UNIVERSITAT POLITÈCNICA DE VALÈNCIA

etsinf
Escola Tècnica Superior d'Enginyeria Informàtica

Escola Tècnica Superior d'Enginyeria Informàtica
Universitat Politècnica de València

Predicción de disponibilidad de bicicletas en estaciones de uso público

Trabajo Fin de Grado

Grado en Ingeniería Informática

Autor: Miguel Fernández-Vázquez Crespo

Tutor: Cèsar Ferri Ramírez

2014-2015

Resumen

Este trabajo fin de grado trata sobre la minería de datos y el aprendizaje automático. Para ello se centra en la realización de un caso de estudio que consiste en la predicción sobre la disponibilidad de bicicletas en servicios públicos de alquiler de las mismas, más concretamente el servicio llamado “Valenbisi” para la ciudad de Valencia, España. A pesar de que existen diversas aplicaciones y webs que informan sobre la disponibilidad en estos servicios, normalmente solo muestran información actual, pero no predicciones para las siguientes horas, lo cual puede resultar de gran interés cuando se quiere planificar varios trayectos.

Palabras clave: minería de datos, aprendizaje automático, regresión, predicción, valenbisi.

Abstract

This bachelor thesis deals about data mining and machine learning. It is focused on a case study involving the prediction of the availability of bicylces in public rental services, and more specifically the service called “Valenbisi” for the city of Valencia, Spain. Although there are many applications and websites that provide information about availability in these services, typically they show only current information, but not predictions for the next few hours, which can be of great interest when you want to plan several trips.

Keywords : data mining, machine learning, regression, prediction, valenbisi.

Tabla de contenidos

1. Introducción

Desde hace unos años el transporte en bicicleta ha ido creciendo a medida que los ciudadanos se han concienciado acerca de sus ventajas y beneficios, y por ello los gobiernos han ido facilitando en cierta medida su uso.

En la actualidad existen servicios públicos de alquiler de bicicletas en numerosas ciudades del mundo, no solo en Europa donde el uso de las bicicletas o popularmente “bicis” está bastante extendido sino también en muchas ciudades de Asia y América [DeMaio, P. y Meddin, R. 2013].

En la mayoría de casos, su negocio consiste en colocar numerosas estaciones repartidas por toda la ciudad. En estas estaciones es donde se encuentran las bicis y las bornetas, que son los lugares donde se pueden dejar estacionadas las bicicletas. Una vez las estaciones físicas están montadas, los usuarios ya pueden coger una bici en alguna de las estaciones de la ciudad y aparcarla en otra próxima al destino donde quieren desplazarse.

Dependiendo de la ciudad donde estén implantadas, las empresas suelen tener un modelo de negocio u otro. Aunque normalmente suelen cobrar una subscripción mensual o anual, además pueden cobrar un precio por tiempo de uso cada vez que te desplazas.

Con gran frecuencia, las empresas que gestionan y mantienen estos servicios, suelen también disponer de herramientas para facilitar el uso a los ciudadanos. Estas herramientas van desde servicios de atención al cliente, hasta páginas web o aplicaciones móviles donde poder consultar la disponibilidad de bicis en sus estaciones.

Por lo general, estás webs y aplicaciones no suelen informar sobre la disponibilidad de bicicletas en el futuro, únicamente se centran en mostrar la información actual. En este aspecto es donde este trabajo fin de grado toma su sentido y su ser.

Enmarcado en el ámbito de la Minería de Datos y el Machine Learning, este trabajo fin de grado pretende realizar un estudio para comprobar qué algoritmos de la minería de datos son capaces de generar mejores predicciones sobre la disponibilidad de bicicletas. Además como objetivo final se desea crear un sistema que, en tiempo real, pueda realizar predicciones sobre la disponibilidad futura de bicicletas en estaciones de uso público.

El trabajo fin de grado se centra en el servicio público de bicicletas existente en la ciudad de Valencia denominado “Valenbisi”.

1.1. Objetivos

La realización de este trabajo fin de grado tiene dos objetivos principales, que son los siguientes:

- El primero consiste en el desarrollo un sistema que genere predicciones en tiempo real del número de bicis disponibles en determinadas estaciones de Valenbisi.

- El segundo es el desarrollo de una aplicación para dispositivos Android que permita a los usuarios consultar estas predicciones en cualquier momento y lugar.

1.2. Motivación

La dificultad del caso del presente estudio, y la creciente importancia del Big Data y la Minería de Datos en el mundo empresarial han sido los dos grandes aspectos motivadores para la realización de este trabajo fin de grado.

Durante los últimos años se ha visto un aumento en la importancia de conseguir información de calidad a partir de los datos guardados por las empresas. Además normalmente los trabajos relacionados con este ámbito suelen estar más comunicados con el área de negocio ya que en muchas ocasiones se utilizan los

resultados de los estudios y predicciones de la Minería de Datos para la toma de decisiones sobre el producto o el resto de la organización.

2. Extracción de conocimiento desde bases de datos

El descubrimiento y la extracción de conocimiento desde bases de datos, en inglés Knowledge Discovery in Databases (KDD), se empezó a usar a finales de la década de 1980 y se refiere al proceso no-trivial de descubrir conocimiento e información potencialmente útil dentro de los datos contenidos en algún repositorio de información [Han, J. y Kamber, M. 2001]. Es un proceso que tomando como entrada grandes volúmenes de datos, extrae información de calidad que puede usarse para tomar decisiones basadas en relaciones o modelos dentro de los datos. La siguiente figura muestra las etapas del proceso KDD:

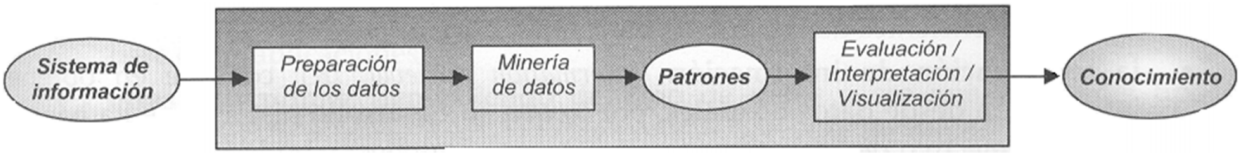

Figura 2.1.- Etapas de la extracción de conocimiento desde bases de datos (KDD)

2.1. Etapas de la extracción de conocimiento desde bases de datos (KDD)

Las fases que normalmente suelen realizarse en un trabajo de KDD son las siguientes:

- Determinar los objetivos del estudio

Primeramente y antes de empezar a realizar ninguna otra acción, se debe de entender el problema y determinar claramente los objetivos que se quieren alcanzar con el estudio de los datos disponibles.

- Preprocesamiento de datos

En muchos casos, se obtienen datos de muchas y muy distintas fuentes, por tal motivo el objetivo de esta fase, es por un lado limpiar los datos de tal manera que se puedan interpretar correctamente datos de distintas fuentes en el dataset final, además de eliminar todos los valores incorrectos, los valores no validos y los desconocidos. Por otro lado se ha de filtrar y transformar los datos necesarios para enriquecer el dataset.

- Generación del dataset

A continuación del preprocesamiento de los datos, se procede a crear el dataset. Para ello, y con la finalidad de reducir el tamaño del dataset, se deben seleccionar de entre las distintas fuentes, aquellos datos que puedan tener una influencia mayor en la organización de los datos. Y así tener las variables que influyen con más fuerza en el problema a solucionar.

- Minería de datos

La minería de datos es un proceso de estudio automático o semiautomático de grandes cantidades de datos. Está formado por un conjunto de técnicas, entre ellas el análisis matemático, la estadística y los algoritmos de búsquedas próximos a la inteligencia artificial para deducir los patrones y tendencias que se encuentran ocultos en los datos. Su objetivo final es extraer el conocimiento para facilitar la toma de decisiones.

En esta fase por tanto es donde se aplica la minería de datos en si y se realiza el aprendizaje, para ello se han de realizar primeramente unos análisis estadísticos, de tal forma que estos permitan la selección de los algoritmos que mejor correspondan de entre todos los diversos algoritmos de Machine Learning.

- Interpretación y evaluación de los resultados

Por último, se deben validar las conclusiones obtenidas al finalizar el proceso de extracción de conocimiento, es decir, comprobar que las conclusiones conseguidas son validas, suficientes y satisfactorias. Al realizar este paso, también se concluye cual es el algoritmo más apropiado para el aprendizaje.

2.2. Machine Learning

Machine Learning es una disciplina científica con el objetivo de que los sistemas aprendan automáticamente. Aprender en este contexto quiere decir identificar patrones complejos en millones de datos. La máquina que realmente aprende es un algoritmo que revisa los datos y es capaz de predecir comportamientos futuros.

2.2.1. Aprendizaje supervisado y no supervisado

Los problemas de Machine Learning pueden ser diferenciados en dos tipos de acorde al tipo de datos del problema y sobre los que va a realizar el aprendizaje.

Por un lado tenemos el aprendizaje supervisado. Este es aquel en el cual los datos están etiquetados, es decir, se conoce la clase a la que pertenecen cada uno de los datos utilizados en la fase de entrenamiento. De esta forma el modelo puede aprender por ejemplo en qué se asimilan los datos de una misma clase y cómo se diferencian de los datos pertenecientes a clases distintas. Este tipo de aprendizaje se utiliza con modelos predictivos, que pueden calcular el valor de una variable dependiente a partir del resto de datos llamados variables independientes.

En cambio, el aprendizaje no supervisado es aquel en el que se desconoce la clase de los datos usados para el entrenamiento. En estos casos, el modelo ha de ser capaz de encontrar las similitudes y patrones en los datos que se le pasan en el entrenamiento, para después en la fase de testing ser capaz de diferenciar los nuevos datos. Este aprendizaje está más relacionado con los modelos descriptivos, ya que estos identifican patrones que explican o resumen los datos, es decir, sirven principalmente para explorar las propiedades de los datos examinados.

2.2.2. Clasificación vs Regresión

Los problemas de clasificación son aquellos en los que el objetivo consiste en diferenciar los datos en distintas clases. Estos problemas son supervisados, ya que

para los datos utilizados durante el entrenamiento de los modelos, se conocen la clase a la que pertenecen. La clasificación asume que hay un conjunto de datos que se caracterizan por algún atributo o rasgo que pertenece a alguna clase en particular, además, las etiquetas de cada clase son valores discretos. El objetivo por tanto es construir los modelos de clasificación, que permitan asignar la etiqueta de clase correcta a datos antes no vistos y sin etiquetas de clase.

Por otro lado, la regresión consiste en un problema muy similar al de clasificación salvo que es este caso el objetivo es el cálculo de una variable continua, es decir un número. Más adelante se explicará mejor este tipo de problemas, y se expondrá un ejemplo de su aplicación.

2.3. Otros trabajos relacionados

Con el objetivo de recoger más información y conseguir una mejor idea sobre los objetivos de este trabajo fin de grado, se consultaron otros trabajos externos relacionados con el tema de la minería de datos y la predicción sobre el uso de bicicletas en servicios de alquiler públicos. A continuación se mencionan los trabajos consultados.

En el proyecto de fin de carrera titulado “Cálculo de modos y tiempos de desplazamiento en una ciudad usando fuentes públicas” [Juan Francisco Ortega Morán 2009] se realiza un sistema que, haciendo uso de técnicas de la minería web y la extracción automática de datos de la web, permite a los usuarios saber el tiempo que se tarda en desplazarse de un lugar a otro, utilizando múltiples servicios de transporte públicos. Este proyecto a servido sobre todo para conocer técnicas sobre la extracción de información de la web y la implementación de scrapers.

