PREDICCIÓN DE LA CARGA DE VEHÍCULOS EN VÍAS URBANAS DE MADRID

TRABAJO DE FIN DE MÁSTER
Máster en Data Science

Autor: Antonio Bravo Muñoz
ANTONIO BRAVO MUÑOZ

KSCHOOL

ÍNDICE DE CONTENIDOS

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 conduce inevitablemente a un incremento 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 hacerlo 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 surge con la intención de optimizar el tiempo del que disponemos. 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.

Las muestras de tráfico son recogidas por una red de puntos de medida que el Ayuntamiento tiene desplegada por toda la cuidad. La información y datos referentes a estos puntos, puede encontrarse a través de este otro enlace.

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.*

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.*

2. DESCRIPCIÓN DE LOS DATOS DE ENTRADA

Los datos usados se clasifican en 2 tipos:

- Datos Históricos de Intensidad de tráfico desde Julio de 2013 (datos de los puntos de medida).

- Ubicación de los puntos de medida de tráfico.

Para el histórico de datos de la intensidad de tráfico, los registros 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 la descarga como archivos comprimidos.

La estructura de cada archivo es la siguiente:

Nombre	Tipo	Descripción
id	Entero	Identificación única del Punto de Medida en los sistemas de control del tráfico del Ayuntamiento de Madrid.
fecha	Fecha	Fecha y hora oficiales de Madrid con formato yyyy-mm-dd hh:mi:ss
tipo_elem	Texto	Nombre del Tipo de Punto de Medida: Urbano o M30.
Intensidad	Entero	Intensidad del Punto de Medida en el periodo de 15 minutos (vehículos/hora).
ocupacion	Entero	Tiempo de Ocupación del Punto de Medida en el periodo de 15 minutos (%).
carga	Entero	Carga de vehículos en el periodo de 15 minutos. Parámetro que tiene en cuenta intensidad, ocupación y capacidad de la vía y establece el grado de uso de la vía de 0 a 100.
vmed	Entero	Velocidad media de los vehículos en el periodo de 15 minutos (Km./h). Sólo para puntos de medida interurbanos M30.
error	Texto	Indicación de si ha habido al menos una muestra errónea o sustituida en el periodo de 15 minutos. N: no ha habido errores ni sustituciones E: los parámetros de calidad de alguna de las muestras integradas no son óptimos. S: alguna de las muestras recibidas era totalmente errónea y no se ha integrado.
periodo_integracion	Entero	Número de muestras recibidas y consideradas para el periodo de integración.

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

De la misma forma que tenemos datos de Intensidad del tráfico, disponemos también de datos relativos a cada punto de medida. En este caso en el portal de datos existen varios ficheros, clasificados por meses, con 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 se va a trabajar están operativos.

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

CAMPO	TIPO	DESCRIPCIÓN
cod_cent	texto	Código de centralización en los sistemas y que se corresponde con el campo <código> de otros conjuntos de datos como el de intensidad del tráfico en tiempo real.
id	entero	Identificador único y permanente del punto de medida.
nombre	texto	Denominación del punto de medida, utilizándose la siguiente nomenclatura: Para los puntos de medida de tráfico urbano se identifica con la calle y orientación del sentido de la circulación. Para los puntos de vías rápida y accesos a Madrid se identifica con el punto kilométrico, la calzada y si se trata de la vía central, vía de servicio o un enlace.
tipo_elem	texto	Descriptor de la tipología del punto de medida según la siguiente codificación: • URB (tráfico URBANO) para dispositivos de control semafórico. • M-30 (tráfico INTERURBANO) para dispositivos de vías rápidas y accesos a Madrid.
x	real	Coordenada X_UTM del centroide de la representación del polígono del punto de medida.
y	real	Coordenada Y_UTM del centroide de la representación del polígono del punto de medida.

De la misma forma que los datos de intensidad de tráfico, estos datos se presentan públicamente para su descarga.

3. METODOLOGÍA

Una vez obtenidos los datos, en este apartado se describe el tratamiento que se aplica a estos. El proceso parte de los datos en crudo (*raw data*) hasta la obtención de un modelo de predicción que sea capaz de estimar, con un cierto error, la carga de tráfico para un itinerario definido y un horario dado.

