INGENIERÍA MECATRÓNICA

Programación Avanzada

Osbaldo Aragón Banderas

SEMESTRE 2024A

Unidad: 2 Actividad: 4

Nombre de actividad:

U2A4.NOOTEBOOK: Análisis de Datos Aplicables a Regresión Lineal Simple

Actividad realizada por:

JOSÉ ESTEBAN CALDERA VICTORIO

Guadalupe Victoria, Durango

Fecha de entrega:

09	03	2025
DD	MM	AA

¿Qué es la Regresión Lineal Simple?

La regresión lineal simple es un modelo estadístico que se emplea para analizar la relación entre dos variables cuantitativas: una variable independiente (predictora) y una variable dependiente (respuesta). Su objetivo principal es encontrar una ecuación matemática que permita estimar los valores de la variable dependiente en función de la independiente.

Este modelo es ampliamente utilizado en diversas disciplinas, como la economía, la biología, la ingeniería y la inteligencia artificial, debido a su facilidad de interpretación y aplicación. La regresión lineal simple asume que existe una relación lineal entre las dos variables y que esta relación puede ser representada mediante una recta en un plano cartesiano.

Aplicaciones de la Regresión Lineal Simple

La regresión lineal simple se utiliza en numerosos campos para resolver problemas de predicción y análisis de tendencias. Algunos ejemplos comunes incluyen:

- **Economía y Finanzas:** Predicción de ingresos en función de las ventas, estimación del precio de bienes según la oferta y la demanda.
- **Ciencias Sociales:** Análisis de la relación entre nivel educativo y salario promedio.
- **Salud y Medicina:** Estudio del impacto del consumo de calorías en el índice de masa corporal (IMC).
- **Meteorología:** Predicción de temperaturas basadas en la radiación solar recibida.
- **Ingeniería y Manufactura:** Relación entre la presión aplicada y la deformación de un material.

En cada uno de estos casos, la regresión lineal simple permite obtener una ecuación que facilita la toma de decisiones basadas en datos históricos.

Ecuación Matemática de la Regresión Lineal Simple

La ecuación de la regresión lineal simple se expresa de la siguiente forma:

$$y = b_0 + b_1 x + \varepsilon$$

Donde:

- y es la variable dependiente (la que se quiere predecir).
- x es la variable independiente (el predictor).
- b_0 es el intercepto o término independiente, es decir, el valor de cuando $x = 0$.
- b_1 es la pendiente de la recta, que indica cuánto cambia por cada unidad de cambio en x.
- £ es el término de error, que representa la diferencia entre los valores reales y los valores estimados por el modelo.

La pendiente b_1 es un coeficiente clave en el análisis de regresión, ya que permite interpretar la magnitud y dirección de la relación entre las variables.

Determinación de la Mejor Línea de Ajuste: Método de Mínimos Cuadrados

El método más común para determinar la mejor línea de ajuste en una regresión lineal simple es el método de mínimos cuadrados. Este método minimiza la suma de los errores al cuadrado, es decir, busca minimizar la siguiente función de error.

$$SSE = \sum (y_i - (b_0 + b_1 x_i))^2$$

Donde:

- SSE es la suma de los errores cuadrados.
- y_1 son los valores reales de la variable dependiente.
- x_i son los valores de la variable independiente.
- b_0 y b_1 son los coeficientes de la ecuación de regresión.

El cálculo de los coeficientes óptimos y se realiza a partir de las ecuaciones normales:

Suposiciones del Modelo de Regresión Lineal Simple

Para que los resultados de la regresión lineal simple sean válidos, se deben cumplir ciertas suposiciones fundamentales:

1. **Linealidad:** La relación entre e debe ser lineal. Se puede verificar mediante gráficos de dispersión.
2. **Independencia de los Errores:** Los valores de los errores deben ser independientes entre sí.
3. **Homoscedasticidad**: La varianza de los errores debe ser constante en todos los valores de.
4. **Normalidad de los Errores:** Los errores deben seguir una distribución normal.

