

**Autor:**

**ANTONIO BRAVO MUÑOZ**

**TRABAJO DE FIN DE MÁSTER**

**Máster en Data Science**

PREDICCIÓN DE LA CARGA DE VEHÍCULOS EN VÍAS URBANAS DE MADRID

**ÍNDICE DE CONTENIDOS**

1. INTRODUCCIÓN AL TRABAJO REALIZADO……………………………………………………………………………………………….
2. DESCRIPCIÓN DE LOS DATOS DE ENTRADA………………………………………………..
3. METODOLOGÍA………………………………………………………………………………………….
4. RESULTADOS…………………………………………………………………………………………….
5. CONCLUSIONES……………………………………………………………………………………….

ANEXO I. DASHBOARDS-GUÍA DE USUARIO………………………………………………………..

1. **INTRODUCCIÓN AL TRABAJO REALIZADO**

En este trabajo de fin de Máster se aborda el problema del tráfico en grandes ciudades como Madrid, concretamente el estudio y la predicción de la carga de tráfico en vías urbanas de esta cuidad.

El crecimiento progresivo de población en las grandes urbes desemboca en un mayor número de vehículos en sus calles, algo que conlleva de forma inevitable a un aumento en los niveles de tráfico. Este trabajo surge a partir de esta idea y de los problemas que conlleva a las personas que se ven forzadas a usar sus vehículos para ir a trabajar en lugar de hacer lo en transporte público.

La idea principal del trabajo es partir de un itinerario ya predefinido y fijo, por ejemplo la ruta que hacemos desde casa al trabajo cada día, y conseguir predecir la carga de vehículos que habrá en un momento dado. Esta idea se basa en la intención de optimizar el tiempo que tenemos. En muchos casos, si por la mañana salimos de casa 15 min más tarde, podemos experimentar un retraso en la llegada considerablemente mayor, de hasta 30 o 45 minutos debido al aumento del tráfico.

Para la realización de este proyecto se usan los datos de intensidad de tráfico que el Ayto. de Madrid (Dirección General de Gestión y Vigilancia de la Circulación) pone a disposición de la ciudadanía a través del siguiente [enlace](https://datos.madrid.es/portal/site/egob/menuitem.c05c1f754a33a9fbe4b2e4b284f1a5a0/?vgnextoid=33cb30c367e78410VgnVCM1000000b205a0aRCRD&vgnextchannel=374512b9ace9f310VgnVCM100000171f5a0aRCRD).

Las muestras de tráfico son recogidas por una red de Puntos de Medida que el Ayuntamiento ha desplegado por toda la cuidad. Es necesario usar los datos referentes a estos puntos, los cuales pueden encontrarse a través de este otro [enlace](https://datos.madrid.es/portal/site/egob/menuitem.c05c1f754a33a9fbe4b2e4b284f1a5a0/?vgnextoid=ee941ce6ba6d3410VgnVCM1000000b205a0aRCRD&vgnextchannel=374512b9ace9f310VgnVCM100000171f5a0aRCRD).

Se han encontrado trabajos relacionados con esta temática en Internet, los cuales han servido para sentar las bases del procedimiento a aplicar. Los más relevantes se citan a continuación.

1. *[Application of data mining techniques for traffic density estimation and prediction. Jithin Raja, Hareesh Bahuleyana, Lelitha Devi Vanajakshia. 11th Transportation Planning and Implementation Methodologies for Developing Countries, 10-12 December 2014, Mumbai, India.](https://cyberleninka.org/article/n/1462742/viewer)*
2. *[Deep Learning-Based Caution Area Traffic Prediction with Automatic Identification System Sensor Data. Kwang-Il Kim and Keon Myung Lee. Department of Computer Science, Chungbuk National University, Cheongju 28644, Korea.](https://www.mdpi.com/1424-8220/18/9/3172/htm)*

