeniería del Software
Universidad Complutense de Madrid

Trabajo Fin de Grado

6 de septiembre de 2022

Tutores

Mercedes García Merayo

Resumen en castellano

El desarrollo de Inteligencia Artificial y el aumento de la capacidad de las máquinas a la hora de realizar cálculos complejos han hecho posible elaborar métodos para conocer con antelación eventos o situaciones futuras basándose en datos anteriores. Dentro del amplio conjunto de campos que abarca de esta idea, este proyecto se centra en predecir los datos del flujo de tráfico aplicando distintos métodos.

Palabras clave

Inteligencia Artificial, Algoritmos de aprendizaje, Predicción

Abstract

The development of Artificial Intelligence and the increasing capacity of machines to perform complex calculations have made it possible to develop methods to anticipate future events or situations based on previous data. Within the broad set of fields covered by this idea, this project focuses on studying traffic flow data by applying different methods.

Keywords

Machine Learning, Traffic, Learning Algorithms

Índice general

Capítulo 1

Introducción

En este trabajo se aborda la posibilidad de predecir el tráfico de una zona localizada de Madrid. Dicha predicción se hizo utilizando métodos de aprendizaje automáticos los cuales fueron alimentados con los datos correspondientes disponibles en las distintas plataformas que nos ofrece la Comunidad de Madrid.

1.1. Motivación

El rápido desarrollo de la potencia de cálculo automático, tanto en el campo del software como del hardware permite abordar problemas que antes habrían sido muy difíciles de llevar a cabo. En los últimos años, se ha visto un notable desarrollo en los algoritmos de *Machine Learning*, el presente trabajo, pretende ser una oportunidad de introducirnos en este campo abordando un mismo problema con cuatro estrategias diferentes, haciendo además una comparación entre las bondades de cada uno de ellos

Estado actual de la inteligencia artificial

Desde la antigüedad el hombre ha tenido la idea de dotar a las máquinas con algún tipo de inteligencia similar a la humana. Sin embargo, durante muchos años no fue posible hacer realidad esta idea debido a que los avances tecnológicos no lo permitían. En estos últimos años, a raíz del desarrollo de máquinas con potentes capacidades de cálculo, esta antigua pretensión se está haciendo realidad a pasos agigantados. Se nota un notable despegue en el desarrollo de lo que hoy conocemos como inteligencia artificial.

El término de inteligencia artificial fue establecido alrededor de 1956 por John McCarthy [1], Marvin Minsky [6] y Claude Shannon [5]. En los años 70 se esperaba que la inteligencia artificial estuviera presente en la mayoría de hábitos de nuestras vidas, sin embargo, el tiempo que empleaban los algoritmos entonces así como la capacidad de cómputo y los datos hicieron que no fuera posible.

Es en la década de los 90 y debido a dos factores esenciales cuando se produce el desarrollo de la inteligencia artificial. El aumento de la capacidad computacional de las máquinas y el avance para poder procesar datos de forma digital fueron claves su progreso.

En la actualidad se ha producido un incremento notable de productos y prestación de servicios basados en ella. Si nos centramos en el campo de los sistemas predictivos de tráfico, ahora somos capaces de determinar qué va a suceder en las carreteras y calles con antelación. Del mismo modo ahora podemos conocer con precisión el tiempo de llegada de un punto a otro. Estas predicciones de tráfico y tiempo estimado son especialmente útiles a la hora de evitar un atasco, si se necesita notificar a amigos y familiares que vamos llegar tarde o si necesitamos salir a tiempo para asistir a una reunión importante, de la misma manera también son útiles para empresas de transporte que necesitan saber los horarios de recogida y entrega o empresas de viajes compartidos, en las que los precios dependen de la duración del viaje. En este contexto nace la idea de nuestro proyecto centrado en la predicción del tráfico de calles de Madrid de las zonas donde confluyen gran parte de los desplazamientos en coche.

1.2. Objetivos

En este proyecto se realizará el estudio de los datos del flujo de tráfico de cuatro calles de Madrid. Estas calles han sido elegidas como muestra representativa de la cantidad de tráfico que hay en la ciudad dependiendo de la zona en la que se ubique.

Dado que la cantidad de coches que circulan por una calle no depende solo de la hora y el día en el que se recogen los datos sino también de las condiciones climáticas del momento o el día en que se tomaron las muestras entre otros factores. Por ello realizaremos el estudio de los datos del flujo de tráfico junto con los datos meteorológicos y el calendario laboral de la ciudad.

Podemos decir que el objetivo principal del proyecto es realizar una predicción del tráfico. Para poder llevarlo a cabo utilizamos técnicas de aprendizaje automático junto con los datos

que hemos mencionado antes tratados debidamente. Uno de los factores más importantes que determinan la eficacia de las técnicas de aprendizaje es la calidad de datos que se le proporcionan ya que que son fundamentales en la fase de entrenamiento. Por esta razón nuestra primera preocupación fue procurarnos de datos adecuados para llevar a cabo el trabajo. Estos datos los obtendremos de las plataformas del Ayuntamiento de Madrid y la Agencia Estatal de Meteorología.

Los pasos que seguiremos se muestran a continuación:

- Obtención de datos: los datos fueron obtenidos de las plataformas del Ayuntamiento de Madrid, la Agencia Estatal de Meteorología.
- Variables de interés: Tenemos que decidir que variables son relevantes para este trabajo.
- Técnicas de aprendizaje: Tenemos que elegir técnicas adecuadas de *Machine learning* que nos permitan alcanzar nuestro objetivo.
- Evaluación: Tenemos que realizar una evaluación comparada de los resultados obtenidos de cada una de las técnicas.

1.3. Estructura de la memoria

En este apartado detallamos los pasos seguidos en el desarrollo del presente trabajo.

- *Formateo de los datos.* Se explica el proceso desde la obtención de los datos hasta que estuvieron preparados para incluirlos en las técnicas de aprendizaje.
- *Fundamentos de los algoritmos utilizados y métodos de evaluación.* Se describen las distintas técnicas de aprendizaje así como los diferentes métodos de evaluación para los resultados de estas técnicas.
- *Ejecución de los métodos de aprendizaje.* Se explica como se ha aplicado cada uno de los métodos
- *Resultados.* Se muestran y explican los resultados obtenidos de las distinta técnicas de aprendizaje y se expone una discusión sobre ellos.

- *Contribuciones individuales al proyecto.* Se exponen las contribuciones de cada uno de los integrantes.
- *Conclusiones y trabajos futuros.* Se exponen las conclusiones de este trabajo y las ideas para poder mejorarlo en un futuro.

Capítulo 1

Introduction

In this chapter we will briefly present what has motivated this project, the objectives and the structure of this paper.

1.1. Motivation

The rapid development of automatic computing power, both in the field of software and hardware, makes it possible to address problems that would have been very difficult to carry out in the past. In recent years, there has been a remarkable development in machine learning algorithms. The present work aims to be an opportunity to introduce ourselves in this field by addressing the same problem with four different strategies. Finally, there will be a discussion of the advantages and disadvantages.

The development of intelligence expressed by machines endowed with some human capability has been subject of study since ancient times. This is the reason behind what we now know as *artificial intelligent*. This term was established around 1956 by John McCarthy, Marvin Minsky and Claude Shannon. In the 1970s, it was expected that artificial intelligent would be present in most of the habits of our lives, however, the time the algorithms used then as the computing power and data made it unfeasible.

It is in the 1990s and due to two essential factors when the development of artificial intelligent occurs. The increase in the computational capacity of machines and the advance to be able to process data digitally were key to the progess of artificial intelligent. Currently, there has been a notable increase in products and services made with artificial intelligent. Focusing on the predictive traffic systems, we are now able to predict what will happen on

roads and streets in advance. Similarly, we can now accurately know the time of arrival from one point to another. These traffic and estimated predictions are especially useful when it comes to avoiding a traffic jam, if you need to notify friends and family that you are going to be late or if you need to leave in time to attend and important meeting, in the same way for transport companies that need to know pick-up and delivery times or ridesharing companies, where prices depend on the lenght of the journey. In this context, the idea of our project focused on the traffic of the streets of Madrid

1.2. Objectives

This project will study the traffic flow data of four streets in Madrid. These streets have been chosen as a representative sample of the amount of traffic in the city depending on the area in which they are located.

Given that the amount of cars circulating on a street does not only depend on the time and day when the data is collected, but also on the weather conditions at the time or on the day the samples were taken, among other factors, we will study the traffic flow data together with the weather data and the working calendar of the city.

We can say that the main objective of the project is to make a traffic prediction. In order to do so, we use machine learning techniques together with the aforementioned data, duly treated. One of the most important factors that determine the effectiveness of the learning techniques is the quality of the data provided, which is fundamental in the training phase. For this reason our first concern was to procure adequate data to carry out the work. These data were obtained from the platforms of the Ayuntamiento de Madrid and the Agencia Estatal de Meteorología de la comunidad de Madrid

The steps we will follow are:

- Data collection: the data were obtained from the platforms of the Ayuntamiento de Madrid, la Agencia Estatal de Meteorología and Comunidad de Madrid

- Variables of interest: We have to decide which variables are relevant for this work.

- Learning techniques: We have to choose appropriate machine learning techniques that allow us to achieve our goal.

- Evaluation: We have to make a comparative evaluation of the results obtained from each of the techniques.

1.3. Work Structure

In this section we detail the steps followed in the development of this work.