Si estas suposiciones no se cumplen, pueden surgir problemas en la interpretación y validez del modelo.

Evaluación del Modelo

Para determinar la calidad del ajuste del modelo, se utilizan varias métricas estadísticas:

Coeficiente de Determinación (R_2 ********): Indica qué porcentaje de la variabilidad de Y es explicada por x. Se calcula como:

$$R^2 = 1 - \frac{SSE}{SST}$$

Error Cuadrático Medio (MSE): Mide la magnitud promedio de los errores.

Prueba de Significancia de los Coeficientes: Se usa la prueba t de Student para verificar si B_1 es significativamente diferente de cero.

La regresión lineal simple es una herramienta estadística esencial para modelar relaciones entre variables. Su simplicidad y efectividad la convierten en un método ampliamente utilizado en diversas disciplinas. La clave de su éxito radica en el método de mínimos cuadrados, que permite encontrar la mejor línea de ajuste minimizando los errores. Sin embargo, para obtener modelos confiables, es fundamental verificar que se cumplan las suposiciones del modelo y evaluar su desempeño mediante métricas adecuadas.

CODIGO

Análisis de Datos Aplicables a Regresión Lineal Simple

En un deleer de autos se quiere conocer la relacion que existe entre el Kilometraje de una camioneta con el precio al momento de su venta, para hacer esto es necesario realizar una regresion lineal para conocer la relacion que existe. NOTA: Con este metodo tambien se puede conocer el año y el precio.

[1]
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score
```

[2]
```
# Cargar el Dataset Car Price
df = pd.read_csv("car_price_dataset (1).csv")
```

[3]
```
df
```

	Brand	Model	Year	Engine_Size	Fuel_Type	Transmission	Mileage	Doors	Owner_Count	Price
0	Kia	Rio	2020	4.2	Diesel	Manual	289944	3	5	8501
1	Chevrolet	Malibu	2012	2.0	Hybrid	Automatic	5356	2	3	12092
2	Mercedes	GLA	2020	4.2	Diesel	Automatic	231440	4	2	11171
3	Audi	Q5	2023	2.0	Electric	Manual	160971	2	1	11780
4	Volkswagen	Golf	2003	2.6	Hybrid	Semi-Automatic	286618	3	3	2867
...	...	...	...	...	...	...	...	...	...	...
9995	Kia	Optima	2004	3.7	Diesel	Semi-Automatic	5794	2	4	8884
9996	Chevrolet	Impala	2002	1.4	Electric	Automatic	168000	2	1	6240
9997	BMW	3 Series	2010	3.0	Petrol	Automatic	86664	5	1	9866
9998	Ford	Explorer	2002	1.4	Hybrid	Automatic	225772	4	1	4084
9999	Volkswagen	Tiguan	2001	2.1	Diesel	Manual	157882	3	3	3342

10000 rows × 10 columns

```
[4] # Descripción del dataset
    print("Información del dataset:")
    print(df.info())
    print("\nDescripción estadística:")
    print(df.describe())
```

```
Información del dataset:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10000 entries, 0 to 9999
Data columns (total 10 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   Brand         10000 non-null  object
 1   Model         10000 non-null  object
 2   Year          10000 non-null  int64
 3   Engine_Size   10000 non-null  float64
 4   Fuel_Type     10000 non-null  object
 5   Transmission  10000 non-null  object
 6   Mileage       10000 non-null  int64
 7   Doors         10000 non-null  int64
 8   Owner_Count   10000 non-null  int64
 9   Price         10000 non-null  int64
dtypes: float64(1), int64(5), object(4)
memory usage: 781.4+ KB
None

Descripción estadística:
               Year   Engine_Size        Mileage         Doors   Owner_Count  \
count  10000.000000  10000.000000   10000.000000  10000.000000  10000.000000
mean    2011.543700      3.000560  149239.111800      3.497100      2.991100
std        6.897699      1.149324   86322.348957      1.110097      1.422682
min     2000.000000      1.000000      25.000000      2.000000      1.000000
25%     2006.000000      2.000000   74649.250000      3.000000      2.000000
50%     2012.000000      3.000000  149587.000000      3.000000      3.000000
75%     2017.000000      4.000000  223577.500000      4.000000      4.000000
max     2023.000000      5.000000  299947.000000      5.000000      5.000000