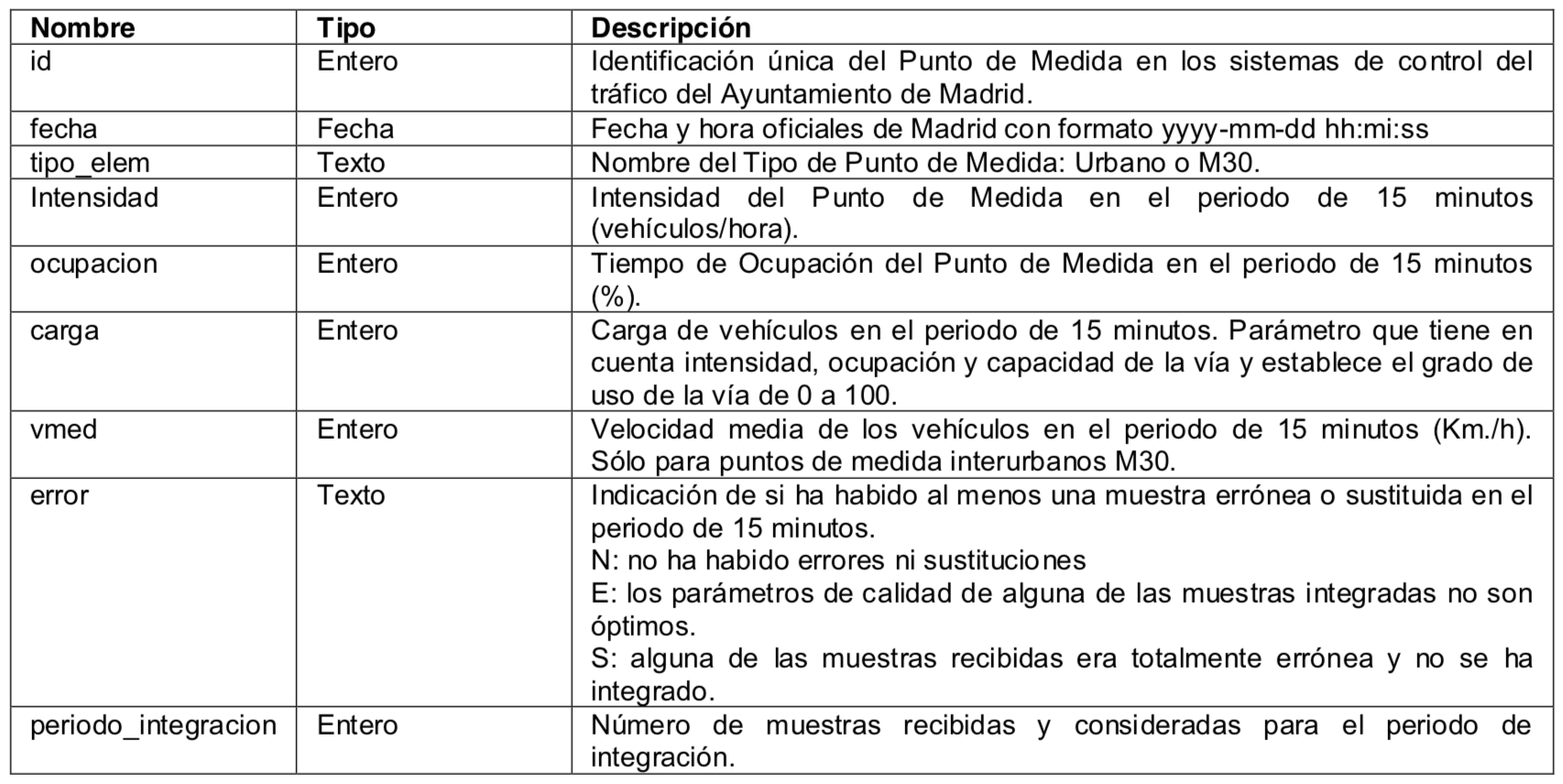
**2. DESCRIPCIÓN DE LOS DATOS DE ENTRADA**

