Facultad de
Cs. de la Computación

Introducción a la Ciencia de Datos.

Proyecto Final: Modelado Predictivo de Alta Precisión para el Sector Automotriz.

Nombre del Alumno: Daniel Carvajal Garcia.
Matrícula: 202502492
Nombre del proyecto: Modelado Predictivo de Alta Precisión para el Sector Automotriz.
Fecha de entrega: 29 de Noviembre de 2025
Nombre del Profesor: Jaime Alejandro Romero Sierra.
Materia: Introducción a la Ciencia de Datos.
Licenciatura: Ingeniería en Ciencia de Datos.
Benemérita Universidad Autónoma de Puebla.

Introducción.

- **Descripción breve del objetivo del proyecto.**
  El objetivo principal de este análisis de datos sobre venta de vehículos es aprovechar la información disponible para mejorar la estrategia comercial de cada fabricante de automóviles, con el propósito de tener una planeación en la toma de decisiones estratégicas dentro del mercado automotriz, a fin de incrementar ventas, mejorar la competitividad y fortalecer el posicionamiento de cada fabricante en un lapso de 12 meses.

- **Justificación y contexto: ¿por qué es importante resolver o estudiar esta problemática?** Con este proyecto se busca resolver las siguientes problemáticas:

  1. Definir estrategias que permita el posicionamiento de cada fabricante.

  Analizar diferentes factores de los cuales incluye: marca, modelo, año, tipo de combustible o si es híbrido, tamaño de motor, de los cuales influye un precio de venta para el consumidor. Tener claro una estrategia de precios que permita ser competitivo, maximice sus ventas y maximice un margen de ganancias para cada segmento.

  2. Detectar tendencias de compra y preferencias de los consumidores.

  Se busca evaluar que fabricantes y modelos de automóviles tienen mayor rentabilidad dentro del mercado, identificar las preferencias de los consumidores en base al tipo de combustible (gasolina o diésel) o si prefiere los autos híbridos y anticipar la evolución de la demanda de cada uno. Reconocer patrones de compra, por ejemplo, cuáles son los meses del año en donde el consumidor llega a comprar un automóvil, con el objetivo de planificar lanzamiento y campañas de marketing.

  3. Impulsar decisiones estratégicas a través de datos.

  Comparar el desempeño de ventas entre los distintos fabricantes y sus diferentes modelos, del mismo modo, detectar cuales las fortalezas y debilidades de cada fabricante, como puede ser el precio, motor, tipo de combustible, apoyar en la definición de estrategias de diferenciación para cada fabricante.

- **Fuentes de datos: descripción de las bases de datos empleadas (origen, cantidad de datos, principales características)**
  Fuente de la base de datos original:
  https://www.kaggle.com/datasets/msnbehdani/mock-dataset-of-second-hand-car-sales
  Este conjunto de datos contiene información detallada sobre la venta de automóviles, abarcando múltiples fabricantes y modelos. Es idóneo para el análisis de datos, la predicción de precios, el análisis de tendencias del mercado, el aprendizaje automático y el análisis exploratorio de datos (EDA).

Metodología.

- En este punto se hablará acerca del seguimiento de limpieza de datos del DataFrame, esto con el propósito de que no se realizo una limpieza adecuada con los datos sucios.

Seguimiento en el Proceso de limpieza

Se continua con el DataFrame que se dejo aparentemente limpio, sin embargo, se debe continuar con la limpieza porque todavía no esta limpia al 100%.

```python
#Extraemos el DataFrame aparentemente limpio,para seguir con la impieza correctamente para hacer un análisis de la mejor manera

import pandas as pd
df = pd.read_csv("Base_limpiaDaniel.csv")
df
```

[1] Python

	Unnamed: 0	Manufacturer	Model	Engine size	Fuel type	Year of manufacture	Mileage	Price
0	0	ford	fiesta	1.0	petrol	2002.0	127300.0	3074.0
1	1	porsche	718 cayman	4.0	petrol	2016.0	57850.0	49704.0
2	2	ford	mondeo	0.0	diesel	2014.0	39190.0	24072.0
3	3	toyota	rav4	1.8	hybrid	1988.0	210814.0	1705.0
4	4	vw	polo	1.0	petrol	2006.0	127869.0	4101.0
...	...	...	...	...	...	...	...	...
54645	59899	porsche	718 cayman	2.0	petrol	2012.0	87055.0	22114.0
54646	59900	porsche	718 cayman	0.0	petrol	2018.0	20634.0	70913.0
54647	59901	vw	polo	0.0	petrol	2011.0	89765.0	7610.0
54648	59902	ford	mondeo	1.6	diesel	2010.0	67468.0	15428.0
54649	59903	ford	focus	0.0	petrol	2020.0	17260.0	30921.0

54650 rows × 8 columns

0 ⚠ 27 Cell 2 of 18

```python
#2 Eliminar columna 'Unnamed: 0', para no arrastrar basura

if 'Unnamed: 0' in df.columns:
    df = df.drop(columns=['Unnamed: 0'])

df.head(10)
```

[2] Python

	Manufacturer	Model	Engine size	Fuel type	Year of manufacture	Mileage	Price
0	ford	fiesta	1.0	petrol	2002.0	127300.0	3074.0
1	porsche	718 cayman	4.0	petrol	2016.0	57850.0	49704.0
2	ford	mondeo	0.0	diesel	2014.0	39190.0	24072.0
3	toyota	rav4	1.8	hybrid	1988.0	210814.0	1705.0
4	vw	polo	1.0	petrol	2006.0	127869.0	4101.0
5	ford	focus	1.4	petrol	2018.0	33603.0	29204.0
6	ford	mondeo	1.8	diesel	2010.0	86686.0	14350.0
7	toyota	prius	1.4	hybrid	2015.0	0.0	30297.0
8	desconocido	polo	1.2	petrol	2012.0	73470.0	9977.0
9	ford	focus	2.0	desconocido	1992.0	262514.0	1049.0

```python
# 3. Estandarización de Texto (Vital para evitar duplicados)

cols_texto = ['Manufacturer', 'Model', 'Fuel type']
for col in cols_texto:
    # Convertir a minúsculas y quitar espacios al inicio/final
    df[col] = df[col].astype(str).str.lower().str.strip()
```

[3] Python

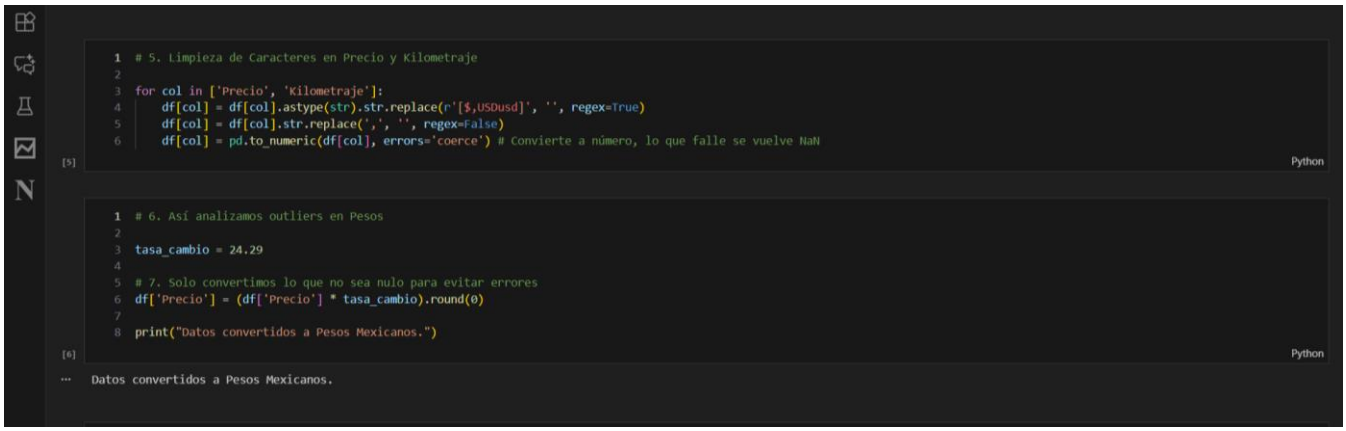
```python
# 4. Renombramos las columnas para un mejor manejo del análisis

df = df.rename(columns={'Manufacturer': 'Fabricante'})
df = df.rename(columns={'Model': 'Modelo'})
df=df.rename(columns= {'Engine size': 'Tamaño del motor'})
df=df.rename(columns={ 'Fuel type': 'Tipo de combustible' })
df=df.rename(columns={'Year of manufacture': 'Año de fabricacion'})
df=df.rename(columns={  'Mileage': 'Kilometraje'})
df = df.rename(columns={'Price': 'Precio'})
df
```

[4] Python

	Fabricante	Modelo	Tamaño del motor	Tipo de combustible	Año de fabricacion	Kilometraje	Precio
0	ford	fiesta	1.0	petrol	2002.0	127300.0	3074.0
1	porsche	718 cayman	4.0	petrol	2016.0	57850.0	49704.0
2	ford	mondeo	0.0	diesel	2014.0	39190.0	24072.0
3	toyota	rav4	1.8	hybrid	1988.0	210814.0	1705.0
4	vw	polo	1.0	petrol	2006.0	127869.0	4101.0
...	...	...	...	...	...	...	...
54645	porsche	718 cayman	2.0	petrol	2012.0	87055.0	22114.0
54646	porsche	718 cayman	0.0	petrol	2018.0	20634.0	70913.0
54647	vw	polo	0.0	petrol	2011.0	89765.0	7610.0
54648	ford	mondeo	1.6	diesel	2010.0	67468.0	15428.0
54649	ford	focus	0.0	petrol	2020.0	17260.0	30921.0

54650 rows × 7 columns

- **Estandarización de Variables y Traducción:**
  Para facilitar la interpretación del análisis, se renombraron todas las columnas al español (ej. 'Manufacturer' a 'Fabricante', 'Price' a 'Precio').
  Normalización de Texto: Se aplicó una conversión a minúsculas y eliminación de espacios en blanco (strip()) en las variables categóricas.
  Justificación: Esto es vital para evitar categorías duplicadas por errores de formato (ej. evitar que "Ford" y "ford " cuenten como marcas distintas).



```
# 5. Limpieza de Caracteres en Precio y Kilometraje

for col in ['Precio', 'Kilometraje']:
    df[col] = df[col].astype(str).str.replace(r'[$,USDusd]', '', regex=True)
    df[col] = df[col].str.replace(',', '', regex=False)
    df[col] = pd.to_numeric(df[col], errors='coerce') # Convierte a número, lo que falle se vuelve NaN
```

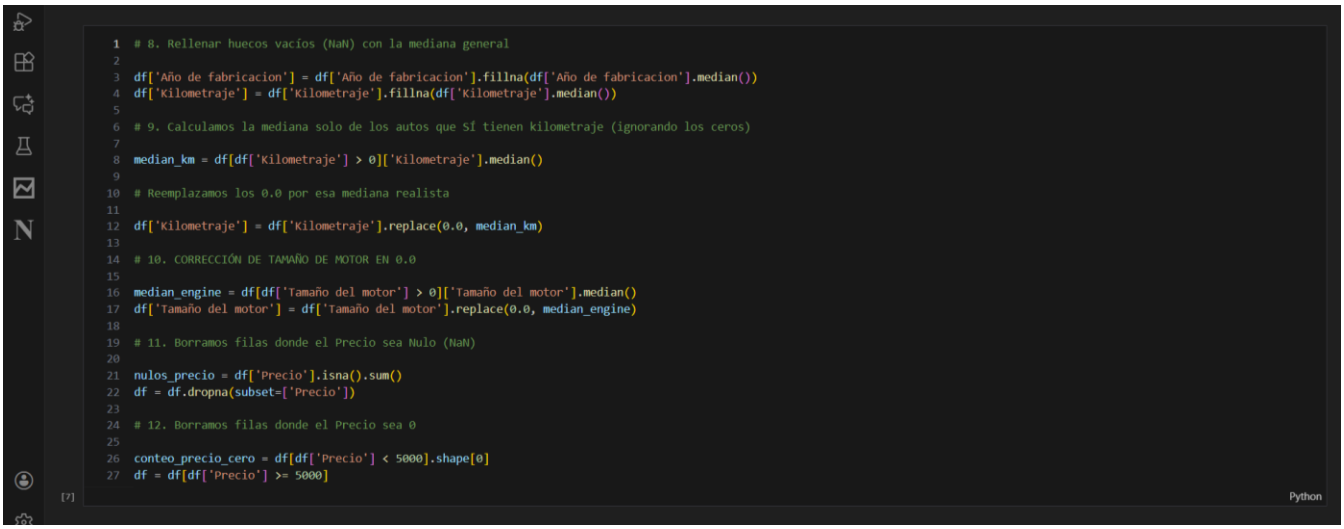
[5] Python

```
# 6. Así analizamos outliers en Pesos

tasa_cambio = 24.29

# 7. Solo convertimos lo que no sea nulo para evitar errores
df['Precio'] = (df['Precio'] * tasa_cambio).round(0)

print("Datos convertidos a Pesos Mexicanos.")
```