“Profiling users of the Vélo’v bike sharing system” [A. Zimmermann, M. Kaytoue, C. Robardet, J. Boulicaut y M. Plantevit, 2015] consiste en un estudio sobre los patrones geográficos de movilidad observados en el servicio de alquiler de bicicletas francés Vélo’v.

3. Recogida y limpieza de datos

3.1 Análisis inicial del problema

A partir del problema anteriormente expuesto en la introducción, y antes de diseñar el sistema, se empezó por la lectura y comprensión teórica de términos y métodos relacionados con el tema.

Una vez que se consideró que los conocimientos teóricos y sobre métodos de la Minería de Datos básicos habían sido comprendidos, se pasó al análisis real del problema.

Por un lado, el dato principal a predecir es el de la disponibilidad de bicicletas en las estaciones, es decir es un número continuo, no una clase. Por lo que el problema es considerado como un problema de regresión.

Por otra parte, entre las fuentes de datos disponibles, se encuentra un histórico con la disponibilidad de las Valenbisi durante los últimos años, hora a hora. Por ello el problema es considerado como supervisado, ya que se encuentran disponibles los datos con sus resultados correctos.

Una de las primeras fases consistió en pensar e ir apuntando cualquier aspecto que pudiera afectar al transporte, y en concreto al transporte en bicicleta. Para ello no se tuvo en cuenta si los datos correspondientes a los aspectos planteados iban a poderse recoger o calcular en los siguientes pasos, sino que el objetivo en este momento fue más bien el de una tormenta de ideas. Es decir conseguir cuantos más aspectos diferentes fuera posible, para después hacer una selección de entre ellos. Se llegó a la conclusión que los aspectos más influyentes eran los siguientes:

- Las condiciones climatológicas.

El transporte en bicicleta al necesitar de esfuerzo físico, requiere que las condiciones climatológicas cumplan unos mínimos, ya que en general nadie cogería una Valenbisi con condiciones climatológicas extremas, ya sea de calor, frío, viento o

lluvia. Así pues se hizo una suposición de que los aspectos que más afectarían serian los grados centígrados, los mililitros por metro cuadrado de lluvia (0 mm^2 en caso de que no llueva) y la velocidad del viento, sin importar su dirección.

Como ya se ha dicho antes, esta suposición inicial no significa que todos estos parámetros fueran tenidos en cuenta más adelante a la hora de generar las predicciones.

- El mes del año, día de la semana y hora del día.

Este aspecto guarda relación con el anterior ya que normalmente en Valencia en los meses de verano suelen alcanzarse temperaturas bastante altas entre las 11:00 y las 17:00 horas. En los de otoño suelen haber más lluvias, aunque no suelen depender de la hora del día. En los de invierno, las temperaturas por la noche y a primeras horas del día suelen ser más bajas, y durante el resto de las horas, suelen ser temperaturas más moderadas, propias también de los meses de primavera.

La fecha también influye en cuanto a que los meses de verano son no lectivos para los estudiantes y además es el periodo vacacional por excelencia. El día de la semana se ha de tener en cuenta, sobre todo para diferenciar entre fines de semana y días entre semana. Por último y más evidente, en las horas nocturnas el uso de Valenbisi es bastante más reducido a lo largo de todo el año, a excepción de aquellas estaciones cercanas a zonas de ocio y fiesta durante las noches de fin de semana.

- El calendario festivo.

Tanto las fiestas nacionales como las locales pueden influir en la movilización de la población. Los trabajadores no tienen que ir a sus empleos/oficinas ni los estudiantes a las universidades/colegios.

- La localización geográfica de cada una de las estaciones.

Dependiendo de donde se encuentren las estaciones de Valenbisi, el comportamiento en cuanto a las bicis disponibles puede ser muy diferente. Por ejemplo, cabría suponer que las estaciones cercanas a las universidades deberían de estar llenas de bicis sobre las 9:00 de la mañana, ya que los estudiantes suelen empezar las clases a partir de las 8:00. En cuanto a esta localización, se pensó que no es tan importante la latitud y longitud de cada una de las estaciones sino más bien alrededor de qué se encuentran.

Dado que Valencia es una ciudad sin apenas desniveles, la altitud no pareció a priori un elemento que pudiera afectar.

- Los eventos deportivos/musicales….

Sobre todo aquellos eventos que movilizan a mucha gente, como por ejemplo los partidos de fútbol, ya que en los minutos previos al comienzo de cada partido, las estaciones más cercanas al estadio deberían de llenarse y al contrario en los minutos posteriores al final del partido.

A continuación del análisis del problema, es necesario recoger los datos de cada una de las distintas fuentes y realizar una limpieza sobre ellos. Esto es debido a que muy frecuentemente, los datos se encuentran en distintos formatos dependiendo de la fuente de donde proceden, o se necesita crear datos derivados a partir de otros más simples.

3.2. Histórico de estaciones de Valenbisi

La principal e indispensable fuente de datos ha consistido en el histórico sobre cada una de las estaciones de Valenbisi que Alex Barros ha ido recolectando desde que se comenzó a usar este servicio público [biciv] y ha cedido al departamento del DSIC.

Estos datos estaban escritos en formato CSV, un archivo por cada estación de Valenbisi. La organización era la siguiente:

station_id	date	hour	avg_available	avg_free	med_available	med_free	src_count
261	2012-05-04	14	**1**	14	1	16	7
261	2012-05-04	15	**1**	16	0	17	59
261	2012-05-04	16	**3**	14	3	14	58
…	…	…	**…**	…	…	…	…

Tabla 3.1.- Fuente histórico de una estación de Valenbisi

Uno de los primeros problemas a resolver fue unificar los datos, ya que para cada estación, los datos estaban divididos en dos archivos distintos. Por un lado el histórico desde 2011 hasta 2014 y por otro los datos desde final de 2014 hasta febrero de 2015. Esto fue fácil de solucionar ya que en todos los casos, ambos archivos mantenían el mismo formato.

De entre estas columnas, se han seleccionado para la generación del dataset las siguientes:

station_id	date	hour	avg_available	avg_free	med_available	med_free	src_count
261	2012-05-04	14	1	14	1	16	7
261	2012-05-04	15	1	16	0	17	59
261	2012-05-04	16	3	14	3	14	58
…	…	…	…	…	…	…	…

Tabla 3.2.- Datos de interés de la fuente del histórico de Valenbisi

A continuación se explican cada una de las columnas elegidas:

- Station_id: es el número de identificación único de cada una de las estaciones de Valenbisi.

- Date: el año, mes y día correspondientes a la medida.

- Hour: la hora de la medida. En los datos disponibles solo hay una medida por hora, por lo que existe una limitación, y es que no se puede saber cómo ha ido cambiando el numero de bicis a cada minuto.

- Avg_available: el número de bicis disponibles en el momento de la consulta.

El resto de columnas fueron desechadas, cabe destacar que la predicción es sobre la cantidad de bicis disponibles, no de bornetas libres (sitios donde aparcar las bicis). Por ello la columna avg_free se desechó.

3.3. Fuentes adicionales

De entre todos los aspectos que se plantearon en el análisis inicial del problema, se paso a hacer una selección de entre ellos, ya que de haber intentado recoger e integrarlos todos, el trabajo fin de grado se hubiera alargado demasiado. Aquellos aspectos que se consideraron más importantes y potencialmente recolectables fueron los siguientes:

3.3.1. Las condiciones climatológicas

Se decidió simplificar el problema y tener en cuenta solo si en algún momento del día correspondiente a cada una de las mediciones había llovido. Es decir un valor verdadero/falso en caso de que haya o no haya precipitaciones. Esta decisión fue tomada principalmente por la poca fiabilidad y la dificultad al acceso sobre datos meteorológicos. Muchas webs solo ofrecen el tiempo actual o el pasado hasta cierto límite, pero en general no permiten acceder gratuitamente al histórico completo sobre el tiempo. Ni tan si quiera webs de organismos públicos como AEMET permiten acceder libremente a estos datos.

Después de buscar exhaustivamente, se encontró la página web de [Freemeteo] en donde están disponibles las mediciones climáticas desde 2008. Pero ir recorriendo manualmente cada uno de los días del año desde Diciembre de 2011 (año del inicio de Valenbisi) hasta Febrero de 2015, e ir anotando si había llovido o no en cada uno de esos días no es una actividad que un informático se plantea.

Así pues para automatizar la extracción de estos datos se diseñó y escribió un programa en Java que, haciendo uso de la librería [Selenium] y de la propia estructura de URLs de la web, iba haciendo peticiones HTTP GET de cada una de las distintas

páginas web ajustando los parámetros debidamente para así poder acceder al tiempo que hizo en cada uno de los días del año desde 2011. Una vez el web scraper cargaba una página, se almacenaban los datos en un archivo con el siguiente formato:

fecha	temp. minima	temp. maxima	precipitaciones (mm)	lluvia (V/F)
01/12/2011	7ºC	19ºC	0mm	rain=false
02/12/2011	10ºC	17ºC	0mm	rain=false
03/12/2011	10ºC	16ºC	7,1mm	rain=true
…	…	…	…	…

Tabla 3.3.- Fuente histórico sobre la información meteorológica

En cuanto a estos datos, y como ya se ha comentado, al final se decidió tener en cuenta solo la fecha y la lluvia, la cual toma un valor verdadero si la cantidad de precipitación en mililitros supera los 0 mililitros, o falso en caso contrario.

3.3.2. La fecha

Esto es, el año, el mes, el día de la semana (entre 1 y 7) y la hora del día. En cuanto a la fecha, se tuvo que realizar una limpieza para utilizar el mismo formato en el dataset final.

3.3.3. El calendario festivo

Dada la facilidad al acceso de estos datos, se decidió generar una lista con las fechas de los días festivos en la ciudad de Valencia desde 2011, la cual se hizo de manera manual ya que era bastante simple.

3.4. Generación del dataset

Para generar el dataset, el dato común en cada una de las fuentes que sirvió de nexo entre ellas fue la fecha. De esta forma se pudo concatenar la información de cada una de las fuentes.

Este es el formato final del dataset:

station	year	month	dayIn Week	hour	holiday	rain	bikes_ 24h	bikes_ 12h	bikes_ 6h	bikes_ 3h	bikes_ 2h	bikes_ 1h	current _bikes	bikes_ 1h_fut
1	2014	02	6	0	FALSE	FALSE	6	8	1	2	4	5	5	4
1	2014	02	6	1	FALSE	FALSE	7	5	0	4	5	5	4	2
1	2014	02	6	2	FALSE	FALSE	7	0	1	5	5	4	2	0
...	…	…	…	…	…	…	…	…	…	…	…	…	…	…

Tabla 3.4.- Formato del dataset generado

La primera columna del dataset no se utilizó en ninguna de las predicciones al tener siempre el mismo valor correspondiente a cada una de las estaciones.

A la hora de generar este dataset para cada una de las estaciones, la mayor problemática consistió en hacer coincidir correctamente las filas de las distintas fuentes, ya que a pesar de que se unificó el formato de la fecha, en ocasiones había fechas en las que por razones que se desconocen no se habían tomado mediciones sobre las bicis o sobre el tiempo.