Los datos usados en este trabajo se clasifican en 2 tipos:

* [Datos Históricos de Intensidad de tráfico desde Julio de 2013 (datos de los puntos de medida).](https://datos.madrid.es/FWProjects/egob/Catalogo/Transporte/Trafico/ficheros/Estructura_DS_Contenido_Trafico_Historico.pdf)
* [Ubicación de los puntos de medida de tráfico.](https://datos.madrid.es/FWProjects/egob/Catalogo/Transporte/Trafico/ficheros/InformacionUbicacionPuntosMedida.pdf)

Para el histórico de datos de la intensidad de tráfico, los datos vienen agrupados por meses desde 2013. Para cada mes se registran lecturas en periodos de 15 minutos de todos los sensores existentes. Los datos se presentan en formato CSV y están disponibles para descargar como archivos comprimidos ZIP.

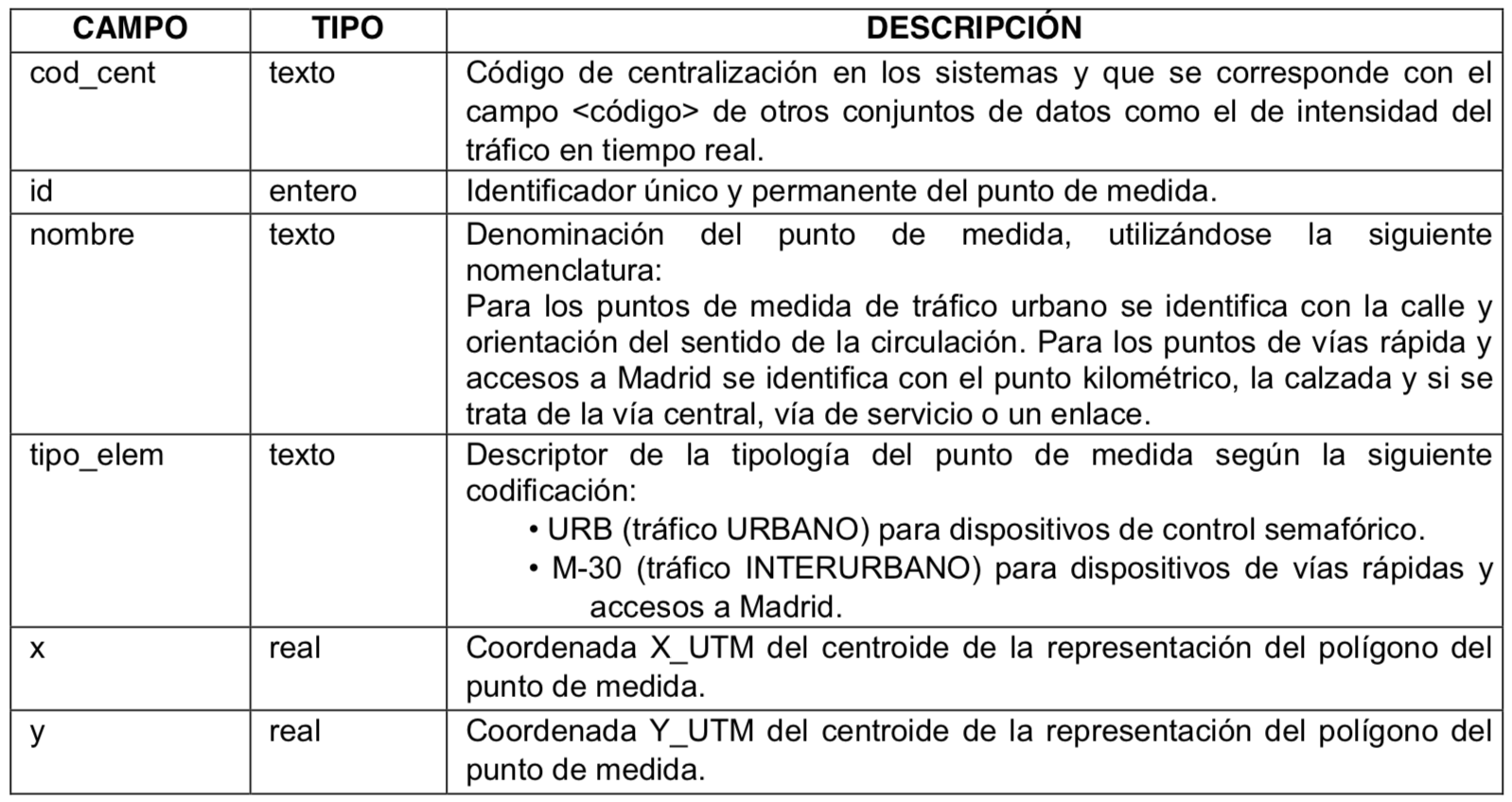
Los datos se estructuran de la siguiente forma:



Para este trabajo, se han usado datos correspondientes a los meses de Enero, Febrero, Marzo, Abril, Mayo, Junio, Julio, Agosto y Septiembre de 2018.

De la misma que los datos de Intensidad del tráfico, tenemos los datos relativos a cada punto de medida. En este caso, en el portal de datos existen varios ficheros, clasificados por meses, con la información relativa a los puntos de medida operativos a final de dicho mes. En nuestro caso, al contar con datos hasta Septiembre, sólo se ha usado el archivo de datos correspondiente. De esta forma nos aseguramos que los puntos de medida sobre los que trabajamos están operativos de acuerdo a la fecha de los datos con los que vamos a trabajar.

Para los puntos de medida, los datos proporcionados tienen la siguiente estructura:



De la misma forma que los datos de intensidad de tráfico, estos datos se presentan para la descarga en formato CSV.

1. **METODOLOGÍA**

Una vez obtenidos los datos con los que se va a trabajar, en este apartado se describe el tratamiento que se ha aplicado a estos. El proceso parte de los datos en crudo hasta la obtención de un modelo de predicción que sea capaz de estimar, con un cierto error, la carga de tráfico para un itinerario que vendrá dado con anterioridad.

**1. ADQUISICIÓN, PREPARACIÓN Y TRANSFORMACIÓN DE LOS DATOS**

**(**Notebooks: Data Acquisition and Preparation\_train.ipynb y Data Acquisition and Preparation\_test.ipynb)

***1.1. Formato de las variables.***

Transformaciones del tipo de variables para su correcto tratamiento.

***1.2. Obtención de la Latitud y la Longitud de cada sensor.***

En el fichero de información de cada punto de medida están registradas las coordenadas UTM X e Y de cada sensor. Mediante una función aplicada a estas dos columnas se obtienen la Latitud y la Longitud. Posteriormente se hace un cruce entre este dataframe

y el dataframe principal (datos de intensidad del tráfico) para incorporar con estas dos columnas.

***1.3. Missing Values****.*

Al realizar el cruce mencionado anteriormente hay algunos sensores de la dataframe principal cuyo identificador no aparece en el dataframe secundario, y como consecuencia se generan valores “Nan”. En este caso eliminaremos estos valores. Esta es una forma de asegurarnos que contamos únicamente con sensores que continúan operativos en el mes actual.

Existen un total de 201.965 registros con valores Nan en características Latitud y Longitud , un 0,22% del total de los datos.

***1.4. Homogenización de las características tras concatenar varios registros de varios meses.***

Los ficheros de datos de Enero no presentan la misma estructura de datos que los de Agosto, por ejemplo. Esto ocurre con la característica “Tipo\_elem”, la cual nos indica si el sensor es del tipo M30 o urbano. Para el fichero de Enero, en lugar de tomar el valor M30 toma “Puntos de Medida M-30”.

Con el fin de homogeneizar la información se aplica una función específica para conseguirlo.