[6] Python

```
Datos convertidos a Pesos Mexicanos.
```

- **Limpieza de Caracteres y Conversión Monetaria:**
  Las columnas numéricas contenían "ruido" en forma de caracteres especiales ($, USD)
  Acción: Se limpiaron estos caracteres y se convirtieron los tipos de datos a numéricos, posteriormente, se realizó una conversión de moneda multiplicando el precio por una tasa de cambio de 24.29 para transformar los valores de dólares a pesos mexicanos (MXN).
  Justificación: Adaptar el análisis al contexto económico local (MXN) hace que los resultados sean comprensibles para la audiencia objetivo en México.



```
# 8. Rellenar huecos vacíos (NaN) con la mediana general

df['Año de fabricacion'] = df['Año de fabricacion'].fillna(df['Año de fabricacion'].median())
df['Kilometraje'] = df['Kilometraje'].fillna(df['Kilometraje'].median())

# 9. Calculamos la mediana solo de los autos que SÍ tienen kilometraje (ignorando los ceros)

median_km = df[df['Kilometraje'] > 0]['Kilometraje'].median()

# Reemplazamos los 0.0 por esa mediana realista

df['Kilometraje'] = df['Kilometraje'].replace(0.0, median_km)

# 10. CORRECCIÓN DE TAMAÑO DE MOTOR EN 0.0

median_engine = df[df['Tamaño del motor'] > 0]['Tamaño del motor'].median()
df['Tamaño del motor'] = df['Tamaño del motor'].replace(0.0, median_engine)

# 11. Borramos filas donde el Precio sea Nulo (NaN)

nulos_precio = df['Precio'].isna().sum()
df = df.dropna(subset=['Precio'])

# 12. Borramos filas donde el Precio sea 0

conteo_precio_cero = df[df['Precio'] < 5000].shape[0]
df = df[df['Precio'] >= 5000]
```

[7] Python

- **Imputación Estadística de Datos Faltantes e Incongruentes:**
  Se detectaron registros con valor 0.0 en 'Kilometraje' y 'Tamaño del motor', así como valores nulos (NaN).
  Estrategia: Se reemplazaron los valores 0.0 y nulos utilizando la mediana específica de cada columna (calculada ignorando los ceros).

Justificación: Se eligió la mediana en lugar del promedio porque es una métrica robusta que no se ve afectada por valores extremos. Esto permitió recuperar registros valiosos en lugar de eliminarlos, manteniendo el tamaño de la muestra.

```
# 12. Borramos filas donde el Precio sea 0

conteo_precio_cero = df[df['Precio'] < 5000].shape[0]
df = df[df['Precio'] >= 5000]
```

Python

```
# 13. Eliminar filas donde el año es ilógico (menor a 1988, lo que incluye el 0)

df = df[df['Año de fabricacion'] >= 1988]
```

Python

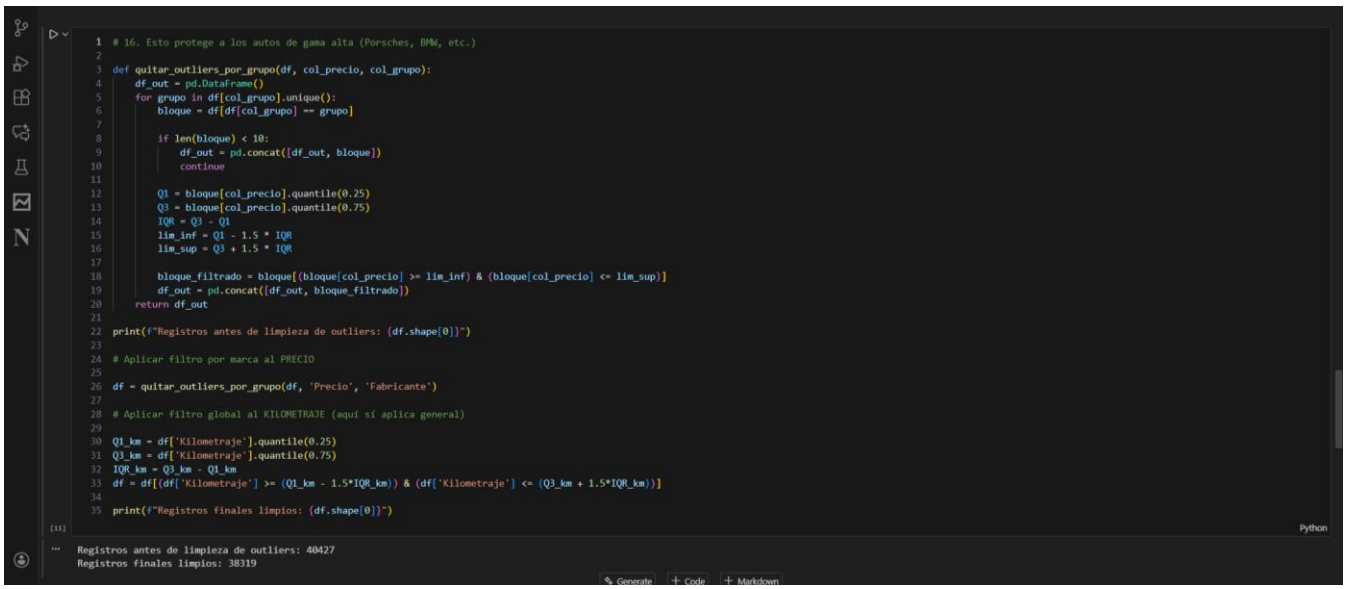
```
# 14. Eliminar registros "desconocido"

filtro_desconocido = (df['Fabricante'] != 'desconocido') & \
                     (df['Modelo'] != 'desconocido') & \
                     (df['Tipo de combustible'] != 'desconocido')
df = df[filtro_desconocido]
```

Python

```
# 15. Eliminar Duplicados
df = df.drop_duplicates()
```

Python

- **Filtrado por Reglas de Negocio**
  Se aplicaron filtros lógicos para eliminar "datos basura" que podrían confundir al modelo:
  Precios Irreales: Se eliminaron autos con precio menor a $5,000 MXN (considerados errores de captura o chatarra) y filas sin precio.
  Antigüedad: Se filtraron vehículos anteriores a 1988, enfocando el análisis en el mercado automotriz moderno y funcional.
  Datos Desconocidos: Se eliminaron registros donde el fabricante o modelo figuraba como "desconocido".

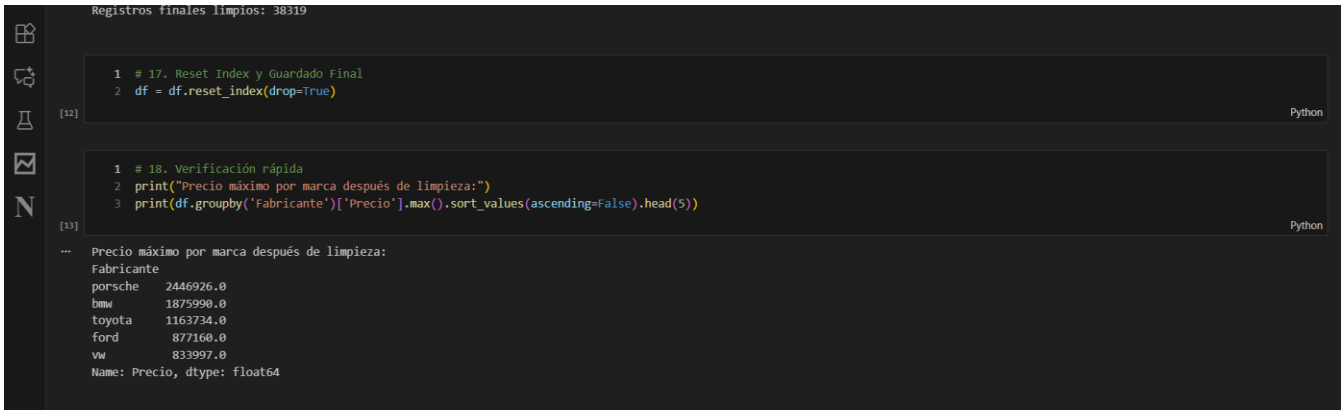


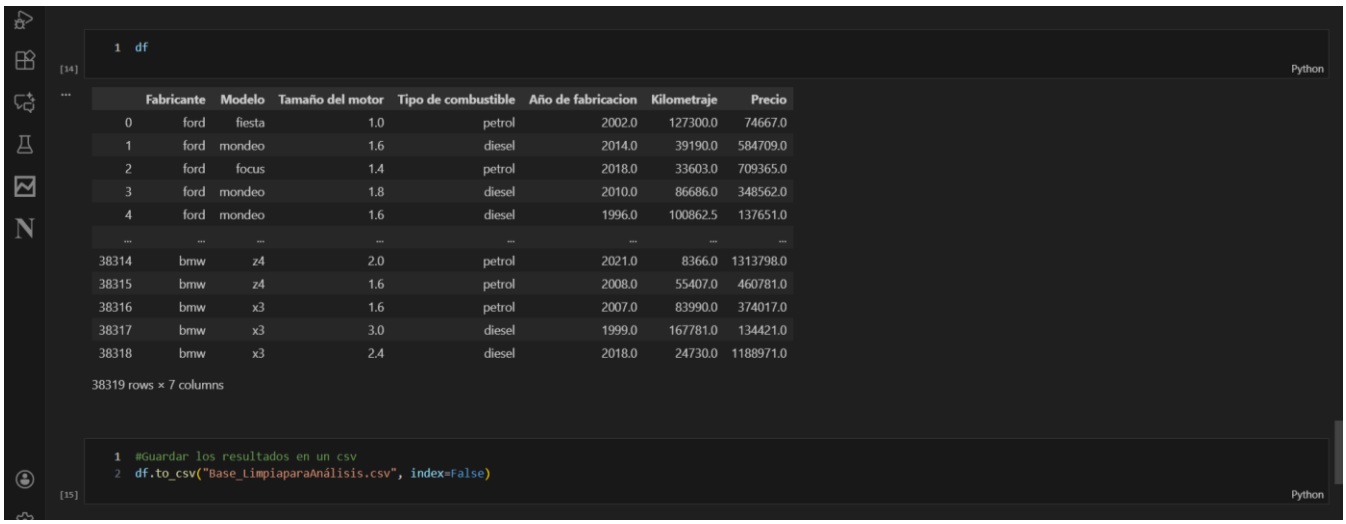
```
# 16. Esto protege a los autos de gama alta (Porsches, BMW, etc.)

def quitar_outliers_por_grupo(df, col_precio, col_grupo):
    df_out = pd.DataFrame()
    for grupo in df[col_grupo].unique():
        bloque = df[df[col_grupo] == grupo]

        if len(bloque) < 10:
            df_out = pd.concat([df_out, bloque])
            continue

        Q1 = bloque[col_precio].quantile(0.25)
        Q3 = bloque[col_precio].quantile(0.75)
        IQR = Q3 - Q1
        lim_inf = Q1 - 1.5 * IQR
        lim_sup = Q3 + 1.5 * IQR

        bloque_filtrado = bloque[(bloque[col_precio] >= lim_inf) & (bloque[col_precio] <= lim_sup)]
        df_out = pd.concat([df_out, bloque_filtrado])
    return df_out

print(f"Registros antes de limpieza de outliers: {df.shape[0]}")

# Aplicar filtro por marca al PRECIO

df = quitar_outliers_por_grupo(df, 'Precio', 'Fabricante')

# Aplicar filtro global al KILOMETRAJE (aquí sí aplica general)

Q1_km = df['Kilometraje'].quantile(0.25)
Q3_km = df['Kilometraje'].quantile(0.75)
IQR_km = Q3_km - Q1_km
df = df[(df['Kilometraje'] >= (Q1_km - 1.5*IQR_km)) & (df['Kilometraje'] <= (Q3_km + 1.5*IQR_km))]

print(f"Registros finales limpios: {df.shape[0]}")
```

Python

```
Registros antes de limpieza de outliers: 40427
Registros finales limpios: 38319
```

- **Eliminación de Valores Atípicos (Outliers) Segmentada:**
  Este fue el paso metodológico más importante.