- *Formatting data*. The process is explained from the time the data were collected until they were ready to be included in the learning techniques.
- *Fundamentals of the algorithms used and methods of evaluation*. The different learning techniques are described as well as the different evaluation methods for the results of these techniques.
- *Implementation of learning methods*.It explains how the different learning methods have been implemented.
- *Results*. The results obtained from the different learning techniques are shown and explained.
- *Discussion on results*.Each of the results of the learning methods is explained.
- *Individual contributions to the project*. The contributions of each of the members are presented.
- *Conclusions and future work* The conclusions of this work are described, as well as ideas for future improvements.

Capítulo 2

Formateo de los datos

Los datos obtenidos no se presentan en la forma adecuada para llevar a cabo nuestro trabajo. Razón por la cual fue necesario realizar un proceso de reestructuración y adecuación de dichos datos.

A partir de los datos obtenidos fue necesario elegir aquellas variables que consideramos necesarias para llevar a cabo nuestro trabajo. Después de un análisis de nuestros objetivos llegamos a la conclusión de que las variables adecuadas son: *día, hora, temperatura, precipitación, día de la semana, laboral/festivo, valor del tráfico medio por hora.*

Los datos obtenidos son variados y con distintos formatos. Para un adecuado procesamiento de los mismos hemos decidido normalizarlos. Más adelante se explicara en detalle el proceso seguido.

2.1. Estudio del *dataset*

El primer paso ha consistido en la obtención de los datos necesarios para el desarrollo del trabajo de diferentes plataformas. Específicamente, datos abiertos del Ayuntamiento de Madrid [3] para el flujo de trafico de las calles, datos abiertos de la Agencia Estatal de Meteorología (AEMET) [4] para recuperar las condiciones meteorológicas, y el calendario laboral. El estudio se ha centrado en los años 2018, 2019, 2020 y hasta julio de 2021.

En el caso del flujo del trafico hemos recuperado el aforo del tráfico medio por día y hora en cada estación, desde 2018 hasta julio de 2021, centrándonos en cuatro estaciones de las disponibles, por considerarlas más relevantes y cubrir diferentes puntos de la ciudad: Paseo de Santa María de la Cabeza, Calle José Abascal, Paseo Infanta Isabel y Paseo de

Figura 2.1: Datos de tráfico ayuntamiento Madrid

Extremadura. Los datos vienen en formato CSV y divididos en ficheros independientes, nombrados por el mes y año, por lo que hemos tenido que agruparlos en un único fichero para su posterior tratamiento, tal y como se refleja en la Figura 2.1.

La información de los ficheros está distribuida en tres hojas: *Datos, Estaciones y Ubicaciones de las estaciones.*

La primera hoja, *Datos*, contiene columnas correspondientes a la fecha, número de la estación, sentido de circulación y horas del día. La fecha viene en formato DD/MM/AAAA e indica cuando fueron tomados los datos, el número de la estación es un identificador que se le asigna a cada calle y el sentido de circulación establece cada uno de los sentidos de la calle con los valores 1 y 2. En cuanto a las horas del día se encuentran divididas en 12 columnas.

La información de esta hoja se estructura de forma que los datos cada estación se distribuyen en cuatro filas. La primera fila corresponde al primer sentido con las 12 primeras horas día, la siguiente fila tiene el mismo sentido y las 12 horas siguientes. De la misma forma se especifica la información para el sentido contrario. Esto ha hecho que sea necesario modificar el documento para disponer de un día completo para cada sentido en una fila. De este modo hemos reducido las filas a la mitad.

La segunda hoja, *Estaciones*, contiene el número y nombre de las estaciones y su ubicación en coordenadas.

Por último, en la tercera hoja, *Ubicaciones estaciones*, se encuentra el nombre de la estación, número, coordenadas de cada una por cada sentido y la orientación.

En el caso de los datos meteorológicos hemos decidido centrarnos en la información de las estaciones meteorológicas de Madrid situadas en Retiro, Cuatro Vientos y Ciudad Universitaria.

Para poder acceder a los datos desde la página de AEMET hemos tenido que obtener una *API Key* proporcionando el correo electrónico, y así generar una clave válida durante cinco días. Con esta clave y dentro de la sección *Acceso general*, en el apartado *Valores Climatológicos* y seleccionando la provincia, la estación y el rango de fechas obtenemos una pagina con el estado de la petición con dos *links*: uno para los datos y otro con los metadatos. Con estos enlaces tenemos toda la información que debemos extraer en ficheros. A partir de ellos, hemos generado tres archivos de texto, uno para cada estación meteorológica, que contienen la información de los años a estudiar.

Los ficheros contienen la fecha de la toma de datos, el identificador de la estación, el nombre de la estación, la provincia, la altitud, temperatura media, precipitación, temperatura mínima, hora de la temperatura mínima, temperatura máxima, hora de la temperatura máxima, dirección del viento, velocidad media del viento, racha , hora de la racha, presión máxima, hora de la presión máxima, presión mínima y hora de la presión mínima.

Por último, el fichero que recoge la información del calendario laboral, cuenta con el día en formato, DD/MM/AAAA, el día de la semana, laborable/festivo que toma los valores "festivo", "laborable", "sábado" o "domingo", el tipo de festivo y la festividad.

2.2. Tratamiento de los datos

A continuación explicamos las consideraciones que se han tenido en cuenta para tratar los datos con el objetivo de poder utilizarnos como entrada de los algoritmos con los que vamos a trabajar.

Flujo de tráfico: Ya que cada fichero corresponde a un mes/año, se han unido todos los correspondientes a un mismo año en un solo documento. De este modo se han obtenido los ficheros: DatosTrafico2018.csv, DatosTrafico2019.csv, DatosTrafico2020.csv y DatosTrafico2021.csv. Estos ficheros se han filtrado para quedarnos solo con las estaciones con las que vamos a trabajar.

Dado que la información de una calle, en un sentido, para un día está dividida en dos filas, cada una de ellas correspondiente a 12 horas, dichas filas se han agrupado en una

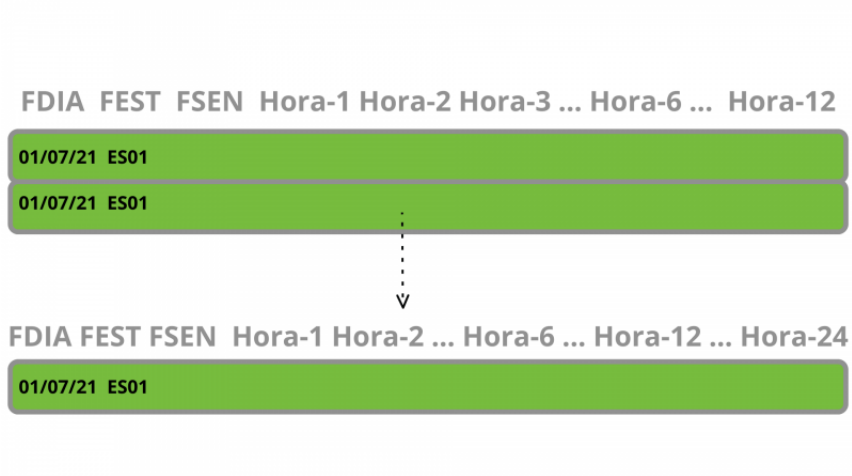

Figura 2.2: Cambio formato ficheros de tráfico

única fila para poder concatenar esta información con la incluida en los archivos de la AEMET y el calendario laboral como se refleja en la Figura 2.2.

Datos meteorológicos: Aunque inicialmente hemos considerado tres estaciones meteorológicas en Madrid, El Retiro, Cuatro Vientos y Ciudad Universitaria, al incorporar los datos a los de las estaciones de trafico se ha detectado una cantidad importante de datos nulos, por lo que hemos descartado la estación de Ciudad Universitaria.

De los datos que proporciona la AEMET hemos decidido trabajar con la fecha, la temperatura mínima, la temperatura máxima y la precipitación. Para cada estación hemos concatenado los documentos de la AEMET de forma que se pueda disponer de la información de cada estación con los datos meteorológicos de todos los años.

Calendario Laboral: De este fichero se necesita únicamente la fecha y el tipo de día. Se han eliminado el resto de columnas. Para el tipo de día se han fijado los valores en "laborable" y "festivo" eliminando los anteriores de "sábado" y "domingo".

Al realizar el procedimiento de unir los datos de tráfico con los meteorológicos hemos tenido que usar los datos de tráfico sin dividir por sentido.

Los archivos finales presentan las siguientes columnas: *fecha, estación,temperatura máxima, temperatura mínima, temperatura media, precipitación, dia de la semana* y *laborable festivo*.

2.2.1. Eliminación de datos

Una vez generados los archivos finales, observamos que el archivo que contenía la estación de José Abascal, disponía de un solo sentido, por lo que eliminamos las filas correspondientes al sentido 2 de ese fichero.

2.2.2. Imputación

Después de la eliminación de las filas correspondientes al archivo que contenía la estación de José Abascal, se encontraron valores NaN (not a number), por lo que decidimos comprobar si en los demás ficheros se hallaban esos valores. Una vez hecha la comprobación empleando el método *isnull().values.any()* y ver que en todos los archivos ocurría lo mismo, decidimos reemplazar esos valores por el valor cero haciendo uso del método *fillna()*. Ambos métodos son proporcionados por la librería Pandas. Una vez reemplazados todos esos valores, se puede empezar a realizar el entrenamiento de los datos correctamente.

2.2.3. Categorización de datos

El desarrollo de nuestro trabajo, implica la manipulación de datos numéricos, por esta razón, en primer lugar nos centramos en la categorización de las variables no numéricas de nuestros archivos.

