# EJERCICIO BMW

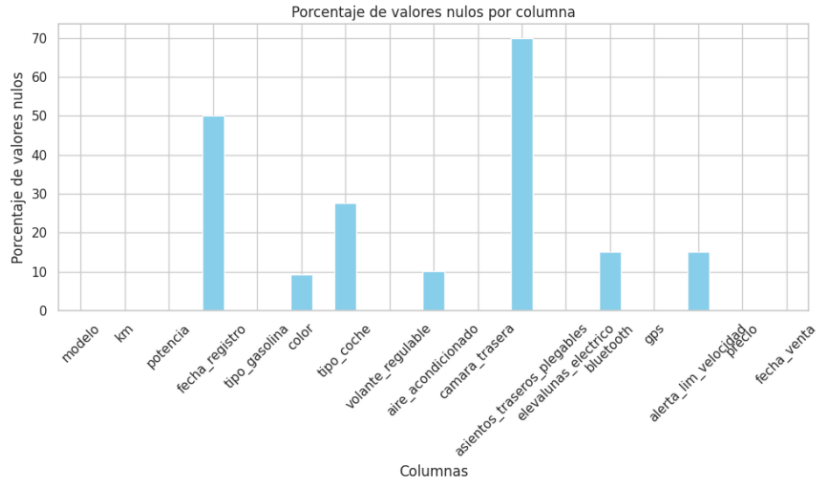
Andrés González Álvarez → https://github.com/andresgit23/BMW_dataset

**¿Qué columnas eliminaron (en caso se haya eliminado)?**

Se elimina la columna categórica **"marca"**, ya que entiendo que es un dataset de coches de la misma marca BMW, y todos los registros indican modelos BMW.

Se han eliminado las columnas categóricas **"asientos_traseros_plegables"** (nulos=70%) y **"fecha_registro"** (nulos=50%). Se justifica antendiendo a que un alto % de nulos se asocia a falta de representatividad, posible impacto en el análisis, reducción del ruido y complejidad del proceso de imputación:

**¿Qué se hizo con los nulos y cómo se limpiaron las columnas?**




**"modelo":** se detectan 3 nulos en modelo (0.06%): a dos de ellos se les imputó "sin_modelo_informado", y al registro nulo restante se imputa el modelo "X3" ya que en la columna tipo_coche aparece "suv", y solo existe el modelo "X3" para éste tipo de coche, es un valor único.

**"km":** se detectan 2 nulos en km (0.04%); ya que el histograma no muestra una distribución normal, reemplazamos los dos null por la mediana de todos los registros.

**"potencia":** se detecta 1 nulo en potencia (0.02%), y ya que el histograma no muestra una distribución normal, reemplazamos los dos null por la mediana de todos los registros.

**"tipo_gasolina":** se detectan 5 nulos (0.10%), se imputa "tipo_gasolina_desconocido".

**"color":** se detectan 445 nulos (9.18%), se imputa a "Sin_color"

**"tipo_coche":** se detectan 1335 nulos (27.56%), se imputa a "Sin_color"

A las columnas **volante_regulable** (4 nulos, 0.08%)**, aire_acondicionado** (486, 10.03%), **camara_trasera** (2 nulos, 0.04%), **elevalunas_electrico** (2 nulos, 0.04%), **bluetooth** (728 nulos, 15.03%), **alerta_lim_velocidad** (728 nulos, 15.03%) les asigno -1 ya que son booleanas al adoptar "True" y "False".

**"fecha_venta":** valoré hacer una resta entre esta fecha y la fecha de registro (eliminada por alto % de nulos), pero imputar la mediana (valor central), incluir una proporción tan alta de valores faltantes puede introducir sesgos o errores significativos en el conjunto de datos. La fecha de registro no se sabe si está referida a la fecha de registro en un comercio de venta o si es la fecha de matriculación, por ejemplo.

