Entregable 1 – Juan Pablo Santangelo

Data set inicial:

	marca	modelo	km	potencia	fecha_registro	tipo_gasolina	color	tipo_coche	volante_regulable	aire_acondicionado	camara_trasera	asientos_traseros_plegables	elevalunas_electrico	bluetooth	gps	alerta_lim_velocidad	precio	fecha_venta
0	NaN	118	140411.0	100.0	1/2/2012	diesel	black	NaN	True	True	False	NaN	True	NaN	True	NaN	11300.0	1/1/2018
1	BMW	M4	13929.0	317.0	NaN	petrol	grey	convertible	True	True	False	NaN	False	True	True	True	69700.0	1/2/2018
2	BMW	320	183297.0	120.0	1/4/2012	diesel	white	NaN	False	False	False	NaN	True	False	True	False	10200.0	1/2/2018
3	BMW	420	128035.0	135.0	NaN	diesel	red	convertible	True	True	False	NaN	True	True	True	NaN	25100.0	1/2/2018
4	BMW	425	97097.0	160.0	NaN	diesel	silver	NaN	True	True	False	False	False	True	True	True	33400.0	1/4/2018

1. Las columnas eliminadas fueron: (1) "asientos_traseros_plegables" (70% de valores NaN) y baja incidencia en el precio del vehículo (dispersión de datos según scatterplot), (2) "fecha_registro" y "fecha_venta" (las utilicé para crear otra variable que mida la diferencia en días desde el registro hasta la venta del vehículo. A su vez, las variables con fechas no serán de gran ayuda en el modelo predictivo), (3) "marca" (eran todas BMW o NaN, por ende, luego de asignárselas al mismo BMW, decidí elimarlas); (4) "bluetooth", porque no tenía información representativa; (5) "modelo", ya que utilicé la columna para agrupar los automóviles según su serie; (6) columnas altamente correlacionadas luego de hacer OneHotEncoder (que básicamente son lo mismo o explican lo mismo): "tipo_gasolina_diesel" y "tipo_coche_suv".

2. Con los nulos hice diferentes tratamientos. Calculé las proporciones de nulos en cada columna y a partir de allí analicé cada uno. La columna "asientos_traseros_plegables" la eliminé porque tenía 70% de nulos. La columna "fecha_registro" tiene 2423 nulos, el 50% de las observaciones y los relleno con la mediana o medida central. La "fecha_venta" tiene un solo NaN y le hago un fillna con la moda. "Tipo_coche" la agrupé según aquellas categorías cuyo valores eran más frecuentes y rellené sus NaN con un string "Desconocido". Creé una nueva variable restando fecha_venta y fecha_registro (la llamé "diferencia", me devuelve la diferencia en días entre un suceso y el otro y rellené sus nulos con la media de sus valores). "Marca" tiene 970 NaN, el 20% del total, se los imputo a la marca "BMW" pero luego decido eliminar la columna marca porque no tiene información sensible. Los NaN de alerta_lim_velocidad son 728, el 15% del total, y los asigné equitativamente a True y False porque tienen proporciones parecidas (utilizando random choices). "Bluetooth" tiene 15% de NaN, 728 en total y los imputo a los valores False luego de calcular sus proporciones, de manera del de mantener la proporción relativa de los valores existentes. "Aire acondicionado", le asigné a los valores NaN un valor entero de 1. (0: no tiene aire, 1: tiene aire, según el análisis anterior, el 79% de los autos posee aire acondicionado). Color tiene 9% de NaN, y los relleno con un string "others". Además, agrupo los colores según la densidad de las observaciones. "Potencia" tiene un solo valor NaN, así que elimino esa fila, lo mismo con "km" (2 Nan), "elevalunas_electrico" (2 NaN), "cámara_trasera" (2 NaN), "volante_regulable" (4 NaN) y "modelo" (tiene 3 NaN, lo agrupé según series de BMW que coloqué en una nueva columna que se llama "categoría" y eliminé la columna "modelo"), "tipo_gasolina" (5 NaN) y precio (6 NaN).

3. Análisis univariable: a priori puede verse que las más representativas para el precio serán "km" y "potencia", aunque "diferencia" guarda una mínima correlación inversa. Sí pueden detectarse outliers, en la columna "km" (97), en la columna "potencia" (587), en "aire_acondicionado" (887), en "cámara_trasera" (971), en "bluetooth" (991), en "gps" (325), en "diferencia" (1089), etc. Por ello utilizo una transformación logarítmica de los outliers para reducir su impacto. Asimismo, agrupé "modelo", "tipo_coche" y "color" para mayor organización, legibilidad y eficiencia del código.