Otra de las dificultades que se resolvieron al generar el dataset fue el rellenar las columnas correspondientes a las bicis disponibles 1, 2, 3, 6, 12 y 24 horas antes de las disponibles actualmente. Esta problemática surgió debido a que en la fuente del histórico de Valenbisi, por cada fila de la tabla solo se encuentra la medición sobre las bicis disponibles en ese momento. Por ello si queremos saber cuántas bicis había 1, 2, 3, 6, 12 o 24 horas antes, nos tenemos que desplazar en la tabla 1, 2, 3, 6, 12, o 24 filas hacia arriba. Lo mismo ocurre con la última columna, que son las bicis disponibles 1 hora después de la medida actual, pero en este caso el dato se consigue en las bicis actuales de la siguiente fila.

4. Modelos de predicción

4.1. Predicción básica

Antes de empezar a hacer predicciones más complejas utilizando técnicas de Machine Learning, se decidió realizar una predicción más básica en la que no hiciera falta entrenar y testear los modelos.

Está predicción consistía en la suposición de que las bicis disponibles dentro de una hora iban a ser las mismas que las disponibles en el momento actual. Este sería un ejemplo tomando los datos de la tabla anterior:

station	year	month	dayInWeek	hour	…	current_bikes	**bikes_1h_fut** (predicción)	**bikes_1h_fut** (valor real)
1	2014	02	6	0	…	5	5	4
1	2014	02	6	1	…	4	4	2
1	2014	02	6	2	…	2	2	0
…	…	…	…	…	…	…	…	…

Tabla 4.1.- Datos a utilizar para las predicciones básicas

La columna “current_bikes” es por tanto la utilizada para realizar la predicción, como se puede ver, la predicción son exactamente los mismos datos que los indicados en esta columna.

Una vez obtenidos estos datos, se procedió a calcular el error en la predicción. Para ello tenemos dos posibles medidas del error. Una es el MAE, es decir la media aritmética del error absoluto de cada una de las predicciones. Otra es el MSE, que también consiste en la media, en este caso del cuadrado de los errores absolutos de cada una de las predicciones.

bikes_1h_fut (predicción)	**bikes_1h_fut** (valor real)	Error	Squared error
5	4	5 - 4 = 1	(5 - 4) ^ 2 = 1
4	2	4 - 2 = 2	(4 - 2) ^ 2 = 4
2	0	2 - 0 = 2	(2 - 0) ^ 2 = 4
...	…	MAE = (1 + 2 + 2) / 3 = 1.66	MSE = (1 + 4 + 4) / 3 = 3

Tabla 4.2.- Cálculo del error para las predicciones realizadas

De igual forma se hizo para calcular la misma predicción pero teniendo en cuenta las bicis que había 2, 3, 6, 12 y 24 horas por separado. Como es de esperar cuanto más pasado en el tiempo, peor es la predicción. En la siguiente gráfica se pueden observar la disponibilidad de bicicletas real y las predicciones realizadas a 1, 2 y 3 horas para la estación 1 de Valenbisi durante un día completo.

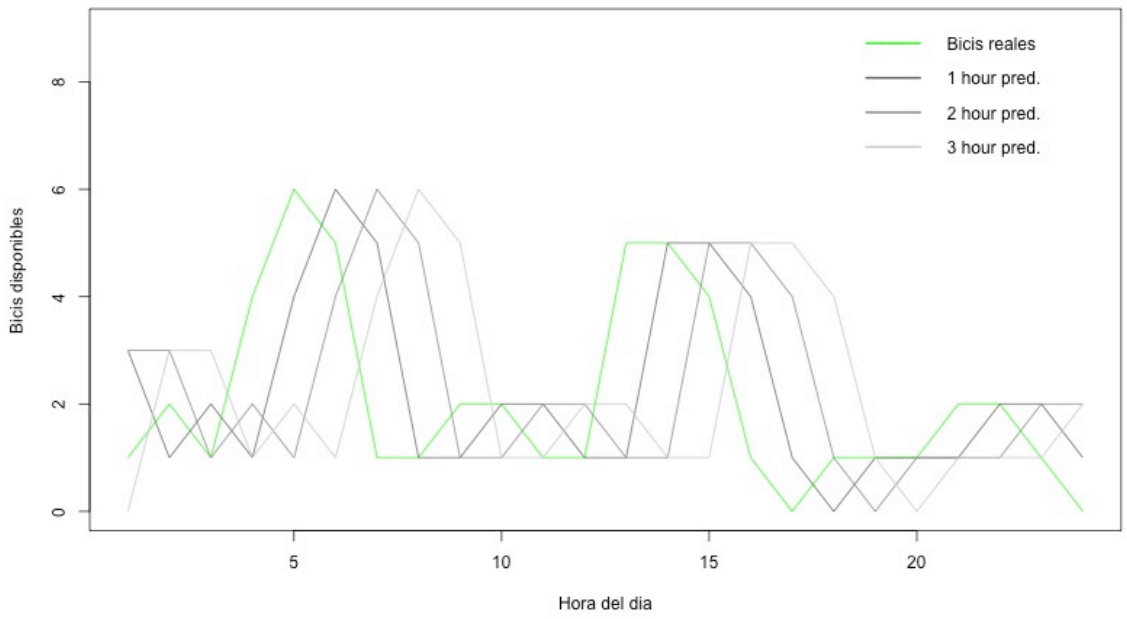

Figura 4.1.- Disponibilidad de bicicletas real y predicho en la estación 1 de Valenbisi durante un día completo

En la siguiente tabla podemos ver los valores del MAE y MSE para el conjunto de todas las estaciones usando la predicción básica:

	Predicción a 1 hora	Predicción a 2 horas	Predicción a 3 horas	Predicción a 6 horas
MAE	1.415635	2.322165	2.994326	4.311014
MSE	5.903399	14.73536	22.96746	41.00966

Tabla 4.3.- Comparación de los errores medios para cada una de las predicciones básicas

4.2. Predicciones con entrenamiento y test de modelos

Para realizar una predicción real, el proceso a realizar es bastante distinto a la predicción básica anterior en la que realmente no se está realizando ninguna predicción inteligente. Se basa principalmente en dos fases:

La primera fase es el entrenamiento del modelo, en la cual se dispone de todas las variables del dataset, incluyendo las correspondientes al resultado correcto de la predicción. Con estos datos el algoritmo es capaz de aprender el modelo correspondiente.

La segunda fase es el testing. En esta fase ya tenemos el modelo entrenado por un lado, y nuevos datos de prueba por otro, con la diferencia de que estos datos no incluyen la variable a predecir. Una vez el modelo calcula el valor de la variable a predecir, se calcula también el error entre el valor real y el predicho.

Por tanto para poder realizar las dos fases enunciadas anteriormente, es necesario dividir el dataset entre datos de entrenamiento y datos de prueba. Esta división se puede hacer de distintas formas, para este trabajo fin de grado se decidió realizar la partición tomando el ochenta por ciento de los datos, empezando por el primer dato, como entrenamiento y el veinte por ciento restante para las pruebas. En otros estudios la partición se realiza de forma aleatoria, para así conseguir una mejor independencia entre los datos.

Existen multitud de algoritmos, de entre los cuales se seleccionaron dos, regresión lineal y k vecinos más cercanos.

4.2.1. Regresión lineal

Regresión lineal es un algoritmo que se suele utilizar para predecir una variable Y a partir de un conjunto de variables X que son capaces de explicar el valor de Y. Este algoritmo asigna un peso a cada una de las variables X que se utiliza para calcular el valor a predecir Y.

Por tanto primero se ha de entrenar el modelo usando la columna de las bicis que habrá disponibles dentro de una hora como variable Y, y el resto de las columnas como las variables X.

De esta forma obtendremos los pesos correspondientes a cada una de las variables X que pueden explicar la variable Y.

Una vez entrenado el modelo, este es capaz de, a partir de un nuevo conjunto de variables explicatorias X, generar las correspondientes predicciones Y. Es decir calcular el valor de Y. A continuación se puede observar un ejemplo de este proceso.

year	month	dayInWeek	hour	holiday	rain	bikes_24h	bikes_12h	bikes_6h	bikes_3h	bikes_2h	bikes_1h	current_bikes	bikes_1h_fut
2014	02	6	0	FALSE	FALSE	6	8	1	2	4	5	5	4
2014	02	6	1	FALSE	FALSE	7	5	0	4	5	5	4	2
2014	02	6	2	FALSE	FALSE	7	0	1	5	5	4	2	???

Tabla 4.4. Datos a utilizar para las predicciones con entrenamiento y test

Para este ejemplo, el dataset se entrenaría con los valores de las filas en el recuadro gris y se obtendría un peso para cada una de las variables explicatorias (variables X) para calcular la variable dependiente (variable Y), por ejemplo:

year * 0,05425 + month * 0,1212 + … + current_bikes * 1,5232 = bikes_1h_fut

Por tanto un modelo de regresión lineal al fin y al cabo no es más que una fórmula donde cada variable tiene asignado un peso y que genera como resultado el valor de la variable que se quiere predecir.

A partir de aquí, solo tenemos que aplicar la fórmula con datos nuevos de testing, es decir la fila morada del ejemplo y calcular el error tanto MAE como MSE. Por ejemplo:

2014 * 0,05425 + 02 * 0,1212 + … + 2 * 1,5232 = 1,06948

Para calcular el error en la predicción tan solo tenemos que realizar la resta (en valor absoluto) entre el resultado predicho y la medida real:

Error = abs(1,06948 - 0) = 1,06948

En las siguientes 7 gráficas podemos ver, para la estación 1 de Valenbisi, cada día de la semana, la media de las bicicletas disponibles reales y los valores predichos a cada hora por el algoritmo de regresión lineal durante un período de 12 semanas:

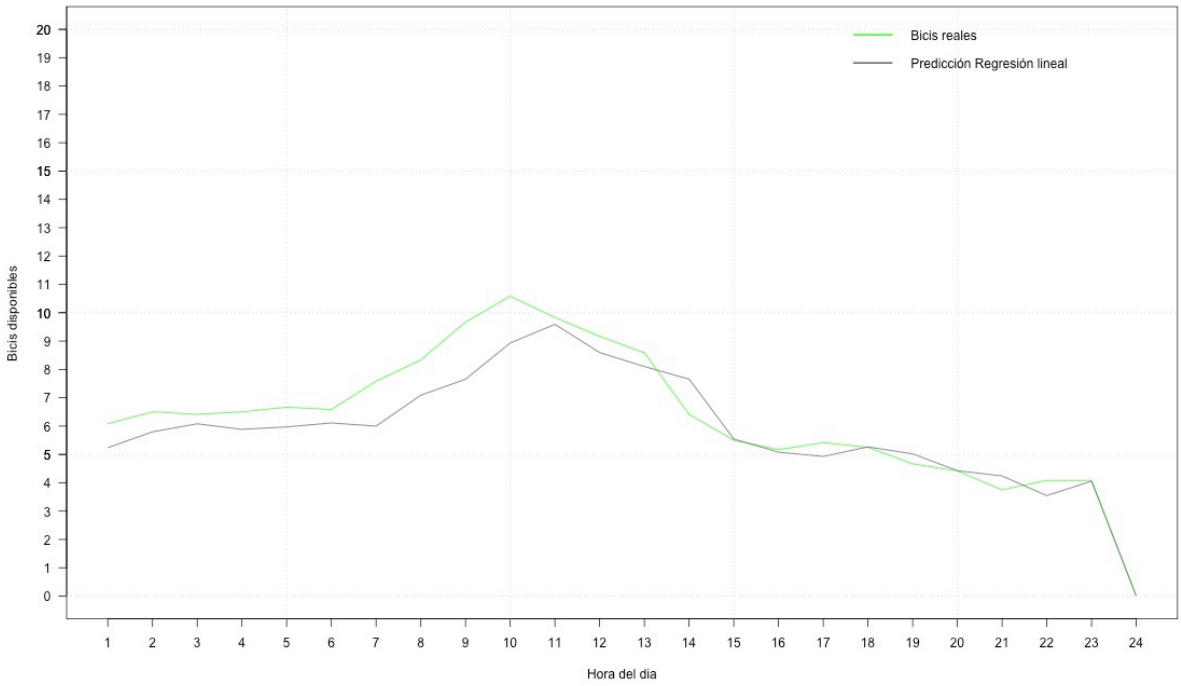

Figura 4.2.- Comparación entre la media de la disponibilidad real de bicicletas y la predicción del algoritmo de regresión lineal en la estación 1 de Valenbisi los lunes a cada hora del día.