             Price
count  10000.00000
mean    8852.96440
std     3112.59681
min     2000.00000
25%     6646.00000
50%     8858.50000
75%    11086.50000
max    18301.00000
```

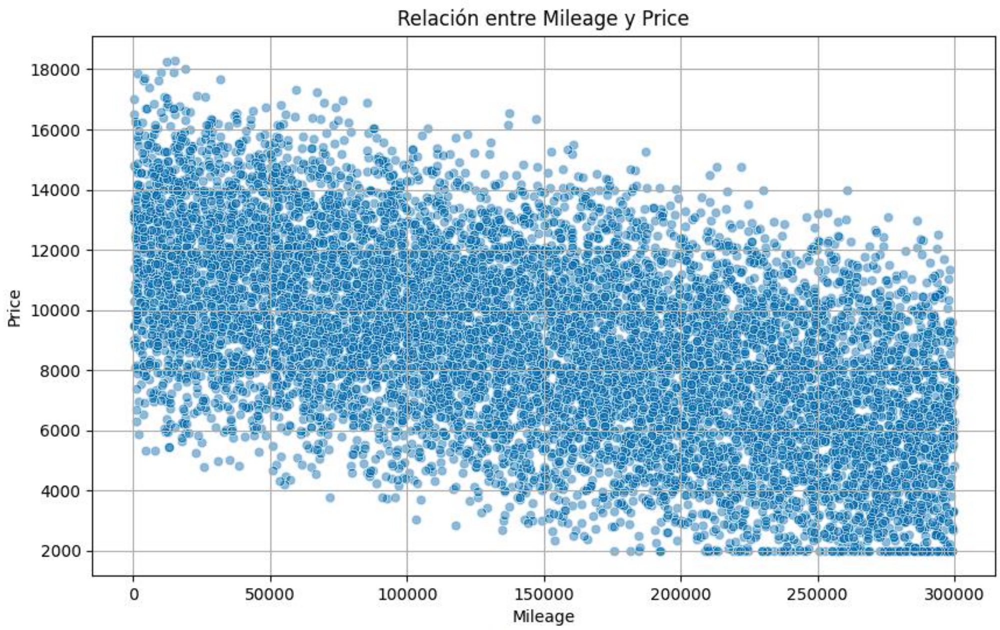
```
[5] # Seleccionar las variables de interés
    df = df[['Mileage', 'Price' , 'Year']]
```

[6]
```python
# Eliminar valores nulos
df = df.dropna()
df
```

