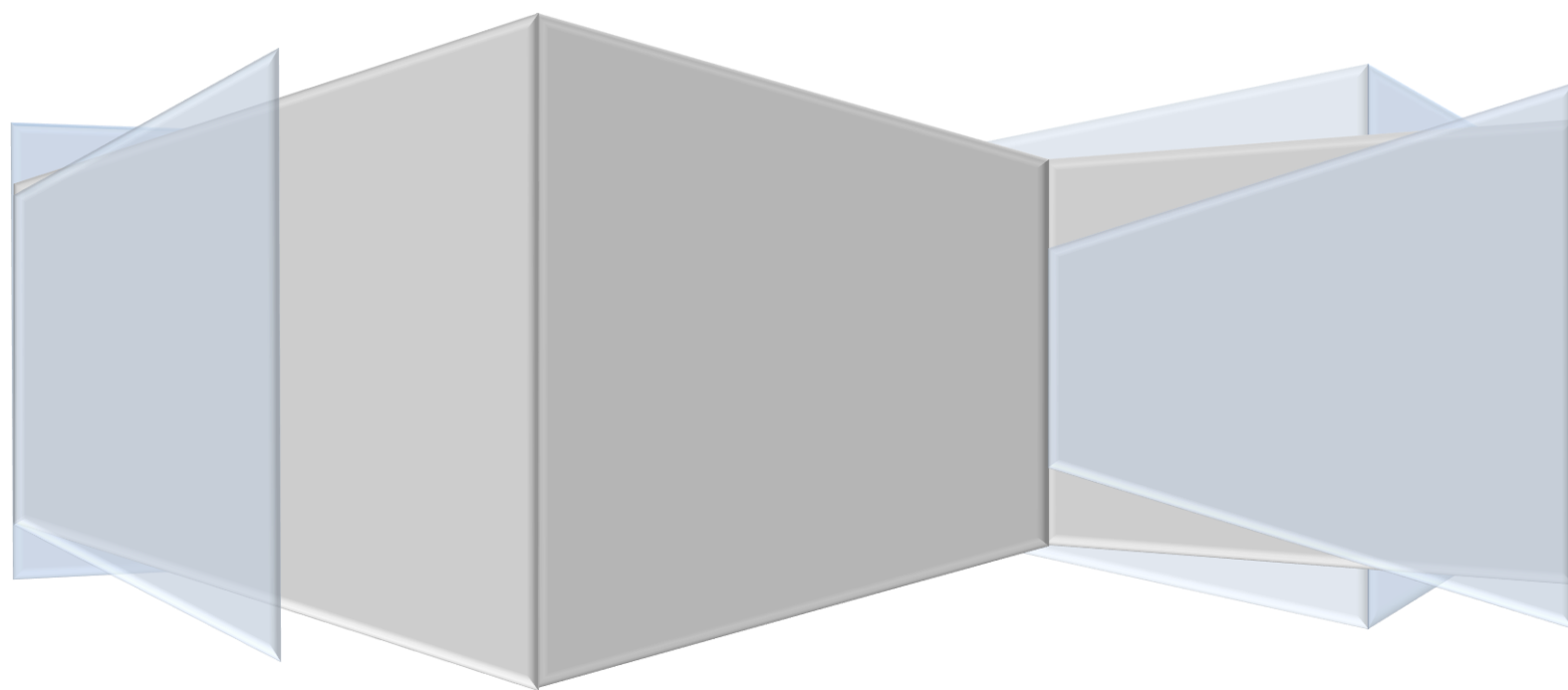


TRANSFORMACIÓN DIGITAL



Optimización de tiempos de carga para BOT's

Borja Rioja Mata



Contenido

EDA (Exploratory Data Analysis).....	2
Consideraciones previas.....	2
Herramientas	2
Acotamiento de datos	2
Objetivo	3
Análisis temporal de carga de trabajo / BOT's activos.....	4
Correlación lineal entre carga de trabajo y número de BOT's activos.....	8
Nº de BOT's – Estaciones de carga.....	9
Tiempo medio de carga de un BOT en un puesto de carga	9
Descarga de batería de un BOT.	11
Frecuencia de nivel de carga de un BOT en la entrada a la fuente de carga.....	13
Conclusiones análisis EDA	15
Selección y aplicación de algoritmos. Predicción y optimización.....	15
Algoritmos predictivos	15
Generación del Dataset necesario y selección de variables del modelo.....	16
Algoritmos basados en árboles.....	16
Redes Neuronales	17
Algoritmos de optimización.....	18
Conclusiones.....	20
Consideraciones futuras.....	20

EDA (Exploratory Data Analysis)

Consideraciones previas

El trabajo realizado para el análisis exploratorio de datos se ha realizado mediante las siguientes fases:

- 1) Análisis y compresión de la BBDD.
- 2) Elaboración de consultas SQL para la extracción de los datos necesarios para el análisis en formato de ficheros CSV. Se proporciona con la solución del trabajo tanto las consultas SQL dentro de la carpeta “.. \TR ASTI\EDA\SQL” así como de los ficheros en la carpeta “.. \TR ASTI\EDA\Ficheros”.
- 3) Elaboración de script en lenguaje Python para realizar un análisis visual de los datos. El script de Python puede encontrarse en la siguiente ruta: “.. \TR ASTI\EDA\Python”.

Herramientas

Las herramientas utilizadas para realizar el análisis exploratorio han sido:

- SQL Server.
- Spyder.
- Python (Librerías Panda, Matplotlib y Seagate).
- Ms Excel.

Acotamiento de datos

Mediante el estudio previo y compresión de las tablas de la BBDD proporcionada se han detectado una serie de registros en los que la calidad del dato no se ha considerado suficiente y que por tanto la inclusión de los mismos podría afectar de cara a la compresión, elaboración de conclusiones y creación del propio modelo predictivo. Es por ello que se ha tomado la decisión de acotar todo el trabajo analítico al último año que respalda esta copia de BBDD, es decir, el periodo comprendido entre el 01/10/2017 y el 30/09/2018.

Se entiende que debido a los propios cambios funcionales y evolución que ha tenido la BBDD durante el año 2017 en la parametrización, fusión de fases y códigos de eventos, como así se refleja en el documento de “*especificaciones de diseño – baterías de litio*”, pueden ser el motivo de la actividad que reflejaban estos registros.

Por ejemplo, se encontraban en las tablas SISTEMA_BMS_BIBERON_ENTRAR, SISTEMA_BMS_BIBERON_SALIR Y SISTEMAS_BMS_ORDENES, registros con datos no informados

en las primeras fases de cargas de datos en las tablas y que podían afectar a la elaboración del trabajo.

```
Select * from SISTEMA_BMS_BIBERON_SALIR order by fecha;  
Select * from SISTEMA_BMS_BIBERON_ENTRAR order by fecha;
```


100 %									
Results Messages									
	IDREGISTRO	FECHA	VEHICULO	INDICE	ESTACION	TEMPERATURA	SOC	TENSION	CONSUMO
1	1	2016-11-15 12:00:37.963	0	0	0	0	0	0	10
2	2	2016-11-15 12:00:37.963	0	0	0	0	0	0	10
3	3	2016-11-15 12:00:37.963	0	0	0	0	0	0	10
4	4	2016-11-15 12:00:37.963	0	0	0	0	0	0	10
5	5	2016-11-15 12:00:37.963	0	0	0	0	0	0	10
6	6	2016-11-15 12:00:37.963	0	0	0	0	0	0	10
7	7	2016-11-15 12:00:37.963	0	0	0	0	0	0	10
8	8	2016-11-15 12:00:37.963	0	0	0	0	0	0	10
9	9	2016-11-15 12:00:37.963	0	0	0	0	0	0	10
10	10	2016-11-15 12:00:37.963	0	0	0	0	0	0	10

Objetivo

El objetivo principal del trabajo es predecir el momento óptimo en el que un BOT debe ir a cargar batería atendiendo a la carga de trabajo actual, el número de BOT's activos, así como sus niveles de carga de batería.