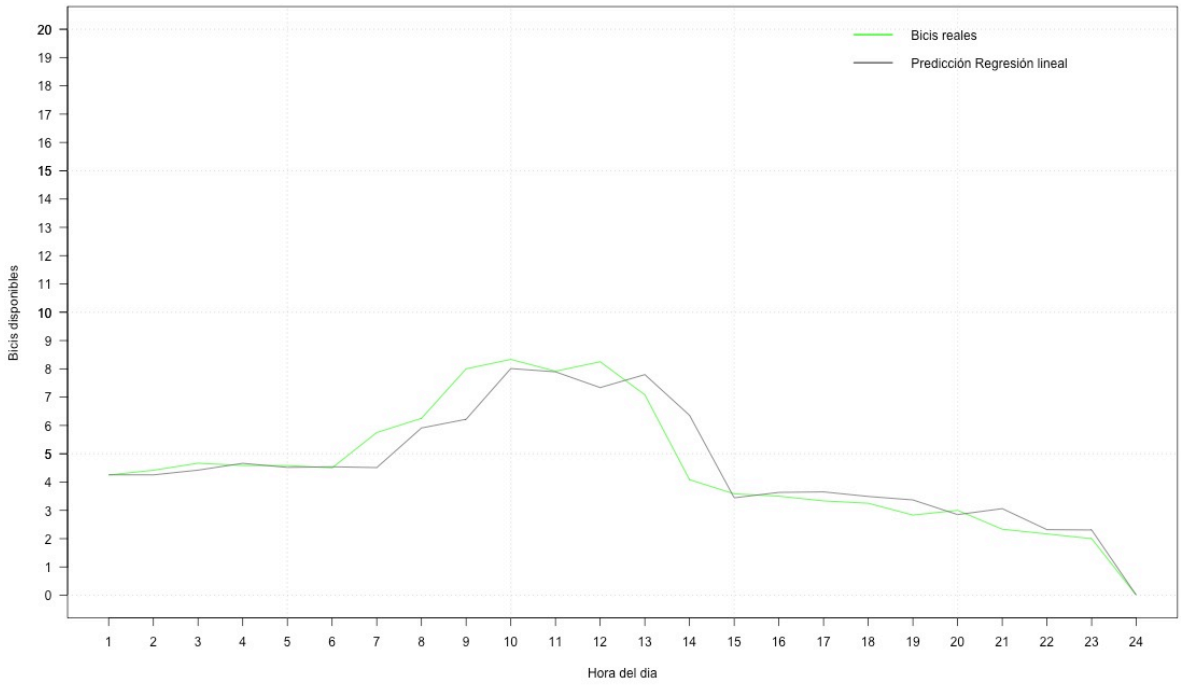

Figura 4.3.- Comparación entre la media de la disponibilidad real de bicicletas y la predicción del algoritmo de regresión lineal en la estación 1 de Valenbisi los martes a cada hora del día.

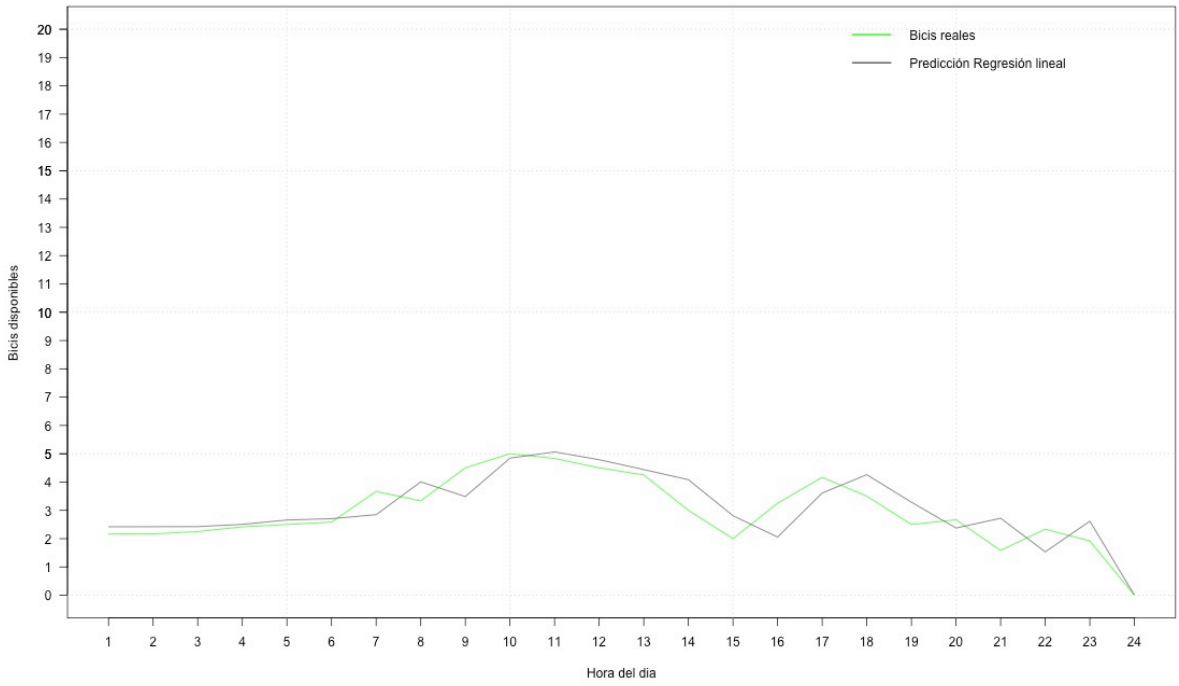

Figura 4.4.- Comparación entre la media de la disponibilidad real de bicicletas y la predicción del algoritmo de regresión lineal en la estación 1 de Valenbisi los miércoles a cada hora del día.

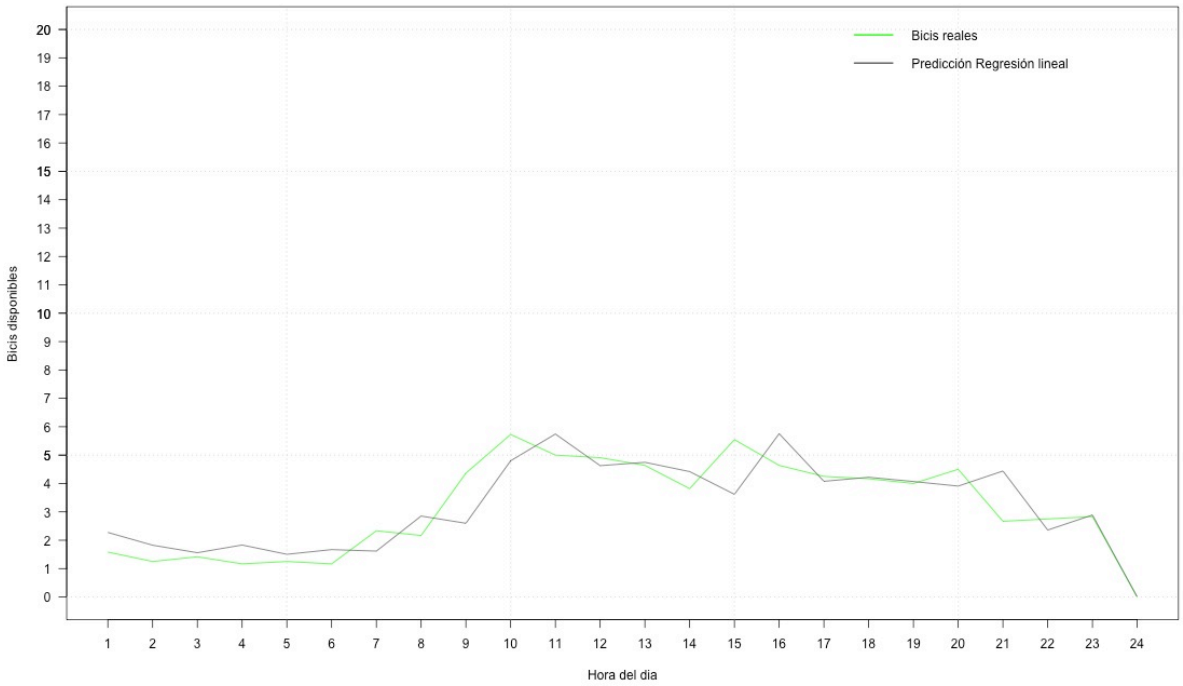

Figura 4.5.- Comparación entre la media de la disponibilidad real de bicicletas y la predicción del algoritmo de regresión lineal en la estación 1 de Valenbisi los jueves a cada hora del día.

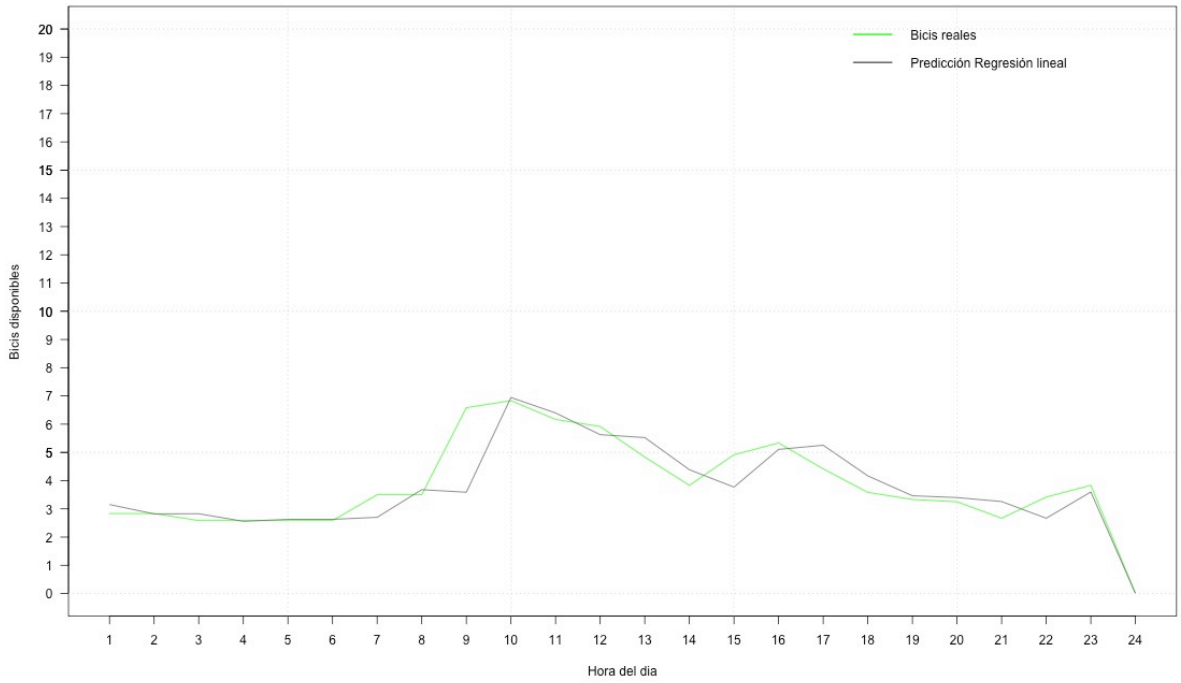

Figura 4.6.- Comparación entre la media de la disponibilidad real de bicicletas y la predicción del algoritmo de regresión lineal en la estación 1 de Valenbisi los viernes a cada hora del día.