Para poder realizar dicha categorización recurrimos a una librería que proporciona Python, Sklearn. De esta librería utilizamos dos clases: *preprocessing* y *LabelEncoder*. Comenzamos aplicando la función *LabelEncoder()* a las variables no numéricas *fecha, estación, día de la semana, laborable-festivo y sentido*, para poder convertir los valores asociados a cada una de ellas a valores numéricos. De este modo obtuvimos nuevas variables numéricas con los valores correspondientes. Una vez hecha la categorización de las variables, nos dimos cuenta de que en las filas de la columna *precipitación* se encontraba el valor *ip*, correspondiente a *inapreciable*, por lo que decidimos reemplazarlo por el valor 0 para no tener problemas posteriormente.

2.2.4. Normalización de datos

Una vez categorizados los datos, tuvimos que normalizarlos dado que teníamos que manipular datos con distintas unidades.

El método que elegimos para la realización del proceso de normalización fue el que proporciona la librería de Python: *fit_ transform()*. Este método calcula la media y la varianza de cada una de las variables para escalar los datos de prueba.

2.3. Estrategia de trabajo

Como ya hemos mencionado, nuestro trabajo pretende evaluar las bondades de cuatro métodos de predicción. Alcanzar estos objetivos requiere disponer de dos conjuntos de datos, a saber , uno que sirva como datos de entrenamiento y otro que sirva como referencia (datos reales) para cuantificar la eficacia de los métodos utilizados.

Estos dos conjuntos fueron obtenidos a partir de los datos originales utilizando el método *train_ test_ split()*. Este método requiere definir el porcentaje de datos que formarán cada uno de los conjuntos. Hemos considerado adecuado dividir en grupos del 50 %.

Los conjuntos generados serán referidos como dataset1 y dataset2. En total, tuvimos que generar siete parejas de conjuntos, cada pareja se diferencia por el sentido del tráfico. Las parejas se distinguen unas de otras por las variables “Estación” y “Calle”.

Capítulo 3

Fundamentos de los algoritmos de clasificación y métodos de evaluación

Una rama dentro de *Machine Learning* es el aprendizaje supervisado, se trata de una técnica que se utiliza para poder predecir resultados futuros partiendo de un conjunto de datos conocidos, llamados datos de entrenamiento. Los datos de entrenamiento constan de una entrada y el resultado.

A continuación presentamos los algoritmos de aprendizaje que se aplican en este trabajo, así como los métodos de evaluación que se utilizan para cuantificar y comparar la eficacia en la predicción de los métodos empleados.

3.1. Algoritmos de aprendizaje

Los algoritmos de aprendizaje supervisado crean una función a partir de los datos de entrenamiento, función que se utiliza para hacer una predicción a partir de una nueva entrada válida.

Para obtener esta función los algoritmos tienen que *generalizar* la información que se le suministra, es decir, intentan buscar relaciones entre los datos de entrenamiento para luego poder detectarlas en las nuevas entradas.

Dependiendo del tipo de salida diferenciamos entre modelos de clasificación y modelos de regresión.Si la salida es un valor categórico se utilizará un método de clasificación y si por el contrario se trata de un valor numérico entonces estamos ante un modelo de regresión.

La clasificación es una subcategoría del aprendizaje automático. En la clasificación se

entrena al algoritmo para identificar a que categoría pertenece un nuevo dato dentro de un conjunto de categorías definido. Podemos distinguir dos tipos de clasificaciones que son las más utilizadas: clasificación binaria y la clasificación multi-clase.

En la clasificación binaria únicamente se puede asignar al nuevo dato dos posibles categorías, como podemos observar en el ejemplo ilustrado de la Figura 3.1. Por el contrario, en la multi-clase, hay diversas categorías a las que puede pertenecer.

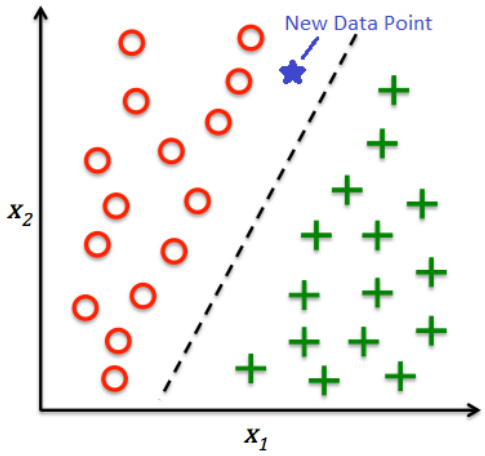

Figura 3.1: Ejemplo de clasificación binaria. [2].

3.1.1. Algoritmo *k-nearest neighbors*

El algoritmo *k-nearest neighbors* (KNN, en español, k vecinos más cercanos) es un tipo de algoritmo de aprendizaje automático supervisado que se puede utilizar tanto para problemas predictivos de clasificación como de regresión. Sin embargo, se utiliza principalmente para la clasificación de problemas predictivos que tienen como salida un valor discreto.

El algoritmo asume que hay objetos similares cerca de la nueva entrada de datos y de esta suposición depende la precisión de sus predicciones. KNN es un algoritmo simple que almacena todos los casos disponibles y clasifica los nuevos datos o casos en función de una medida de similitud, que puede ser la distancia entre puntos.

Es un algoritmo de aprendizaje perezoso y no paramétrico. Es perezoso porque no hace ningún entrenamiento. Tan solo almacena el conjunto completo de datos. No paramétrico significa que no hay suposiciones para la distribución de datos subyacente, es decir, la estructura del modelo se determina a partir del conjunto de datos.

En la clasificación KNN, la salida es una pertenencia a una clase. Un objeto se asigna a la clase más común entre sus k vecinos más cercanos, siendo k entero positivo, como podemos ver en el ejemplo de la Figura 3.2. En el caso concreto de k = 1 el objeto simplemente se asigna a la clase de ese vecino más cercano, por ello es fundamental elegir el valor correcto de k. A medida que disminuimos el valor de k hacia 1, nuestras predicciones se vuelven menos estables y podemos obtener resultados incorrectos. De la misma manera al aumentar k realizamos predicciones más precisas, pero aumenta el número de errores. Generalmente se toma un valor impar de k, para aquellos casos en los que haya un empate a la hora de clasificar un dato nuevo. Para elegir el valor óptimo de k, se ejecuta varias veces el algoritmo con diferentes valores y optamos por el que tenga menor error.

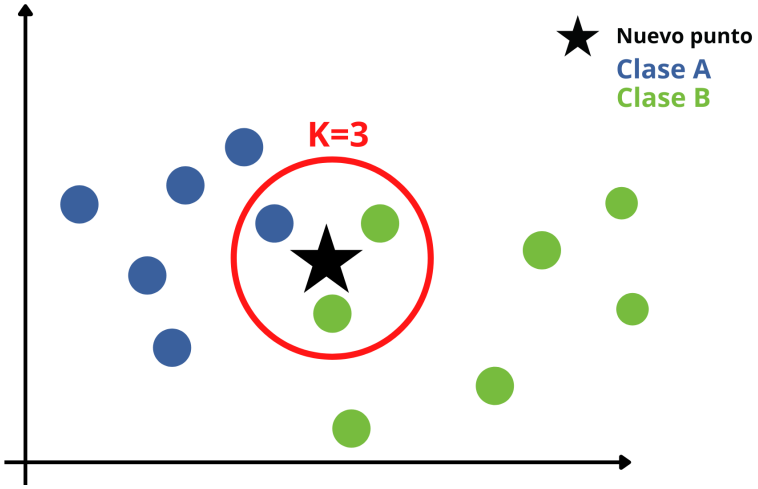

Figura 3.2: Ejemplo de clasificación del algoritmo KNN con k=3.

Las ventajas que más destacan son:

- Es simple de entender e interpretar
- Es útil para los datos no lineales al no haber suposiciones sobre los datos en este algoritmo
- Se puede usar tanto para la clasificación como para la regresión

Y sus desventajas más destacadas son:

- Es computacionalmente un algoritmo costoso ya que almacena todos los datos de entrenamiento.

- Se requiere un alto almacenamiento de memoria en comparación con otros algoritmos de aprendizaje supervisado.

- Es muy sensible a la escala de los datos, así como las características irrelevantes.

3.1.2. Algoritmo *Random Forest*

Random Forest es un tipo de algoritmo de aprendizaje supervisado conjunto, es decir, formado por una combinación de árboles predictores, se utiliza para resolver problemas de regresión y clasificación. Dicho de otro modo, el método de aprendizaje conjunto es una técnica que combina predicciones de varios algoritmos de aprendizaje automático para hacer una predicción más precisa que un solo modelo.

El *Random Forest* utiliza la idea de los árboles de decisión para construir sistemas de clasificación o regresión. Esta idea consiste en dividir el conjunto de datos en subconjuntos cada vez más pequeños.

El algoritmo crea cada árbol a partir de una muestra diferente de datos de entrada. En cada nodo, se selecciona una muestra diferente de características para dividir. Las predicciones de cada uno de los árboles se promedian para producir un único resultado, que es la predicción del *Random Forest* como refleja la figura 3.3. Construir de esta forma el modelo hace que los árboles compensen entre sí sus errores individuales.

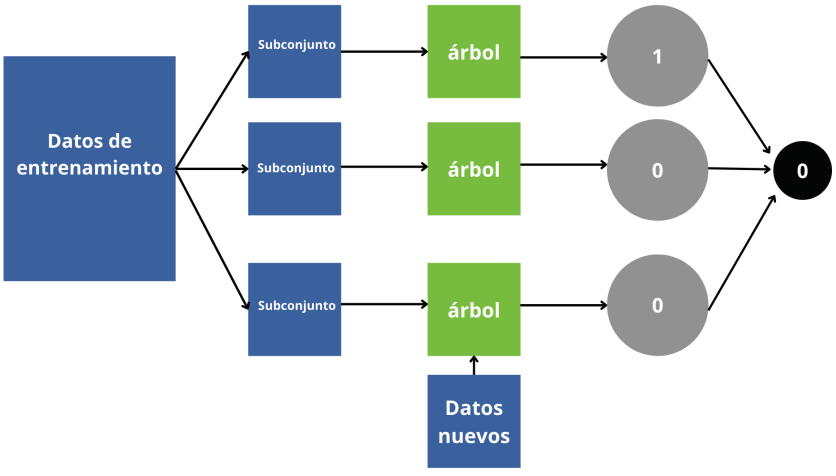

Figura 3.3: Estructura del algoritmo Random Forest.

