**EJERCICIO ENTREGABLE BMW – Benjamin Monrabal Orts**

1. **Qué columnas eliminaron**

* Columna “marca”, debido a que se presupone que es un dataset de BMW, y todos los registros son marca BMW.
* A la hora de imputar nulos, se detecta que la columna “asientos\_traseros\_plegables”, tiene un 70% de nulos, por lo que se considera que esta variable no aporta valor al futuro modelo, y por lo tanto, se ha decidido eliminarla.
* Al hacer la normalización del target (“log\_precio”), se elimina la columna original (“precio”).

1. **Qué se hizo con los nulos y cómo se limpiaron las columnas**

Se han tomado diferentes criterios, según la variable:

* **“modelo”:** se detectan 3 nulos en el modelo, y debido a que la cantidad de nulos es menos a un 0,062%, se decide eliminarlos.
* **“km”:** se detectan 2 nulos en el modelo, y debido a que la cantidad de nulos es menos a un 0,04%, se decide hacer una mediana.
* **“potencia”:** se detectan 1 nulo en el modelo, y debido a que la cantidad de nulos es menos a un 0,02%, se decide hacer una mediana. **“tipo\_gasolina”:** se detectan 5 nulos en el modelo, y se decide no eliminarlos, e imputarlos a “sin\_tipo\_gasolina”.
* **“color”:** se detectan 445 nulos en el modelo, y se decide no eliminarlos, e imputarlos a “sin\_color”.
* **“tipo\_coche”:** se detectan 1460 nulos en el modelo, y se decide no eliminarlos, e imputarlos a “sin\_tipo\_coche”.
* **“volante\_regulable”:** se detectan 4 nulos en el modelo, y debido a que la cantidad de nulos es menos a un 0,08%, se decide eliminarlos.
* **“aire\_acondicionado”:** se detectan 484 nulos en el modelo, y debido a que esta columna tiene registros booelanos, se decide imputarlos a “-1”.
* **“camara\_trasera”:** se detectan 2 nulos en el modelo, y debido a que la cantidad de nulos es menos a un 0,04%, se decide eliminarlos.
* **“asientos\_traseros\_plegables”:** se detectan 3391 nulos en el modelo, y debido a que eso es un 70% de toda la columna, se decide eliminar la columna entera, debido a que no aporta al valor al modelo.
* **“elevalunas\_electrico”:** se detectan 2 nulos en el modelo, y debido a que la cantidad de nulos es menos a 0,04%, se decide eliminarlos.
* **“bluetooth”:** se detectan 725 nulos en el modelo, y se decide no eliminarlos, e imputarlos a “-1”.
* **“alerta\_lim\_velocidad”:** se detectan 725 nulos en el modelo, y se decide no eliminarlos, e imputarlos a “-1”.
* **“tiempo\_venta”:** columna creada diferencia entre fecha\_venta y fecha\_registro”, se detectan 2243 nulos y se decide imputar los nulos al valor de la mediana.
* **“precio”:** se detectan 6 nulos, y se decide imputar los nulos al valor de la mediana.

Gráfico

Descripción generada automáticamente

1. **Análisis univariable: comentarios, outliers, agrupar…**

* Se ha decidido agrupar el índice con la columna “modelo”, y eliminar la columna “modelo”

nuevo\_indice = bmw6.index.astype(str) + '\_' + bmw6['modelo'].astype(str)

bmw6.index = nuevo\_indice

* Se ha realizado la diferencia entre “fecha\_registro” y “fecha\_venta” (y eliminados esas dos columnas), para obtener la columna tiempo\_venta.
* En la columna tipo\_gasolina, cambiamos el registro Diesel por diesel

bmw["tipo\_gasolina"]=bmw["tipo\_gasolina"].replace("Diesel", "diesel", regex=True)

* Se detectan 87 outliers en la columna “km” (incluido valores negativos) que se imputan al valor de la mediana.
* Se detectan 591 outliers en la columna “potencia” que se imputan al valor de la mediana.
* En la columna “precio” se consideran precios por encima de 100000€ o por debajo de 2000€, como outliers, un total de 100 outliers, que se imputan al valor de la mediana.

outliers2\_precio = bmw3[(bmw3["precio"]<= 2000) | (bmw3["precio"] > 100000)]

* En la columna tipo\_gasolina: se han agrupado aquellos valores que tienen menos de 100 registros.
* En la columna color: se han agrupado aquellos valores que tienen menos de 400 registros.
* En la columna tipo\_coche: se han agrupado aquellos valores que tienen menos de 100 registros.

1. **Análisis de Correlación inicial**

Se ve cierta correlación entre precio y potencia (63.9%) pero no produce alerta roja de que sean variables altamente correlacionadas.

Imagen que contiene nombre de la empresa

Descripción generada automáticamente

1. **Análisis variable vs target**

A mayor kilometraje, hay una tendencia descendente del precio, y viceversa, coches con menos km, mayor precio. Con potencia 120, hay una gran variedad de rango de kilometraje y de precio. Los coches de muy baja o muy alta potencia, hay menos rango de precio. Mucha variedad de precio con coches diesel, seguido por coches de petróleo (que en general son de precios bajos cuando hay algo de kilometraje).

1. **Transformación de categóricas a numéricas**

* La variable tiempo\_venta, con formato timedelta64[ns], la convierto a numérico.

bmw8["tiempo\_venta\_int"] = bmw8["tiempo\_venta"]/np.timedelta64(1, 's')

* Para pasar de valores true/false a 1 y 0, utilizamos la función np.where:

bmw9["volante\_regulable\_int"]=np.where(bmw9["volante\_regulable"]==True,1,0 )

* Utilizo la función get\_dummies de pandas, y se transforman 3 variables dentro de las categóricas, lcat = [ 'tipo\_gasolina', 'color', 'tipo\_coche'], y obtener valores 0 y 1.

bmw12 = pd.get\_dummies(data=bmw11, columns=lcat2)\*1

1. **Normalizar variables numéricas**

Se utiliza un MinMaxScaler para las tres variables numéricas: “km”, “potencia” y “tiempo\_venta\_int”. Se hace cada normalización por separado. Ejemplo de km:

bmw13[lnum2]= minMaxResultado\_km.fit\_transform(bmw13[lnum2])

1. **Análisis de correlación final, hay alguna variable correlacionada?**

Ningún tipo de correlación importante. Algo de correlación tipo\_gasolina\_electrica y modelo\_i3 (70%). Cierta correlación entre el log\_precio y la columna “alerta\_lim\_velocidad\_int” (42%).

1. **Dataset limpio y preprocesado**

Imagen que contiene texto, placa, tabla, grande

Descripción generada automáticamente Interfaz de usuario gráfica

Descripción generada automáticamente con confianza media