Seguidamente, se describen las 3 etapas en las que se ha dividido el trabajo.

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 geográficas 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, se generan valores nulos. Existen algunos sensores del dataframe principal cuyo identificador no aparece en el dataframe secundario. En este caso eliminaremos estos valores para asegurarnos que contamos únicamente con sensores que continúan operativos en el mes actual.

Existen un total de 201.965 registros con valores nulos en *latitud* y *longitud*, un 0,22% del total de los datos. Vemos que se pueden eliminar sin ocasionar muchos problemas, ya que constituyen una proporción muy baja respecto al universo total de registros.

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. 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 dos valores diferentes la primera y tres la segunda. Para que los algoritmos de ML puedan interpretarlas estas características, se procede a codificarlas a través de *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 conteniendo la misma información.

1.9. Exportando los datos procesados.

Finalmente se exporta el dataframe procesado para que el algoritmo trabaje con él desde otro notebook. También se guarda una versión de este antes del paso 1.8, la cual contendrá información adecuada 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.
- Perceptrones 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 han comparado los siguientes 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, finalmente se ha usado *Random Forest*.

2.3. Train.

Antes de realizar el entrenamiento del modelo, se han escogido los hiperparámetros que minimizan el error usando "*gridsearchcv*" de *Scikit-Learn*. Una vez determinados estos, se procede 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 *Random Forest* o *XGBoost*, requiriendo incluso periodos superiores a 2 días para completar el proceso, 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 superiores. Tras el entrenamiento en la nube, se exporta el modelo entrenado para evaluarlo en local a través del paquete "*joblib*".

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 para la primera semana de este mes 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 la predicción, 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.

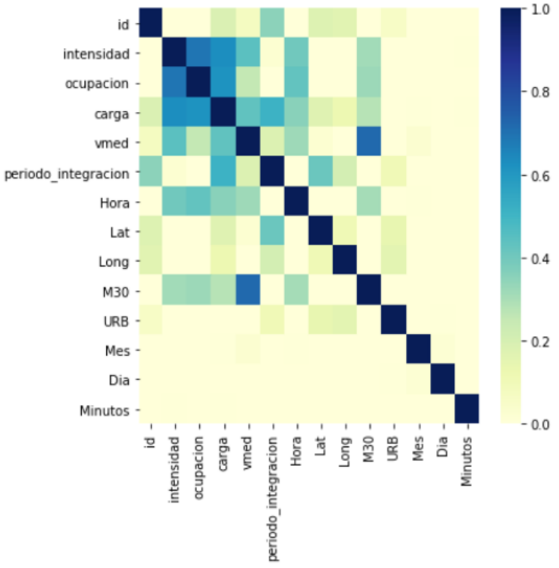
3. FRONT-END. DASHBOARDS INTERACTIVOS.
(*Notebooks: Data Visualization.ipynb, Measurement Points Location Visualization.ipynb y predictions_visualizations.twb*)

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 por otro lado *Tableau*. El funcionamiento e interpretación de la información en estos archivos se detalla en el apartado "*ANEXO I. Dashboards-Guía de usuario*".

4. RESULTADOS

Una vez realizado el proceso anterior se han obtenido los siguientes resultados:

- Se ha podido observar la relación entre las distintas variables. Definiendo, como era de esperar que la relación más fuerte está entre *Intensidad*, *Ocupación*, *Carga* y *Vmed*.



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

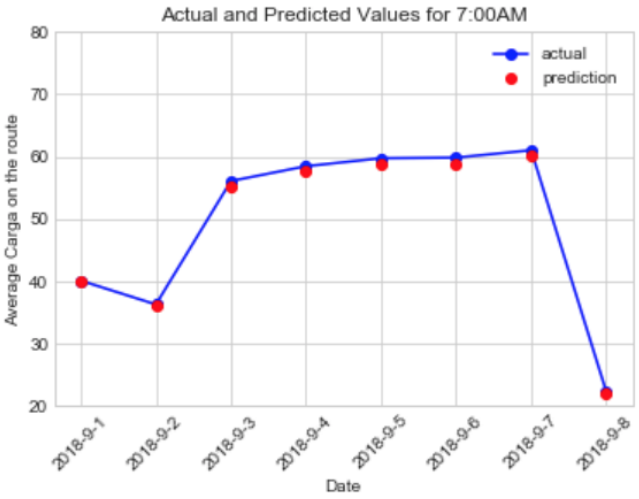
- En el proceso de ***train*** se ha usado el modelo ***Random Forest Regressor***, entrenándolo con los siguientes parámetros óptimos (obtenidos con *gridsearchcv*):