Las ventajas más importantes de este algoritmo son:

- Se ejecuta de manera eficiente en grandes conjuntos de datos.
- Tiene una mayor precisión en comparación a otros algoritmos.
- Tiene un método eficaz para estimar los datos que faltan y mantiene la precisión cuando falta una gran proporción de los datos.

Una de sus desventajas es que puede dar lugar a un ajuste excesivo para algunos conjuntos de datos.

3.1.3. Algoritmo *Linear regression*

La regresión lineal es una forma de calcular la relación entre dos variables. Asume que existe una correlación directa entre las dos variables y que esta relación se puede representar con una línea recta.

De estas dos variables a una se le denomina *variable independiente* y a la otra *variable dependiente*. La *variable independiente* se llama así porque el modelo asume que puede comportarse como quiera y no depende de la otra variable por ningún motivo. La *variable dependiente* es lo contrario; el modelo asume que es un resultado directo de la *variable independiente*, su valor depende en gran medida de esta última.

La regresión lineal proporciona una relación matemática entre estas dos variables. Permite calcular la predicción de la *variable dependiente* si se conoce la *variable independiente*.

Dado que es una forma de regresión tan simple, la ecuación para la regresión lineal también es bastante simple. Se define de la siguiente forma:

$$y = B * x + A \tag{3.1}$$

donde y es la *variable dependiente*, x la *variable independiente* y A y B son coeficientes que determinan la pendiente y la intersección de la ecuación. Estos coeficientes se calculan con el criterio de mínimos cuadrados, es decir, un criterio que minimiza el error entre las predicciones de los modelos y los datos reales. Existe un parámetro matemático denominado coeficiente de correlación lineal que cuantifica la bondad del ajuste siendo un buen ajuste cuando dicho parámetro se acerca a ± 1.

En conclusión, la regresión lineal es una herramienta sencilla para estudiar las relaciones matemáticas entre dos variables diferentes cuando hay razones suficientes que indican que dicha relación es lineal

3.1.4. Redes Neuronales LSTM

La manera de operar de las LSTM (*Long Short Term Memory*, en español, memoria a corto plazo) es bastante complejo y sobrepasa de largo los objetivos de este trabajo, sin embargo, en lo que sigue describiremos cualitativamente su modo de actuar.

Una red neuronal recurrente es un tipo de red artificial que utiliza datos secuenciales o datos temporales. Se distinguen por su memoria ya que toman información de entradas anteriores para influir tanto en la entrada como en la salida posterior.

Las LSTM son un tipo especial de redes recurrentes, es decir, de aquellas redes en las que la información puede persistir introduciendo bucles en el diagrama de red, esto significa que de alguna manera pueden "recordar" estados previos y utilizar esta información para predecir cual puede ser el estado siguiente. Esta característica las presenta como muy adecuadas en el manejo de series cronológicas. El siguiente gráfico nos muestra la idea que subyace en el proceder del método.

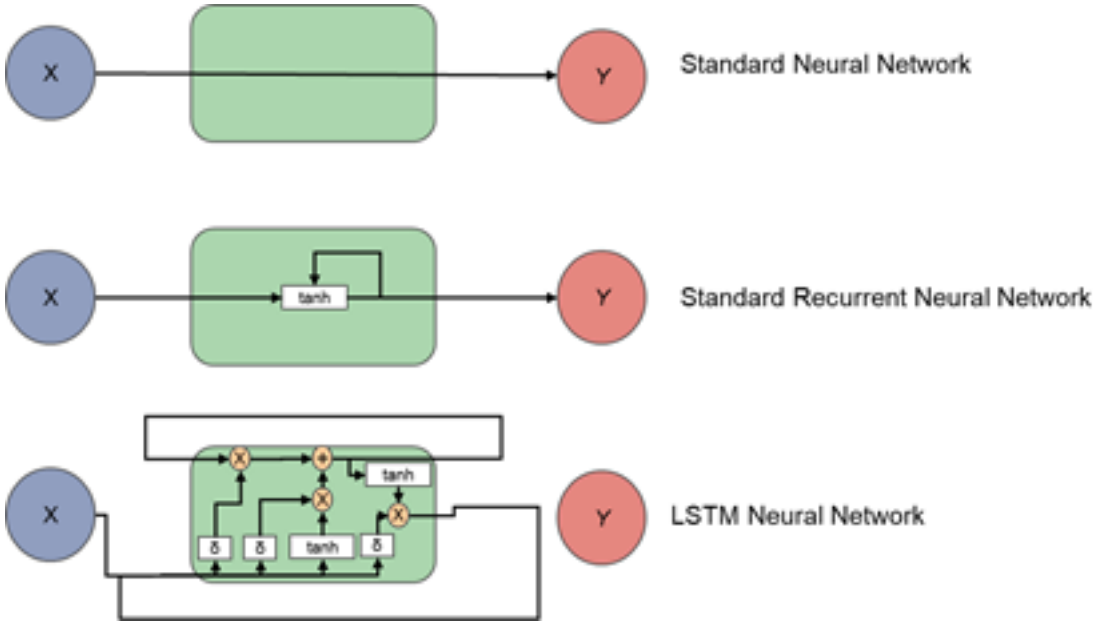

Figura 3.4: Esquema de tres tipos de redes neuronales

En la figura 5.8 se muestra esquemáticamente el grado de complejidad de tres distintas redes neuronales. Se nota claramente que el LSTM es una técnica más elaborada, razón por la cual sus resultados son mejores. El LSTM como ya hemos dicho, a parte de tener una propiedad recurrente tiene la cualidad de "recordar" datos de interés que mejoran la predicción.

En resumen podemos decir que las redes neuronales LSTM es una variedad de las RNN (*Recurrent Neural Network*) que son capaces de aprender dependencias a largo plazo, especialmente en problemas de predicción de secuencias. Las redes neuronales LSTM tienen conexiones de retroalimentación, es decir, es capaz de procesar la secuencia completa de datos como voz o vídeo, además de puntos de datos individuales como imágenes. Las redes recurrentes LSTM son un tipo especial de RNN, que muestra un rendimiento sobresaliente en una gran variedad de problemas.

3.2. Métodos de evaluación

Aplicados los métodos de aprendizaje nos queda evaluar los resultados para poder determinar cual de los métodos empleados es el más eficaz en la predicción. Para ello se ha decidido utilizar como instrumento el *Mean Squared Error*, *R-Squared* y el *Error Porcentual Medio*.

3.2.1. *Mean Squared Error*

Mean Squared Error (MSE, en español, error cuadrático medio) se define como el cociente entre la suma de las diferencias al cuadrado entre los valores predichos y los valores reales y el numero de datos considerados, es decir, la media aritmética de las diferencias elevadas al cuadrado.

$$MSE = \frac{\sum_{i=1}^{N}(y_i - \bar{y}_i)^2}{N} \tag{3.2}$$

Donde N es el número de valores observados, y_i el valor objetivo e $\bar{y}_i$ es el valor objetivo predicho.

Una desventaja del MSE es que, si se aplica a un conjunto de datos con unos pocos valores atípicos, puede penalizar fuertemente el modelo.

El MSE es una de las herramientas más populares para medir la precisión de los modelos en estadística y en aprendizaje automático.

3.2.2. *R-Squared*

La varianza es un término estadístico que determina la dispersión de nuestros datos y nos indica cuántos valores atípicos hay en ellos. Una vez definida la varianza, se puede entender mejor esta métrica de evaluación.

R-Squared Error (R^2) es el porcentaje de la variación de la variable de respuesta que se explica mediante un modelo lineal: variación explicada por el modelo/variación total.

R^2 está siempre entre el 0 % y el 100 %: El 0 % indica que el modelo no explica nada de la variabilidad de los datos de respuesta en torno a su media. El 100 % indica que el modelo explica toda la variabilidad de los datos de respuesta en torno a su media. En conclusión, cuanto mayor sea R^2, mejor se ajustará el modelo a los datos.

3.2.3. *Error Porcentual Medio*

Con el fin de disponer de un parámetro más intuitivo que muestre la bondad de la predicción hemos considerado interesante calcular un error al cual hemos denominado, Error porcentual medio (EPM). Éste se define por la siguiente fórmula:

$$EPM = \sum_{i=1}^{i=N} \frac{(V_{predicho} - V_{real}) * 100}{V_{real}} \tag{3.3}$$

Capítulo 4

Ejecución de los métodos de aprendizaje

En este capítulo pondremos en funcionamiento los distintos métodos de aprendizaje que este trabajo utiliza. Para ello haremos uso de cada uno de los datasets generados.

Como ya dijimos los métodos utilizados son: KNN, Linear Regression, Random forest Regressor y Redes neuronales LSTM.