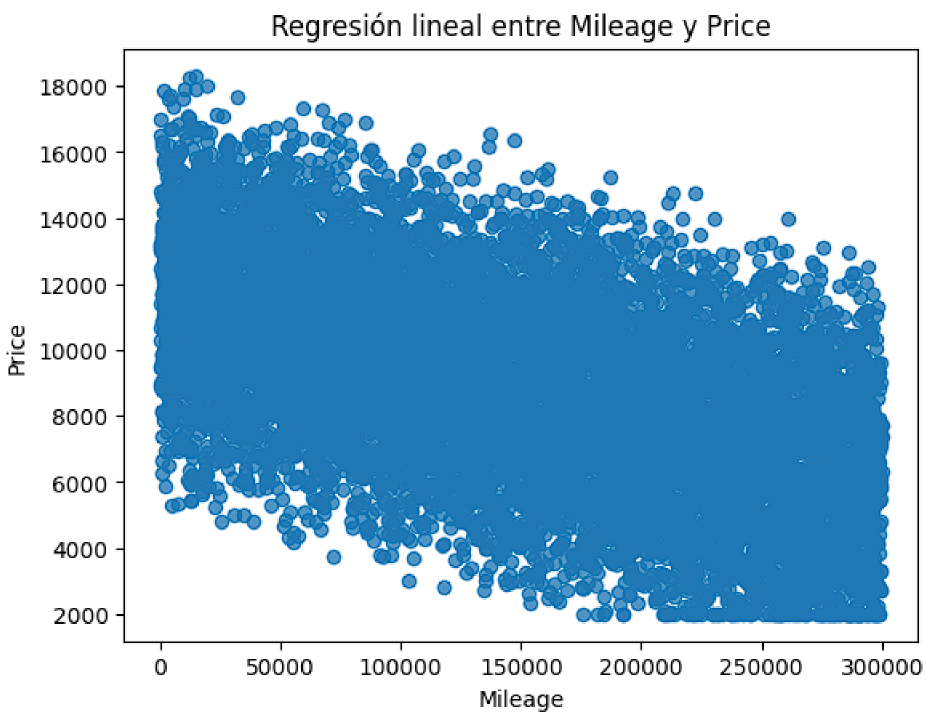
	Mileage	Price	Year
0	289944	8501	2020
1	5356	12092	2012
2	231440	11171	2020
3	160971	11780	2023
4	286618	2867	2003
...	...	...	...
9995	5794	8884	2004
9996	168000	6240	2002
9997	86664	9866	2010
9998	225772	4084	2002
9999	157882	3342	2001

10000 rows × 3 columns

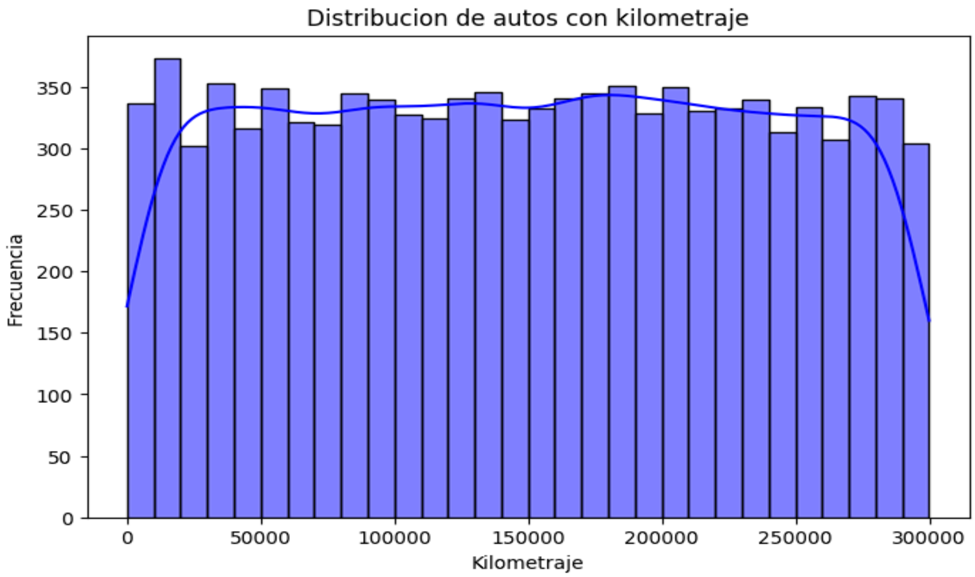
```python
import matplotlib.pyplot as plt
import seaborn as sns