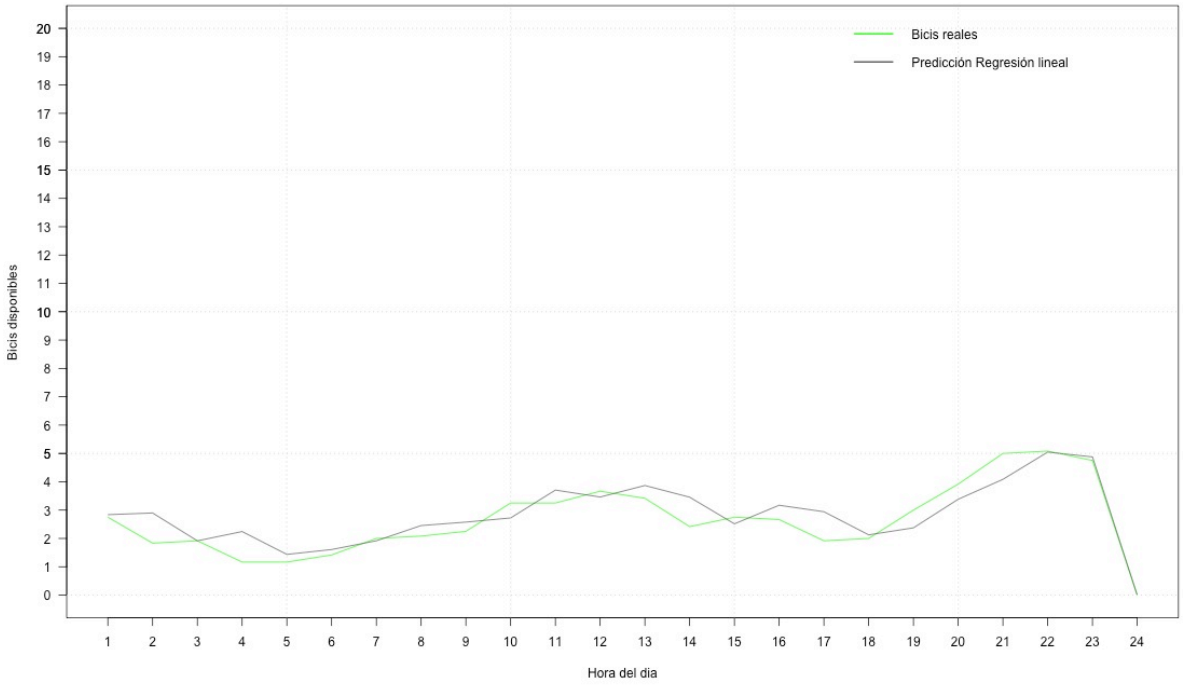

Figura 4.7.- Comparación entre la media de la disponibilidad real de bicicletas y la predicción del algoritmo de regresión lineal en la estación 1 de Valenbisi los sábados a cada hora del día.

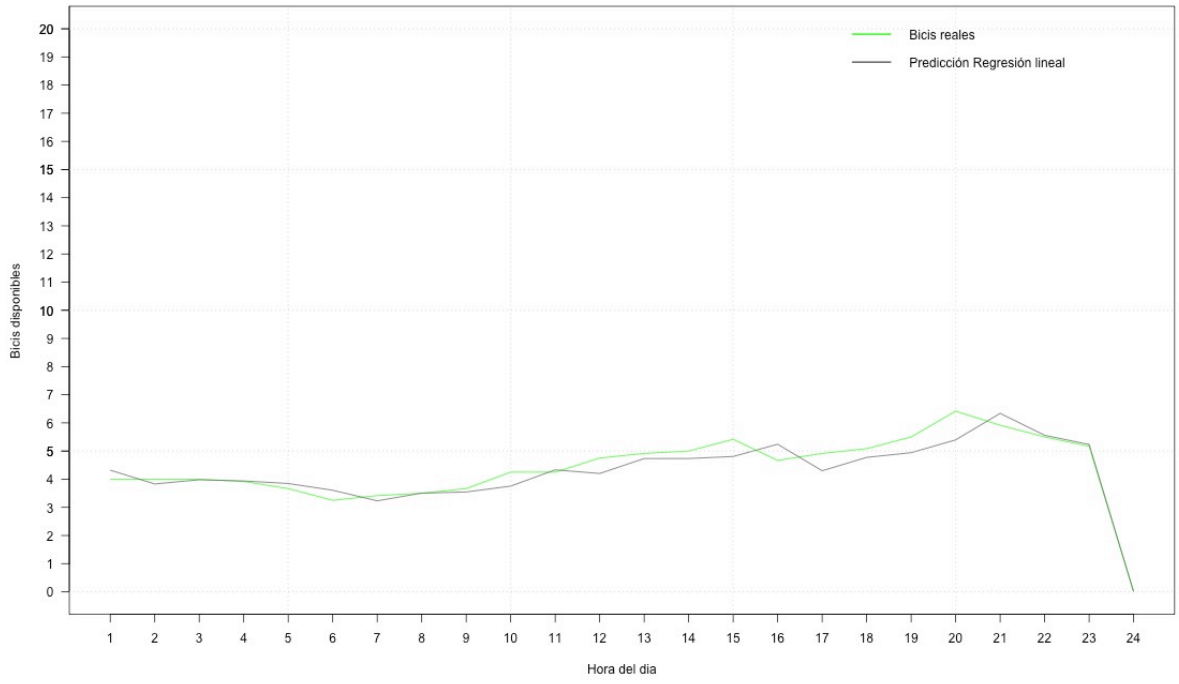

Figura 4.8.- Comparación entre la media de la disponibilidad real de bicicletas y la predicción del algoritmo de regresión lineal en la estación 1 de Valenbisi los domingos a cada hora del día.

4.2.2. K vecinos más cercanos

El algoritmo de aprendizaje Knn, al igual que el de regresión lineal, se basa en asignar pesos a unas determinadas variables para así después poder calcular el valor de la variable a predecir.

En este caso, se asignan pesos a los K vecinos más cercanos dependiendo de su distancia a la muestra. Cuanto menor sea la distancia entre la muestra a calcular y cualquiera de los vecinos ya calculados previamente, mayor será el peso de ese vecino.

En las siguientes 7 gráficas podemos ver, para la estación 1 de Valenbisi, cada día de la semana, la media de las bicicletas disponibles reales y los valores predichos a cada hora por el algoritmo de K vecinos más cercanos durante un período de 12 semanas:

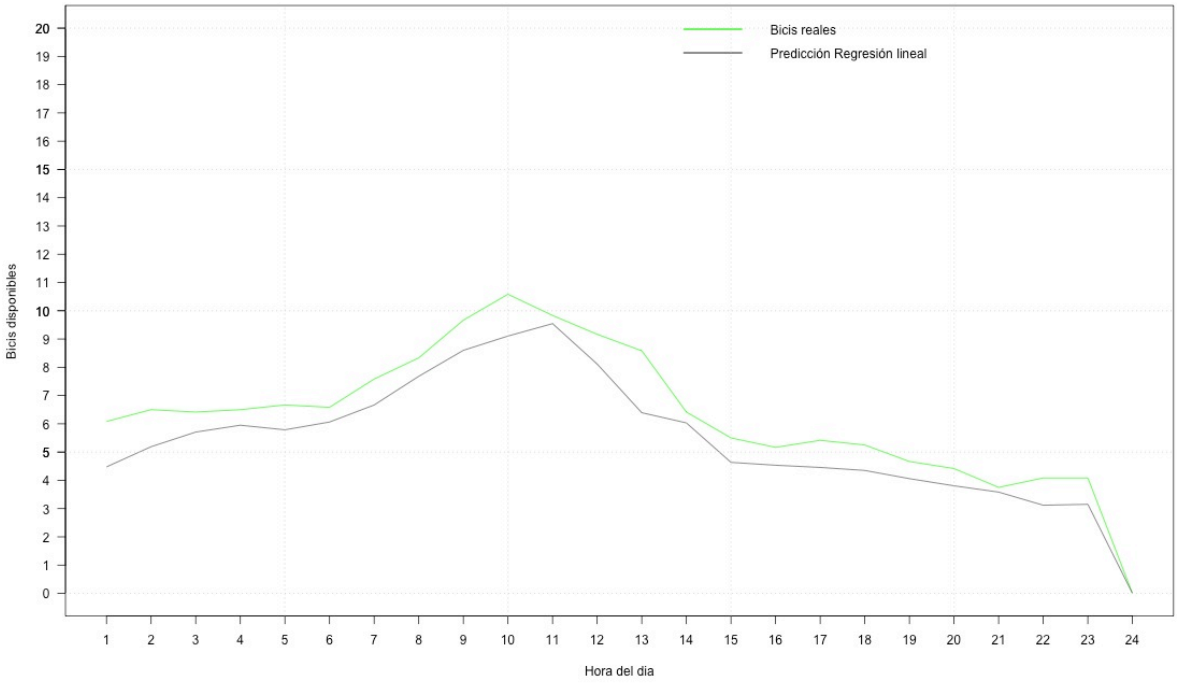

Figura 4.9.- Comparación entre la media de la disponibilidad real de bicicletas y la predicción del algoritmo de K vecinos más cercanos en la estación 1 de Valenbisi los lunes a cada hora del día.