***1.5. Tratamiento de registros con información errónea.***

Atendiendo a indicaciones de la fuente de datos, pueden existir registros cuyos valores sean negativos lo que implicaría la ausencia de datos. De la misma forma, se dice que si existen registros cuya velocidad media es 0 y existen datos para el resto de variables (intensidad, carga o ocupación) puede considerarse como un error en la toma de datos por algún fallo en el detector.

En este caso se suprimen estos registros del conjunto de datos.

***1.6. Outliers.***

Se procede a comprobar la existencia de outliers en las características numéricas (intensidad, ocupación, carga y vmed) mediante 2 aproximaciones: de forma visual; y de forma cuantitativa a través del parámetro Z-Score. Se eliminarán los registros cuyo Z-score se sitúe fuera de un intervalo determinado.

***1.7. Codificación de variables categóricas.***

Existen dos variables categóricas en el conjunto de datos: tipo\_elem y error. Pudiendo tomar 2 valores diferentes la primera y 3 la segunda. Para que los algoritmos de ML puedan interpretarlas, se procede a codificarlas mediante la técnica One-Hot Encoding.

***1.8. Adecuación de los datos de entrada a algoritmos de ML.***

Para proporcionar al modelo de predicción un archivo limpio y optimizado se transforman los tipos de algunas variables. Este es el caso de la variable “Date”, de tipo “datetime”, que es dividida en campos día, mes, año, hora, minutos y segundos todos ellos de tipo entero.

Revisando los datos tratados nos damos cuenta que se pueden eliminar ciertos campos que no serían representativos para el modelo ML por ser constates, es el caso de “segundos”, “error” y “año”. De esta forma, el conjunto de datos ocupará menos espacio en memoria, agilizando el funcionamiento del algoritmo y conteniendo la misma información.

***1.9. Exportando los datos procesados.***

Finalmente se exporta el dataframe procesado y adecuado para que el algoritmo trabaje con el desde otro notebook. También se guarda una versión de este antes del paso 1.8, la cual contendrá mas información que puede ser de utilidad en el proceso de visualización.

**2. PREDICCIÓN CON ALGORITMOS DE ML.**

**(**Notebooks: Data Modeling and Predictions-full dataset.ipynby Data Modeling Colab-full dataset.ipynb)

***2.2. Algoritmos Machine Learning.***

En una primera aproximación se barajó la posibilidad de usar dos tipos de modelos predictivos:

- Predictores basados en Árboles de Decisión.

- Perceptrón Multicapa (Deep-Learning).

Finalmente y tras no disponer de los recursos computacionales ni del tiempo requerido para poder usar Deep Learning, se dejó esta opción a un lado y nos centramos en el uso de modelos basados en Árboles de Decisión.

Se ha procedido a comparar distintos algoritmos basados en Árboles de Decisión: Decisson Tree Regressor, Random Forest Regressor y XGBoost Regressor.

Evaluando cada uno de ellos buscando los hiperparámetros que producen un resultado óptimo en cada caso, se ha elegido Random Forest como el algoritmo que mejor resultado ha mostrado.

***2.3. Train.***

Antes de realizar el entrenamiento del modelo, se han escogido los hiperparámetros que minimizan el error con la ayuda del paquete “gridsearchcv” de Scikit-Learn. Una vez determinados, se produce a entrenar el modelo con los datos de entrada y los parámetros definidos.

El conjunto de datos de train se compone de los registros correspondientes a los meses desde Enero hasta Agosto de 2018, estos datos contemplan información de toda la red de sensores.

En este apartado hay que mencionar que debido al elevado número de registros del dataset, ha sido imposible entrenar el modelo cuando este se basa en algoritmos como RF y XGBoost, contemplando incluso periodos superiores a 2 días incluso usando “grids” con un número de parámetros reducido. Para estos caso ha sido necesario realizar el entrenamiento en un entorno virtualizado con Google Colab, donde las capacidades de procesamiento son algo superiores. Tras el entrenamiento en la nube, se exporta el modelo entrenado para evaluarlo en local a través del paquete “pickle”.