El proceso seguido tiene como esquema general el siguiente:

- Se proporciona los datos de entrenamiento a los programas. Utilizamos el método *fit()* para ajustar el modelo correspondiente a los datos.
- Utilizamos el método *predict()* con cada modelo obtenido con el fin de predecir los valores.
- En base a los datos predichos y los datos reales (datos no utilizados en el "entrenamiento") calculamos parámetros que nos permiten determinar la eficacia de cada método. Estos son el *mean_squared_error()* y *r2_score()*

4.1. KNN

Dicho a grandes rasgos, este método necesita estimar una distancia entre dos eventos. Entendiendo por evento un punto multidimensional en el que cada coordenada representa una variable de las que hemos elegido para el presente estudio. Por ejemplo:

$$(Temperatura, Precipitacion, Trafico, ...) = (x_1, x_2, x_3, ...)$$

Para calcular la distancia entre dos eventos considerando que tienen distintas unidades los valores correspondientes han sido categorizados y normalizados previamente. Esto quiere decir que la distancia se define como:

$$d = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + ... + (x_n - y_n)^2} \tag{4.1}$$

Basado en el cálculo de estas distancias el método selecciona el número de vecinos más cercanos para realizar su predicción. El número de vecinos más cercanos es un parámetro que nosotros tenemos que introducir y se le denomina *n_ neighbors*. El presente trabajo ha explorado la predicción para distintos valores de *n_ neighbors*, específicamente los valores utilizados van del 1 al 20. Lo descrito anteriormente ha sido implementado para el método *KNeighborRegressor* facilitado por el módulo *neighbors* de la librería *sklearn*.

4.2. Random Forest

En el apartado 3.1.2 hemos explicado de modo básico en que consiste este método, sin embargo, podemos añadir que este método utiliza la clase *RandomForestRegressor* que nos proporciona el módulo *ensemble* de la librería *sklearn*.

El parámetro de entrada más importante de esta clase es el parámetro llamado *n_ estimators*, que es un número positivo el cual se puede definir como el número de *árboles* que se incluyen en el modelo. Con el objetivo de saber cuál es el mejor valor del parámetro *n_ estimators*, emplearemos un rango de valores que se corresponde entre 1 y 40.

4.3. Linear Regression

En el apartado 3.1.3 ya hicimos un comentario acerca de este método. En lo que sigue podemos decir que el algoritmo utilizado es específicamente, el *linear_ model* de la librería *sklearn*.

La clase utilizada para este algoritmo es *LinearRegression*. Con esta herramienta y con parte de los datos de "entrenamiento" se determinan los parámetros de la recta de regresión, recta que nos permite determinar los valores predichos.

4.4. Redes neuronales *LSTM*

En el apartado 3.1.4 se ha explicado en la medida que nos fue posible el modo de proceder de la red neuronal LSTM. Tal y como se ha indicado se ha proporcionado a esta red los datos que disponíamos acerca del tema en estudio. Nos hemos asegurado también en la medida de lo posible de no cometer errores en la implementación. Decimos en la medida de lo posible, dado que la profundidad del tema no es del todo accesible a nuestro actual de conocimiento.

Capítulo 5

Resultados

En este capítulo presentaremos los resultados obtenidos con los cuatro métodos de predicción que en este trabajo se han utilizado.

5.1. KNN

Las gráficas que se muestra a continuación presentan en el eje horizontal los distintos valores de K vecinos más cercanos, en tanto que en el eje vertical para la primera gráfica muestra el *mean squared error* y para la segunda el error porcentual medio.