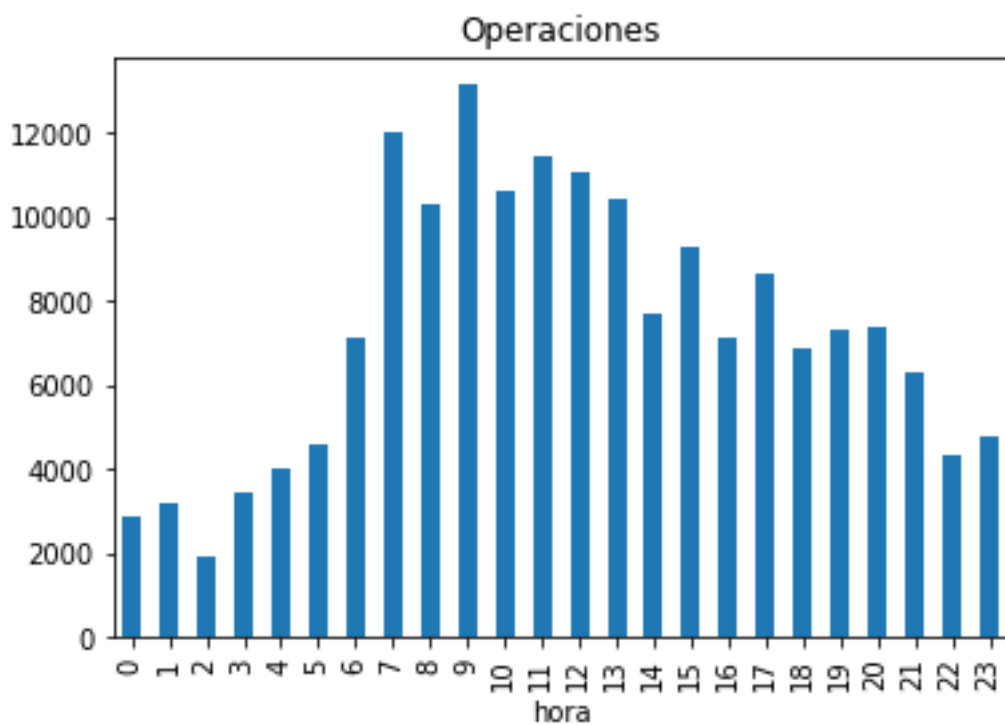
Se entiende por "óptimo" el número de BOT's activos necesarios que puedan cubrir una demanda de trabajo en momento determinado, por lo que será necesario controlar los niveles de batería de los mismos así como sus tiempos de carga y descarga para cubrir esa demanda.

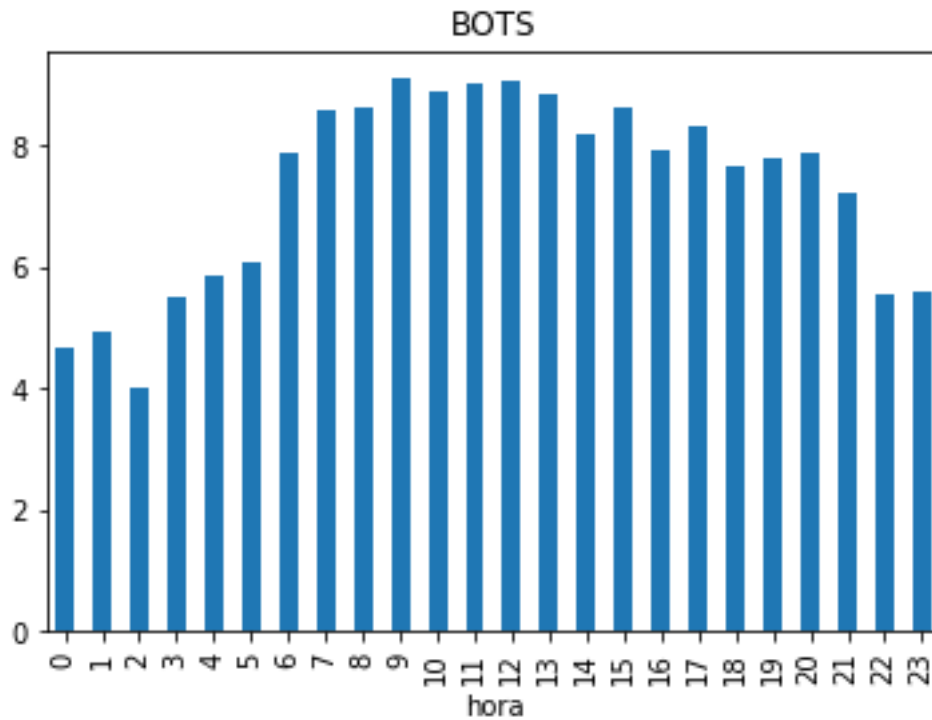
Se observa que las tablas SISTEMA_BMS_BIBERON_ENTRAR, SISTEMA_BMS_BIBERON_SALIR Y SISTEMAS_BMS_ORDENES de la base de datos proporcionada contiene el campo SOC (State Of Charge) que nos indican el estado de la batería en distintas fases o momentos. También se observa que las mismas tablas tienen el campo VEHICULO por lo que podemos obtener el número de BOT's que han realizado actividad en un intervalo de tiempo (hora/día/mes/ etc). Sin embargo, la variable "carga de trabajo actual" no viene reflejada en ningún dato implícito de ninguna de las tablas por lo que es necesario definir un KPI o medida que nos indique la "carga de trabajo actual". Para ello se ha considerado medir la carga de trabajo actual como el número de ordenes finalizadas por los BOT's en un intervalo de tiempo, dato que se puede obtener de los registros de la tabla SISTEMAS_BMS_ORDENES donde precisamente se registra un único registro por orden finalizada por un BOT.

Análisis temporal de carga de trabajo / BOT's activos

Se ha considerado realizar un análisis temporal para analizar la distribución de la carga de trabajo así como de los BOT's activos en función de la hora del día, día de la semana y mes con objeto de incluir estas variables o no en un posterior modelo / algoritmo que nos ayude a encontrar el punto óptimo en el que un BOT debe de cargar su batería. Para ello se ha realizado una extracción de los datos mediante la consulta *"Carga_Trabajo_Bots.sql"* para su posterior análisis visual en Python.