***2.4. Test.***

Una vez el modelo ha sido entrenado se realiza una predicción de la carga sobre los datos de Septiembre de 2018, concretamente sobre la primera semana de este mes y en horario de 7:00 AM. Estos datos no han sido usados en el proceso de entrenamiento, por lo que el modelo no los ha contemplado en ningún momento, suponiendo así un caso real e inmejorable para evaluar el modelo.

Cabe mencionar que para realizar el test, sólo se tienen en cuenta los sensores situados dentro del itinerario que hayamos establecido. De esta forma obtendremos el error de test asociado nuestro modelo.

**3. FRONT-END. DASHBOARDS INTERACTIVOS.**

**(**Notebooks Data Visualization.ipynb, Measurement Points Location Visualization.ipynb y predictions\_visualizations.twb)

Para transformar la información contenida en los datos en información útil e interpretable para el usuario final, se elaboran principalmente 3 dashboards interactivos usando los paquetes de python “folium” y “altair” y un dashboard interactivo en Tableau accesible a través del siguiente enlace ([tableau\_dashboard](https://public.tableau.com/views/CargaPredictions_Dashboard/Dashboard1?:embed=y&:display_count=yes&publish=yes))). Su contenido es el siguiente

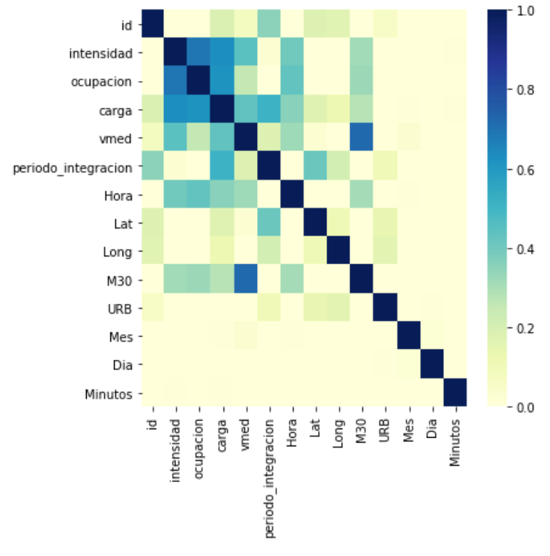
* Mapa interactivo con información de la red completa de sensores. (hecho en Folium)
* Panel interactivo con gráficos que proporcionan información relativa a al tráfico en el itinerario seleccionado para los meses desde Enero hasta Agosto. (Elaborado con Altair)
* Dashboard interactivo en Tableau con información sobre la predicción de carga en la ruta seleccionada para la primera semana de Septiembre de 2018. Adicionalmente se muestra el histograma del valor de carga y un gráfico en el que se compara el valor predicho con el valor real.

El funcionamiento e interpretación de la información en estos archivos se detalla en el apartado “ANEXO I. Dashboards-Guía de usuario”.

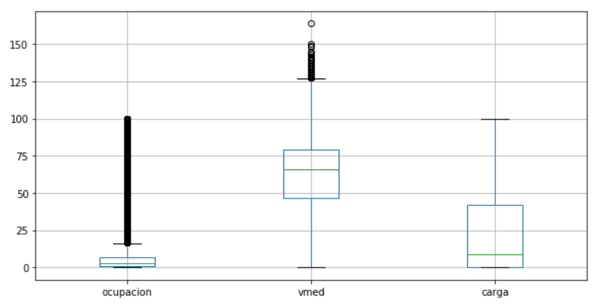
1. **RESULTADOS**

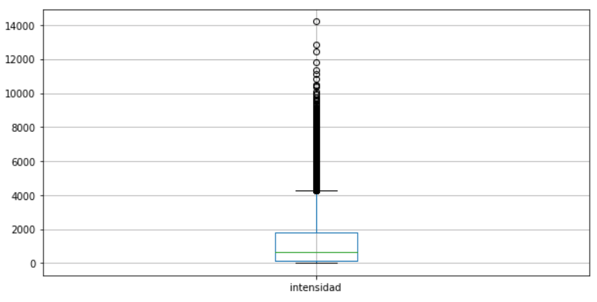
Se han obtenido los siguientes resultados tras la realización de este trabajo

* Se ha podido observar la relación entre las distintas variables. Constatamos que la relación más fuerte está entre Intensidad, Ocupación, Carga y Vmed.