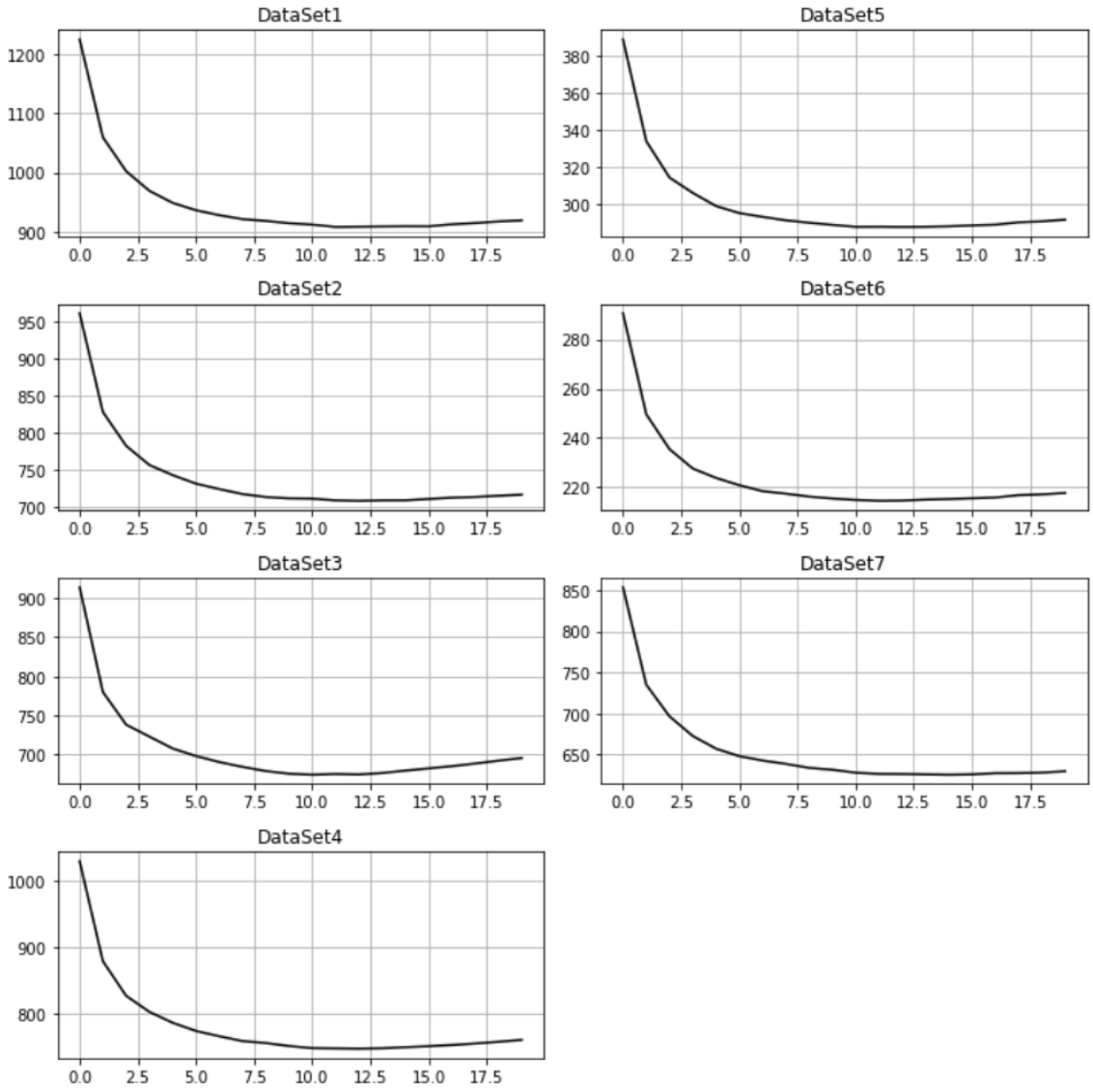

Figura 5.1: Gráficos MSE

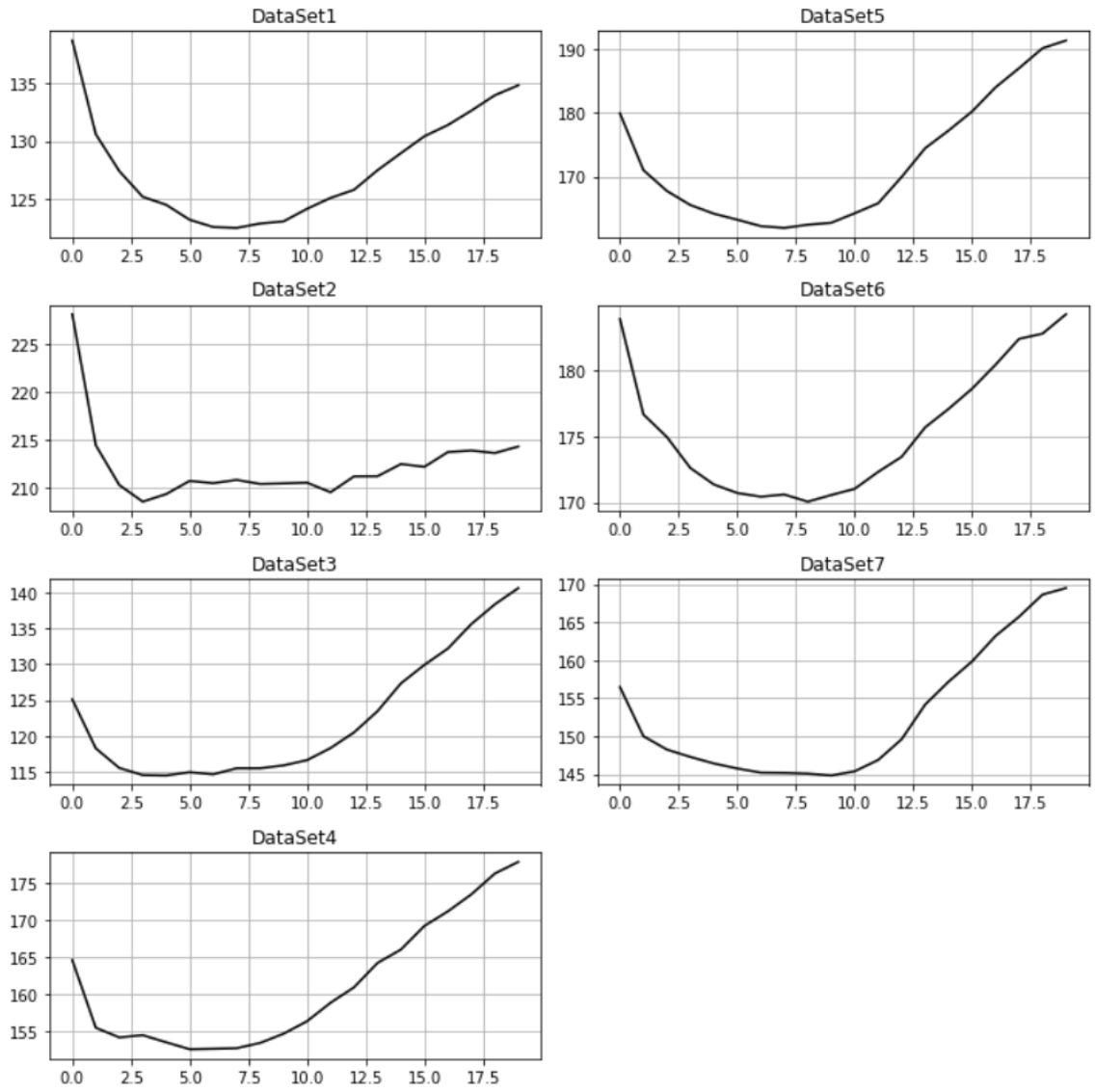

Figura 5.2: Gráficos EPM

Dijimos con anterioridad que otro de los métodos que hemos utilizado para caracterizar la bondad de la predicción es el R^2. Esto presupone, como ya comentamos, la existencia de una relación lineal entre las variables (valor predicho, valor real). Sin embargo, una representación gráfica entre el valor predicho y el valor real nos muestra claramente que dicha relación lineal no existe. Por tanto, asignar un valor de R^2 a cada una de las pruebas hechas (para cada K) carece de sentido. Por esta razón no presentamos los resultados de este método.

5.2. Random Forest Regressor

Los resultados obtenidos con éste método son mostrados a continuación gráficamente:

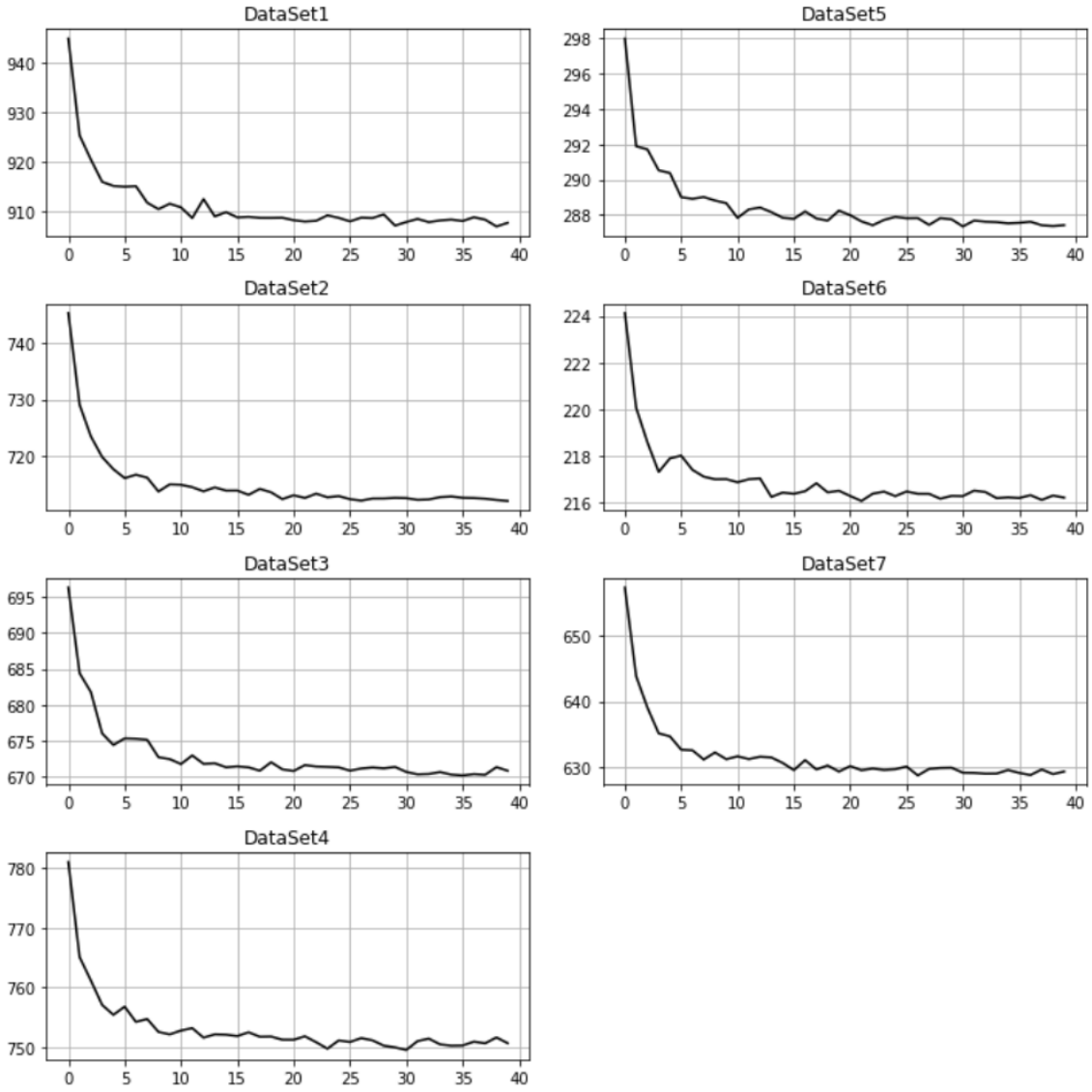

Figura 5.3: Gráfico MSE

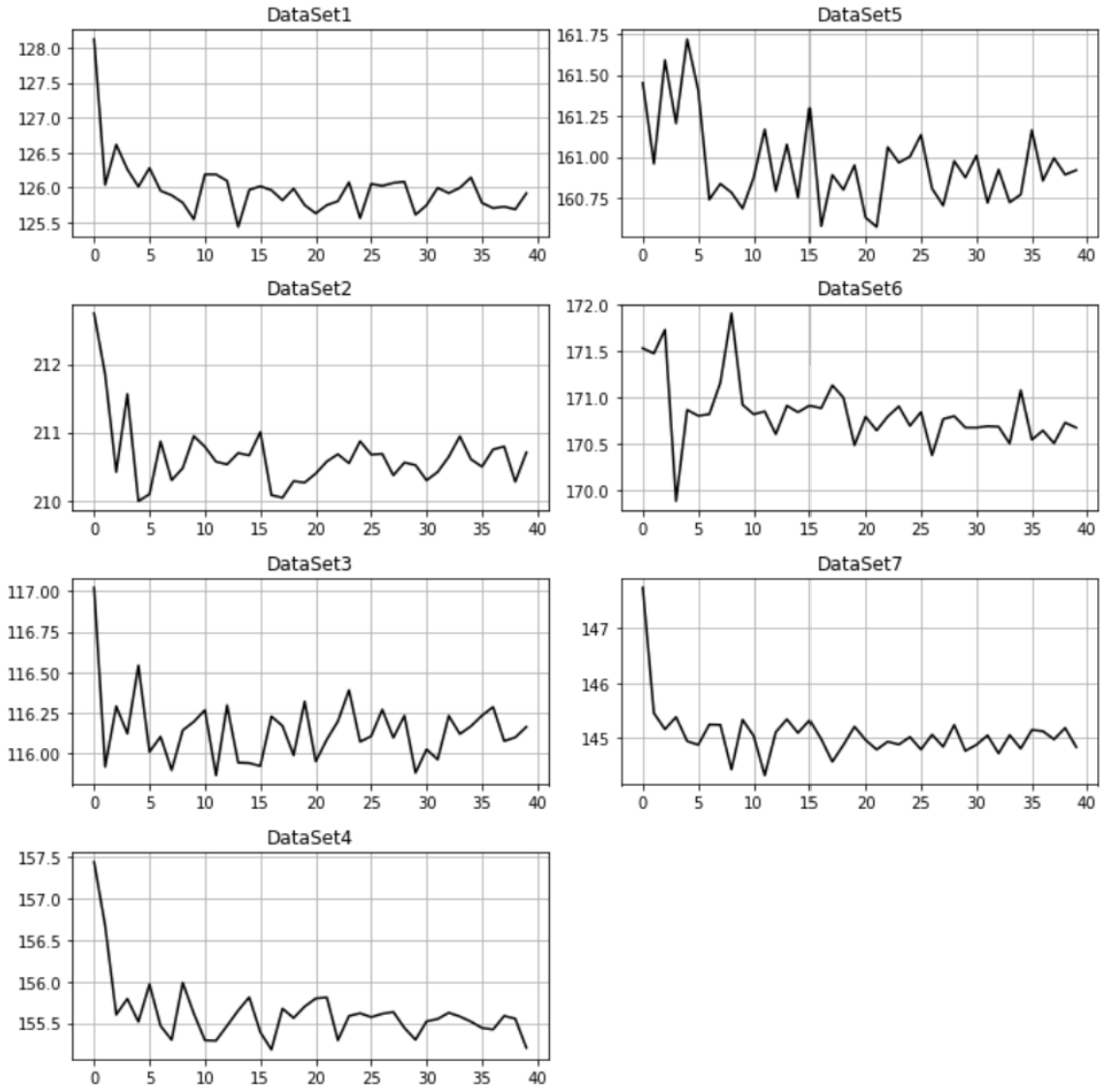

Figura 5.4: Gráfico EPM

5.3. Linear Regressor

Como ya comentamos en el apartado 5.1 los resultados obtenidos con este método se muestran en la tabla 5.7 indicando que no es un buen método de predicción.