El resultado final fue una base de datos reducida a 38,319 registros limpios. Este proceso fue exitoso porque logró equilibrar la eliminación de ruido (errores de captura) con la conservación de información crítica (autos de lujo), generando un dataset confiable para el entrenamiento del modelo predictivo.

Justificación para el Punto 4: Análisis de Valores Atípicos (Outliers).

Para dar cumplimiento al Punto 4 (Análisis de Valores Atípicos), se implementó una estrategia de detección y tratamiento en dos niveles dentro del código de limpieza:

Detección Lógica: Se identificaron valores extremos imposibles para el mercado, como vehículos con precios menores a $5,000 MXN o precios nulos. Estos fueron eliminados directamente por considerarse errores de captura o datos 'basura' que distorsionarían el entrenamiento.

Tratamiento Estadístico Segmentado (El Aporte Clave): En lugar de eliminar outliers usando un Rango Intercuartílico (IQR) global —lo cual habría borrado erróneamente vehículos de gama alta legítimos—, se aplicó una función de limpieza agrupada por 'Fabricante'.

Justificación: Se calculó el IQR específico para cada marca. Esto permitió que el algoritmo conservara precios altos (ej. $2 Millones) si pertenecían a marcas como Porsche o BMW (donde

esos precios son normales), pero eliminara precios similares si aparecían en marcas económicas como Ford (donde serían claramente un error).

Resultado: Se eliminó el ruido estadístico sin sacrificar la información de los segmentos de lujo, cumpliendo con el requisito de justificar la decisión de 'mantener' ciertos valores atípicos valiosos.

Justificación para el Punto 5: Análisis de Valores Faltantes

El cumplimiento del Punto 5 (Análisis de Valores Faltantes) se abordó mediante una estrategia de imputación estadística robusta, identificando no solo los valores nulos (NaN), sino también los 'nulos ocultos' (registros con valor 0.0 en campos donde es imposible, como kilometraje o motor).

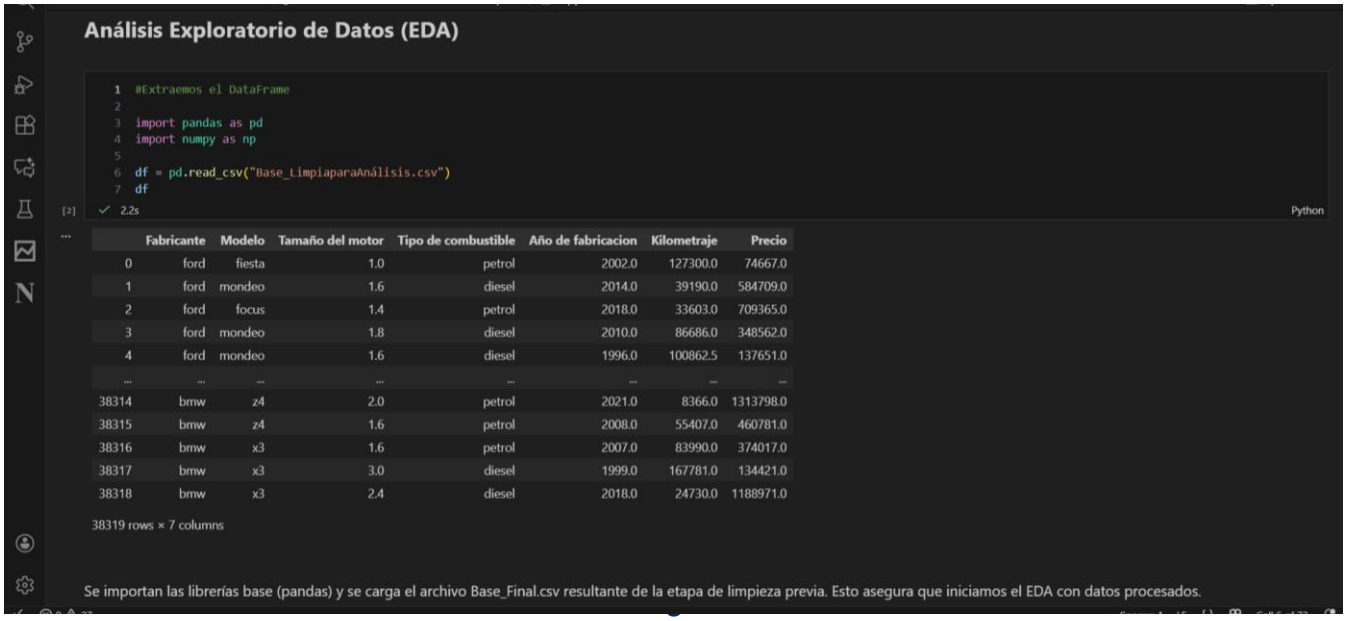
Estrategia Implementada:

Imputación por Mediana: Para las variables 'Kilometraje' y 'Tamaño del Motor', se detectaron valores en cero. Se optó por reemplazar estos ceros con la mediana de la variable (calculada excluyendo los ceros).

Justificación: Se seleccionó la mediana en lugar de la media (promedio) debido a que el dataset presentaba sesgos y valores atípicos. La mediana es una medida de tendencia central más resistente a estos extremos, garantizando que el valor imputado fuera representativo del 'auto típico' en el mercado.

Eliminación Selectiva: Para la variable objetivo ('Precio'), se eliminaron las filas con valores nulos, ya que imputar el precio (la respuesta que buscamos predecir) introduciría un sesgo artificial inaceptable en el modelo supervisado."

Análisis Exploratorio de Datos (EDA)



	Fabricante	Modelo	Tamaño del motor	Tipo de combustible	Año de fabricacion	Kilometraje	Precio
0	ford	fiesta	1.0	petrol	2002.0	127300.0	74667.0
1	ford	mondeo	1.6	diesel	2014.0	39190.0	584709.0
2	ford	focus	1.4	petrol	2018.0	33603.0	709365.0
3	ford	mondeo	1.8	diesel	2010.0	86686.0	348562.0
4	ford	mondeo	1.6	diesel	1996.0	100862.5	137651.0
...	...	...	...	...	...	...	...
38314	bmw	z4	2.0	petrol	2021.0	8366.0	1313798.0
38315	bmw	z4	1.6	petrol	2008.0	55407.0	460781.0
38316	bmw	x3	1.6	petrol	2007.0	83990.0	374017.0
38317	bmw	x3	3.0	diesel	1999.0	167781.0	134421.0
38318	bmw	x3	2.4	diesel	2018.0	24730.0	1188971.0

38319 rows × 7 columns

ANÁLISIS EXPLORATORIO (EDA)

El Análisis Exploratorio de Datos (EDA) es una fase esencial para comprender la estructura, patrones y relaciones dentro del dataset antes de construir el modelo. Esta sección debe dividirse en los siguientes apartados y acompañarse con gráficos, tablas, y explicaciones breves

1. Descripción general de los datos.

Visión General

```
1 df.shape
```

[7] ✓ 0.0s Python

```
(38319, 7)
```

Ejecución de df.shape para dimensionar el volumen de datos (registros vs variables) y confirmar el tamaño de la muestra, se confirma que el dataset tiene un total de 38319 datos y estan divididos en 7 columnas.

```
1 df.dtypes
```

[8] ✓ 0.1s Python

```
Fabricante              object
Modelo                  object
Tamaño del motor       float64
Tipo de combustible     object
Año de fabricacion       int64
Kilometraje              int64
Precio                   int64
dtype: object
```

Se rectifica que la columna: Fabricante, Modelo, Tipo de Combustible → son categóricas

Año de fabricación → es numérica

Kilometraje → es numérica

Precio → es numérica

Tamaño de motor → es numérica con punto flotante

3. Resumen estadístico

```
1 df.describe
```

[9] ✓ 0.6s Python

```
<bound method NDFrame.describe of      Fabricante  Modelo  Tamaño del motor Tipo de combustible  \
0           ford  fiesta              1.0             petrol
1           ford  mondeo              1.6             diesel
2           ford   focus              1.4             petrol
3           ford  mondeo              1.8             diesel
4           ford  mondeo              1.6             diesel
...          ...     ...              ...                ...
38314        bmw      z4              2.0             petrol
38315        bmw      z4              1.6             petrol
38316        bmw      x3              1.6             petrol
38317        bmw      x3              3.0             diesel
38318        bmw      x3              2.4             diesel

       Año de fabricacion  Kilometraje   Precio
0                    2002       127300    74667
1                    2014        39190   584709
2                    2018        33603   709365
3                    2010        86686   348562
4                    1996       100862   137651
...                   ...          ...      ...
38314                2021         8366  1313798
38315                2008        55407   460781
38316                2007        83990   374017
38317                1999       167781   134421
38318                2018        24730  1188971

[38319 rows x 7 columns]>
```

```
1 df.describe()
```

[17] ✓ 0.2s Python

	Tamaño del motor	Año de fabricacion	Kilometraje	Precio
count	38319.000000	38319.000000	38319.000000	3.831900e+04
mean	1.744490	2004.280253	109885.795402	3.017309e+05
std	0.694533	8.815940	61836.589280	3.084592e+05
min	1.000000	1988.000000	1000.000000	6.243000e+03
25%	1.400000	1997.000000	61474.000000	8.035100e+04
50%	1.600000	2004.000000	100862.000000	1.931300e+05
75%	2.000000	2012.000000	149751.500000	4.307470e+05
max	5.000000	2022.000000	287301.000000	2.446926e+06

Resumen estadístico para variables categoricos

```
df["Fabricante"].value_counts()
```

[11] ✓ 0.3s Python

```
Fabricante
vw          11454
ford        11452
toyota       9659
bmw          3761
porsche      1993
Name: count, dtype: int64
```

```
df["Modelo"].value_counts()
```

[12] ✓ 0.0s Python

```
Modelo
golf          3974
polo          3952
fiesta        3908
focus         3775
mondeo        3769
passat        3528
yaris         3297
prius         3254
rav4          3108
z4            1359
x3            1319
m5            1083
718 cayman     681
cayenne        676
```

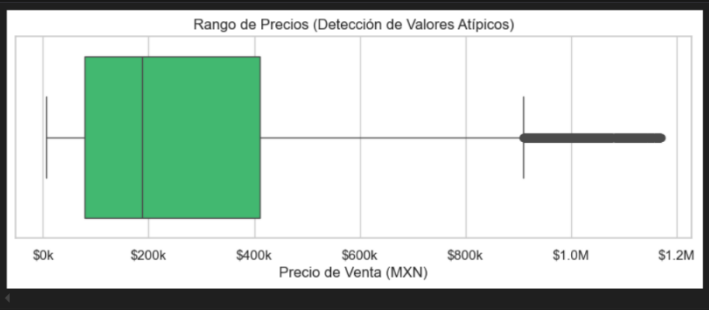
Spaces: 4 LF { } Cell 26 of 73

Creación de los gráficos.