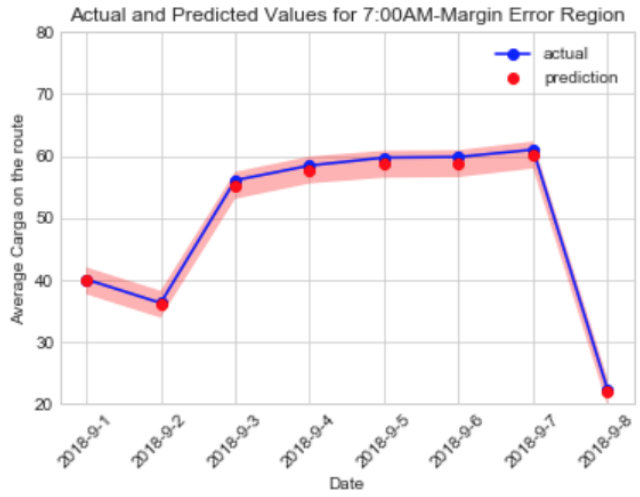
    - *N_estimators:100*
    - *Min_sample_leaf: 20*
    - *Max_depth: 29*

  Obteniendo el siguiente error:

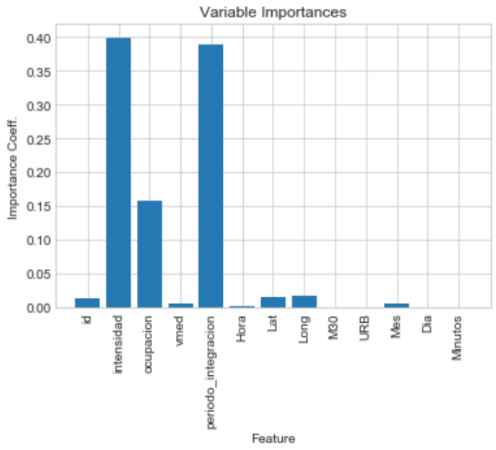
$$\boldsymbol{MSE = 3.979}$$
$$\boldsymbol{sqrt(MSE) = 1.994}$$

- En lo que respecta al ***test***, se obtiene un coeficiente de determinación (R^2) de 0.97, un error medio absoluto (*MAE*) de 2.17 y un *Accuracy* del 94.74 %.





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



- Tanto para la visualización de los datos usados como la de los datos generados (predicción) se elaboran 3 dashboards interactivos. Su contenido es el siguiente:

    - Mapa con información de la localización red completa de sensores, distinguiendo a través de su color si se ubican en la M30 (rojo) o en zonas urbanas (azul). Ruta del archivo: *(Figures_and_dashboards/Final_MeasurementPoints_involved_location.html)*

    - Panel interactivo con 4 gráficos que proporcionan información relativa al tráfico en el itinerario seleccionado para los meses desde Enero hasta Agosto. Concretamente se puede observar información de la carga para cada sensor (sus valores e histograma), la carga media de cada sensor en cada mes (Enero hasta Agosto) y una relación entre la velocidad media y la ocupación de la vía, donde se observa cómo ambas variables presentan una relación inversamente proporcional. Ruta del archivo: *(Figures_and_dashboards/Interactive_Dashboard.html)*

    - Dashboard interactivo en *Tableau* con información sobre la predicción de carga en la ruta seleccionada para la primera semana de Septiembre de 2018 a las 7:00AM (pudiendo filtrar por día). Adicionalmente se muestra el histograma de carga y un gráfico en el que se compara el valor predicho con el valor real donde se puede observar la precisión del valor predicho. Este archivo se encuentra accesible a través del siguiente enlace en *Tableau Public*: *tableau_dashboard*

5. 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 una serie de fases intermedias para asegurar su completa y correcta 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, evaluaciones de estos para trabajar con aquel que mejor resultados proporcione y finalmente la interpretación de los resultados obtenidos.