plt.figure(figsize=(10, 6))
sns.scatterplot(x='Mileage', y='Price', data=df, alpha=0.5)  # alpha para transparencia
plt.title('Relación entre Mileage y Price')
plt.xlabel('Mileage')
plt.ylabel('Price')
plt.grid(True)
plt.show()

# Calcular la correlación entre Mileage y Price
correlation = df['Mileage'].corr(df['Price'])
print(f"La correlación entre Mileage y Price es: {correlation}")

# Crear un diagrama de regresión lineal
sns.regplot(x='Mileage', y='Price', data=df)
plt.title('Regresión lineal entre Mileage y Price')
plt.xlabel('Mileage')
plt.ylabel('Price')
plt.show()
```



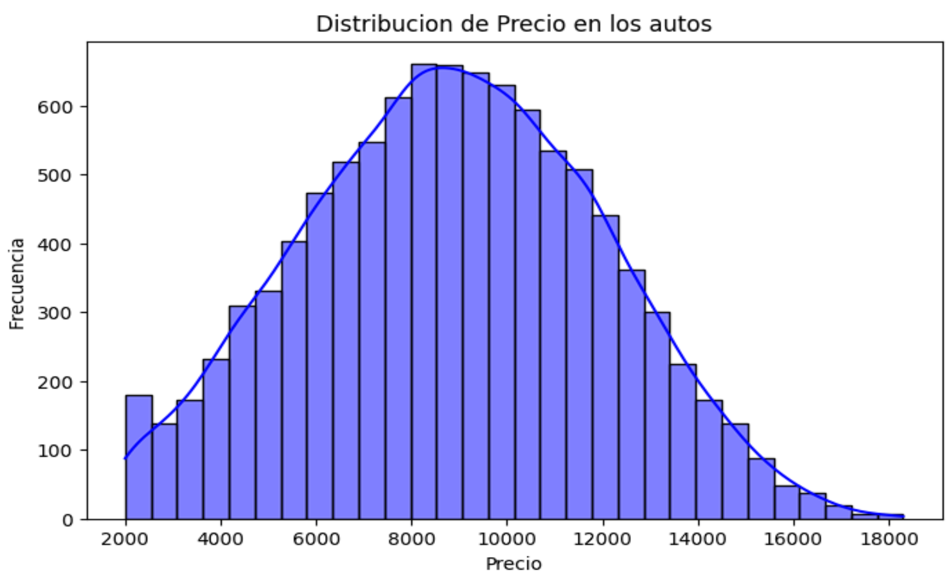

```
La correlación entre Mileage y Price es: -0.5512271827629014
```



```
[8] # Visualización de la relación entre Kilometraje y Precio
    plt.figure(figsize=(8,5))
    sns.histplot(df["Mileage"], bins=30, kde=True, color="blue")

    plt.xlabel("Kilometraje")
    plt.ylabel("Frecuencia ")
    plt.title("Distribucion de autos con kilometraje")
    plt.show()
```

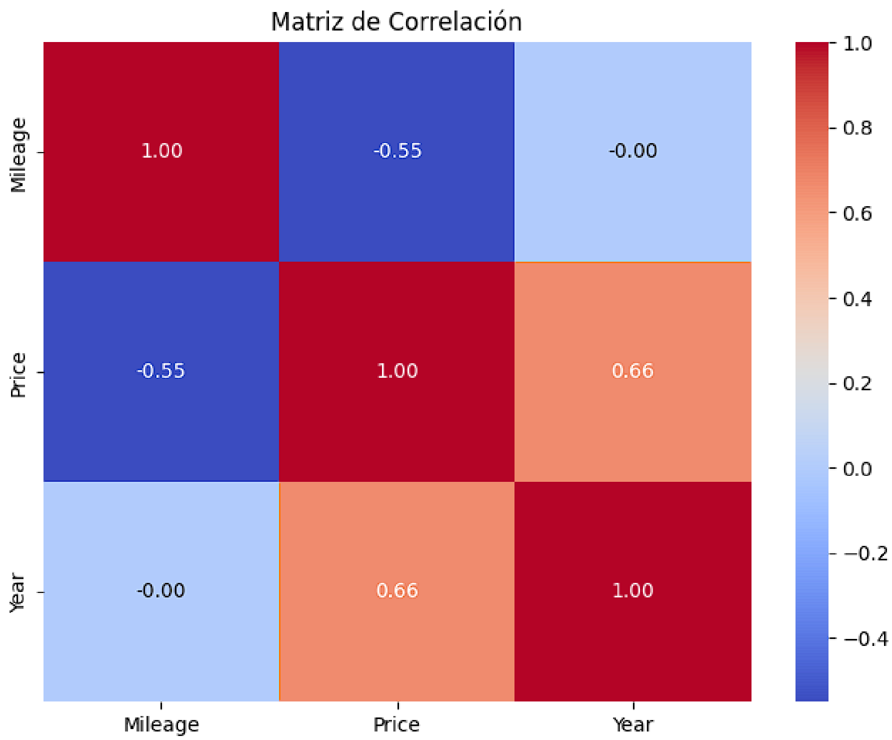