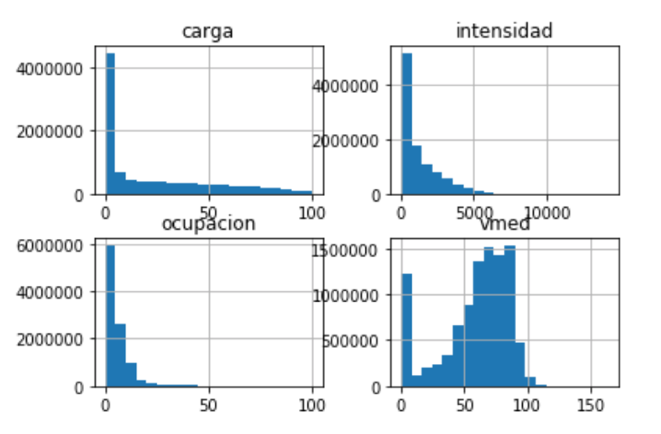
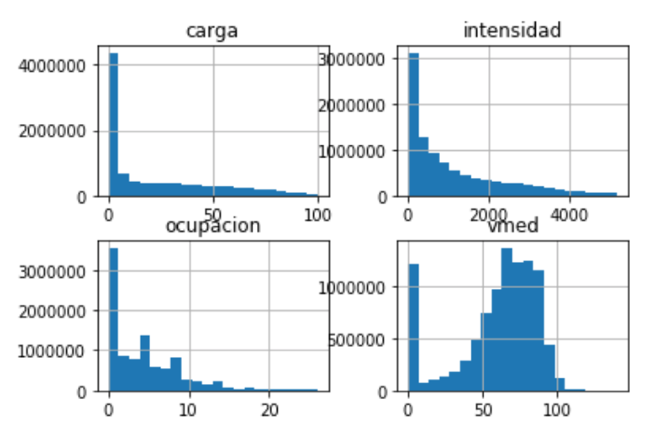


* Tras analizar y procesar los datos en crudo, se detectan algunos outliers en estas cuatro variables. Aunque al usar algoritmos basados en árboles de decisión se supone que los outiers no deberían tener mucha influencia en el resultado del modelo, se decide eliminarlos del conjunto de datos.





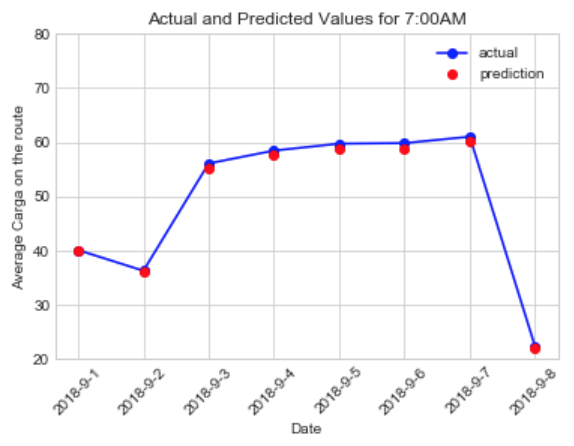
Observamos las distribuciones antes (izquierda) y después (derecha), apreciando como las colas se reducen.