BOXPLOST e HISTOGRAMA de Precio

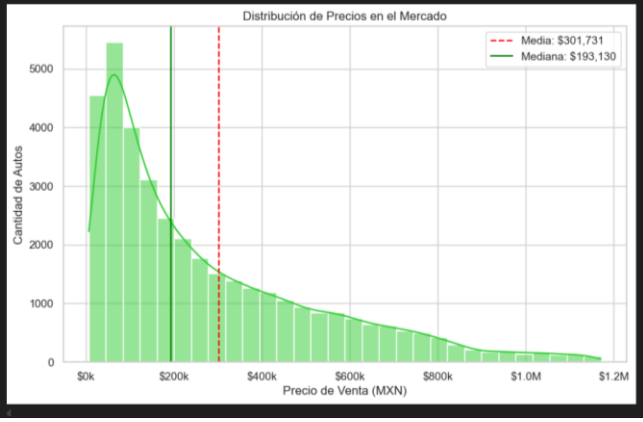
```
# Filtro visual
data_zoom = df[df['Precio'] <= df['Precio'].quantile(0.98)]

plt.figure(figsize=(10, 3))
sns.boxplot(x=data_zoom['Precio'], color="#2ecc6b")

plt.title('Rango de Precios (Detección de Valores Atípicos)')
plt.xlabel('Precio de Venta (MXN)')
plt.gca().xaxis.set_major_formatter(ticker.FuncFormatter(formato_dinero))
plt.show()
```

[20] ✓ 3.7s Python

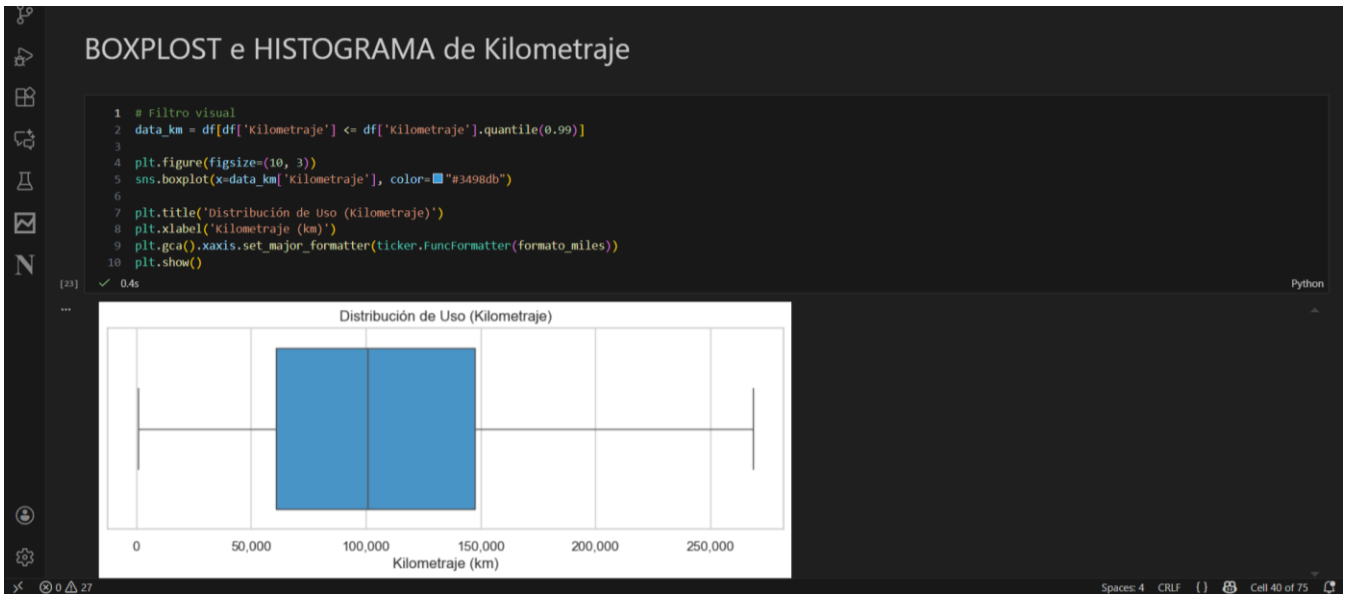


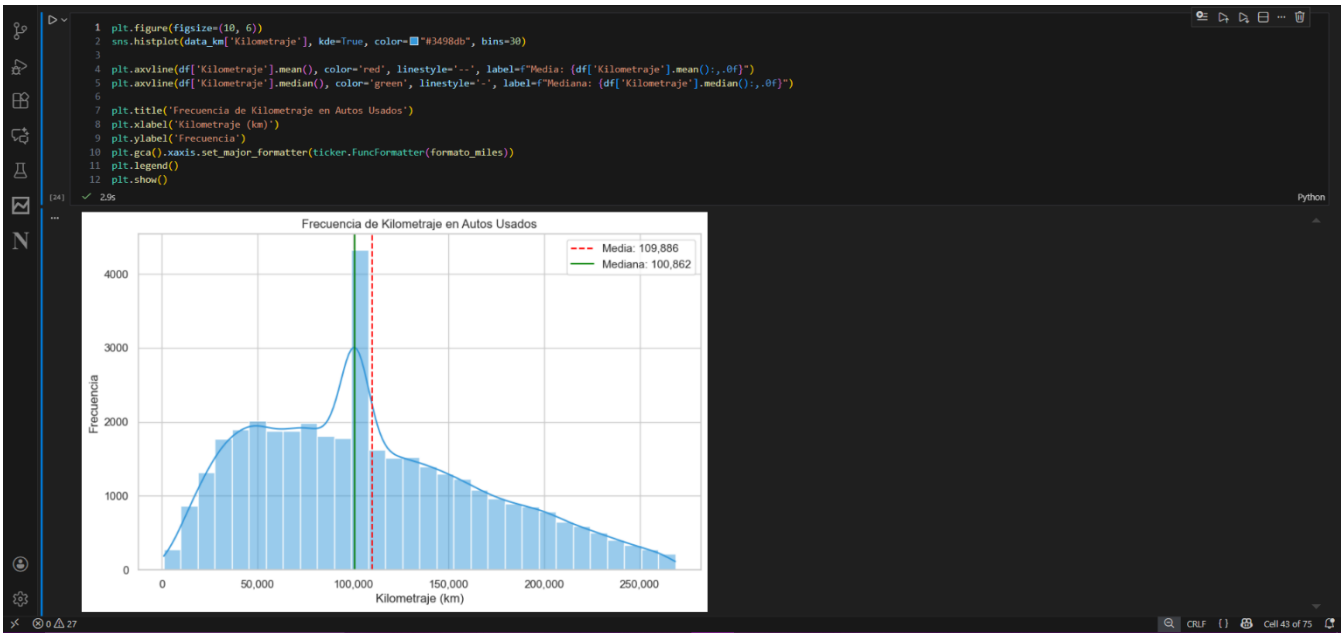

```
plt.figure(figsize=(10, 6))
sns.histplot(data_zoom['Precio'], kde=True, color="#2ecc31", bins=30)

plt.axvline(df['Precio'].mean(), color='red', linestyle='--', label=f"Media: ${df['Precio'].mean():,.0f}")
plt.axvline(df['Precio'].median(), color='green', linestyle='-', label=f"Mediana: ${df['Precio'].median():,.0f}")

plt.title('Distribución de Precios en el Mercado')
plt.xlabel('Precio de Venta (MXN)')
plt.ylabel('Cantidad de Autos')
plt.gca().xaxis.set_major_formatter(ticker.FuncFormatter(formato_dinero))
plt.legend()
plt.show()
```

[22] ✓ 5.2s Python



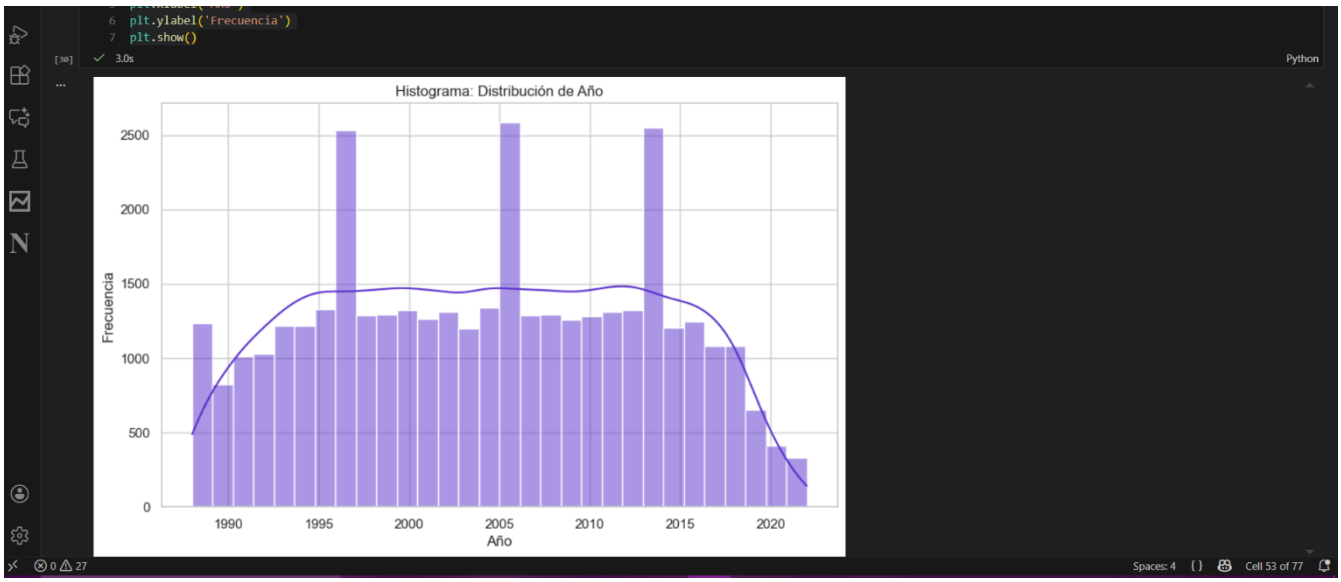
La variable objetivo ('Precio') presenta una distribución sesgada a la derecha. La mayor concentración de vehículos (moda) se encuentra en el rango de $100k a $300k MXN, lo que define el mercado masivo. Hallazgo de Negocio: El Boxplot superior muestra 'outliers' hacia la derecha. Estos no son errores, sino vehículos de gama alta (BMW, Porsche) que, aunque son menos frecuentes, tienen precios exponencialmente más altos.
Hallazgo: Se decidió aplicar un filtro visual del 98% para este gráfico, lo que nos permite observar el comportamiento del mercado masivo sin la distorsión de los superdeportivos, confirmando que la base de datos es representativa del comprador promedio.