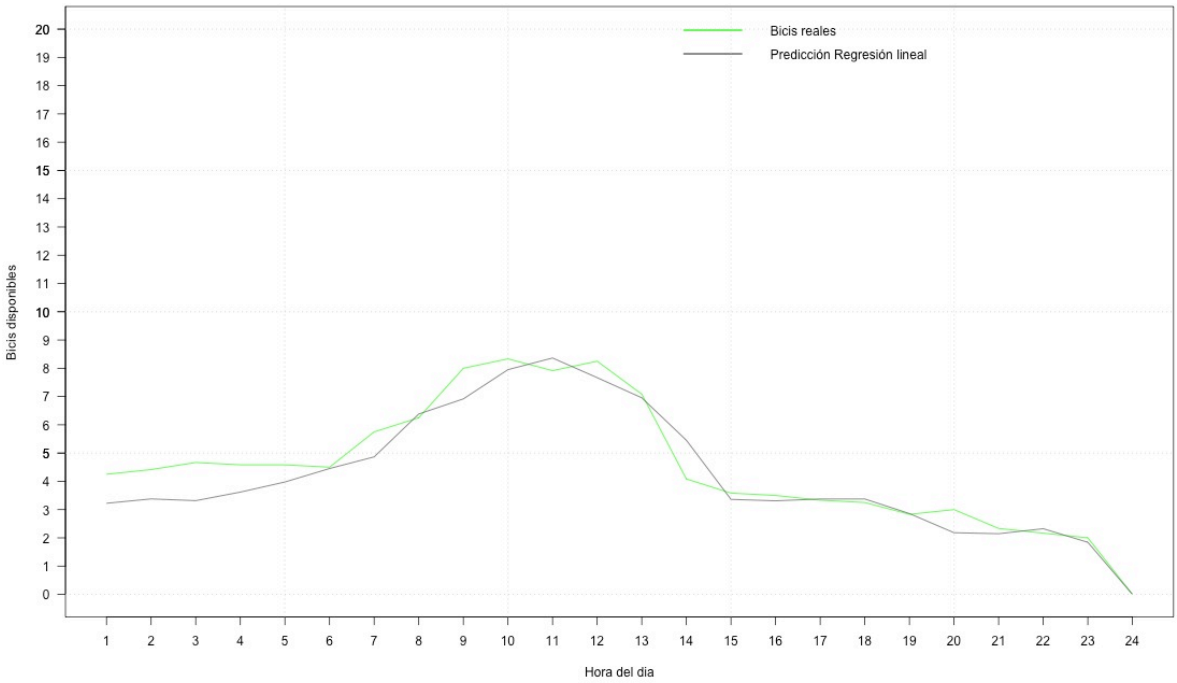

Figura 4.10.- Comparación entre la media de la disponibilidad real de bicicletas y la predicción del algoritmo de K vecinos más cercanos en la estación 1 de Valenbisi los martes a cada hora del día.

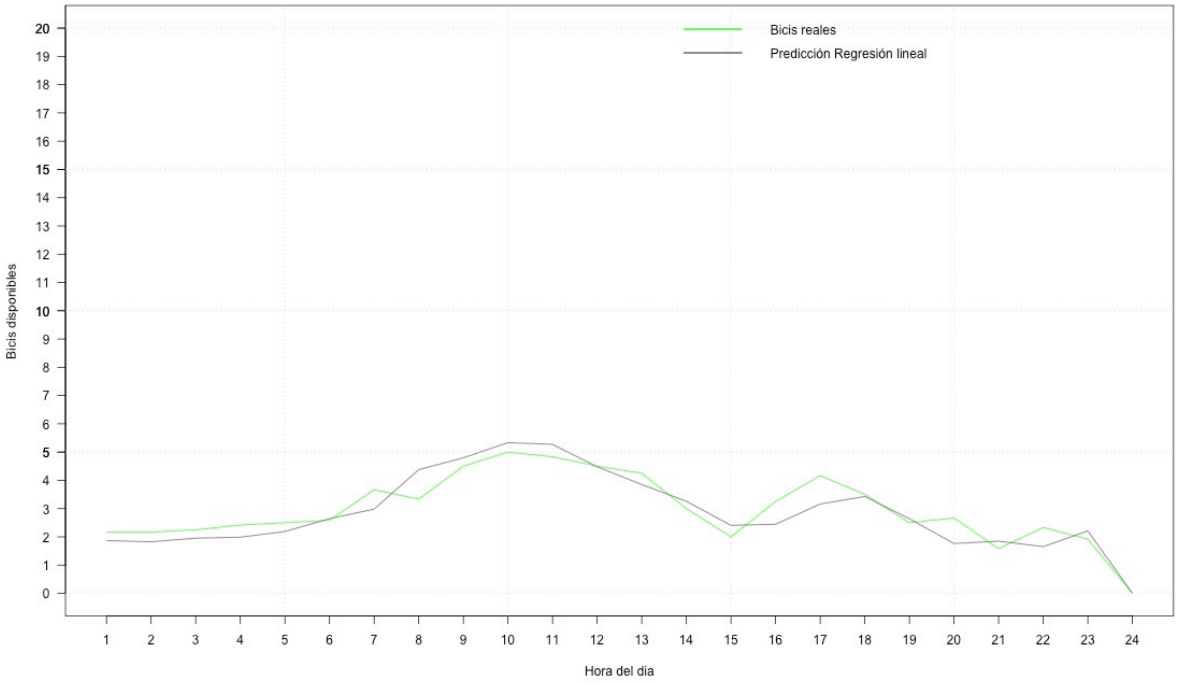

Figura 4.11.- Comparación entre la media de la disponibilidad real de bicicletas y la predicción del algoritmo de K vecinos más cercanos en la estación 1 de Valenbisi los miércoles a cada hora del día.

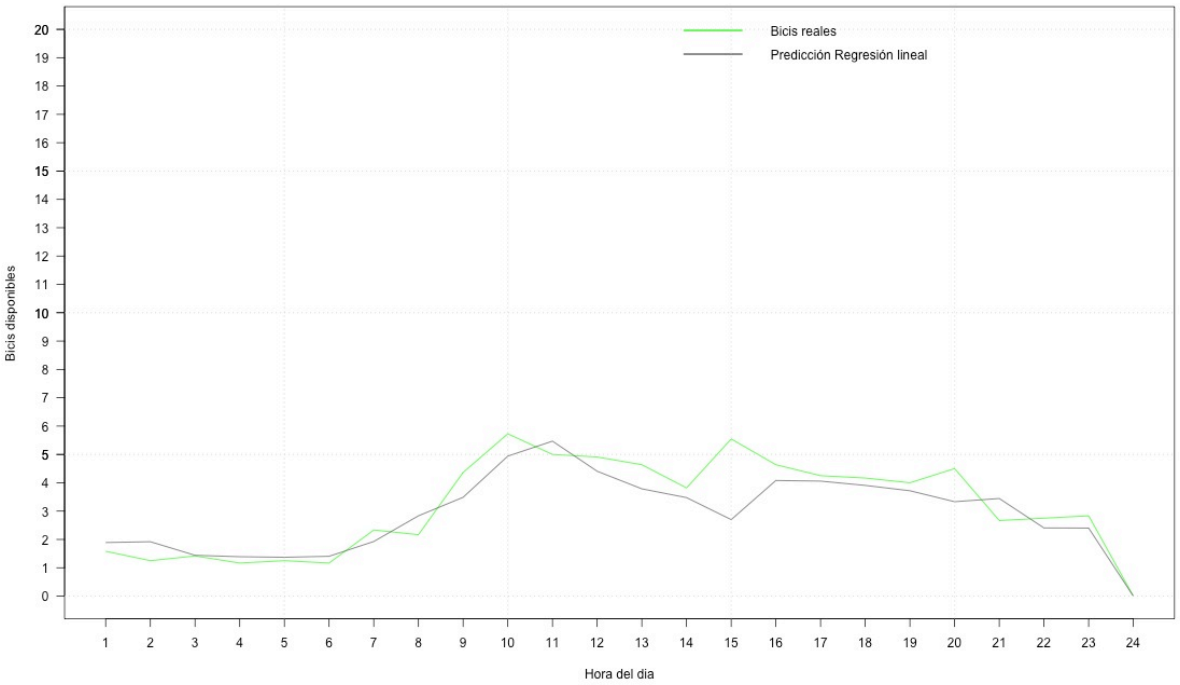

Figura 4.12.- Comparación entre la media de la disponibilidad real de bicicletas y la predicción del algoritmo de K vecinos más cercanos en la estación 1 de Valenbisi los jueves a cada hora del día.

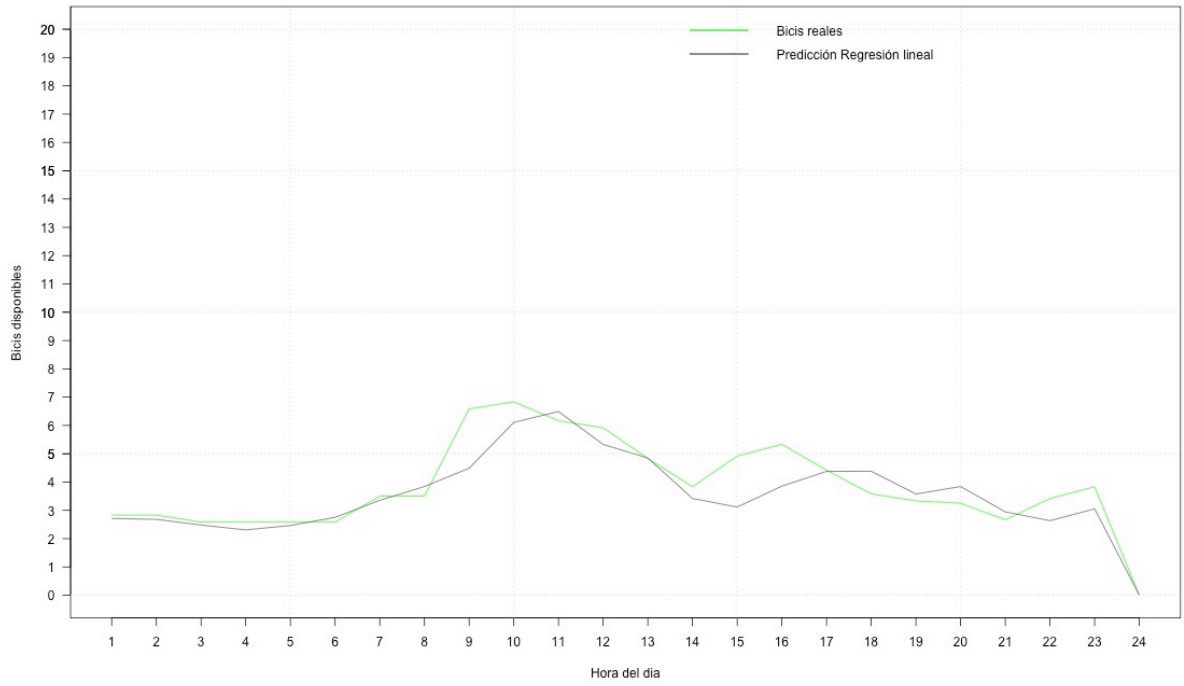

Figura 4.13.- Comparación entre la media de la disponibilidad real de bicicletas y la predicción del algoritmo de K vecinos más cercanos en la estación 1 de Valenbisi los viernes a cada hora del día.