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

Durante la elaboración, se han encontrado algunos problemas que, finalmente, se han conseguido solucionar. Por ejemplo, la falta de capacidad computacional para la fase de entrenamiento se solventa con la migración de este proceso a la plataforma virtual *Google Colab*. Posteriormente se descarga el modelo en local para poder evaluarlo en la fase de test.

Como líneas de mejora cabe mencionar que si se hubiera entrenado adicionalmente con datos de meses del pasado posiblemente el algoritmo ofrecería mejores resultados al poder contemplar sucesos típicos que ocurren anualmente, como puentes, periodos donde la gente suele cogerse vacaciones, etc. Igualmente, si se asume el crecimiento en complejidad del proceso, se podría entrenar cada sensor con datos de otros que se encuentren alejados para eliminar la posible correlación espacial que pudiera existir. Incluso se podrían probar otros modelos como *XGBoost* y *Redes Neuronales*, a priori, más potentes y que podrían ofrecer unos mejores resultados.

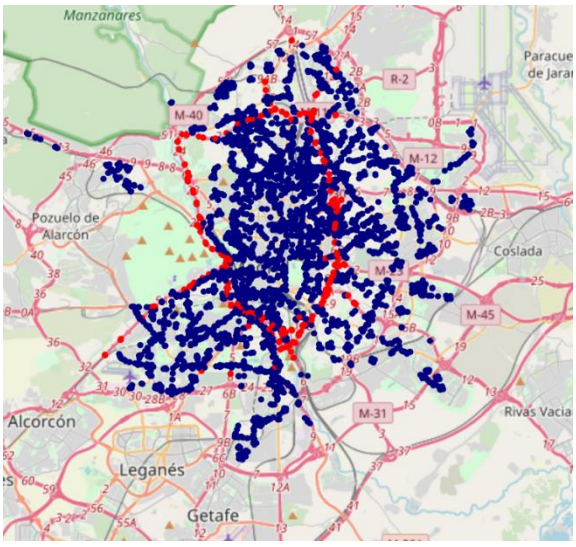
Finalmente, como último apunte y con el propósito de mantener el modelo con un funcionamiento óptimo, sería aconsejable ir re-entrenándolo con nuevos datos conforme pase el tiempo.

ANEXO I. Dashboards-Guía de usuario

En este anexo se detalla el funcionamiento del *front-end* de cara al usuario con el fin de que este entienda su funcionamiento y saque el mayor partido de la información recogida en las visualizaciones.

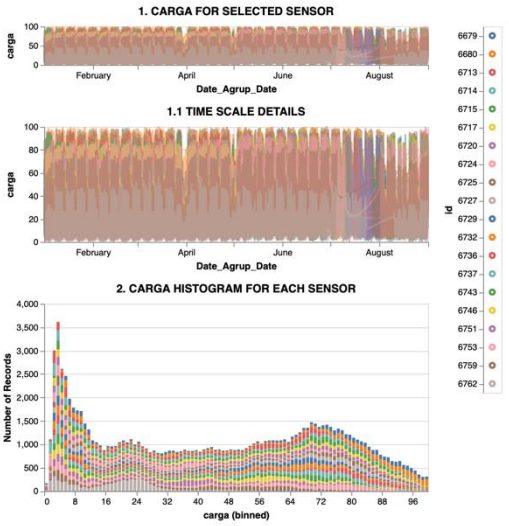
1. **Final_MeasurementPoints_involved_location.**