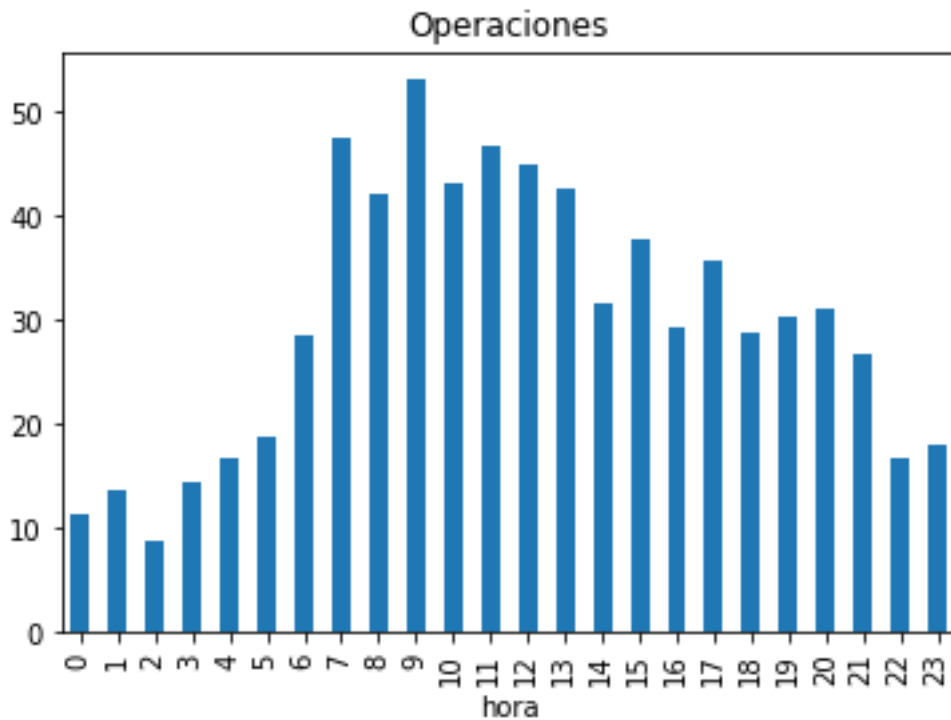
Análisis de la carga de trabajo y número medio de BOT's activo por hora



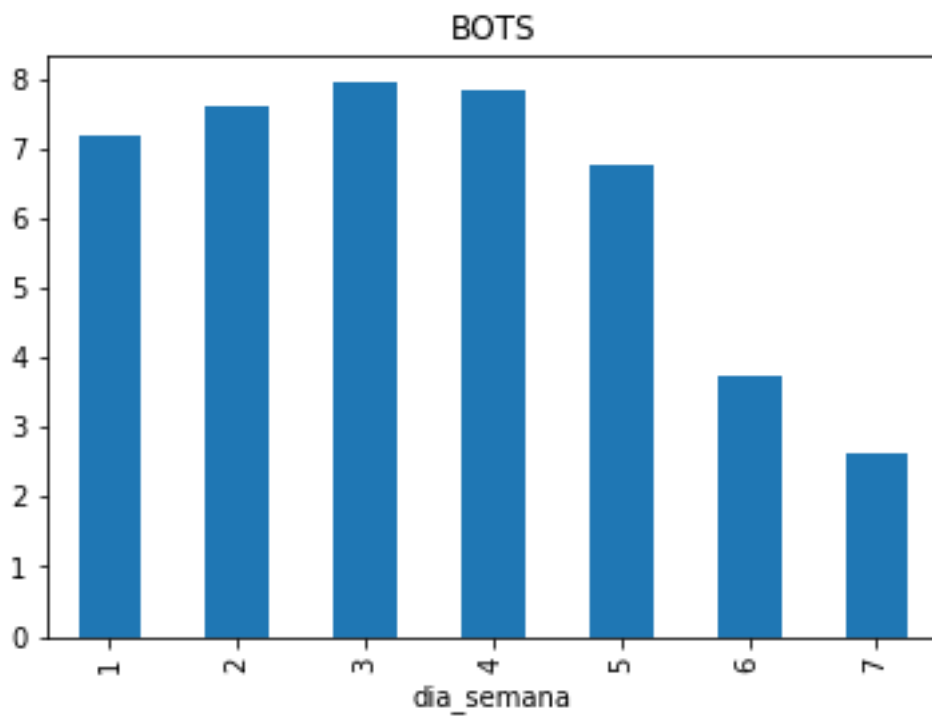
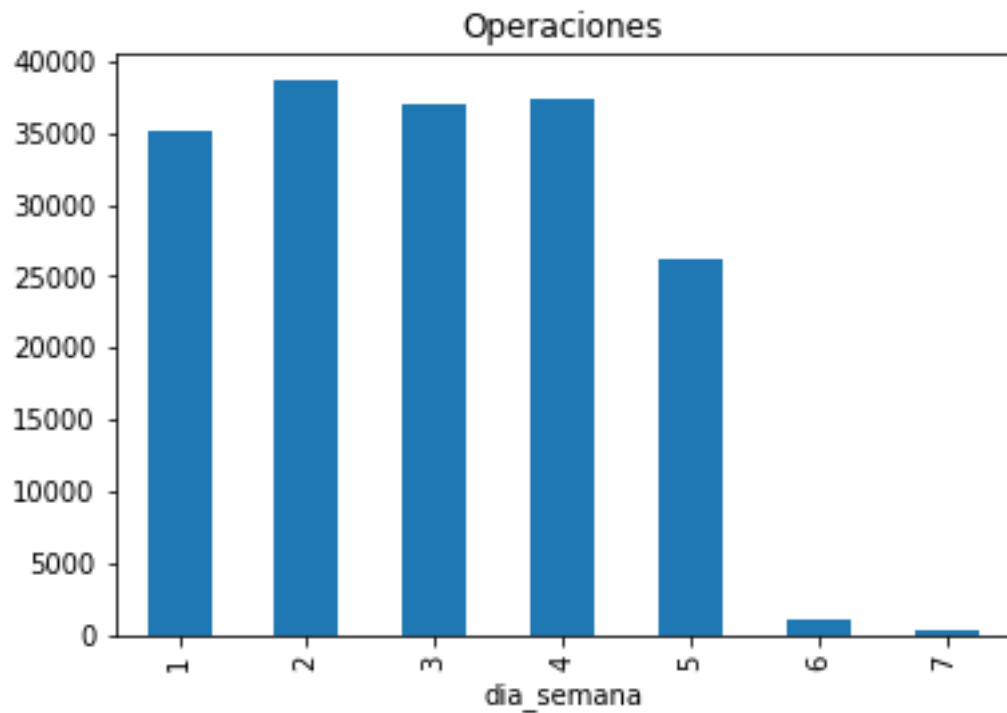


En ambas gráficas se observa que la actividad de los BOT's así como la media del número de BOT's varía dependiendo de la hora del día, por tanto, la variable hora influirá directamente en el momento óptimo en el que un BOT pueda o no ir a cargar su batería.

Para confirmar la teoría se puede visualizar el número de operaciones media por hora donde se observa el mismo efecto.