En “fecha_venta”, se dividió la columna en [año_venta, mes_venta, día_venta]. [día_venta] se eliminó (aparecen los días uno de cada mes únicamente) y [año_venta] se convirtió a [año_venta_2018] y se convirtió a dummy de acuerdo el valor. Los nulos de mes_venta y año venta se imputaron con -1.

**“precio”:** imputamos los nulos con la mediana ya que la distribución no es normal.

**Comentarios del análisis univariable, están todas ok? Hay alguna con outliers? Hay alguna por agrupar?**

**“km”:** tenemos valores outliers (limite superior: 283618.5, límite inferior: -5509.5) y un negativo que son tratados imputando la mediana ya que la distribución no es normal. El límite superior para outliers se situó en 1.000.000.

**“potencia**”: en outliers (límite superior: 187,5 y límite inferior_ 66,5), se imputa la mediana pero el límite superior se estableció en 200.

**“fecha_registro” y “fecha_venta”:** para [fecha_venta], se dividió la columna en [año_venta, mes_venta, día_venta]. [día_venta] se eliminó (solo unos) y [año_venta] se convirtió a [año_venta_2018] y se convirtió a dummy de acuerdo el valor.

**“precio” (target):**

Se creó una variable aparte para normalizar la original mediante logaritmo (log_precio) para almacenar la transformación; la distribución se acerca más a una normal pero está influenciada por los coches con precio menor a 20.000 euros. que hacen que la distribución no sea tan normal como se puede apreciar a continuación:



**¿Análisis de Correlación inicial, hay alguna variable correlacionada?**

Se aprecia cierta correlación entre precio y potencia (63.9%) pero no produce alerta de que sean variables altamente correlacionadas (establezco un límite de alerta de 80-90% de correlación).

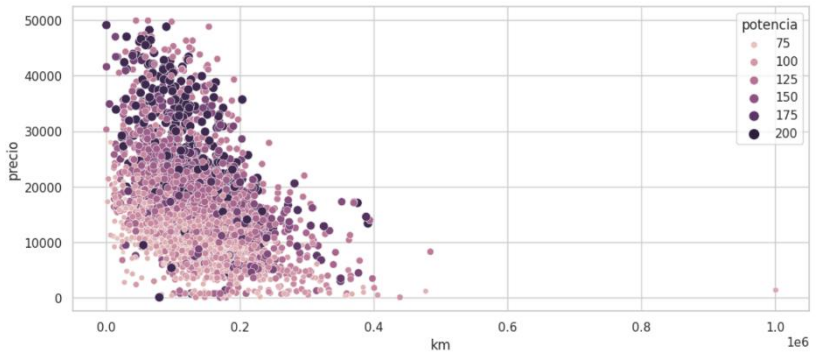
**¿Análisis variable vs target, hay algún insight interesante?**

No hay correlaciones entre variables ni contra el target.

**Km vs precio:** se puede estimar que a mayor [km] el coche tiene menor precio.

**potencia vs precio:** una mayor potencia, a simple vista, no implica a un mayor precio.

**potencia y Km vs precio:** al comparar simultáneamente potencia y km con precio, para coches con un registro de km similar, la incidencia supone un plus o diferencia a mayores.




**modelo vs precio:** el modelo BMw X3 es algo más caro que el resto.

**tipo_gasolina vs precio;** el grupo tipo_gasolina_otro (agrupa coches eléctricos e híbridos), es algo más caro que los otros tipos de gasolina.

**¿Transformación de categóricas a numéricas, que variables van a transformar? técnica se va usar?**

Se transforman 4 variables categóricas; 'modelo', 'tipo_gasolina', 'color', 'tipo_coche'. Se aplica One Hot Encoding para tipo_coche, y Label Encoder para modelo, color, tipo_gasolina.

**Normalizar variables numéricas**

Se utiliza un MinMaxScaler para las tres variables numéricas: “km”, “potencia”. Se hizo un Label Encoder para [modelo, color, tipo_gasolina], un MinMax Scaler para [km, potencia] y One Hot Encoding para [tipo_coche]. El resto de las columnas se dejó con sus valores numéricos. No hay correlación final entre variables.