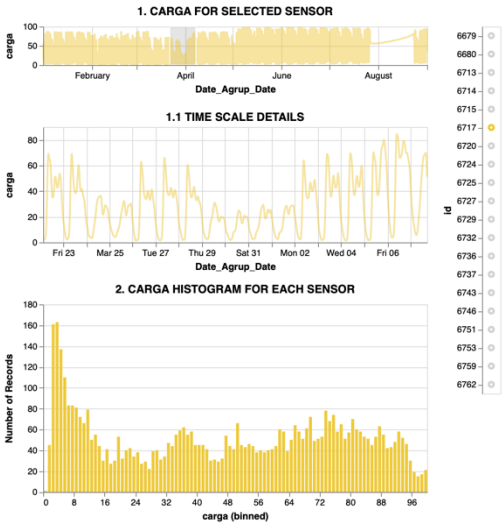
   Se trata de un mapa interactivo desde el cual se puede observar la ubicación en el espacio de cada detector. Contiene información de si el sensor es tipo M30 o URB según el color con el que se codifica. Su funcionalidad se reduce a que si pinchamos encima de cada punto, se muestra su ID, teniendo la posibilidad de hacer zoom sobre el área de interés.



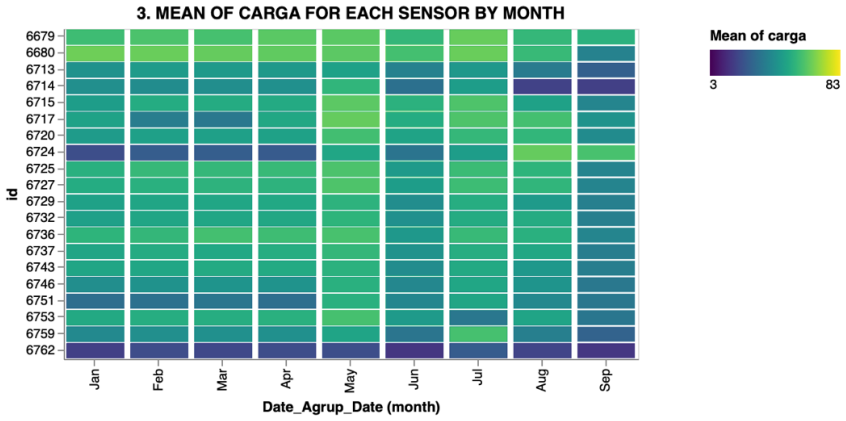
2. **Interactive_Dashboard.html**

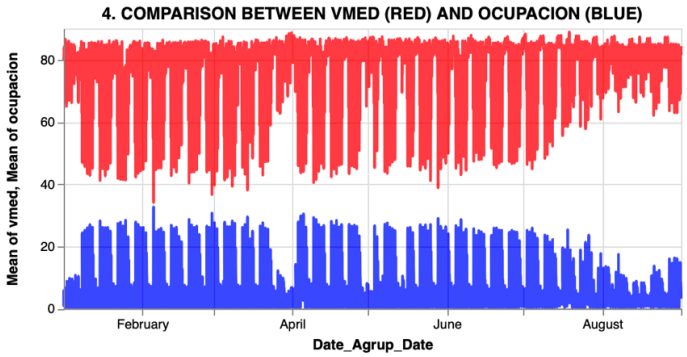
   Este panel muestra información de 4 gráficos bien diferenciados. En el gráfico 1 y 2 se puede seleccionar sobre qué detector se quiere obtener información de *carga*. En el panel "leyenda" del margen izquierdo, si se selecciona el sensor por ID pinchando encima del color correspondiente, se obtendrá información individualizada de cada uno de ellos, pudiendo hacer zoom temporal sobre el gráfico superior para observar con mayor detalle el valor de la *carga* en el gráfico inferior 1.1.





En los gráficos 3 y 4 puede observarse la carga media de cada sensor en cada mes desde Enero hasta Agosto y una relación entre la velocidad media y la ocupación de la vía respectivamente.





3. tableau_dashboard

En este dashboard interactivo de *Tableau* se presenta información sobre la predicción de carga en la ruta seleccionada para la primera semana de Septiembre de 2018 en horario de 7:00AM, pudiendo filtrar por día que el usuario desee. Adicionalmente se muestra el histograma de *carga* y un gráfico en el que se compara el valor predicho con el valor real donde se puede observar la precisión del valor predicho. Si se selecciona un día en concreto, los 3 gráficos se ven afectados por este filtro, mostrando la información correspondiente en cada caso.