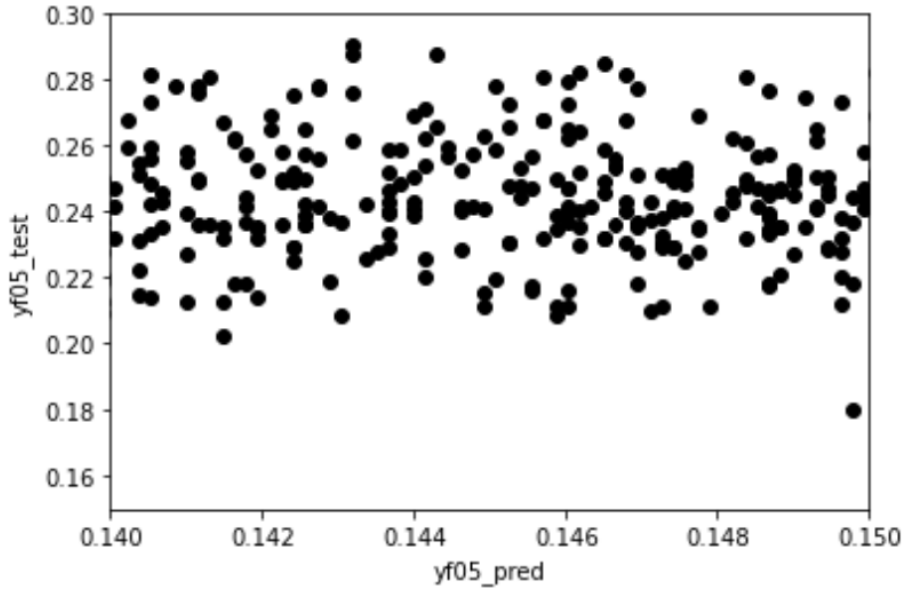

Figura 5.6: Rango reducido-detallado

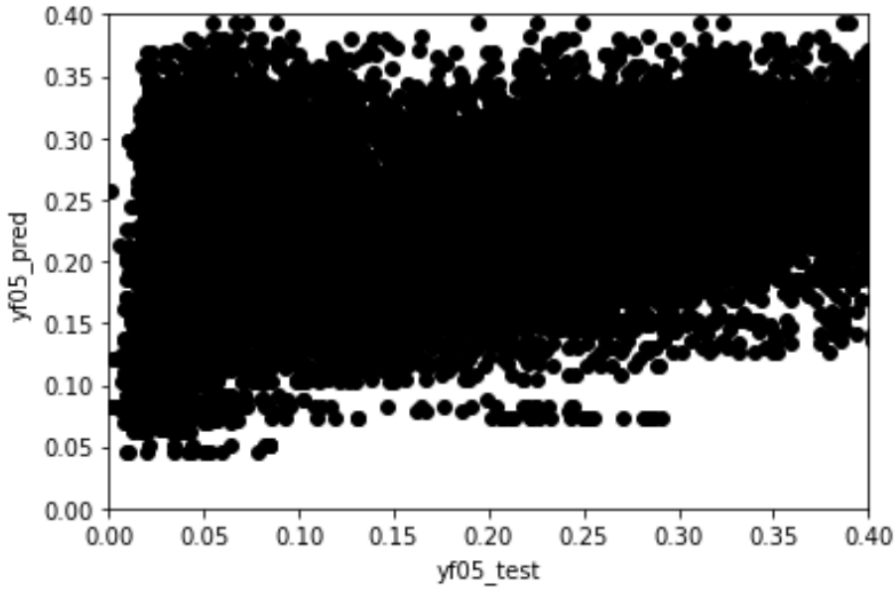

Figura 5.5: Rango Completo

DATASETS	Score
Dataset01	0.00619
Dataset02	0.01112
Dataset03	0.00954
Dataset04	0.00729
Dataset05	0.01052
Dataset06	0.00785
Dataset07	0.00829

Figura 5.7: Tabla Resultados Score

5.4. Redes Neuronales LSTM

A continuación se muestran gráficamente los resultados obtenidos de cada uno de los datasets con la aplicación de éste método. El eje y expresa la pérdida producida en la predicción en relación a los datos que se quieren predecir. Específicamente se trata del error medio absoluto en función de las (épocas) que se han explorado.

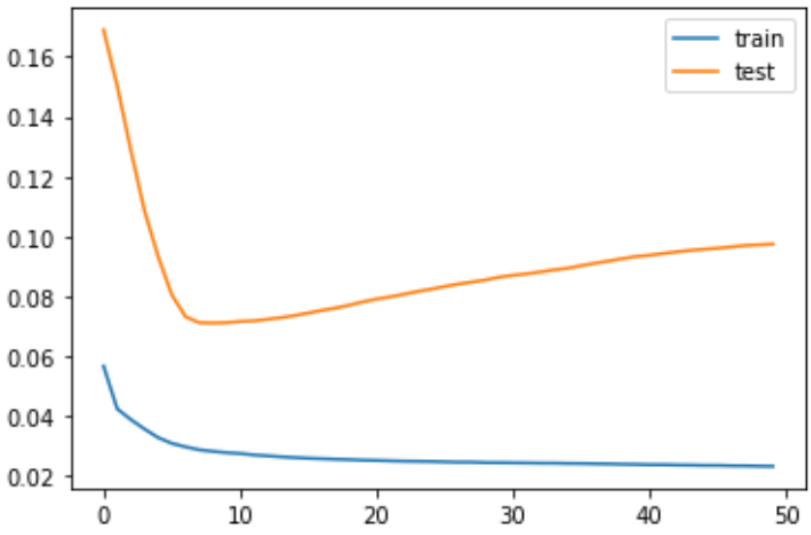

Figura 5.8: Grafico LSTM Dataset01

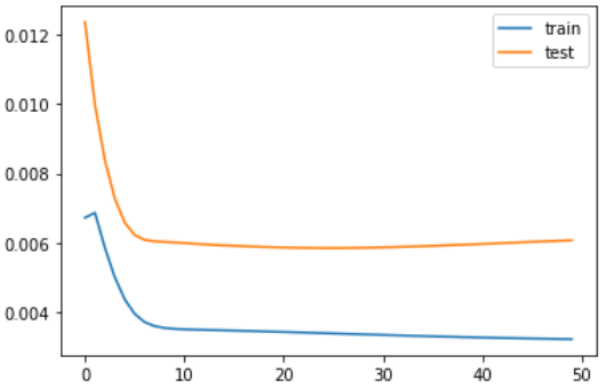

Figura 5.9: Grafico LSTM Dataset02

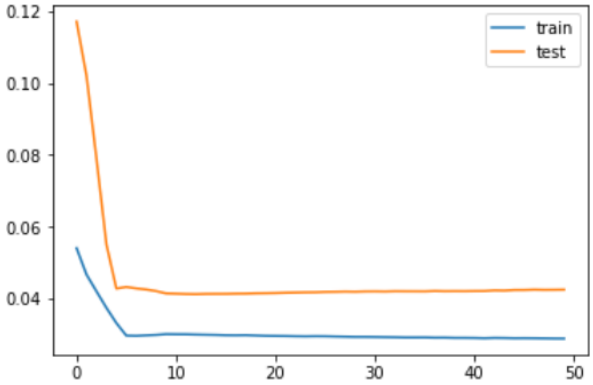

Figura 5.10: Grafico LSTM Dataset03

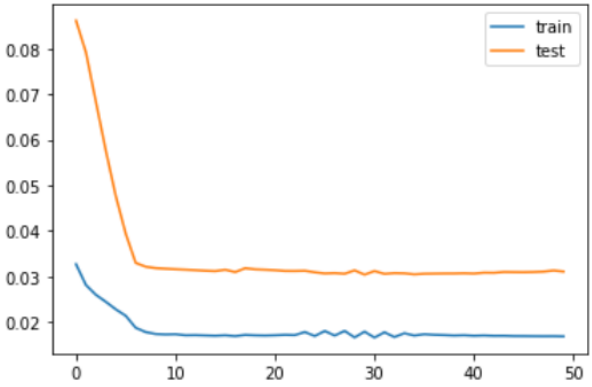

Figura 5.11: Grafico LSTM Dataset04

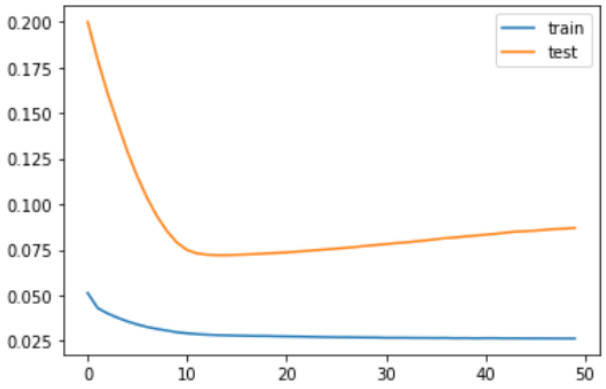

Figura 5.12: Grafico LSTM Dataset05

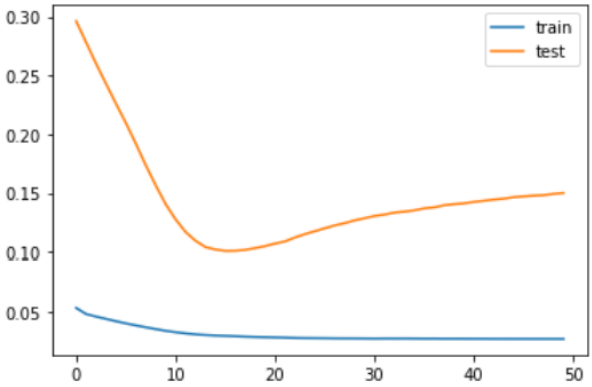

Figura 5.13: Grafico LSTM Dataset06

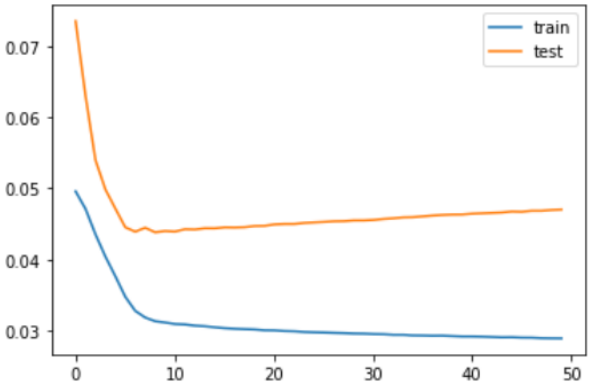

Figura 5.14: Grafico LSTM Dataset07

5.5. Discusión sobre los resultados

En relación al primer método evaluado cuyos resultados se muestran en el apartado 5.1 podemos observar que existe un valor óptimo para el cual es mínimo el error cometido. En nuestro caso el valor de los k-vecinos más cercanos óptimo se sitúa alrededor de 8. Por otro lado, si nos fijamos en los valores que cuantifican el error comprobamos que éste es alto, superando el 100 %.