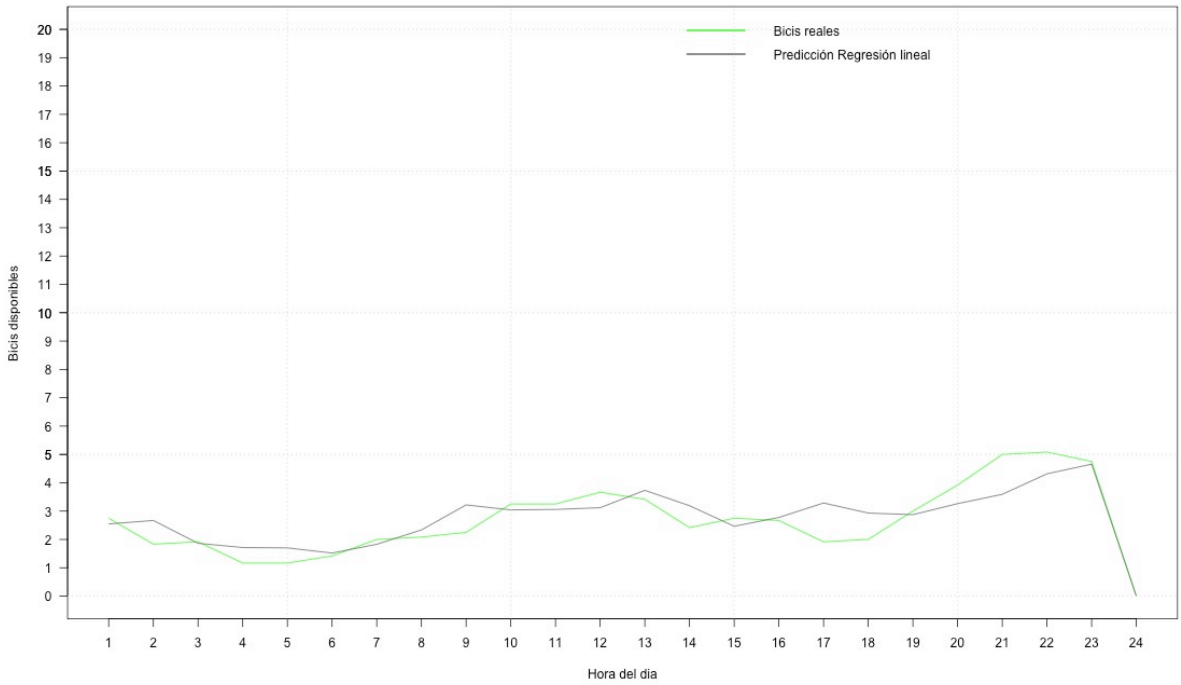

Figura 4.14.- Comparación entre la media de la disponibilidad real de bicicletas y la predicción del algoritmo de K vecinos más cercanos en la estación 1 de Valenbisi los sábados a cada hora del día.

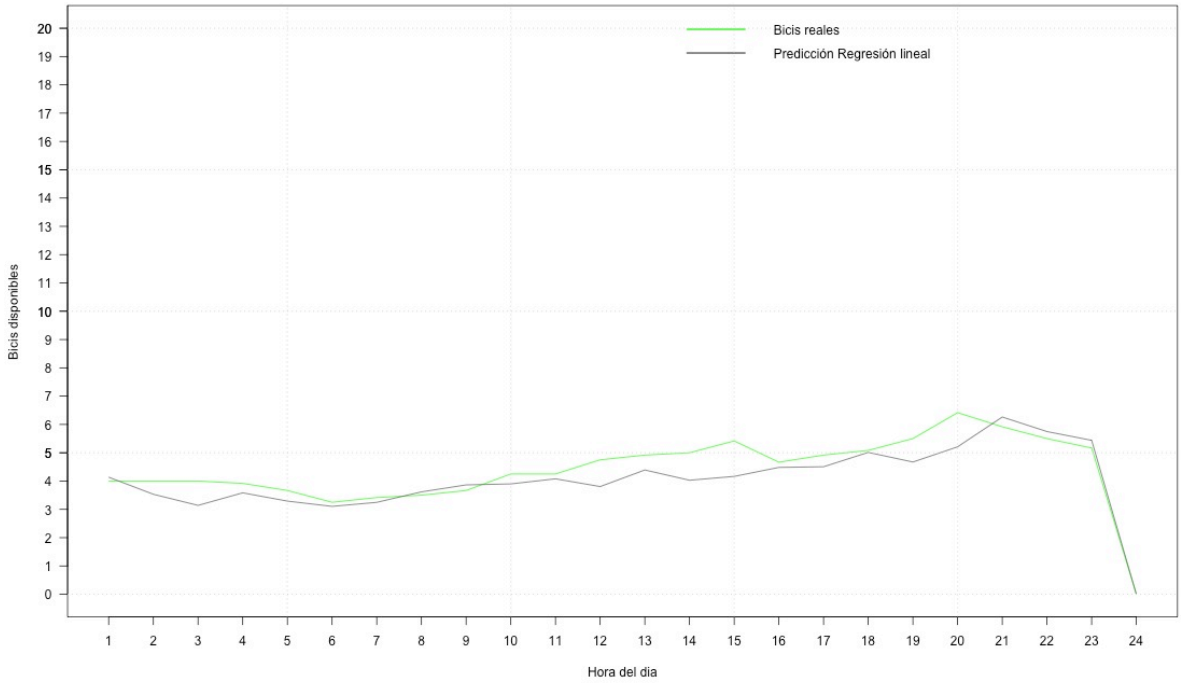

Figura 4.15.- Comparación entre la media de la disponibilidad real de bicicletas y la predicción del algoritmo de K vecinos más cercanos en la estación 1 de Valenbisi los domingos a cada hora del día.

Y en la siguiente tabla podemos ver el MAE y MSE de ambos algoritmos, en este caso para todas las estaciones durante todo el periodo del que se disponen datos:

	Regresión lineal	K vecinos más cercanos
MAE	1.371536	1.523779
MSE	4.192462	4.666395

Tabla 4.5.- Comparación del los errores medios para las predicciones con entrenamiento y pruebas

4.2.3. Resultados del estudio

A continuación se muestra una la misma tabla con los resultados del MAE y MSE que anteriormente pero en este caso, comparando la mejor predicción básica del subcapítulo anterior (la predicción a 1 hora) con las predicciones con entrenamiento y pruebas usando los algoritmos de regresión lineal y k vecinos más cercanos.

	Regresión lineal	K vecinos más cercanos	Predicción a 1 hora
MAE	1.371536	1.523779	1.415635
MSE	4.192462	4.666395	5.903399

Tabla 4.6.- Comparación entre las mejores técnicas para el cálculo de las bicis disponibles.

Podemos observar que la regresión lineal consigue el error medio más bajo, aunque la predicción básica a 1 hora obtiene prácticamente el mismo error. Esto puede significar que el peso que el algoritmo de regresión lineal dé a las bicis disponibles 1 hora antes de la predicción sea bastante grande. El aspecto negativo de esto es que el valor predicho depende en gran parte de las bicis que había disponibles una hora antes, por lo que un cambio drástico y rápido en las bicis disponibles en una estación (por ejemplo porque la empresa administradora del servicio mueva las bicis de un barrio a otro con un camión) podría generar malas predicciones.

Como conclusión de este estudio, se decidió usar el algoritmo de regresión lineal para los modelos que se utilizarían en las predicciones en tiempo real.

5. Desarrollo de una aplicación para dispositivos moviles basada en los modelos predictivos

El resultado final del trabajo fin de grado ha consistido en usar el modelo con el error más pequeño según los estudios, para realizar predicciones sobre la disponibilidad de las Valenbisi en tiempo real y mostrar esas predicciones en una aplicación Android.

A continuación se explican cada uno de los pasos que se llevaron a cabo para hacer el resultado final del TFG.

5.1. Modificaciones en la generación del dataset

Hasta este momento, el dataset utilizado para generar cada uno de los modelos de predicción y calcular sus errores, era un dataset estático. Es decir, se creó al principio juntando los datos de cada una de las distintas fuentes y desde ese momento, el dataset no cambió ni fue aumentando su tamaño.

Para el resultado final del trabajo fin de grado, queríamos poder generar predicciones sobre las bicis disponibles en tiempo real, y que estas predicciones estuvieran disponibles en todo momento del día. Por ello era necesario que el dataset se fuera generando automáticamente en cada momento con los datos actualizados.

Los datos del dataset se sacarían por tanto directamente de la página web de Valenbisi, y los metereológicos de la web de [OpenWeatherMap].

El primer obstáculo que hubo que solucionar fue poder acceder a los datos sobre la disponibilidad de las Valenbisi, ya que a partir de cierta cantidad de peticiones HTTP GET por minuto de su web realizadas desde la misma dirección IP, su sistema

te bloquea. Debido a esto, se decidió reducir la cantidad de las estaciones sobre las que se realizarían predicciones de 275 a 5 estaciones. De esta manera la cantidad de peticiones HTTP GET se redujo también considerablemente.

El otro obstáculo a solucionar fue decidir cómo montar el sistema para que los resultados de las predicciones pudieran ser accedidos desde cualquier cliente (Android en este caso). Se decidió optar por un servidor VPS con Windows Server instalado. Este servidor tendría un acceso remoto de forma que instalar cualquier programa necesario fuera bastante sencillo, como usar cualquier otro sistema operativo con interfaz de usuario.

El sistema entero está compuesto de tres partes. A continuación se explican cada una de las partes.

Primeramente un programa en Java que cada 3 minutos consulta las bicis disponibles en la página web de Valenbisi y la información meteorológica y escribe el resultado al final del fichero del dataset. Este programa en Java guarda en memoria los datos de las últimas 24 horas, de forma que puede componer el dataset añadiendo la información sobre las bicis disponibles 24 horas antes de la última ejecución.

En paralelo y también ejecutándose cada 3 minutos, un programa escrito en R toma como entrada los datos recogidos en la última fila del dataset y, usando cada modelo correspondiente a cada estación, calcula las predicciones sobre la disponibilidad de bicis 1 hora en el futuro. Este programa escribe los resultados al final de un fichero donde se van almacenando todas las predicciones realizadas.

Por último, otro programa en Java necesario para recopilar y guardar en un fichero únicamente los datos y resultados visibles para la aplicación Android. Este es un ejemplo del fichero de salida accesible para Android:

station 1: pred +1h future -> 0.342533313290133; now/pred now -> 0/0.126690607721344; 1h ago/pred -> 0/0.351411186321035; 2h ago/pred -> 0/0.237756294775329; 3h ago/pred -> 0/0.170496922966522

station 2: pred +1h future -> 0.176736101116044; now/pred now -> 0/0.143258498975852; 1h ago/pred -> 0/0.230030198916908; 2h ago/pred -> 0/-0.00904648567605482; 3h ago/pred -> 0/0.986867222432029

station 3: pred +1h future -> 3.6261153905011; now/pred now -> 4/4.71488179821163; 1h ago/pred -> 4/3.445664127035942; 2h ago/pred -> 0/-0.484211687045954; 3h ago/pred -> 0/1.65594965058947

station 4: pred +1h future -> 3.63906015398089; now/pred now -> 3/2.13902188700491; 1h ago/pred -> 2/1.89892356906841; 2h ago/pred -> 3/4.59145510051108; 3h ago/pred -> 6/4.38792220368142

station 5: pred +1h future -> 0.673708124306398; now/pred now -> 0/0.823947984877581; 1h ago/pred -> 0/0.515476780658323; 2h ago/pred -> 0/1.70799668552703; 3h ago/pred -> 1/0.751673004032568

Se decidió que debido a su facilidad de uso, los ficheros se guardarían en una carpeta pública en Dropbox, de tal forma que la información puede ser accedida desde una URL. De esta forma se eliminaba el trabajo extra que hubiera podido surgir en caso de acceder a los datos mediante una API Restful.

5.2. Aplicación en Android

Para finalizar el TFG, se quería que el sistema para predecir la disponibilidad futura de Valenbisis tuviera una manera de ser accesible fácilmente. Primeramente se pensó en realizar una interfaz web desde la que poder acceder a dichas predicciones en cualquier momento. Pero dado que los conocimientos de desarrollo web no eran tantos como los de desarrollo en Android, se decidió optar por la opción de desarrollar una app para Android. Otro de los beneficios es que la app podría ser accedida más fácilmente una vez descargada, además de poder añadir múltiples funcionalidades en el futuro, como por ejemplo notificaciones automáticas.