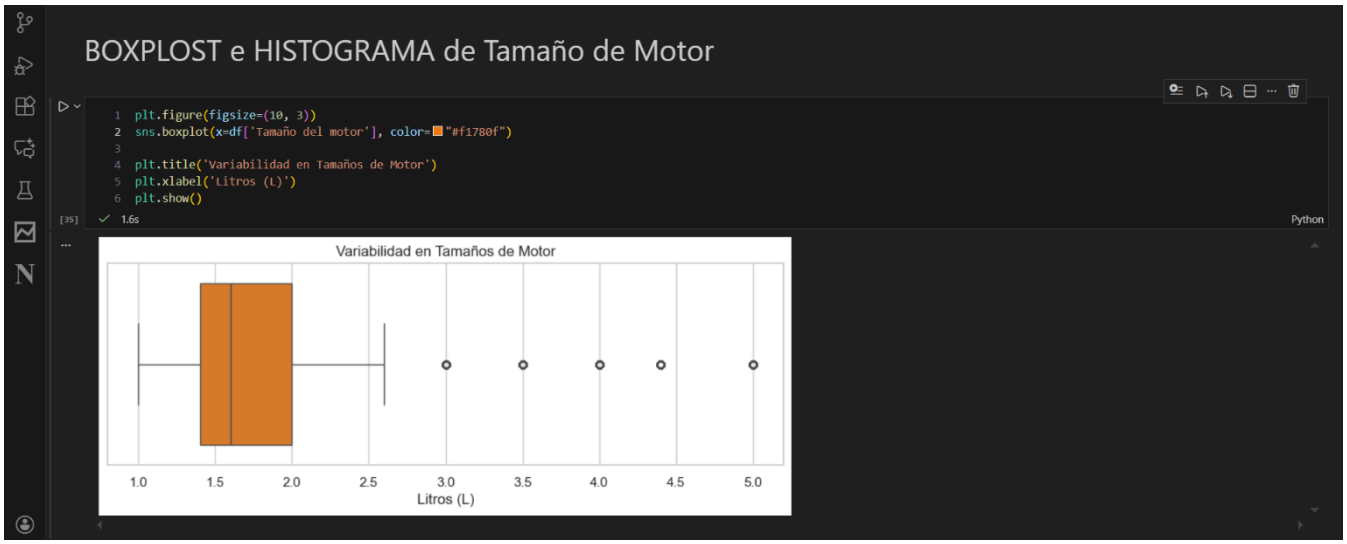


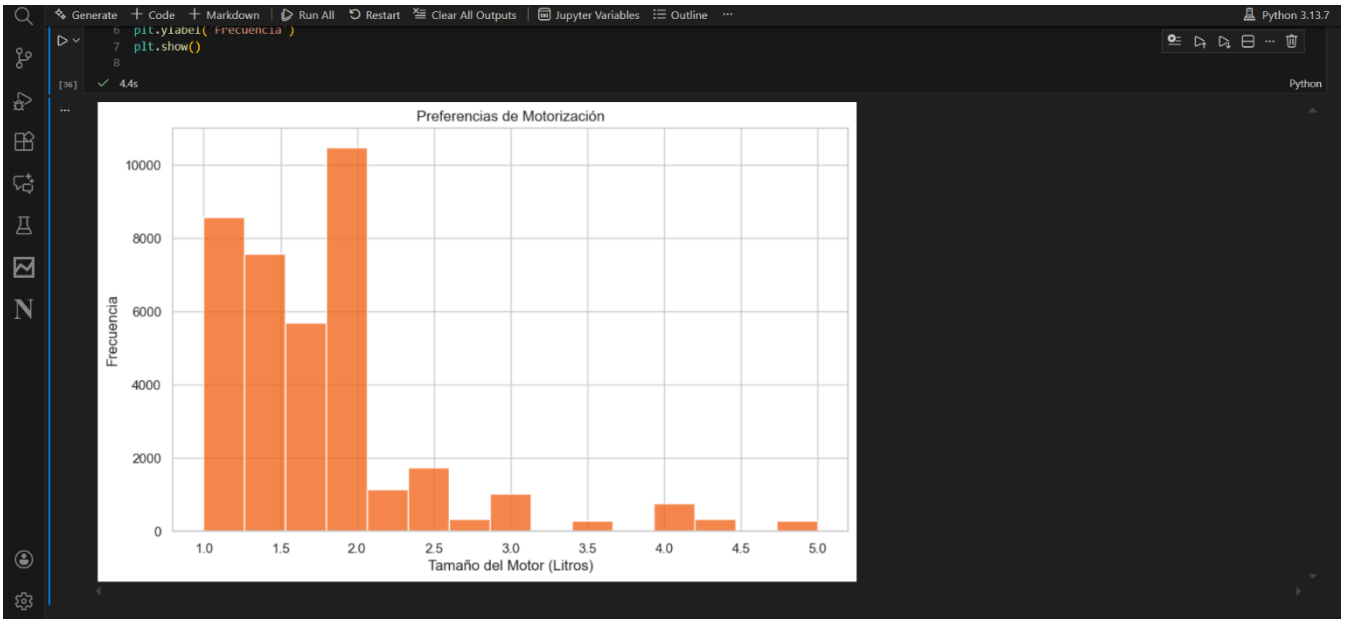
El histograma muestra que la distribución del kilometraje se concentra fuertemente alrededor de los 90,000 a 100,000 km (donde se ubican la media y la mediana). A diferencia de lo que se esperaría en un lote de autos 'seminuevos', la curva está desplazada hacia la derecha.

La caja central del diagrama (que representa el 50% de los datos) abarca un rango amplio, indicando una gran variabilidad en el uso de los vehículos. Se observan valores desde autos con muy poco uso hasta unidades que superan los 150,000 km.



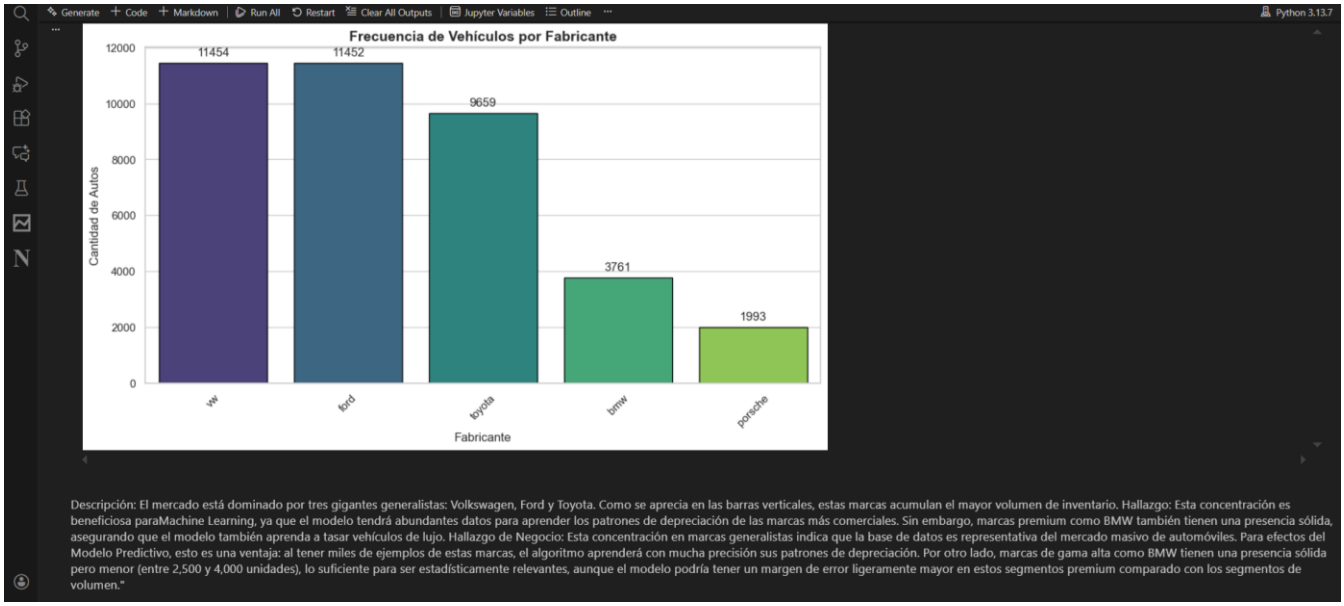
Se observa una distribución con sesgo a la izquierda, lo que indica que la gran mayoría de los datos se concentran en años pasados. El pico de la distribución se encuentra en los modelos de los últimos 20 a 10 años, disminuyendo progresivamente conforme avancemos en el tiempo. Los autos con mayor antigüedad (modelos de los 90s o principios de los 2000s) tienen una frecuencia alta y baja a la vez.



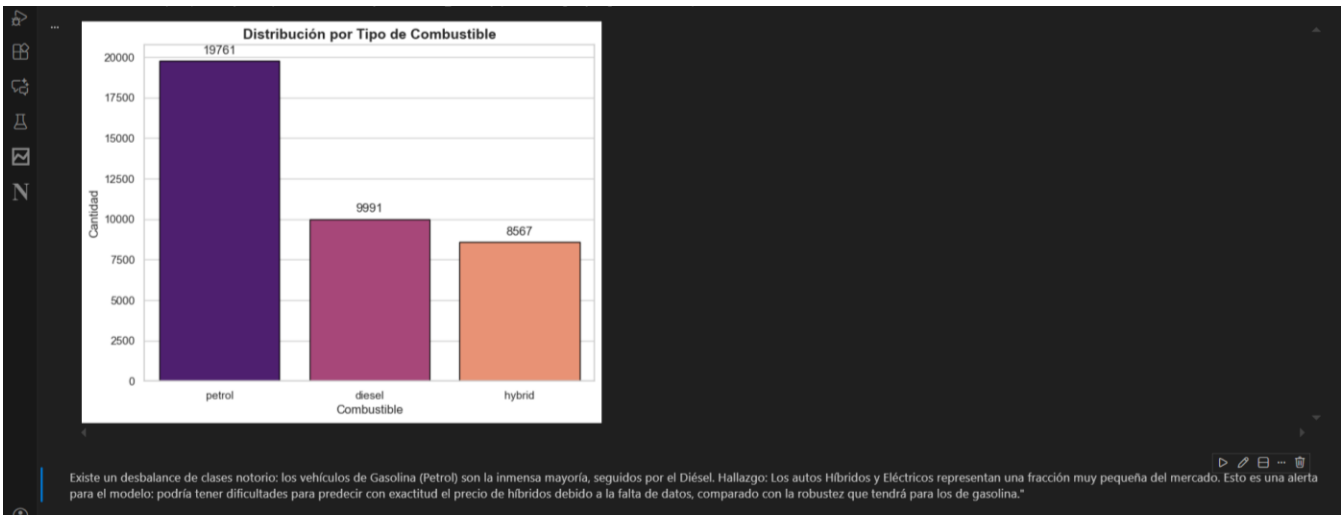


La distribución del tamaño del motor no es continua, sino multimodal, concentrándose en valores estándar de la industria (1.0L, 1.2L, 1.6L y 2.0L). Se observa un claro predominio de motores pequeños (entre 1.0 y 2.0 litros), lo cual es consistente con la alta presencia de marcas generalistas como Ford y VW en el dataset. El diagrama de caja muestra valores atípicos hacia la derecha (motores de 3.0L o más), que corresponden a los vehículos deportivos o de lujo que decidimos conservar.

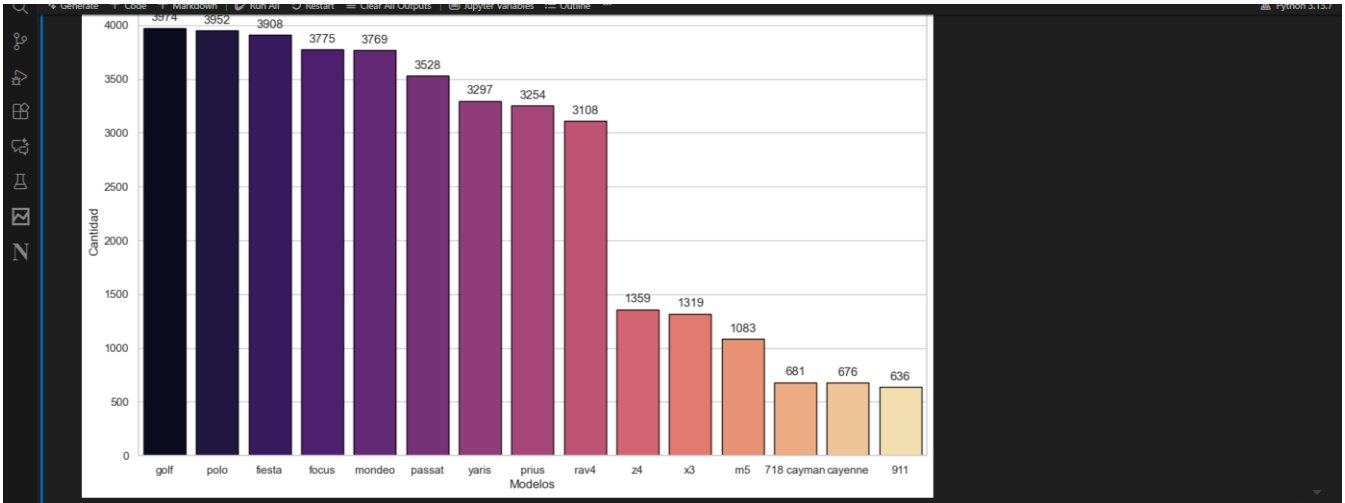
Grafico de barras de variables categóricas.