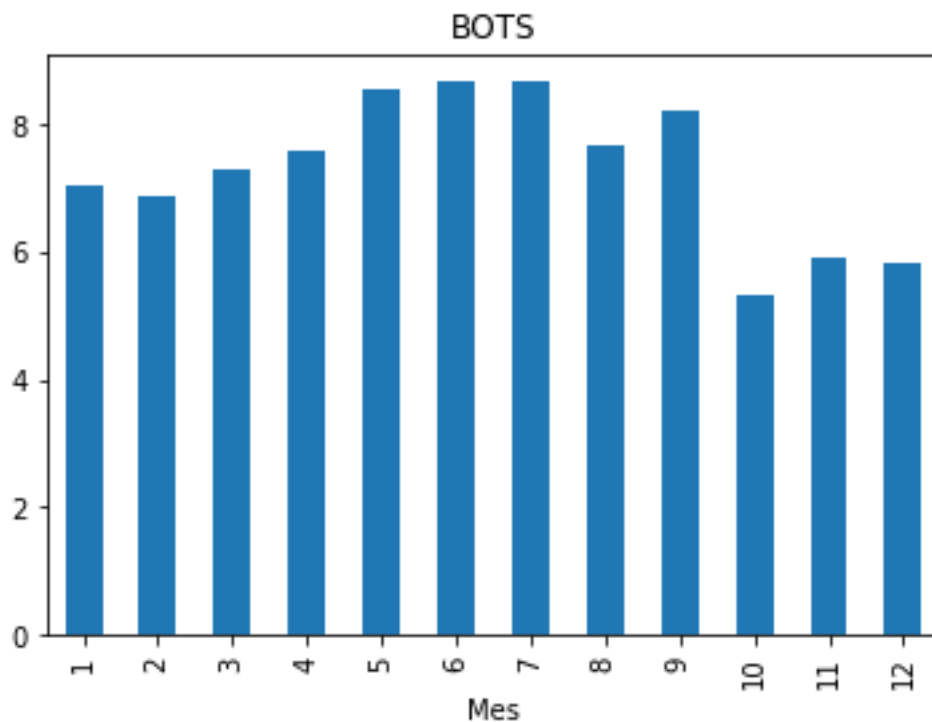
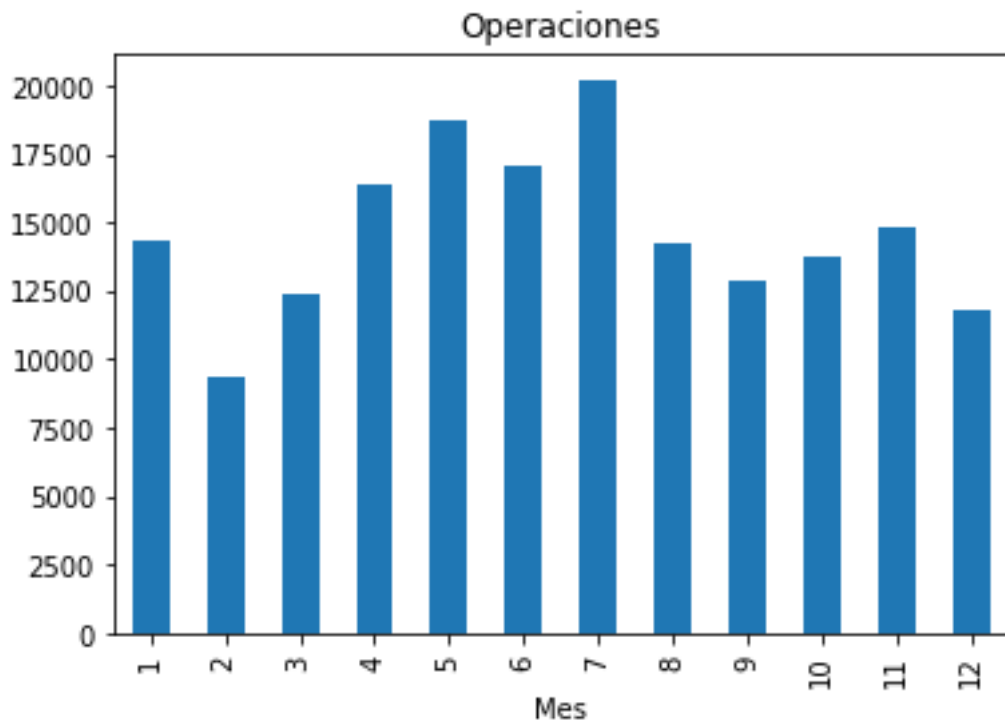


Análisis de la carga de trabajo y número medio de BOT's activo en función del día de la semana.



Se observa que tanto la carga de trabajo como el número de BOT's medio trabajando disminuye considerablemente los fines de semana, lo cual indica que deberemos tener en cuenta esta variable de cara a parametrizar el momento ideal de carga de un BOT.

Análisis de la carga de trabajo y número medio de BOT's activo día por mes.



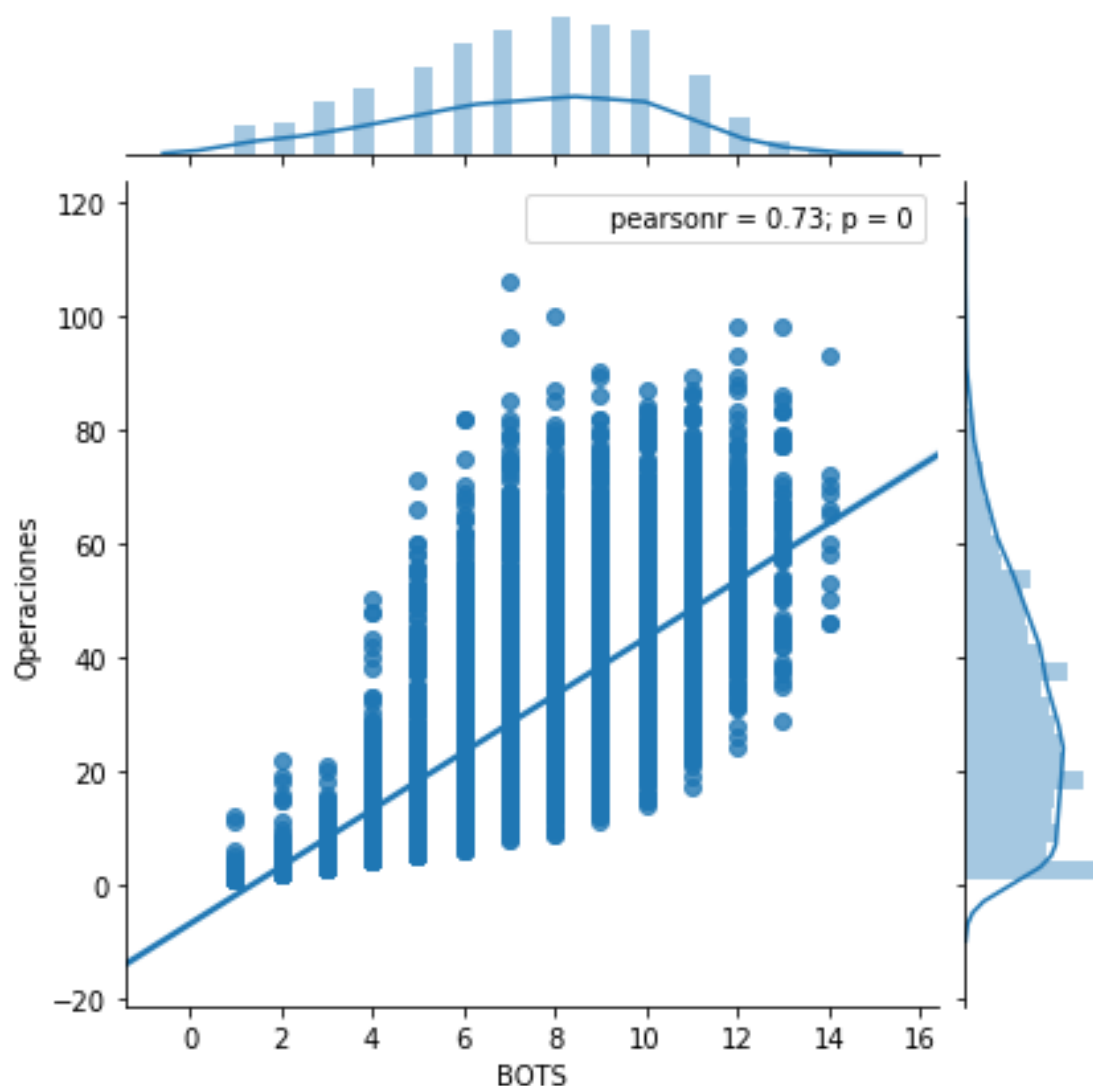
Se observa que la carga de trabajo puede tener variaciones considerables en función del mes de trabajo, sin embargo, no existen variaciones tan fuertes en el número medio de BOT's utilizados en un mes. Esto puede deberse a que la carga de trabajo puede ser repartida entre diferentes BOT's a lo largo de un máximo número de BOT's por lo que medir el número medio de BOT's activos en un mes puede no tener demasiado sentido en un intervalo de tiempo tan

amplio. No obstante, se considera que la variable mes es relevante para incluir en una posterior modelización del problema, ya que si se observa la influencia de la misma en la carga de trabajo.