A continuación se pueden ver diversas capturas de pantalla de la app:

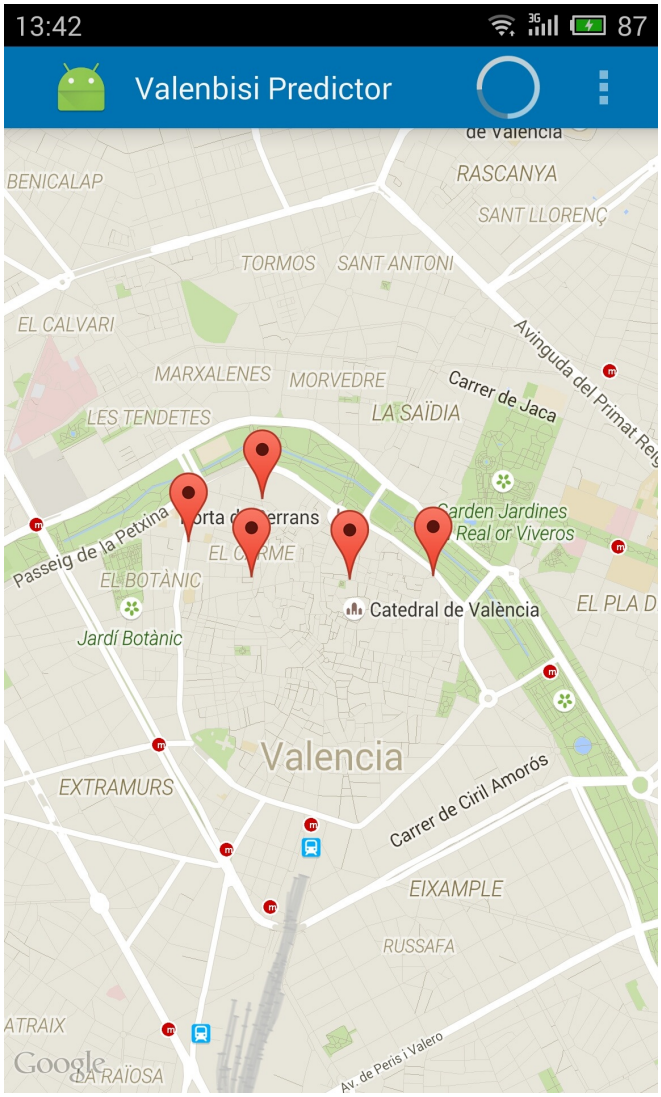

Figura 5.1.- Interfaz de usuario principal de la aplicación Android

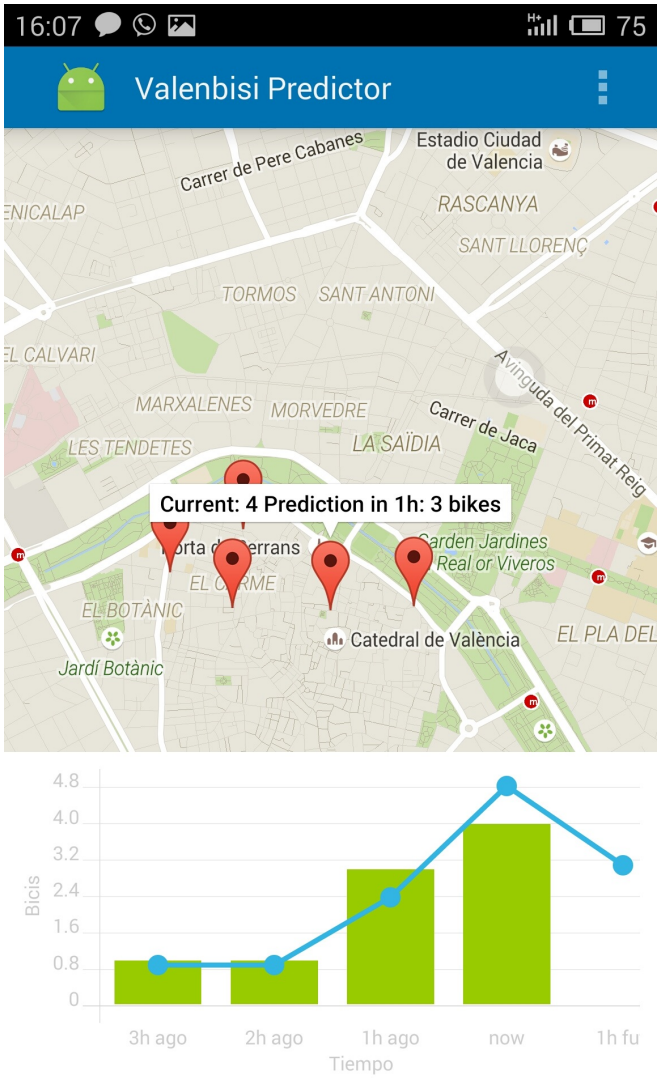

Figura 5.2.- Interfaz de usuario de la aplicación Android con gráfica

En la primera imagen se muestra la interfaz principal de la aplicación. Aquí se muestra un mapa que ocupa todo el espacio y en donde cada pin del mapa corresponde a una estación de Valenbisi.

Al iniciarse la app, esta consulta automáticamente el archivo anterior guardado en Dropbox con los resultados de las predicciones. Esto se realiza mediante una petición HTTP GET de la siguiente URL (https://dl.dropboxusercontent.com/u/26280031/TFG/android_final_data.txt) de forma asíncrona para no bloquear el hilo principal de la interfaz.

Mientras se descarga el fichero, se muestra la barra circular de progreso para indicar al usuario que los datos se están descargando. Una vez acabada la descarga, al pulsar sobre el marcador de cada estación aparece una gráfica en la que ver los resultados.

La gráfica está compuesta por barras verdes y lineas azules. Las barras son las bicis reales que había 3, 2 y 1 hora antes de las actuales, y la última barra son las bicis actuales. Las líneas azules con puntos corresponden a las previsiones que se habían realizado para cada una de las barras correspondientes. Además, el último punto azul corresponde con la predicción sobre las bicis disponibles que habrá dentro de una hora.

El trabajo fin de grado completo esta disponible en Dropbox en el siguiente enlace:
https://www.dropbox.com/sh/brzfmsekxxo9die/AAAsrSrG1PPTIQH6_TI1bs5Ma?dl=0

6. Conclusión

Como conclusión personal del trabajo fin de grado, considero que ha sido en general muy satisfactorio, ya que personalmente creo que he cumplido con los objetivos que me planteé. Principalmente quería hacer un trabajo fin de grado que no estuviera demasiado relacionado con aspectos de la informática y del software que ya dominara bastante, de forma que su realización necesitara de un mayor esfuerzo. He aprendido teoría, práctica y técnicas no solo sobre el machine learning y la minería de datos y su proceso, sino también sobre el Web scraping.

Aunque este trabajo fin de grado se ha centrado en el caso de estudio de la disponibilidad de bicicletas en Valenbisi, me ha aportado una visión más amplia sobre los diferentes ámbitos en donde se pueden aplicar las técnicas aprendidas en este trabajo fin de grado, sobre todo en el entorno empresarial.

7. Glosario

Android: sistema operativo inicialmente pensado para teléfonos móviles. Está basado en Linux, un núcleo de sistema operativo libre, gratuito y multiplataforma.

Big Data: sistemas informáticos basados en la acumulación a gran escala de datos y de los procedimientos usados para identificar patrones recurrentes dentro de esos datos.

CSV: Del inglés “comma-separated values”, son un tipo de documento en formato abierto sencillo para representar datos en forma de tabla, en las que las columnas se separan por comas y las filas por saltos de línea.

Dataset: colección de datos habitualmente tabulada. En su versión más simple, un conjunto de datos corresponde a los contenidos de una única tabla de base de datos o una única matriz de datos estadística.

Dirección IP: etiqueta numérica que identifica, de manera lógica y jerárquica, a un elemento de comunicación de un dispositivo dentro de una red que utilice el protocolo IP (Internet Protocol).

Dropbox: servicio de alojamiento de archivos multiplataforma en la nube. Permite a los usuarios almacenar y sincronizar archivos en línea y entre ordenadores y compartir archivos y carpetas con otros usuarios y con tabletas y móviles.

Selenium: entorno de pruebas de software para aplicaciones basadas en la web. Provee una herramienta de grabar/reproducir para crear pruebas sin usar un lenguaje de scripting para pruebas.

Valenbisi: servicio de alquiler de bicicletas públicas de la ciudad de Valencia implantado desde el 21 de Junio de 2010, promovido por el Ayuntamiento y gestionado por la empresa JCDecaux.

Valor discreto: variable que sólo puede tomar algunos valores dentro de un mínimo conjunto numerable, es decir, no acepta cualquier valor, sólo aquellos que pertenecen al conjunto.

VPS: Un servidor virtual privado (Virtual Private Server) es un método de particionar un servidor físico en varios servidores de tal forma que todo funcione como si se estuviese ejecutando en una única máquina. Cada servidor virtual es capaz de funcionar bajo su propio sistema operativo y además cada servidor puede ser reiniciado de forma independiente.

Web scraper: programas de software que extraen información de sitios web. Normalmente, estos programas simulan la navegación de un humano en la World Wide Web ya sea utilizando el protocolo HTTP manualmente, o incrustando un navegador en una aplicación.

8. Bibliografía

Alex Barros Biciv. URL http://biciv.com/

DeMaio, P. y Meddin, R. The bike-sharing world map, 2013. URL https://www.google.com/maps/d/viewer?hl=en&oe=UTF8&msa=0&ie=UTF8&mid=zGPlSU9zZvZw.kmqv_ul1MfkI.

Han, J.; Kamber, M. 2001. Data Mining: Concepts ans Techniques. Morgan Kaufmann Publishers, USA.

Juan Francisco Ortega Morán 2009 . Cálculo de modos y tiempos de desplazamiento en una ciudad usando fuentes públicas. URL http://e-archivo.uc3m.es/bitstream/handle/10016/5837/PFC;jsessionid=8F57813978554898381887942FB35DA6?sequence=1

A. Zimmermann, M. Kaytoue, C. Robardet, J. Boulicaut y M. Plantevit, 2015. Profiling users of the Vélo'v bike sharing system.

Página web de Freemeteo. URL http://freemeteo.es/eltiempo/valencia/historial/historial-diario/?gid=2509953&station=2899&language=spanish&country=spain

Selenium, Automatización de navegadores. URL http://www.seleniumhq.org/docs/

OpenWeatherMap. URL http://api.openweathermap.org/data/2.5/weather?id=2509954

https://es.wikipedia.org/wiki/Aprendizaje_autom%C3%A1tico

Maurizio Alejandro R.M., Inteligencia Empresarial combinando técnicas de Minería de Procesos y Minería de Datos, 2014.

Sinnexus Business Intelligence, Informática Estratégica. URL http://www.sinnexus.com/business_intelligence/datamining.aspx

Extracción de Conocimiento en BBDD. URL http://weblidi.info.unlp.edu.ar/catedras/MD_SI/01_Extraccion%20de%20conocimiento.pdf

Juan Francisco O.M., Cálculo de modos y tiempos de desplazamiento en una ciudad usando fuentes públicas, 2009

CleverTask. URL http://www.clevertask.com/conceptos-basicos-machine-learning/

Mr. Edwin Chen Blog. URL http://blog.echen.me/2011/04/27/choosing-a-machine-learning-classifier/