



Proyecto grupo 12



Álgebra (Regresión
lineal simple y múltiple)



Regresión Lineal Aplicado en Python

Herramientas Útiles para crear una Regresión Lineal en python:

- Matplotlib (es una biblioteca de visualización de datos para Python que permite crear gráficos y visualizaciones de manera sencilla).

- Pandas** y **Scikit-learn** (son dos bibliotecas de Python muy populares en el análisis de datos y aprendizaje automático, cada una con su propósito específico en el procesamiento y modelado de datos).

- Visual Studio Code** (VS Code es un editor de código fuente ligero y de código abierto).



1. Importación de bibliotecas necesarias


```
# Importamos las bibliotecas necesarias
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score
```

- **pandas**: Se usa para manipulación y análisis de datos. Aquí se utiliza para manejar el conjunto de datos de las viviendas.
- **numpy**: Utilizado para cálculos matemáticos.
- **matplotlib.pyplot**: Biblioteca para generar gráficos.
- **sklearn.model_selection.train_test_split**: Se usa para dividir el conjunto de datos en dos partes: entrenamiento y prueba.
- **sklearn.linear_model.LinearRegression**: Modelo de regresión lineal.
- **sklearn.metrics**: Contiene funciones para evaluar el rendimiento del modelo (error absoluto, cuadrático, R2).

2. Simulación de datos de casas

```
# Simulamos datos de casas (supongamos un conjunto de datos simplificado)
data = {
    'Area': [1500, 1600, 1700, 1800, 1900, 2000, 2100, 2200, 2300, 2400],
    'Habitaciones': [3, 3, 3, 4, 4, 4, 5, 5, 5, 6],
    'Edad': [10, 15, 20, 15, 10, 5, 20, 5, 10, 1],
    'Precio': [300000, 320000, 340000, 360000, 380000, 400000, 420000, 440000, 460000, 480000]
}
df = pd.DataFrame(data)
```


- Se crea un conjunto de datos simulado con 4 columnas: **Area**, **Habitaciones**, **Edad** y **Precio**. Este conjunto de datos es usado para el modelo.



3. Separación de variables independientes y dependientes

```
# Separar las variables independientes (X) y la variable dependiente (y)
X = df[['Area', 'Habitaciones', 'Edad']]
y = df['Precio']
```

- **X** contiene las características (variables independientes) que predicen el precio (área, número de habitaciones y edad).
- **y** contiene la variable dependiente, que es el precio de la vivienda.



4. División del conjunto de datos en entrenamiento y prueba

```
# Dividimos los datos en conjuntos de entrenamiento y prueba
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)
```

- Se separan los datos en dos conjuntos: `X_train` y `y_train` para entrenamiento, y `X_test` y `y_test` para prueba.
- El tamaño de prueba es el 30% de los datos, y la semilla para la aleatorización está fijada con `random_state=42` para asegurar reproducibilidad.



5. Creación y entrenamiento del modelo de regresión lineal

```
# Crear y entrenar el modelo de regresión lineal
model = LinearRegression()
model.fit(X_train, y_train)
```

- Se crea una instancia del modelo de regresión lineal (`LinearRegression()`).
- Luego, el modelo es entrenado utilizando el conjunto de entrenamiento (`X_train, y_train`).



6. Predicción en el conjunto de prueba

```
# Realizamos predicciones en el conjunto de prueba  
y_pred = model.predict(X_test)
```

- Usamos el modelo entrenado para predecir los precios de las viviendas en el conjunto de prueba (`X_test`).



7. Evaluación del modelo

```
# Evaluación del modelo
mae = mean_absolute_error(y_test, y_pred)
mse = mean_squared_error(y_test, y_pred)
rmse = np.sqrt(mse)
r2 = r2_score(y_test, y_pred)
```

- **Mean Absolute Error (MAE)**: Error absoluto medio entre los valores predichos y los reales.
- **Mean Squared Error (MSE)**: Error cuadrático medio, penaliza errores más grandes más fuertemente.
- **Root Mean Squared Error (RMSE)**: Raíz cuadrada del MSE, expresado en las mismas unidades que la variable dependiente (precio).
- **R-squared (R2)**: Medida de la calidad del ajuste del modelo, que indica la proporción de la varianza explicada por el modelo.

8. Impresión de las métricas de evaluación

```
# Imprimir métricas de evaluación
print("Mean Absolute Error (MAE):", mae)
print("Mean Squared Error (MSE):", mse)
print("Root Mean Squared Error (RMSE):", rmse)
print("R-squared (R2):", r2)
```

```
Mean Absolute Error (MAE): 1.9402553637822468e-11
Mean Squared Error (MSE): 1.1293772630057337e-21
Root Mean Squared Error (RMSE): 3.3606208697288864e-11
R-squared (R2): 1.0
```

- Se imprimen las métricas de evaluación calculadas anteriormente.