Descripción: El mercado está dominado por tres gigantes generalistas: Volkswagen, Ford y Toyota. Como se aprecia en las barras verticales, estas marcas acumulan el mayor volumen de inventario. Hallazgo: Esta concentración es beneficiosa paraMachine Learning, ya que el modelo tendrá abundantes datos para aprender los patrones de depreciación de las marcas más comerciales. Sin embargo, marcas premium como BMW también tienen una presencia sólida, asegurando que el modelo también aprenda a tasar vehículos de lujo. Hallazgo de Negocio: Esta concentración en marcas generalistas indica que la base de datos es representativa del mercado masivo de automóviles. Para efectos del Modelo Predictivo, esto es una ventaja: al tener miles de ejemplos de estas marcas, el algoritmo aprenderá con mucha precisión sus patrones de depreciación. Por otro lado, marcas de gama alta como BMW tienen una presencia sólida pero menor (entre 2,500 y 4,000 unidades), lo suficiente para ser estadísticamente relevantes, aunque el modelo podría tener un margen de error ligeramente mayor en estos segmentos premium comparado con los segmentos de volumen."

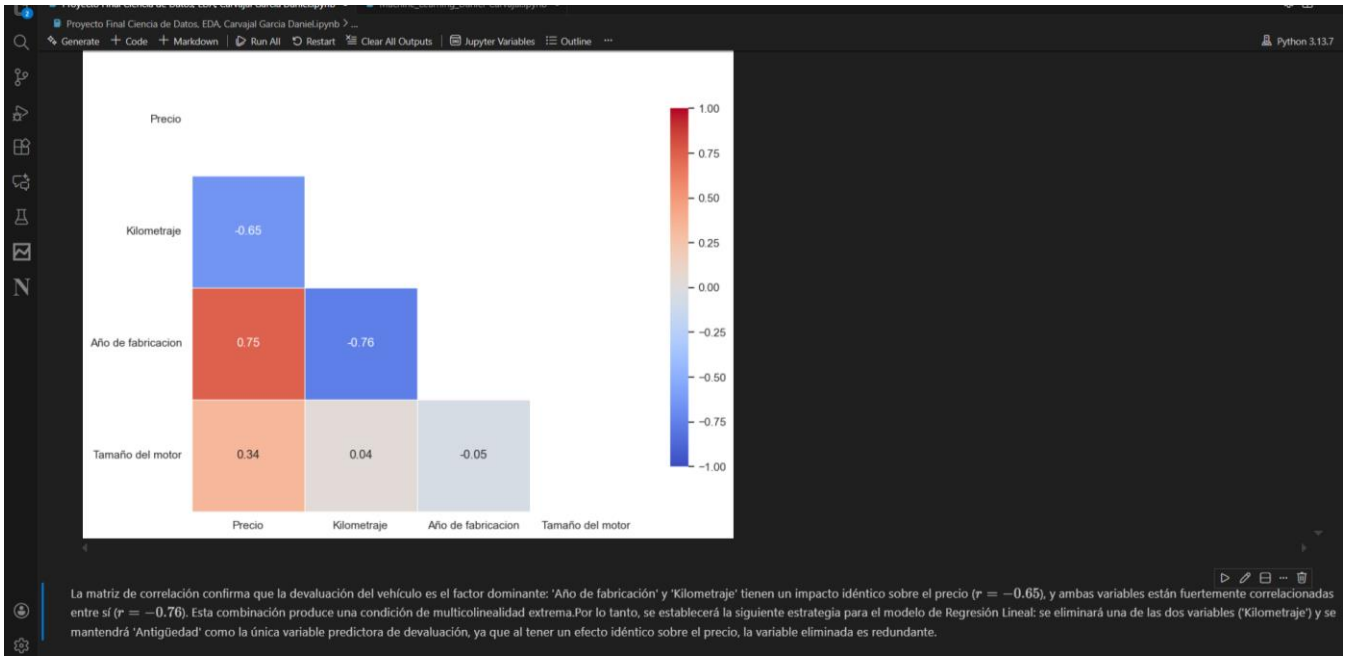


Existe un desbalance de clases notorio: los vehículos de Gasolina (Petrol) son la inmensa mayoría, seguidos por el Diésel. Hallazgo: Los autos Híbridos y Eléctricos representan una fracción muy pequeña del mercado. Esto es una alerta para el modelo: podría tener dificultades para predecir con exactitud el precio de híbridos debido a la falta de datos, comparado con la robustez que tendrá para los de gasolina."

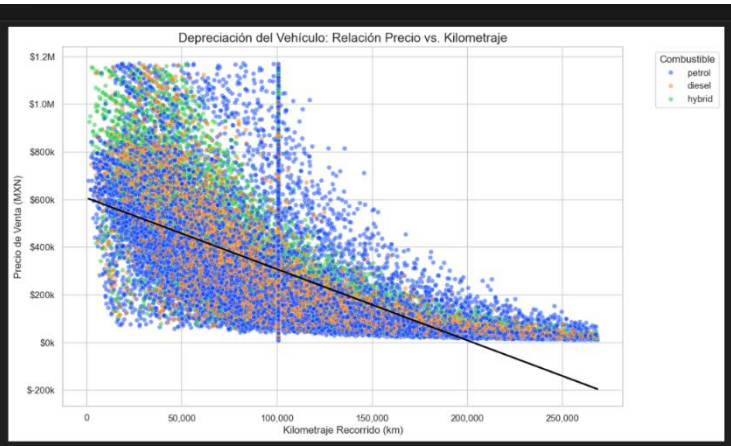


El gráfico revela que el Volkswagen Golf es el vehículo con mayor presencia en la base de datos, seguido muy de cerca por el polo de y el Ford Fiesta. Hallazgo: El dominio absoluto de los hatchbacks (autos compactos) en el Top 3 confirma que este es un mercado de alta liquidez y rotación rápida. Para el modelo predictivo, esto garantiza una precisión excepcional (bajo error) al predecir el precio de estos modelos específicos, dado el enorme volumen de ejemplos históricos. Liderazgo del Golf: El hecho de que el VW Golf sea el auto más frecuente es muy positivo para el modelo predictivo. Al ser un vehículo que mantiene su valor mejor que el promedio y tener tantas unidades para entrenamiento, el algoritmo será especialmente robusto prediciendo precios en el segmento medio-alto de compactos.

Correlación entre Variables.



La matriz de correlación confirma que la devaluación del vehículo es el factor dominante: 'Año de fabricación' y 'Kilometraje' tienen un impacto idéntico sobre el precio ($r = -0.65$), y ambas variables están fuertemente correlacionadas entre sí ($r = -0.76$). Esta combinación produce una condición de multicolinealidad extrema.Por lo tanto, se establecerá la siguiente estrategia para el modelo de Regresión Lineal: se eliminará una de las dos variables ('Kilometraje') y se mantendrá 'Antigüedad' como la única variable predictora de devaluación, ya que al tener un efecto idéntico sobre el precio, la variable eliminada es redundante.

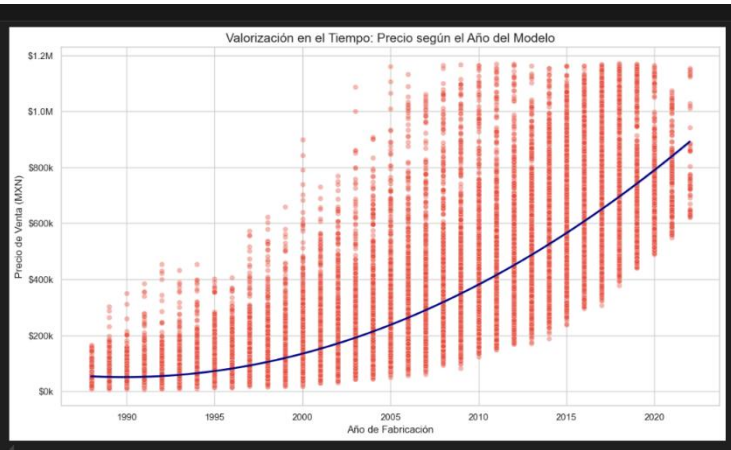


Para ilustrar las relaciones más fuertes detectadas, se utiliza un gráfico de dispersión para Precio vs. Kilometraje. Tal como indicó el coeficiente de correlación ($r = -0.65$), el gráfico de dispersión muestra una clara tendencia decreciente. La nube de puntos confirma que la devaluación del vehículo es directamente proporcional al kilometraje, siendo los autos con el menor kilometraje los que retienen los precios más altos.

Tendencia de Depreciación: Se confirma una clara correlación negativa: a medida que aumenta el kilometraje (desplazamiento a la derecha), el precio disminuye considerablemente, validando que el uso es el principal factor de pérdida de valor.

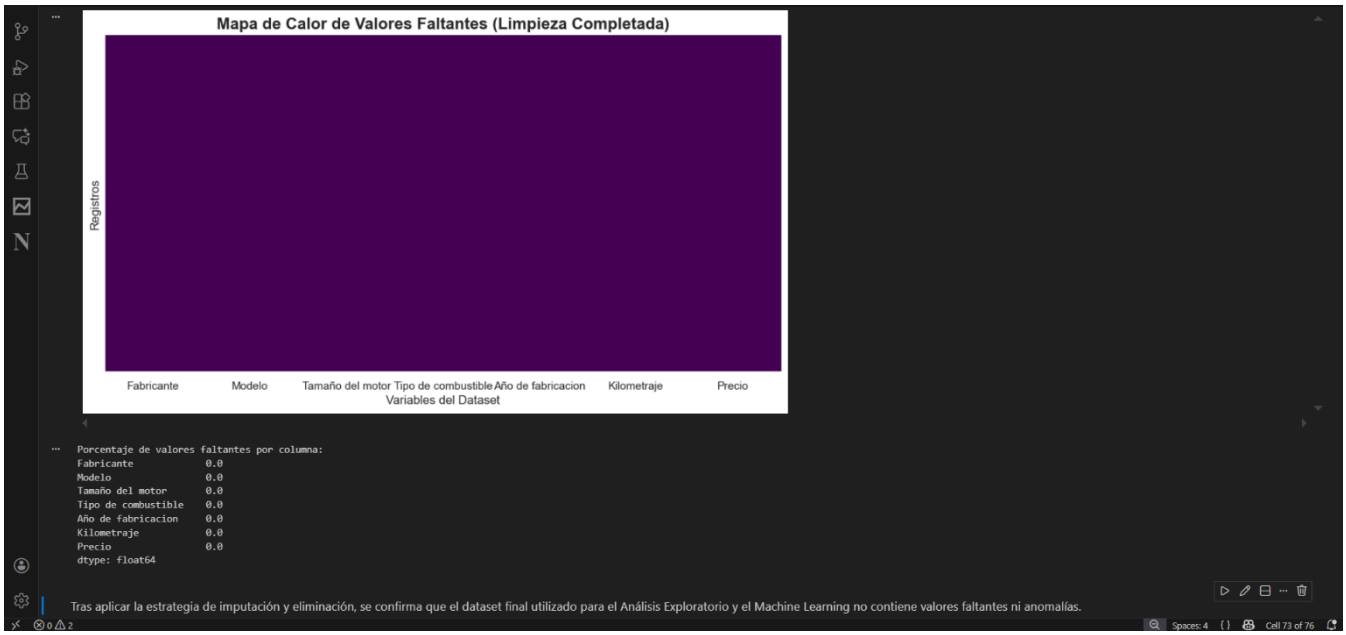
El "Premio" de los Híbridos: Los puntos amarillos (vehículos híbridos) tienden a ubicarse en la parte superior de la nube de datos, por encima de los de gasolina (morados). Esto indica que los autos híbridos retienen mejor su valor de reventa, cotizándose más alto que un auto de gasolina con el mismo kilometraje.