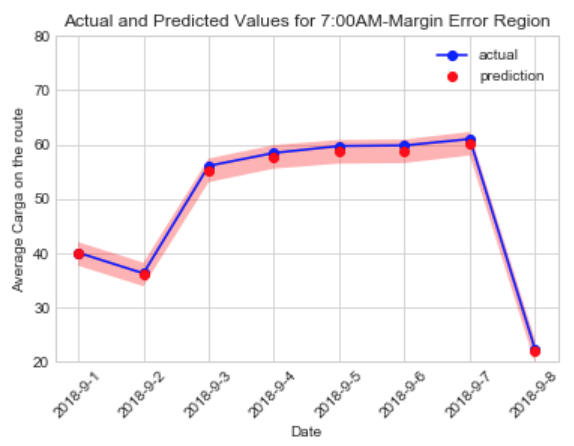


* Se ha usado un modelo **Random Forest Regressor**, entrenándolo con los parámetros óptimos (obtenidos con gridsearchcv):
  + *N\_estimators:100*
  + *Min\_sample\_leaf: 20*
  + *Max\_depth: 29*

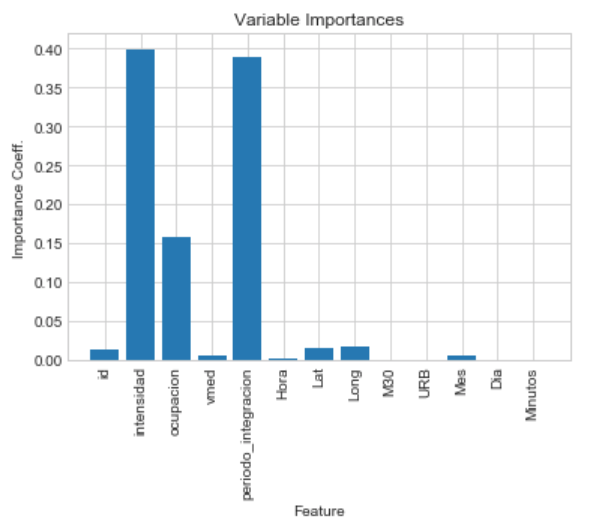
Produciendo el siguiente error:

* En lo que respecta al test se obtiene un coeficiente de determinación () de 0.97, un error medio absoluto (MAE) de 2.17 y un *Accuracy* del 94.738 %.





* Las variables más representativas en el resultado del modelo son intensidad, periodo de integración y ocupación.



1. **CONCLUSIONES**

El presente proyecto se ha trabajado en el proceso que abarca desde el la obtención de los datos de tráfico en crudo hasta la obtención de un resultado en forma de predicción de la variable objetivo “carga”, pasando por todas las fases intermedias correspondientes para asegurar su completa consecución.

Se han estudiado y elaborado los procedimientos que un trabajo de Data Science debe contener, desde el procesamiento y adecuación de los datos, valoración de los modelos/algoritmos a usar y su correspondiente evaluación para trabajar con aquel que mejor resultados, y finalmente la interpretación de los resultados obtenidos.

Tras la realización del trabajo y a la vista de los resultados obtenidos, se puede afirmar que el procedimiento cumple con creces la función para la que fue pensado y diseñado, ofreciendo un resultado que cumple con las expectativas esperadas.

Durante su realización se han encontrado numerosos problemas que se han conseguido resolver, por ejemplo con la falta de capacidad computacional para la fase de entrenamiento, para la cual se ha usado la plataforma virtual Google Colab.

Como puntos de interés se observa como existen algunas variables más importantes que otras y que consecuentemente influyen de una forma más relevante en el resultado final.

Conclusiones sobre el trabajo, problemas encontrados, puntos de interés…

Sugerencias de mejora, mejoras para futuros trabajos, incluir más datos de otros años para contemplar los posibles acontecimientos que puedan ser un patrón en los años.

Comentar como se observan que los fines de semana disminuye el tráfico, también en navidad(reyes) y en semana santa, también para San Isidro.

* Porqué arboles de decisión…. Y por qué RF y XGboost
* Poner diagramas donde
* Problemas encontrados con la capacidad computacional.

**ANEXO I. Dashboards-Guía de usuario**

* Poner capturas de los 3 dashboards