9. Impresión de los coeficientes del modelo

```
# Imprimir coeficientes del modelo
print("\nCoeficientes del modelo:")
for i, col in enumerate(X.columns):
    print(f"{col}: {model.coef_[i]}")
print("Intercepto:", model.intercept_)
```

```
Coeficientes del modelo:
Area: 200.00000000000006
Habitaciones: -2.71679233382239e-12
Edad: 5.261247622466191e-13
Intercepto: -1.1641532182693481e-10
```

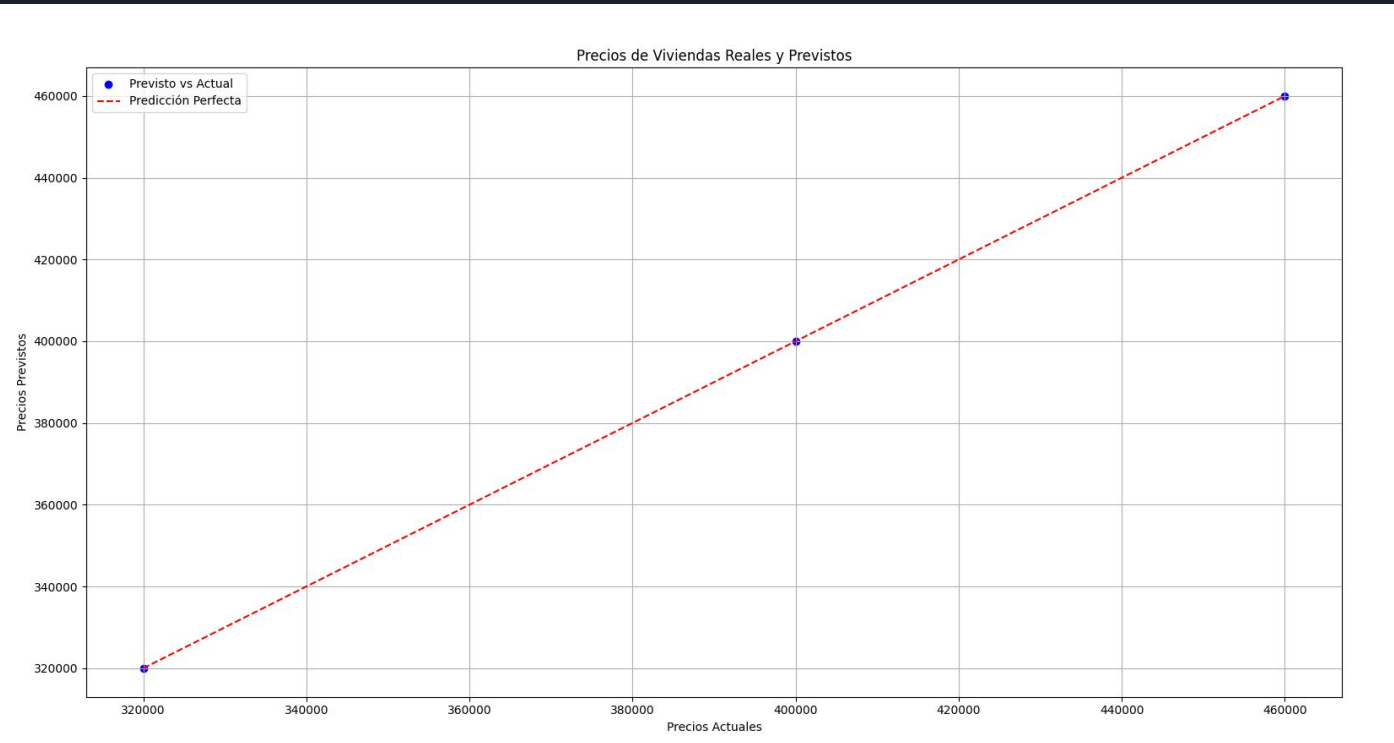
- Se muestran los **coeficientes** que indican el peso de cada característica en el modelo de regresión lineal (cómo influye cada variable en el precio).
- Se **imprime** también el valor del intercepto del modelo (el valor predicho cuando todas las características son cero).

10. Gráfico de dispersión de precios reales vs. predichos

```
# Gráfico General de predicción actual y el previsto
plt.figure(figsize=(10, 6))
plt.scatter(y_test, y_pred, color='blue', label='Previsto vs Actual')
plt.plot([min(y_test), max(y_test)], [min(y_test), max(y_test)], color='red', linestyle='--', label='Predicción Perfecta')
plt.xlabel('Precios Actuales')
plt.ylabel('Precios Previstos')
plt.title('Precios de Viviendas Reales y Previstos')
plt.legend()
plt.grid(True)
plt.show()
```

- Se crea un gráfico de dispersión con los precios actuales (`y_test`) en el eje X y los precios predichos (`y_pred`) en el eje Y.
- La línea roja muestra la predicción perfecta (donde los precios reales coinciden con los predichos).