**¿Análisis de correlación final, hay alguna variable correlacionada?**

Se detecta una correlación entendible por la transformación de precio a log_precio;  correlación muy alta entre precio y log_precio (91.9)%. Correlación tipo_gasolina_electrica y modelo_i3 (70%)

```
df_bmw14.head().T
```

| | 0 | 1 | 2 | 3 | 4 |
|---|---|---|---|---|---|
| modelo | 1.000000 | 22.000000 | 6.000000 | 9.000000 | 22.000000 |
| km | 0.140413 | 0.013987 | 0.183280 | 0.128043 | 0.097118 |
| potencia | 0.253731 | 0.402985 | 0.402985 | 0.514925 | 0.701493 |
| tipo_gasolina | 0.000000 | 2.000000 | 0.000000 | 0.000000 | 0.000000 |
| color | 2.000000 | 6.000000 | 10.000000 | 8.000000 | 9.000000 |
| volante_regulable | 1.000000 | 1.000000 | 0.000000 | 1.000000 | 1.000000 |
| aire_acondicionado | 1.000000 | 1.000000 | 0.000000 | 1.000000 | 1.000000 |
| camara_trasera | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| elevalunas_electrico | 1.000000 | 0.000000 | 1.000000 | 1.000000 | 0.000000 |
| bluetooth | -1.000000 | 1.000000 | 0.000000 | 1.000000 | 1.000000 |
| gps | 1.000000 | 1.000000 | 1.000000 | 1.000000 | 1.000000 |
| alerta_lim_velocidad | -1.000000 | 1.000000 | 0.000000 | -1.000000 | 1.000000 |
| año_venta_2018 | 1.000000 | 1.000000 | 1.000000 | 1.000000 | 1.000000 |
| mes_venta | 1.000000 | 2.000000 | 2.000000 | 2.000000 | 4.000000 |
| log_precio | 4.053078 | 4.843233 | 4.008600 | 4.399674 | 4.523746 |
| tipo_coche_convertible | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| tipo_coche_coupe | 0.000000 | 0.000000 | 0.000000 | 1.000000 | 0.000000 |
| tipo_coche_estate | 0.000000 | 0.000000 | 1.000000 | 0.000000 | 0.000000 |
| tipo_coche_hatchback | 1.000000 | 0.000000 | 0.000000 | 0.000000 | 1.000000 |
| tipo_coche_sedan | 0.000000 | 1.000000 | 0.000000 | 0.000000 | 0.000000 |
| tipo_coche_subcompact | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| tipo_coche_suv | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| tipo_coche_van | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |

```
[253] df_bmw14.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4843 entries, 0 to 4842
Data columns (total 23 columns):
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   modelo                  4843 non-null   int8
 1   km                      4841 non-null   float64
 2   potencia                4843 non-null   float64
 3   tipo_gasolina           4843 non-null   int8
 4   color                   4843 non-null   int8
 5   volante_regulable       4843 non-null   int8
 6   aire_acondicionado      4843 non-null   int8
 7   camara_trasera          4843 non-null   int8
 8   elevalunas_electrico    4843 non-null   int8
 9   bluetooth               4843 non-null   int8
 10  gps                     4843 non-null   int8
 11  alerta_lim_velocidad    4843 non-null   int8
 12  año_venta_2018          4843 non-null   int8
 13  mes_venta               4843 non-null   int8
 14  log_precio              4843 non-null   float64
 15  tipo_coche_convertible  4843 non-null   uint8
 16  tipo_coche_coupe        4843 non-null   uint8
 17  tipo_coche_estate       4843 non-null   uint8
 18  tipo_coche_hatchback    4843 non-null   uint8
 19  tipo_coche_sedan        4843 non-null   uint8
 20  tipo_coche_subcompact   4843 non-null   uint8
 21  tipo_coche_suv          4843 non-null   uint8
 22  tipo_coche_van          4843 non-null   uint8
dtypes: float64(3), int8(12), uint8(8)
memory usage: 208.2 KB
```