Correlación lineal entre carga de trabajo y número de BOT's activos

Tras observar la relación de carga de trabajo y número de BOT's a lo largo de diferentes fases temporales se ha considerado analizar la correlación existente entre ambas variables obteniendo un índice de Pearson de 0'73, lo que indica que existe una correlación notable entre las mismas. Sin embargo, visualizando de forma gráfica esta correlación, se pueden obtener algunas conclusiones interesantes.

A pesar de que se observa una relación clara entre el número de operaciones realizadas y el número de BOT's activos, se observa que se alcanzan algunos máximos de número de operaciones con tan solo la mitad de BOT's funcionando (7 BOT's alcanzan las 106 operaciones finalizadas el día 20/10/2017). Lo cual indica que se pueden optimizar los flujos de trabajo para **soportar más carga de trabajo con menos BOT's operativos.**



Nº de BOT's – Estaciones de carga

Es importante considerar que existan estaciones de carga libres cuando un BOT requiera recargar su batería.

Analizando nuestro dataset con los datos acotados, tal como se comentó anteriormente, observamos que se han utilizado 7 estaciones de carga y 14 BOT's han realizado actividad. Por lo que será necesario tener en cuenta las estaciones libres antes de mandar un BOT a cargar su batería. Es decir, será otro parámetro más a considerar en el modelo.

```
In [165]: Bib_E.VEHICULO.nunique()
Out[165]: 14

In [166]: Bib_E.ESTACION.nunique()
Out[166]: 7
```

Tiempo medio de carga de un BOT en un puesto de carga

Se plantea como objetivo conocer la variación del porcentaje de carga de una batería en función del tiempo cuando un BOT se encuentra en una estación de carga. Es decir, se desea conocer el porcentaje de incremento de carga / tiempo para realizar estimaciones sobre cuánto tiempo debe estar cargando un BOT para alcanzar un porcentaje necesario para poder abordar una carga de trabajo "X".

Para ello, se ha generado la consulta SQL "*Tiempos_Carga.sql*" mediante la cual se pueda observar el incremento del porcentaje de batería de un BOT durante una carga, entendiendo como una carga el tiempo que pasa el BOT en la estación de carga desde que entra hasta que sale de la misma. La consulta extrae los tiempos de entrada y salida de los BOT's a la estación de carga de la tabla SISTEMA_BMS_BIBERON_ENTRAR y busca su respectiva salida de la estación en la tabla SISTEMA_BMS_BIBERON_SALIR

Cabe mencionar que se encuentran registros en la tabla de SISTEMA_BMS_BIBERON_ENTRAR en las que se observa que un BOT genera varios registros de entrada al biberón sin generar un registro de salida en la tabla de SISTEMA_BMS_BIBERON_SALIR.

	IDREGISTRO	FECHA	VEHICULO	INDICE	ESTACION	TEMPERATURA	SOC	TENSION	T1_CONSUMO	T1_DIST
6	52216	2018-08-28 17:38:30.243	31	213	3716	14	34	2586	-4831014	72420
7	52218	2018-08-28 17:51:48.167	31	43	3716	14	34	2599	-3386146	74
8	52220	2018-08-28 19:57:38.607	31	184	3716	14	32	2583	-29945136	19925
9	52222	2018-08-28 20:31:23.063	31	129	3716	14	31	2597	-4293221	235
10	52224	2018-08-28 20:35:10.860	31	129	3716	14	30	2578	-4078530	74731
11	52225	2018-08-28 20:53:13.737	31	106	3716	14	30	2581	-426235	1706
1	58414	2018-08-28 20:31:48.483	31	129	3716	0	31	2582	0	

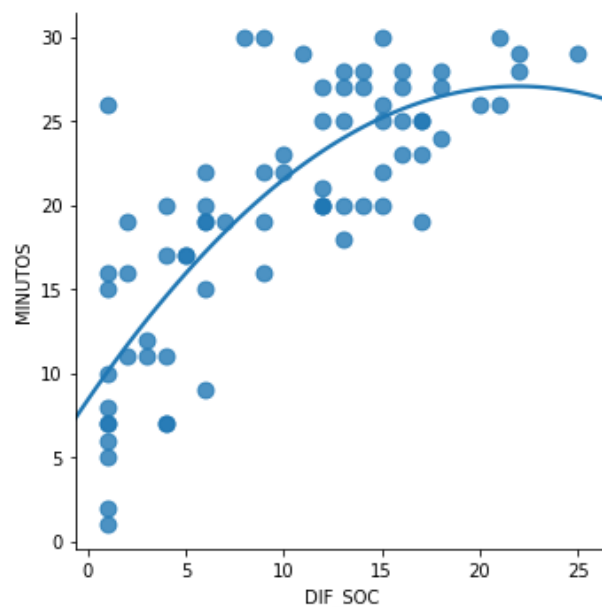
También se observan casos en los que se genera una salida pero no se ha aumentado la carga.

Se ha decidido eliminar estos casos del análisis ya que el objeto de la prueba es medir los aumentos de porcentaje de las baterías y estos casos podían afectar en los cálculos, por lo que solo se analizarán los casos en los que si se produce un aumento de batería entre la entrada y la salida del BOT de la estación de carga.

Debido a que se observan muchos casos en los que los BOT's alcanzan porcentajes, al parecer, máximos en sus baterías y en los que permanecen largos periodos de tiempo en las estaciones de carga, se ha decidido seleccionar una muestra de caso para el análisis basándonos en aquellos BOT's que entran con porcentajes inferiores al 50% a la estación y permanecen menos de 30 minutos en la misma. Solo de esta forma podíamos hacernos una idea general del porcentaje "real" de carga que podían tener los mismos. Para ello se ha generado un segundo fichero "Tiempos_Carga_30MIN.csv".

Si visualizamos de forma gráfica la relación entre ambas variables de estos últimos casos, se observa una relación lineal con cierta tendencia exponencial en el tiempo entre las mismas.

Esto podría indicar que **no siempre ciertos rangos niveles de batería se cargan a la misma velocidad**. Es decir, entre los niveles 30-40 de batería la velocidad de carga puede no ser la misma que en los nivel 40-50 de batería.



Dado que se considera imposible determinar con los datos existentes los niveles de carga en función de los rangos de batería con los datos proporcionados (solo se reflejan niveles de entrada y de salida y no tenemos una referencia del ciclo de carga mientras los BOT's cargan en las tablas) se ha decidido obtener una **media del tiempo que pasan los BOT's en la estación de carga**. Para ello se han discriminado aquellos BOT's que están más de 10 horas en la estación (considerando estos casos como BOT's que ya han alcanzado sus niveles máximo de

batería y puedan encontrarse parados sin salir de la estación) y se observa que la media de tiempo que pasa un BOT en la estación de carga es de **2'32 horas** con lo que se consideraría una carga completa.