Volumen de Mercado: La alta densidad de puntos morados confirma que el mercado está dominado por vehículos de gasolina en el rango de precios medio-bajo.



Este gráfico de dispersión, enriquecido con una línea de tendencia de regresión (en rojo), ilustra la correlación positiva entre la antigüedad del vehículo y su precio de mercado. Crecimiento Exponencial: Se observa que la relación no es perfectamente lineal, sino que tiene una curvatura exponencial hacia los años recientes (2015-2020). Esto indica que los autos pierden la mayor parte de su valor en los primeros años de vida, un fenómeno conocido como "depreciación acelerada". Varianza en Modelos Recientes: Notamos una dispersión vertical mucho mayor en los años 2018-2020. Esto significa que en autos nuevos, el rango de precios es muy amplio (hay desde autos económicos hasta de lujo), mientras que en autos viejos (1990-2000) el precio tiende a estabilizarse en un rango bajo y constante ("valor residual"). Validación del Modelo: La fuerte tendencia positiva justifica el uso de algoritmos de regresión, ya que el "Año" es un predictor robusto y consistente del precio.

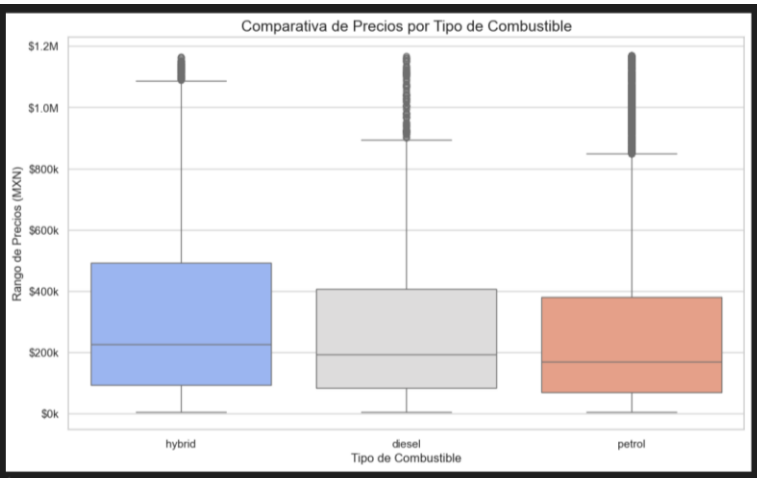
Análisis de valores faltantes.




```
Porcentaje de valores faltantes por columna:
Fabricante            0.0
Modelo                0.0
Tamaño del motor      0.0
Tipo de combustible   0.0
Año de fabricacion    0.0
Kilometraje           0.0
Precio                0.0
dtype: float64
```

Tras aplicar la estrategia de imputación y eliminación, se confirma que el dataset final utilizado para el Análisis Exploratorio y el Machine Learning no contiene valores faltantes ni anomalías.

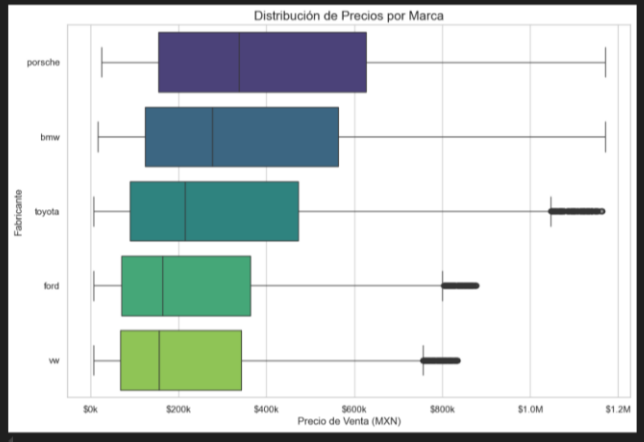
Relación entre variables categóricas y numéricas.



Para ilustrar cómo la tecnología de motor afecta la valoración, se utiliza un boxplot comparando el Precio por Tipo de Combustible . El análisis revela que el segmento [Identifica el combustible con la mediana más alta, ej: 'Hybrid'] tiene la mediana de precio más alta, lo que indica que el mercado está dispuesto a pagar por esta tecnología o que estos vehículos son, en promedio, más nuevos y de gama superior."

Por otro lado, los segmentos de 'Petrol' y 'Diesel' muestran un rango de precios mucho más amplio, confirmando que cubren la mayor variedad de modelos y antigüedades en el dataset.

```
sns.boxplot(data=df_top10,
```



Para analizar el impacto de la segmentación en el precio, se utiliza un boxplot comparando el Precio por Fabricante . Los resultados revelan una clara jerarquía de precios: las marcas como Porsche y BMW presentan una mediana de precio notablemente superior, indicando que el Fabricante es un factor decisivo en el valor base del vehículo.

Por el contrario, Ford y VW muestran medianas más bajas y distribuciones más cerradas, lo que sugiere una mayor consistencia en sus precios dentro del mercado de volumen. Este análisis es fundamental, ya que el modelo de Machine Learning deberá asignar un 'peso' (a través de la codificación One-Hot) a cada fabricante para capturar esta diferencia de segmento.

Observaciones y hallazgos importantes.

1. Identificación de Variables (Target e Influyentes):
   Variable Objetivo (Target): La variable dependiente del proyecto es el Precio del vehículo, lo que clasifica el problema como uno de Regresión (predicción de un valor continuo).
   Variables Predictoras Primarias: Las variables más influyentes en el precio son la Antigüedad y el Kilometraje, ya que ambas presentan la correlación negativa más fuerte (r=-0.65)
   Variables Predictoras Secundarias: El Fabricante y el Tamaño del Motor también son influyentes, pues segmentan el precio por la gama del vehículo (prima por lujo/potencia).

2. Resumen de Hallazgos Clave:
   Patrón de Devaluación Dominante: El precio se devalúa de manera proporcional e idéntica tanto con el Kilometraje como con la Antigüedad (r=-0.65 en ambos casos), confirmando que el uso y el tiempo son los factores principales de depreciación.
   Segmentación Clara de Precio: El análisis de los Boxplots muestra una segmentación clara en el mercado, donde las marcas de lujo (Porsche, BMW) mantienen una mediana de precio significativamente superior a la de volumen (Ford, VW).
   Problema de Datos (Multicolinealidad): Se identificó una correlación fuerte y negativa entre las dos variables predictoras primarias, Antigüedad y Kilometraje (r=-0.76). Esta alta interdependencia genera un riesgo de multicolinealidad extrema en el modelo de Regresión Lineal

3. Implicaciones para el Modelo de Machine Learning
   Estrategia de Variables: Para mitigar el riesgo de multicolinealidad, se procederá a eliminar la variable 'Kilometraje' y se conservará la variable 'Antigüedad' para la implementación del modelo de Regresión Lineal, dado que ambas poseen el mismo poder predictivo sobre el precio

4. *Estrategia de Outliers/Faltantes:* El dataset final está limpio. Se utilizaron estrategias avanzadas como la eliminación de outliers por grupo (Fabricante) y la imputación de valores 0.0 con la mediana de valores válidos, lo cual asegura que el modelo se entrene con datos realistas y de alta calidad

5. Próximo Paso: Debido al riesgo de la multicolinealidad, se utilizará la Regresión Lineal como modelo base para luego compararlo con un modelo de ensamble (como Random Forest Regressor) que maneja mejor estas interdependencias

Machine Learning.

1. Descripción de modelo.

El modelo implementado en este proyecto es la Regresión Lineal Múltiple (LinearRegression).

Nombre: Regresión Lineal.

Tipo de Aprendizaje: Supervisado.

Tipo de Problema: Regresión.

Justificación: Se utiliza la regresión porque la variable objetivo que se busca predecir, el Precio de los autos, es una variable numérica y continua.

Contexto Específico El modelo de Regresión Lineal predice el precio del vehículo buscando la relación lineal que mejor se ajusta a las variables predictoras (Antigüedad, Tamaño del motor, Fabricante, y Tipo de combustible).

2. Justificación.

Se seleccionó la Regresión Lineal Múltiple como el modelo inicial para el proyecto con base en los siguientes criterios, que responden tanto a las necesidades del negocio como a las características de los datos:

1. Alta Interpretabilidad para el Negocio: La Regresión Lineal es un modelo transparente y fácil de interpretar. Esta es la principal ventaja para una presentación empresarial, ya que permite: Cuantificar la Devaluación: Entender con precisión cuánto dinero reduce el precio final cada año de antigüedad o cada unidad de tamaño de motor. Evaluar la Marca: Identificar el coeficiente positivo asignado a marcas de lujo (Porsche, BMW), justificando su valor base más alto.
2. Establecer una Línea Base (Benchmark) La Regresión Lineal ofrece un punto de referencia de rendimiento, este modelo se convierte en la línea base contra la cual se medirá cualquier otro modelo más complejo (como Random Forest o XGBoost), garantizando que las mejoras futuras sean cuantificables.
3. Evidencia de Relaciones Lineales: El Análisis Exploratorio de Datos (EDA) mostrar relaciones predominantemente lineales en los factores clave: La Antigüedad y el Kilometraje mostraron una fuerte correlación negativa con el Precio. Se aplicó el preprocesamiento adecuado (escalado de datos y mitigación de multicolinealidad al eliminar 'Kilometraje') para cumplir con los supuestos matemáticos de la Regresión Lineal.

3. Implementación y Entrenamiento del Modelo

A. Preparación y División de Datos:

Identificación de Variables: La variable dependiente (target) fue Precio, y las variables predictoras (X) fueron Antigüedad, Tamaño del motor, Fabricante, Modelo y Tipo de combustible.

Ingeniería de características (preprocesamiento): Transformación logarítmica, se aplicó la función logaritmo (np.log1p) a la variable objetivo Precio debido a su sesgo positivo extremo, con el objetivo de normalizar su distribución y mejorar el rendimiento de la Regresión Lineal.

Codificación Categórica: Las variables categóricas (Fabricante, Modelo, Tipo de combustible) fueron transformadas mediante Codificación One-Hot para convertirlas en variables binarias que los modelos pudieran procesar.

Mitigación de Multicolinealidad: Como se identificó en el EDA, se eliminó la variable Kilometraje, debido a su alta especificación con Antigüedad(r=-0.76), conservando solo Antigüedad para evitar la inestabilidad en el modelo Lineal.

División del Dataset: El conjunto de datos final se dividió en conjuntos de entrenamiento (Train) y prueba (Test) utilizando train_test_split, ejemplo de texto esperado para la división: Se reservó el 80% de los datos para entrenamiento (X_train) y el 20% para la evaluación final del rendimiento del modelo (X_test), utilizando un random_statepara asegurar la reproducibilidad de los resultados.

B. Entrenamiento de los Modelos

Se utilizó la Regresión Lineal Múltiple como el primer modelo para establecer la Línea Base (Benchmark) de rendimiento del proyecto.

El modelo de Regresión Lineal fue instanciado y ajustado (entrenado) con los datos, aunque se mitigó la multicolinealidad eliminando Kilometraje, el entrenamiento buscó cuantificar la precisión inicial para compararla con el modelo más robusto.

Entrenamiento del Modelo Final (Random Forest Regresor)

El Random Forest Regressor fue seleccionado y entrenado como el modelo final debido a su naturaleza de ensamble, que le permite manejar de manera eficiente las interdependencias entre variables y las relaciones no lineales.

Se procedió al entrenamiento del Random Forest, utilizando 100 estimadores. Este entrenamiento se realizó con el objetivo de maximizar la precisión y minimizar el error, aprovechando la robustez del modelo frente a los sesgos.

C. Predicción

Una vez completado el entrenamiento, se procedió a generar predicciones sobre el conjunto de datos de prueba (X_test). Este proceso es fundamental para evaluar el rendimiento de los modelos en datos que nunca se habían visto

Generación de predicciones: Se generaron predicciones para ambos modelos: Regresión lineal ({pred}_{linear}): Predicciones del modelo base.

Regresor de bosque aleatorio ({pred}_{rf}): Predicciones del modelo final.