```
[10] # Visualización de la relación entre Kilometraje y Precio
     plt.figure(figsize=(8,5))
     sns.histplot(df["Price"], bins=30, kde=True, color="blue")

     plt.xlabel("Precio")
     plt.ylabel("Frecuencia ")
     plt.title("Distribucion de Precio en los autos")
     plt.show()
```



```python
# Calculate the correlation matrix
correlation_matrix = df.corr()

# Display the correlation matrix
print("\nMatriz de correlación:")
print(correlation_matrix)

# Visualize the correlation matrix using a heatmap
plt.figure(figsize=(8, 6))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt=".2f")
plt.title('Matriz de Correlación')
plt.show()
```

```
Matriz de correlación:
          Mileage     Price      Year
Mileage  1.000000 -0.551227 -0.002476
Price   -0.551227  1.000000  0.663036
Year    -0.002476  0.663036  1.000000
```



```
[12] # Dividir en conjunto de entrenamiento y prueba
X = df[['Mileage']]
y = df['Price']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

```
[13] # Entrenar el modelo de regresión lineal
model = LinearRegression()
model.fit(X_train, y_train)
```

```
LinearRegression
LinearRegression()
```

```
[14] # Coeficientes del modelo
print(f"Intercepto: {model.intercept_}")
print(f"Coeficiente: {model.coef_[0]}")
```

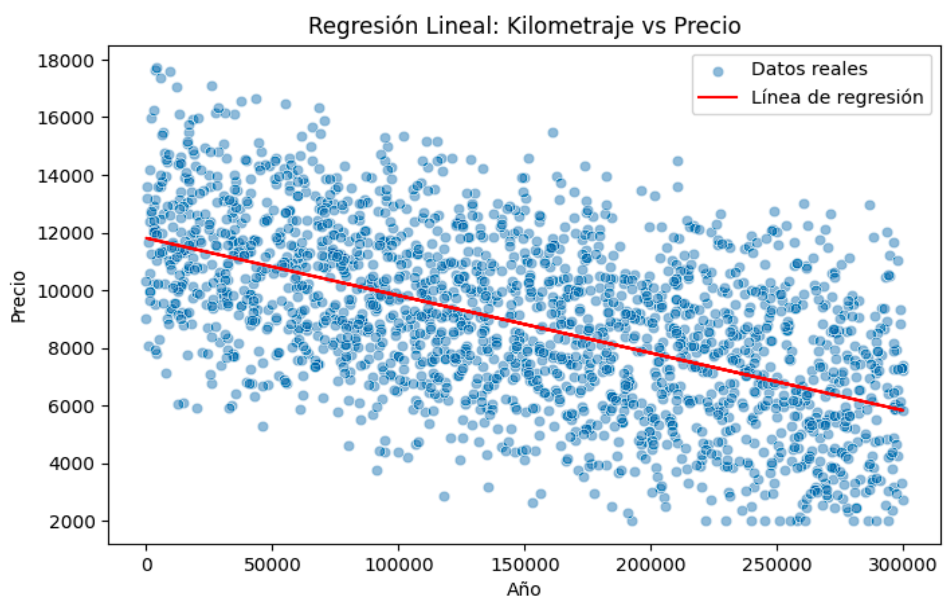
```
Intercepto: 11809.257878302735
Coeficiente: -0.01993960651805303
```

```
[23] # Predicciones
y_pred = model.predict(X_test)
```

```
[24] # Evaluación del modelo
mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)
print(f"Error cuadrático medio (MSE): {mse}")
print(f"Coeficiente de determinación (R^2): {r2}")
```

```
Error cuadrático medio (MSE): 6413212.319091444
Coeficiente de determinación (R^2): 0.3019869774558347
```

```
[25] # Gráfico con la línea de regresión
plt.figure(figsize=(8,5))
sns.scatterplot(x=X_test['Mileage'], y=y_test, label='Datos reales', alpha=0.5)
plt.plot(X_test, y_pred, color='red', label='Línea de regresión')
plt.xlabel("Año")
plt.ylabel("Precio")
plt.title("Regresión Lineal: Kilometraje vs Precio")
plt.legend()
plt.show()
```