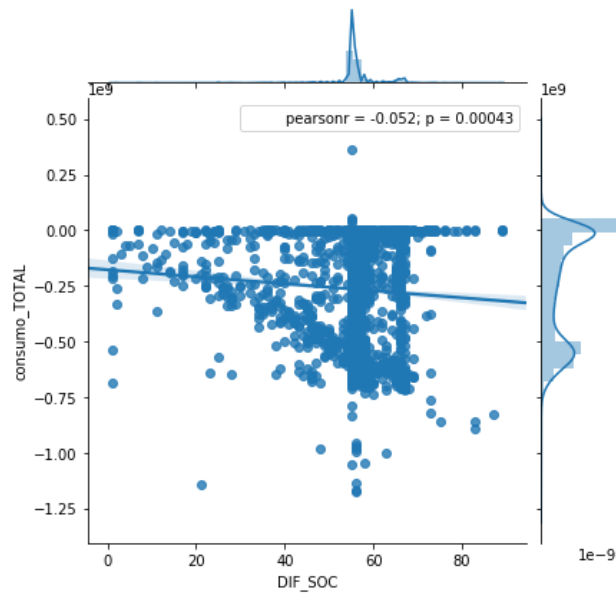
Descarga de batería de un BOT.

Por otro lado, se ha considerado oportuno realizar una medición de los tiempos en los que los BOT's están funcionando para lo que se ha generado un fichero mediante la consulta SQL "*Tiempos_Desarga.sql*" en el cual se reflejan los datos de entrada de los BOT's a la fuente de carga así como los datos de su salida inmediatamente anterior de una fuente de carga.

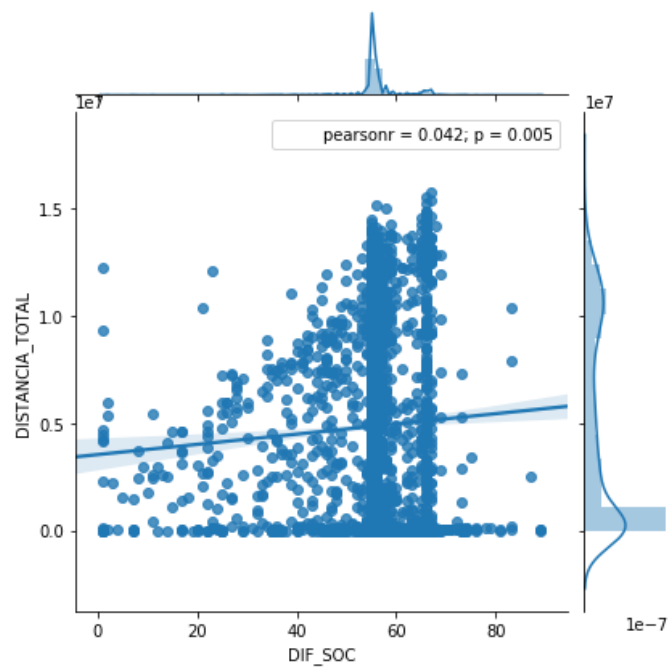
En el fichero generado se puede observar el tiempo que los BOT's han estado funcionando desde la última carga, la variación de su porcentaje de batería así como el número de operaciones realizadas por un BOT en ese intervalo de tiempo, el consumo total realizado entre cargas y la distancia total recorrida. Para estas dos últimas variables se ha tenido en cuenta la suma los 4 valores aportados por las 4 fases informados en los registros (T1+T2+T3+T4).

Se ha considerado eliminar de la prueba aquellos registros en los que el BOT ha estado parado, es decir, donde no ha realizado un consumo mínimo de batería.

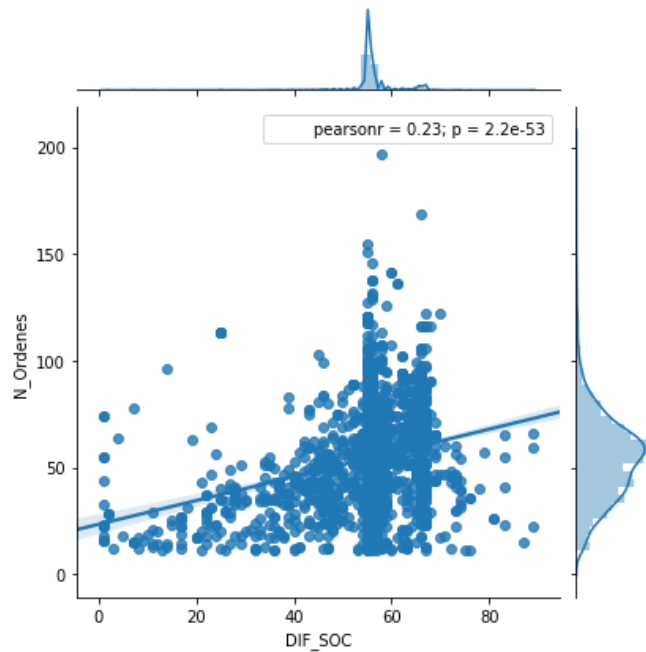
El objetivo de esta prueba será encontrar una relación entre el porcentaje de batería descargado y alguna de las otras variables, de tal forma que podamos medir el desgaste de batería de un BOT ante una carga de trabajo. Para ello se han analizado las correlaciones entre las variables "consumo total", "distancia total recorrida" y "número de operaciones realizadas" con sus respectivos niveles de descarga de batería, teniendo en cuenta solo aquellas observaciones en las que hay realmente actividad (se ha considerado solo aquellas en las que se han realizado un mínimo de 10 operaciones finalizadas y se ha superado la distancia recorrida de 50 metros) obteniendo los siguientes resultados:



No se encuentra una correlación lineal entre el consumo total y la descarga de batería.



No se encuentra una correlación lineal entre la descarga de batería y la distancia total recorrida.