4. No, en el análisis de correlación inicial (antes de realizar OneHotEncoder y min max scaler) no he detectado variables altamente correlacionadas. Utilicé una función para detectar y dropear las columnas correlacionadas y también la varianza mínima entre variables. A priori puede observarse que "potencia" y "precio" tienen una correlación de 0.65 y que "precio" y "km" tienen una correlación de -0.4. No son representativas, pero son los valores más altos que exhibe esta primera matriz de correlación.

5. Sí, hay varios aspectos a destacar. Primeramente, una correlación inversa (aunque poco significativa) entre “diferencia” y “precio” y también entre “km” y “precio”, lo cual nos sugiere que, a mayor cantidad de días entre venta y registro o mayor cantidad de km, menor será el precio. Por otro lado, la columna potencia guarda una relación positiva con el precio, lo cual es esperable dado que, a mayor potencia del auto, mayor será su valor. Respecto de los booleanos transformados a integer, no arrojan demasiada información excepto por alerta límite de velocidad (cuando el automóvil posee esta característica, el precio parece elevarse). Luego, dentro de la columna tipo gasolina, se destaca “hybrid petrol” que parece ser la que guarda mayor relación con el precio.

6. Creé una función para generar lista numéricas y categóricas. Una vez que hice ello, utilicé las listas categóricas para hacer un OneHotEncoder a través de get dummies. Transformé las siguientes variables: ['tipo_gasolina', 'color', 'tipo_coche', 'categoria']

7. Hay variables luego de hacer el OneHotEncoder que están altamente correlacionadas. Ejemplo: tipo_gasolina_petrol y tipo_gasolina_diesel (-0.95) y tipo_coche_suv y categoría_suv (0.8). Esto tiene que ver con los agrupamientos anteriores a la correlación y la utilización de técnicas OneHotEncoder y min max scaler, ya que se cruzan tipos de gasolina y tipos de coche con categorías similares. En este caso, opté por eliminar uno de cada par ya que estas columnas prácticamente están explicando lo mismo, se asemejan.

8. `Data columns (total 34 columns):`

```
 #   Column                        Non-Null Count  Dtype
---  ------                        --------------  -----
 0   precio                        4817 non-null   int32
 1   tipo_gasolina_Diesel          4817 non-null   int32
 2   tipo_gasolina_electro         4817 non-null   int32
 3   tipo_gasolina_hybrid_petrol   4817 non-null   int32
 4   tipo_gasolina_petrol          4817 non-null   int32
 5   color_black                   4817 non-null   int32
 6   color_blue                    4817 non-null   int32
 7   color_grey                    4817 non-null   int32
 8   color_others                  4817 non-null   int32
 9   tipo_coche_Desconocido        4817 non-null   int32
 10  tipo_coche_estate             4817 non-null   int32
 11  tipo_coche_otros              4817 non-null   int32
 12  tipo_coche_sedan              4817 non-null   int32
 13  categoria_BMWi                4817 non-null   int32
 14  categoria_Deportivo           4817 non-null   int32
 15  categoria_SUV                 4817 non-null   int32
 16  categoria_Serie 1             4817 non-null   int32
 17  categoria_Serie 2             4817 non-null   int32
 18  categoria_Serie 3             4817 non-null   int32
 19  categoria_Serie 4             4817 non-null   int32
...
 32  minMax_alerta_lim_velocidad   4817 non-null   float64
 33  minMax_diferencia             4817 non-null   float64
dtypes: float64(10), int32(24)
```

Aquí hay un pantallazo final del DataFrame:

	precio	tipo_gasolina_Diesel	tipo_gasolina_electro	tipo_gasolina_hybrid_petrol	tipo_gasolina_petrol	color_black	color_blue	color_grey	color_others	tipo_coche_Desconocido	tipo_coche_estate	tipo_coche_otros	tipo_coche_sedan	categoria_BMWi	categoria_Deportivo	categoria_SUV	categoria_Serie 1	categoria_Serie 2	categoria_Serie 3	categoria_Serie 4	categoria_Serie 5	categoria_Serie 6	categoria_Serie 7	categoria_Z line	minMax_km	minMax_potencia	minMax_volante_regulable	minMax_aire_acondicionado	minMax_camara_trasera	minMax_elevalunas_electrico	minMax_bluetooth	minMax_gps	minMax_alerta_lim_velocidad	minMax_diferencia
0	11300	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0.139949	0.236407	1.0	1.0	0.0	1.0	0.0	1.0	1.0	0.193767
1	69700	0	0	0	1	0	0	1	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0.013454	0.749409	1.0	1.0	0.0	0.0	1.0	1.0	1.0	0.145474
2	10200	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0.162839	0.283688	0.0	0.0	0.0	1.0	0.0	1.0	0.0	0.190879
3	25100	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0.125752	0.319149	1.0	1.0	0.0	1.0	1.0	1.0	1.0	0.145474
4	33400	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0.096631	0.378251	1.0	1.0	0.0	0.0	1.0	1.0	1.0	0.151349