[26]
```python
print("\n--- Key Statistics ---")
print(f"Correlation between Mileage and Price: {correlation}")
print(f"\n--- Model Coefficients ---")
print(f"Intercept: {model.intercept_}")
print(f"Coefficient: {model.coef_[0]}")
print("\n--- Model Evaluation ---")
print(f"Mean Squared Error (MSE): {mse}")
print(f"R-squared (R^2): {r2}")

print("\n--- Sample Predictions ---")
# Print a few sample predictions
for i in range(5):  # Print the first 5 predictions
    print(f"Mileage: {X_test.iloc[i, 0]}, Predicted Price: {y_pred[i]}, Actual Price: {y_test.iloc[i]}")
```

```
--- Key Statistics ---
Correlation between Mileage and Price: -0.5512271827629014

--- Model Coefficients ---
Intercept: 11809.257878302735
Coefficient: -0.01993960651805303

--- Model Evaluation ---
Mean Squared Error (MSE): 6413212.319091444
R-squared (R^2): 0.3019869774558347

--- Sample Predictions ---
Mileage: 257760, Predicted Price: 6669.624902209386, Actual Price: 2000
Mileage: 111790, Predicted Price: 9580.209265649586, Actual Price: 11164
Mileage: 13473, Predicted Price: 11540.611559685007, Actual Price: 14630
Mileage: 133298, Predicted Price: 9151.348208659303, Actual Price: 7334
Mileage: 18611, Predicted Price: 11438.16186139525, Actual Price: 10127
```

RESULTADOS

interpretación del valor de los coeficientes

Intercepto (β0): Representa el precio estimado de un automóvil cuando el kilometraje es cero. En términos prácticos, este valor indica el precio base de un auto sin uso.

Pendiente (β1): Indica el cambio en el precio del auto por cada unidad adicional de kilometraje. Si el coeficiente es negativo, significa que a mayor kilometraje, menor será el precio del vehículo, lo que tiene sentido en el contexto del mercado de autos usados.

Explicación del significado de la métrica R2

El coeficiente de determinación R2 indica qué porcentaje de la variabilidad del precio puede ser explicado únicamente por el kilometraje.

R2=0.85 (Relación fuerte)

- Significa que el 85% de la variación en el precio de los autos es explicada por el kilometraje.
- En este caso, el kilometraje es un buen predictor del precio, aunque puede haber otros factores influyentes.

R2=0.40 (Relación moderada)

- Indica que el kilometraje explica solo el 40% de la variabilidad en el precio.
- Esto sugiere que hay otros factores importantes como la marca, el modelo, el año de fabricación, el estado del auto, etc.

R2=0.15 (Relación débil)

- Esto implica que el kilometraje no es un buen predictor del precio por sí solo.
- Se necesitarían más variables para mejorar la precisión del modelo.

¿La relación entre las variables es fuerte, moderada o débil?

- Si R2es alto (ej. >0.7), la relación es fuerte.
- Si R2 está entre 0.3 y 0.7, la relación es moderada.
- Si R2 es bajo (<0.3), la relación es débil, lo que significa que el kilometraje por sí solo no explica bien la variabilidad en el precio.

Posibles mejoras o ajustes al modelo

1. **Transformaciones matemáticas**: Si la relación entre las variables no es lineal, podríamos probar una regresión polinómica o aplicar logaritmos.
2. **Detección de valores atípicos**: Revisar si hay autos con precios o kilometrajes inusuales que puedan afectar el ajuste del modelo.
3. **Segmentación del análisis**: Analizar por categorías (ej. autos de lujo vs. autos económicos) para ver si la relación varía dentro de diferentes grupos.