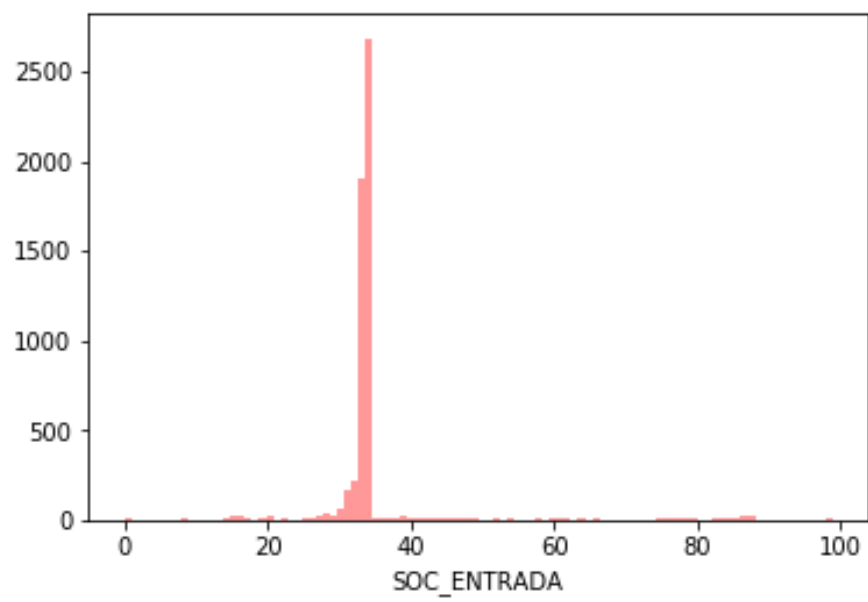
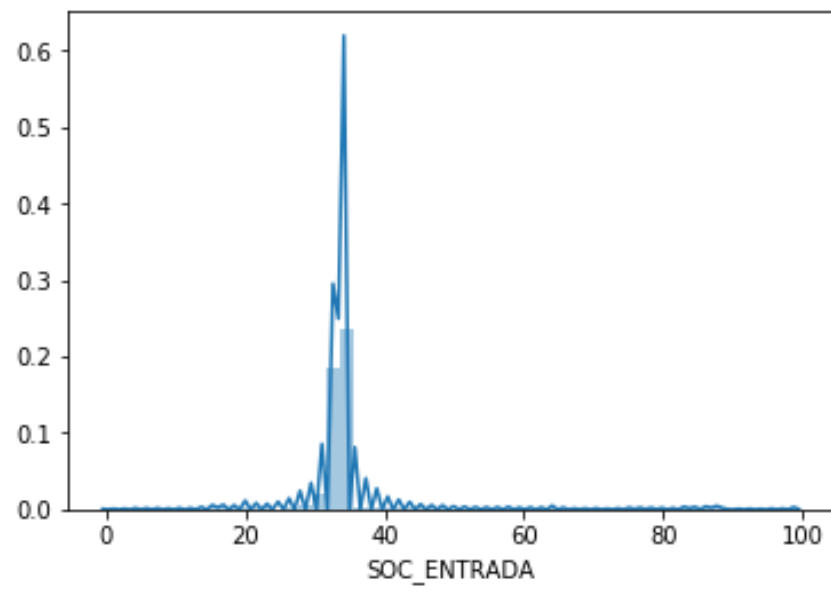


Se encuentra una ligera correlación lineal entre el número de órdenes finalizadas y el desgaste de batería del BOT, sin embargo, no se considera suficiente como para determinar el momento óptimo de enviar un BOT a recargar después de un número determinado de órdenes.

Dado que ninguna de las variables analizada se considera determinante para obtener una referencia de la descarga de los BOT's, se ha decidido obtener la referencia de descarga en función del tiempo obteniendo un promedio entre tiempo de cargas de los **BOT's de 1871 minutos o 31'18 horas**.

Frecuencia de nivel de carga de un BOT en la entrada a la fuente de carga

Se ha considerado interesante analizar la frecuencia en la que BOT ha entrado en la estación de carga en el histórico de datos. Para ello se ha generado en Python dos histogramas donde se observa que en la mayoría de observaciones en las que un BOT entra a cargar su batería el porcentaje de carga se encuentra entre el 30 y 40 por ciento de la misma.



Conclusiones análisis EDA

Para conseguir obtener el momento óptimo en el que un BOT debe ser enviado a cargar su batería se deben tener en cuenta los siguientes aspectos:

- 1) Será necesario parametrizar el algoritmo en función a la **hora, día de la semana y mes**, ya que la carga de trabajo varía en función de estas tres variables temporales.
- 2) Será necesario incluir una variable que indique el número de BOT's activos.
- 3) Será necesario incluir una variable que indique la carga de trabajo.
- 4) Será necesario tener en cuenta el número de puestos de cargas libres.
- 5) Será necesario parametrizar el algoritmo en función de los tiempos de carga de un BOT establecido en **2'32 horas**.
- 6) Será necesario parametrizar el algoritmo en función del tiempo medio que un BOT puede estar trabajando, tiempo medio establecido en **31'8 horas**.
- 7) Será necesario controlar que el BOT entre a recargar cuando si nivel de batería descienda de **X %**.

Selección y aplicación de algoritmos. Predicción y optimización

El objetivo principal del problema es predecir el momento óptimo en el que un BOT debe ir a cargar batería atendiendo a la carga de trabajo actual, el número de BOT's activos, así como sus niveles de carga de batería.

Para la resolución del problema y dar respuesta a predecir el momento se ha optado por encontrar diferentes soluciones basándonos en algoritmos predictivos (orientados a predecir una variable objetivo en función de datos históricos) y algoritmos de optimización (orientados a la mejora de un proceso concreto).

Algoritmos predictivos

Es posible predecir el resultado de una situación futura **basándonos en los datos históricos**, es decir, se puede configurar un algoritmo que determine el momento en el que un BOT deba acudir a recargar su batería teniendo en cuenta su nivel de carga de batería, el número de BOT's activos y la carga de trabajo actual basándonos en los patrones históricos en los que manual o automáticamente un BOT ha sido enviado a una estación de carga.

De esta forma cuando un BOT se encuentre bajo ciertas circunstancias puede ser enviado o no a la estación de carga de forma automática, ya que el algoritmo evaluará una serie de reglas basadas en una serie de comportamientos anteriores para determinar una salida: ir o no a recargar su batería.

Por ello se puede plantear el problema como un problema de clasificación en base a una serie de variables.

Por ejemplo, se puede construir un algoritmo basado en árboles de decisión que sea capaz de determinar una salida (ir a recargar la batería / no ir a recarga la batería) después de cada orden terminada por un BOT basándonos en la actividad histórica de los mismos.

Este tipo de algoritmos determinarán una serie de reglas basada en los datos de entrenamiento para **tomar la mejor decisión posible en base a una serie de circunstancias o patrones**.

Generación del Dataset necesario y selección de variables del modelo

Para ello se ha construido un Dataset mediante la SQL (`..\TRASTI\MODELO\Ficheros\Fichero_Base.sql`) generando el fichero (`..\TRASTI\MODELO\Ficheros\Fichero_Base.csv`) a partir de las tablas de la base de datos con la información a nivel de orden terminada donde figuran todos los datos relevantes de la tabla de `SISTEMAS_BMS_ORDENES` y con la información añadida de:

- **Carga_Trabajo:** variable que mide el número de operaciones realizadas en los últimos 60 minutos.
- **Bots_Activos:** número de BOT's que han finalizado una operación en los últimos 60 minutos.
- **MCA_CARGA:** variable objetivo que marca si después de esa operación el vehículo ha sido enviado a recargar su batería o no.
- **Mes:** variable mes.
- **Día_semana:** variable que informa del día de la semana.
- **Hora:** variable hora.
- **DISTANCIA_TOTAL:** distancia total recorrida por el vehículo en base los parámetros T1,T2,T3,T4.
- **CONSUMO_TOTAL:** consumo total recorrida por el vehículo en base los parámetros T1,T2,T3,T4.

Dado que tenemos un conjunto de datos etiquetados donde sabemos cuándo un robot ha sido enviado a cargar tras realizar una última operación, podemos aplicar algoritmos de clasificación que detecten las reglas o patrones que han llevado al BOT a ser enviado al puesto de carga y por consiguiente aplicar esas reglas ante nuevas observaciones con las mismas características.

Algoritmos basados en árboles

Los algoritmos basados en árboles de decisión pueden ayudarnos a resolver el problema, ya que entrenándolos con un conjunto de datos suficientemente grande, podrán establecer las

reglas necesarias para que un BOT sea o no enviado a cargar después de realizar una operación en base a las variables de entrada identificadas en el EDA.

Por ello, se han construido 3 modelos predictivos basados en un **árbol de clasificación**, el algoritmo **“Random Forest”** y el algoritmo **“Gradient Boosting”** para predecir el momento en el que un BOT debe ser enviado a cargar su batería tras realizar una operación.