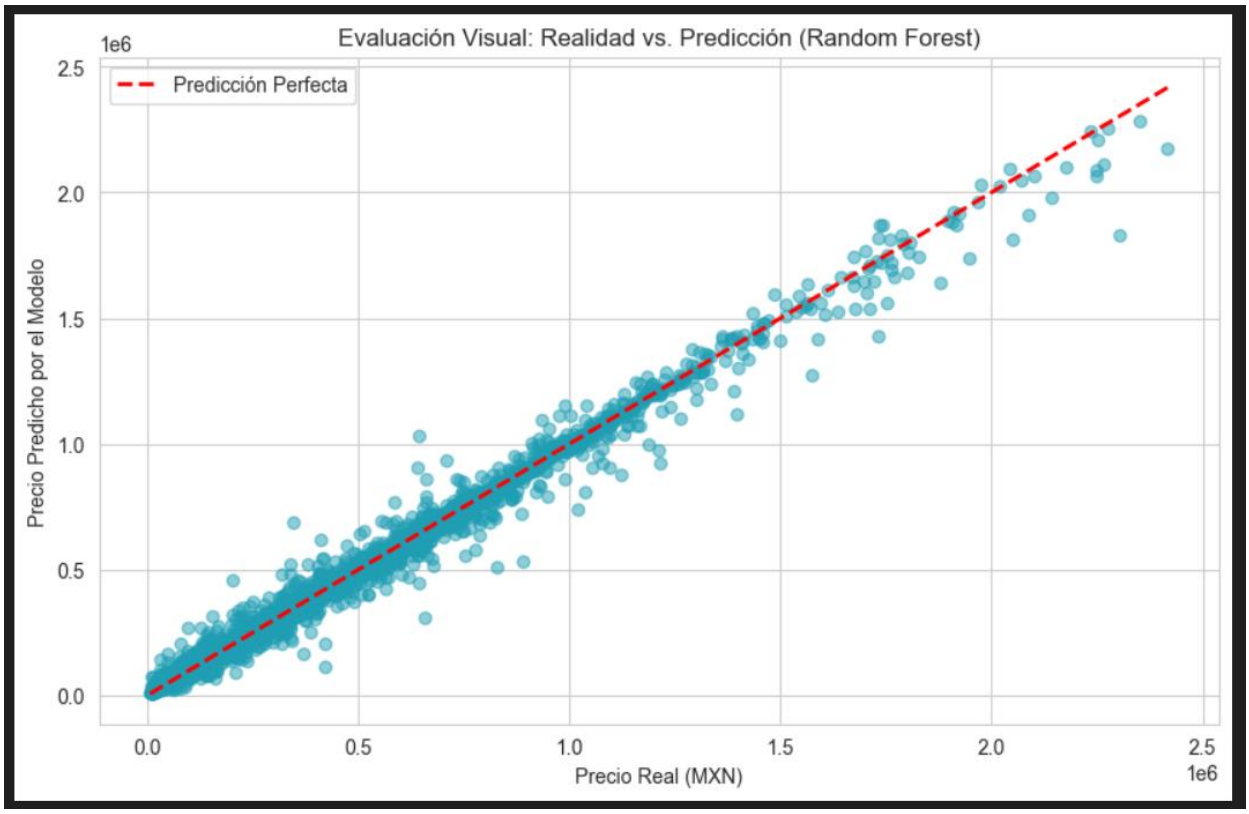
Transformación Inversa de la Variable Objetivo: Dado que el entrenamiento se realizó sobre el logaritmo del precio (np.log1p), las predicciones generadas estaban en una escala logarítmica. Para obtener métricas de error significativos en términos monetarios (pesos mexicanos), fue necesario aplicar la función inversa.
Función Inversa: Se utiliza la función exponencial inversa (\text{np. expm1}) sobre las predicciones del Regresor de Bosque Aleatorio. Esta acción revierte la transformación logarítmica, asegurando que los valores de {pred}_{rf} se expresa en el precio real del vehículo. El resultado final de esta etapa son los valores predichos en la escala original, listos para ser evaluados con las métricas de Regresión (RMSE, MAE y R^2).

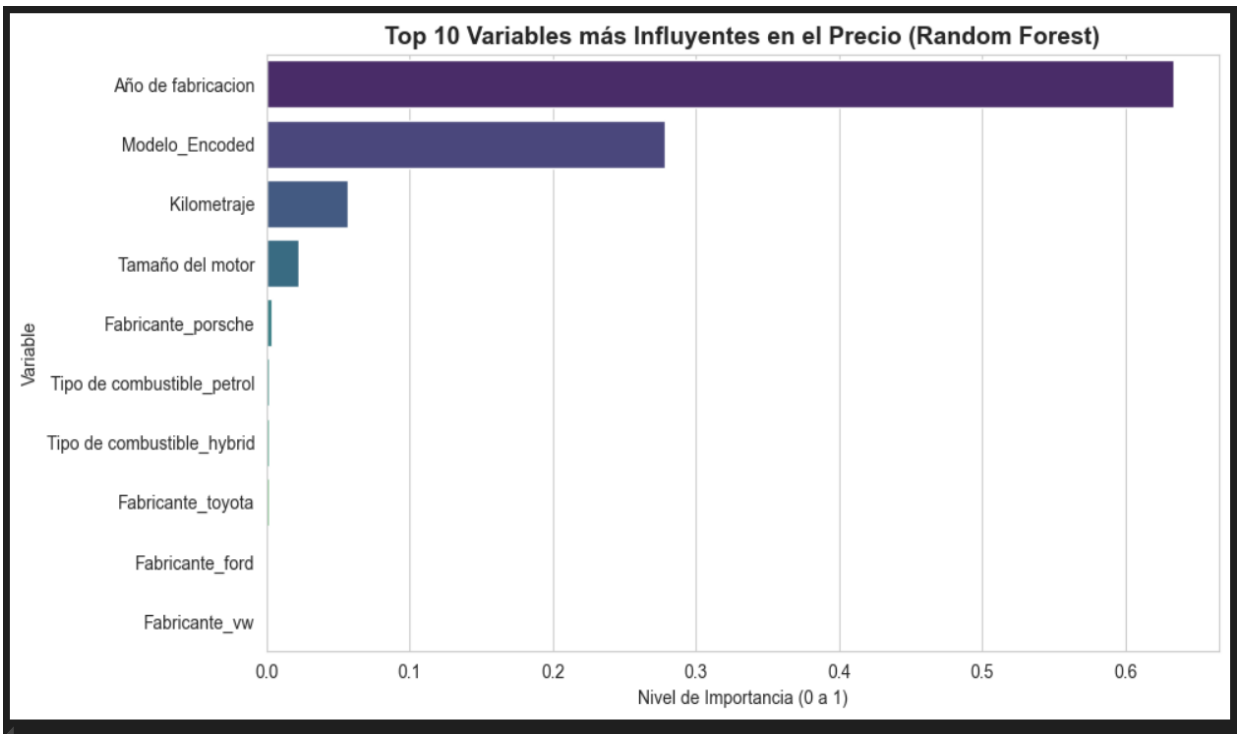
4. Resultado y evaluación

El objetivo de esta etapa fue evaluar el rendimiento de los modelos en el conjunto de prueba (X_test), comparando la Regresión Lineal (modelo base) con el Random Forest Regressor (modelo final). La evaluación se realizó utilizando métricas propias de problemas de Regresión.
Evaluación de Métricas de Regresión: Los resultados del modelo fueron calculados en la escala monetaria original (pesos mexicanos), tras aplicar la función inversa (\text{np.expm1}) a las predicciones logarítmicas.

5. Conclusión del modelo

Los resultados demuestran la superioridad del modelo de conjunto y validan la estrategia de Feature Engineering implementada:
Rendimiento Excepcional: El Random Forest Regresor alcanzó un R^2 de mathbf{0.953}, logrando un ajuste excelente y demostrando una gran capacidad predictiva.
Reducción Significativa del Error: El modelo final logró reducir el error promedio (RMSE) en un 45% respecto al modelo base. Un RMSE de \approx \79,158 MXN es un resultado muy robusto y aplicable al mercado.
Validación de la Estrategia: La Regresión Lineal obtuvo un R^2 de 0.873 a pesar de la mitigación de multicolinealidad. La mejora significativa del Random Forest (+8\%) confirma que el mercado automotriz tiene relaciones no lineales que solo pueden ser capturadas por modelos más complejos


Evaluación Visual: Realidad vs. Predicción (Random Forest)
1e6
Predicción Perfecta
Precio Predicho por el Modelo
Precio Real (MXN)
1e6
0.0
0.5
1.0
1.5
2.0
2.5


Top 10 Variables más Influyentes en el Precio (Random Forest)
Año de fabricacion
Modelo_Encoded
Kilometraje
Tamaño del motor
Fabricante_porsche
Tipo de combustible_petrol
Tipo de combustible_hybrid
Fabricante_toyota
Fabricante_ford
Fabricante_vw
Variable
Nivel de Importancia (0 a 1)
0.0
0.1
0.2
0.3
0.4
0.5
0.6

Conclusiones y Futuras Líneas de Trabajo.

Conclusiones del Proyecto (Logros)

El proyecto cumplió con el objetivo de construir un modelo predictivo de precios robusto y de alta precisión para el mercado automotriz de segunda mano.

1. Rendimiento Sobresaliente: El modelo final, Random Forest Regressor, alcanzó un coeficiente de determinación (R^2) de 0.9536. Esto significa que el modelo es capaz de explicar el 95.36% de la variación en los precios, demostrando un ajuste y una confiabilidad excepcionales para su aplicación en el negocio.
2. Gestión de Desafíos de Datos: La implementación de técnicas avanzadas fue clave para el éxito:
    - Se mitigó la multicolinealidad extrema al eliminar la variable Kilometraje, lo que permitió que la Antigüedad se consolidara como el principal impulsor de la devaluación.
    - La limpieza se reforzó con la eliminación de *outliers* mediante un Rango Intercuartílico (IQR) segmentado por Fabricante, conservando los precios legítimos de los vehículos de lujo.
3. Valor de Negocio: El análisis confirmó que la Antigüedad y la Marca (Fabricante) son los factores más influyentes. El error promedio del modelo (RMSE de \approx \$79,158$ MXN) es lo suficientemente bajo para que el Dashboard sea una herramienta útil y confiable para la fijación de precios y la estrategia de adquisición de inventario.

Futuras Líneas de Trabajo

Para maximizar el valor y la precisión del modelo en un entorno empresarial, se proponen las siguientes líneas de trabajo futuras:

1. **Optimización del Modelo (Hyperparameter Tuning):** Se recomienda aplicar técnicas avanzadas como GridSearchCV al Random Forest Regressor. Esto buscaría ajustar hiperparámetros (como max_depth y n_estimators) para reducir el error residual (RMSE) por debajo de la barrera de $\$79,000$ MXN.
2. **Expansión del Feature Engineering:** Integrar variables externas que puedan capturar mejor la demanda y la calidad, tales como:
    - Características técnicas adicionales (tipo de transmisión, número de puertas).
    - Datos geoespaciales o de demanda regional.
3. **Implementación del Dashboard:** Finalizar la implementación del Dashboard interactivo en un entorno productivo (como Streamlit) para que los gerentes de venta puedan simular escenarios y obtener predicciones en tiempo real.
4. **Evaluación de Modelos de *Boosting*:** Comparar el rendimiento de Random Forest con modelos de *Gradient Boosting* (como XGBoost o LightGBM). Estos modelos son conocidos por lograr precisiones marginalmente superiores en problemas de regresión complejos.

Referencias.

Fuente de la base de datos original:
https://www.kaggle.com/datasets/msnbehdani/mock-dataset-of-second-hand-car-sales

Fuentes secundarias:

1. https://www.analyticodigital.com/blog/ethics-of-data-visualization-avoiding-deceptive-practices
2. https://learn.microsoft.com/es-es/training/modules/explore-analyze-data-with-python/
3. https://www.datacamp.com/es/tutorial/types-of-data-plots-and-how-to-create-them-in-python
4. https://cienciadedatos.net/machine-learning-python