Podemos concluir para este método que si es adecuado como instrumento para la predicción del tráfico dado que existe un valor óptimo. Consideramos que los errores pueden disminuirse significativamente si aplicamos el método a la predicción del tráfico de calles donde éste no fluctúe demasiado. En caso contrario, es decir, donde el tráfico fluctúa mucho, quizás sea recomendable ampliar el intervalo de tiempo considerado en la predicción. Por ejemplo cada 3 horas, 4 horas, etc.

En relación al método *Random Forest Regressor* se observa que es posible minimizar el error en la predicción determinando un número óptimo de árboles. En nuestro caso hemos podido encontrar que el error se minimiza a partir de la consideración de 10 árboles en adelante.

Los errores obtenidos son similares a los obtenidos con el método KNN, es decir, son altos. Pensamos que las mismas consideraciones indicadas en la discusión del método anterior son pertinentes también en este caso.

Los resultados obtenidos con el método *Linear Regressor* nos muestran que los datos de tráfico no tienen una dependencia lineal, de allí que los valores de R^2 obtenidos son cercanos a 0 indicando que el método es inadecuado para este tipo de datos.

Tal y como se muestra en la figura 5.6 quizás si dividimos la predicción para cortos periodos de tiempo podría mejorar en algo. Sin embargo el gráfico que muestra un detalle de los datos nos dice que no es recomendable este método.

Si nos fijamos en los resultados obtenidos con las redes neuronales LSTM, podemos observar que existe un valor dado de "épocas" para los cuales el error de la predicción es mínimo. Esto nos indica que el método tiene posibilidades de ser útil. Las consideraciones que hemos discutido en el primer apartado siguen siendo, a nuestro juicio, aplicables también a este método.

Capítulo 6

Contribuciones individuales al proyecto

6.1. Contribuciones de Miguel Portocarrero Sánchez

Uno de los puntos más importantes de este proyecto es el preprocesamiento de los datos, en el cuál invertimos la gran cantidad de tiempo. Para poder realizarlo correctamente, nuestro tutor nos guió sobre los campos más importantes. Como se explica en 2.1, todos los datos se encuentran en ficheros independientes divididos por meses, por lo que me encargué primeramente de unir todos los ficheros del calendario laboral desde enero de 2018 hasta julio de 2021 en cuatro ficheros que se corresponden con el año. Posteriormente, filtré cada uno de esos ficheros por la estación correspondiente. Al filtrar cada uno de los ficheros, me di cuenta de que estaban divididos por sentidos, por lo que decidí dividir cada fichero en dos, uno por cada sentido, obteniendo así ocho ficheros. Como una estación sólo era de un sentido, eliminé el fichero quedando únicamente con siete. Cada uno de estos ficheros se estructuraba en las siguientes columnas:

- Fecha y hora del día correspondiente
- El valor del tráfico medio por ese día
- El sentido
- La estación
- El día de la semana
- Laborable/festivo

- Tipo de festividad
- Nombre de Estación
- Temperatura media
- Precipitación
- Temperatura Mínima
- Temperatura Máxima

Como las columnas del día de la semana, tipo de festividad y el nombre de la estación eran irrelevantes, pude eliminarlas sin que tuviera algún efecto en el conjunto de datos. A continuación me encargué de las tareas de procesamiento de los datos, es decir, como se tratan los datos categóricos. También llevé a cabo el proceso de la eliminación, imputación, categorización y normalización de los datos reflejado en 2.2.1 y 2.2.2, 2.2.3 y 2.2.4.

Una vez hecho todo el procesamiento de los datos tuvimos una reunión con nuestro tutor, en la que nos recomendaba qué métodos de aprendizaje utilizar para aplicarlos a nuestros datasets.

Mi tarea fue investigar el funcionamiento de cada uno de los métodos ya implementados por la librería *Sklearn* y aplicarlos a nuestros datasets. El fundamento teórico de estos métodos se pueden encontrar en la sección 3.

Después de aplicar los distintos métodos de aprendizaje, me encargué de evaluar los resultados de cada uno de ellos. Para hacer esta evaluación, investigué sobre las distintas métricas de evaluación que se explican en la sección 3.2 y seleccioné las más comunes para este tipo de problemas. También me encargué de escribir los resultados y realizar los gráficos que éstos representan y así llevar a cabo una comparación sobre cuál es el mejor modelo para la predicción. Estos resultados se pueden observar en la sección 5. Además traduje el capítulo 1.

6.2. Contribuciones de Ricardo Suquillo Muzo

El procesamiento de datos es un apartado fundamental para el desarrollo del proyecto por ello como primer paso el objetivo era recopilar los datos necesarios para más adelante poder procesarlos.

En primer lugar tuve que obtener los datos de tráfico, que se encuentran disponibles en la web del Ayuntamiento de Madrid, sin embargo fue necesario descargar ficheros uno a uno ya que estaban divididos por mes del año. El intervalo de fecha fue de enero de 2018 a julio de 2021. Como continuación a la parte de recopilación de datos y una vez obtenidos los de tráfico, mi siguiente paso era acceder a los datos meteorológicos que se encuentra en el sitio web de la AEMET. Para obtener los datos de la agencia fue necesario el intervalo y la estación meteorológica de la zona en la que se encuentra las calles que vamos a estudiar.

Una vez recopilados los tres grupos datos necesarios había que quitar las columnas que no eran necesarias, cambiar el formato en el que venia dado las fechas en los documentos de la agencia meteorológica a un formato común a todos los archivos. Además debido a que había datos incompletos fue necesario aplicar un valor a estos campos vacíos.

El siguiente paso era aplicar los métodos que se consideraron para este proyecto. Me encargué del algoritmo KNN y Random Forest, primero de la parte teórica buscando información sobre estos métodos y de la forma de aplicarlos y segundo usándolos con los datos que habíamos generado anteriormente.

Capítulo 7

Conclusión y trabajo futuro

La realización de este trabajo nos ha servido como un primer contacto con las técnicas de predicción que en la actualidad hay disponibles. También nos ha permitido comprobar que es recomendable hacer un estudio previo, como el que presentamos, de cara a elegir la herramienta adecuada para un cierto tipo de predicción que se desee hacer.

Sugerimos que un trabajo que continúe a éste debería estar dirigido a estudiar los posibles factores que influyeron en que los errores absolutos hayan sido altos. Algunas sugerencias ya hemos hecho y seguramente un estudio más minucioso permitiría detectar otras variables que también influyen. Finalmente, queremos decir que para nuestra formación la realización de éste trabajo ha sido de gran importancia.

Capítulo 7

Conclusions and Future work

This work has been a first contact with the prediction techniques that are currently available. It has also allowed us to verify that it is advisable to carry out a previous study, such as the one we present here, in order to choose the most suitable tool for a certain type of prediction we wish to make.

We suggest that further work should be directed at studying the possible factors that influenced the high absolute errors. Some suggestions have already been made, and surely a more thorough study would allow us to detect other variables that also play a role. Finally, we would like to say that for our career, carrying out this work has had a great importance.

Bibliografía

[1] David Alandete. John mccarthy, el arranque de la inteligencia artificial [internet]. https://elpais.com/diario/2011/10/27/necrologicas/1319666402_850215.html, 2011. El País.

[2] Medium. 2019 [citado 8 abril 2022]. Aprendizaje supervisado: Introducción a la clasificación y principales algoritmos [internet]. https://medium.com/datos-y-ciencia/aprendizaje-supervisado-introduccion-a-la-clasificacion-y-principales-algoritmos-dad

[3] Ayuntamiento de Madrid. Tráfico. histórico de datos del tráfico desde 2013. https://datos.madrid.es/portal/site/egob.

[4] Agencia Estatal de Meteorología. Datos meteorológicos de madrid. http://www.aemet.es/es/datos_abiertos/AEMET_OpenData.

[5] Javier Jiménez. La historia de claude shannon: el hombre que creó la información. https://www.xataka.com/historia-tecnologica/un-pequeno-homenaje-a-claude-shannon-el-hombre-que-creo-la-informacion, 2016. Xataka.

[6] Javier Sampedro. Marvin minsky, cerebro de la inteligencia artificial. https://elpais.com/elpais/2016/01/26/ciencia/1453809513_840043.html, 2016. El Pais.