Los algoritmos han sido entrenados con el dataset *“Fichero_base.csv”* citado anteriormente dividiendo el conjunto de datos con un 60% de observaciones en data de entrenamiento y un 40 % en datos test.

La tasa de predicción (% de observaciones bien clasificadas) ofrecida por cada uno de los algoritmos ha sido la siguiente:

- Árbol de decisión: **98,20 % de acierto.**
- Random Forest: **98,92 % de acierto.**
- Gradient Boosting: **98,92 % de acierto.**

Estos datos nos sugieren que los modelos ofrecidos tienen una capacidad predictiva muy alta a la hora de enviar o no un BOT a recargar después de terminar cierta acción.

Para la solución de este problema seleccionaríamos o bien **“Random Forest”** o bien **“Gradient Boosting”** ya que los dos nos ofrecen la misma tasa de precisión.

El **momento óptimo para enviar un BOT a recargar la batería**, en este caso, vendrá determinado por las reglas o decisiones establecidas por los árboles generados en función siguientes variables:

'SOC','Carga_Trabajo','Bots_Activos','Mes','dia_semana','hora','CONSUMO_TOTAL','DISTANCIA_TOTAL'.

La conclusión obtenida es que el uso de este tipo de algoritmos pueden dar una buena solución al problema enviando al BOT a recargar la batería en el momento oportuno en un 98% de las veces de forma correcta.

La implementación de los algoritmos viene recogida en el script de Python *“Prediccion_Arboles.py”* y puede encontrarse dentro de la carpeta *“.. \ TR ASTI\MODELO\Python”*.

Cabe destacar que el algoritmo podría ser mejorado mediante el estudio profundo de las mejores configuraciones del mismo que se adapten a los datos aportados.

Redes Neuronales

Se ha decidido explorar el uso de redes neuronales que nos ayuden a resolver el problema. Una red permite buscar la mejor combinación de parámetros que mejor se ajustan a resolver

un problema. Es por ello que se ha decidido realizar uso de este tipo de algoritmos para encontrar una solución al mismo.

En nuestro caso, los parámetros serían nuestras variables input del dataset “Fichero_Base.csv” y la variable objetivo MCA_CARGA la resolución del problema. Una vez la red esta entrada, podría ser utilizada para aplicar la mejor combinación encontrada. Es decir, el momento óptimo en el que BOT debe ir a cargar su batería.

Para ello se ha utilizado una red neuronal artificial perceptron multicapa del paquete de Python SKLearn, que nos permitirá predecir el momento óptimo en base a las variables de entrada seleccionadas

Se ha creado un programa en Python que crea una serie de redes neuronales que nos aproximen a la solución del problema. El modelo obtenido tiene una precisión del 98 % por lo que se puede afirmar que el modelo generado puede predecir el momento preciso de cuando un BOT debe ir a cargar su batería con un alto grado de precisión.

El **momento óptimo para enviar un BOT a recargar la batería** en este caso vendrá determinado por la combinación de las siguientes variables: 'SOC', 'Carga_Trabajo', 'Bots_Activos', 'Mes', 'dia_semana', 'hora', 'CONSUMO_TOTAL', 'DISTANCIA_TOTAL'.

La conclusión obtenida es que el uso de redes neuronales puede ayudarnos a predecir el momento óptimo en el que un BOT debe acudir a cargar su batería.

La implementación de los algoritmos viene recogida en el script de Python “Red_Neuronal.py” y puede encontrarse dentro de la carpeta “.. \ TR ASTI\MODELO\Python”.

Al igual que en la creación de los algoritmos basados en árboles, es importante destacar que la modelización es un arte y la propia red neuronal puede ser mejorada mediante un testeo mucho más profundo de la configuración de la red neuronal.

Algoritmos de optimización

Tal como hemos comprobado, podemos predecir el momento en el que un BOT debe ser enviado a cargar su batería en función de los patrones recogidos de un histórico de datos. Sin embargo, el hecho de predecir un resultado basado en lo que se ha hecho en una situación anterior, **puede o no ser óptimo**, ya que si nos basamos en datos históricos para entrenar un algoritmo, este tomará la mejor decisión posible en base a lo que ha pasado en circunstancias pasadas que pueden o no ser las idóneas de cara resolver un problema.

¿Qué se entiende cómo óptimo? Dada la definición del problema planteado se entiende como óptimo tener el mayor número de BOT's disponibles cuando la curva de demanda de BOT's alcance su mayor actividad. Como **no tenemos una referencia de la curva de la demanda** para conocer cómo debemos atender esta situación, debemos basarnos en el KPI definido anteriormente como carga de trabajo, es decir, deberemos basarnos en el número de

operaciones realizadas por los BOT's en un intervalo de tiempo para alcanzar una aproximación de esa carga de trabajo así como los BOT's necesarios para atender esa carga.

Si deseamos encontrar el momento óptimo en el que un BOT debe acudir a una estación de carga, estamos ante un claro **problema de optimización**. Para plantear un problema de optimización es necesario establecer los límites del sistema que deseamos optimizar. Tal como hemos observado en el EDA, una limitación podría ser el nivel de batería aceptable para que un BOT esté trabajando. Por otro lado, es necesario definir ciertas variables que definan el modelo a obtener (variables identificadas en el EDA).

Dadas las premisas se ha definido el siguiente modelo para resolver el problema:

t = tiempo.

a = batería que pierde en una hora un BOT.

b = mínima batería consentida.

d = demanda de BOT's.

i = número de BOT's.

M = constante definida para resolver la función = 3000.

X(it) = estado de la batería.

Y(it) = (1: Si está trabajando, 0: si está cargando).

Función objetivo:

$$Z = \sum t [(\sum_i Y(it)) - dt]$$

Restricción:

$$X(it) - b \geq M (Y(it) - 1) \Rightarrow X(it) - b \geq 0$$

Mediante la función objetivo se pretende obtener el número de BOT's necesarios en función del tiempo y la demanda.

La restricción aplicada nos permitirá controlar el nivel de carga de la batería.

Nota: No ha sido posible crear un algoritmo de optimización con Python debido a la falta de experiencia en este tipo de algoritmos y el tiempo máximo para realizar la prueba no me ha permitido interiorizar más en el uso de los mismos.

Conclusiones

Se ha comprobado que utilizando algoritmos de clasificación basados en árboles de decisión, se puede determinar el momento óptimo en el que enviar un BOT a recargar su batería. Del mismo modo se ha comprobado que una Red Neuronal puede darnos solución al problema.

Se entiende que la selección de variables realizada a partir del EDA es la correcta para alimentar los algoritmos en ambos casos, debido a que la respuesta en ambos casos nos ha ofrecido una tasa de precisión muy alta.

Aunque, como mencionaba en ambos casos, modelizar es un arte y la configuración de los parámetros puede ser mejorada profundizando en la creación de los mismos, una vez seleccionado un modelo concreto, el algoritmo podría ser implantado para que **el sistema que envía los BOT's al biberón sea automatizado usando bien una red neuronal o un algoritmo basado en arboles de decisión.**

Consideraciones futuras

Como comentaba en el apartado de algoritmos de optimización, no ha sido posible implementar un algoritmo de optimización con Python debido a la falta de experiencia en este tipo de algoritmos y el tiempo disponible para realizar la prueba.

Es por ello que considero que, de cara a dar una continuación, este trabajo sería idóneo implementar algoritmos de optimización que nos permitan comparar su uso para la resolución de este tipo de problemas con el del uso de algoritmos predictivos.