Gráfico Estadístico Real y Previsto



11. Gráficos de dispersión para cada característica con respecto al precio

```
# Gráficos de dispersión de cada característica en función del precio
fig, axes = plt.subplots(1, 3, figsize=(18, 5), sharey=True)
fig.suptitle('Relación entre Características y Precio de la Vivienda')

# Area vs Precio
axes[0].scatter(df['Area'], df['Precio'], color='blue')
axes[0].set_xlabel('Área')
axes[0].set_ylabel('Precio')

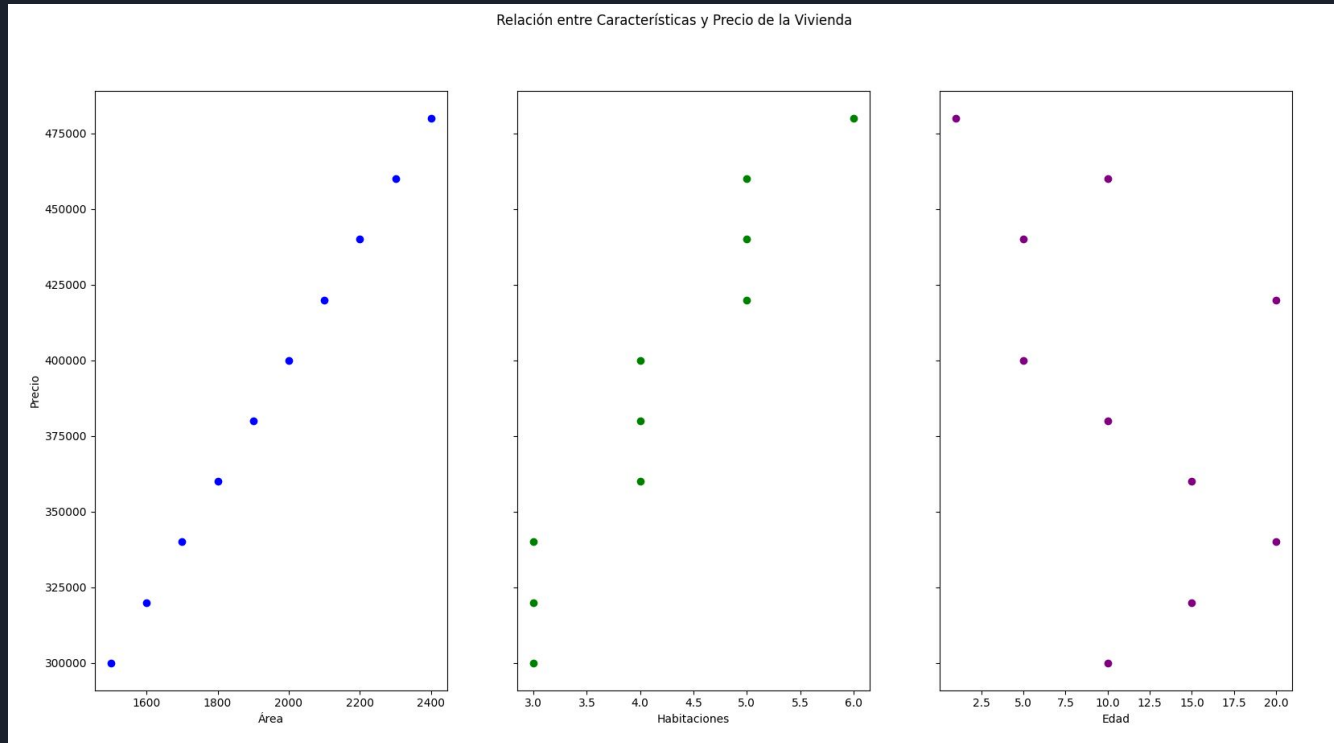
# Habitaciones vs Precio
axes[1].scatter(df['Habitaciones'], df['Precio'], color='green')
axes[1].set_xlabel('Habitaciones')

# Edad vs Precio
axes[2].scatter(df['Edad'], df['Precio'], color='purple')
axes[2].set_xlabel('Edad')

plt.show()
```

- Estos gráficos muestran cómo se relacionan las características individuales con el precio de la vivienda.

Gráfico de Características y Precios de la Vivienda






Regresión Lineal Múltiple

Introducción al concepto de regresión lineal múltiple

El modelo de regresión lineal múltiple es una extensión de la regresión lineal simple, donde en lugar de una sola variable independiente (predictora), se usan múltiples variables (en este caso, como el número de habitaciones, baños, y varios tamaños en pies cuadrados) para predecir el precio de una casa.



se usa **álgebra matricial** para manipular los datos. Los datos se organizan en matrices, donde cada fila representa una vivienda y cada columna una variable predictora.

MATRICES

Concepto

Se llama matriz de orden $m \times n$ a todo conjunto de elementos a_{ij} dispuestos en m líneas horizontales (filas) y n verticales (columnas) de la forma:

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1j} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2j} & \cdots & a_{2n} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ a_{i1} & a_{i2} & \cdots & a_{ij} & \cdots & a_{in} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mj} & \cdots & a_{mn} \end{bmatrix}$$

Nuestro proyecto HOME VALUE ESTIMATOR

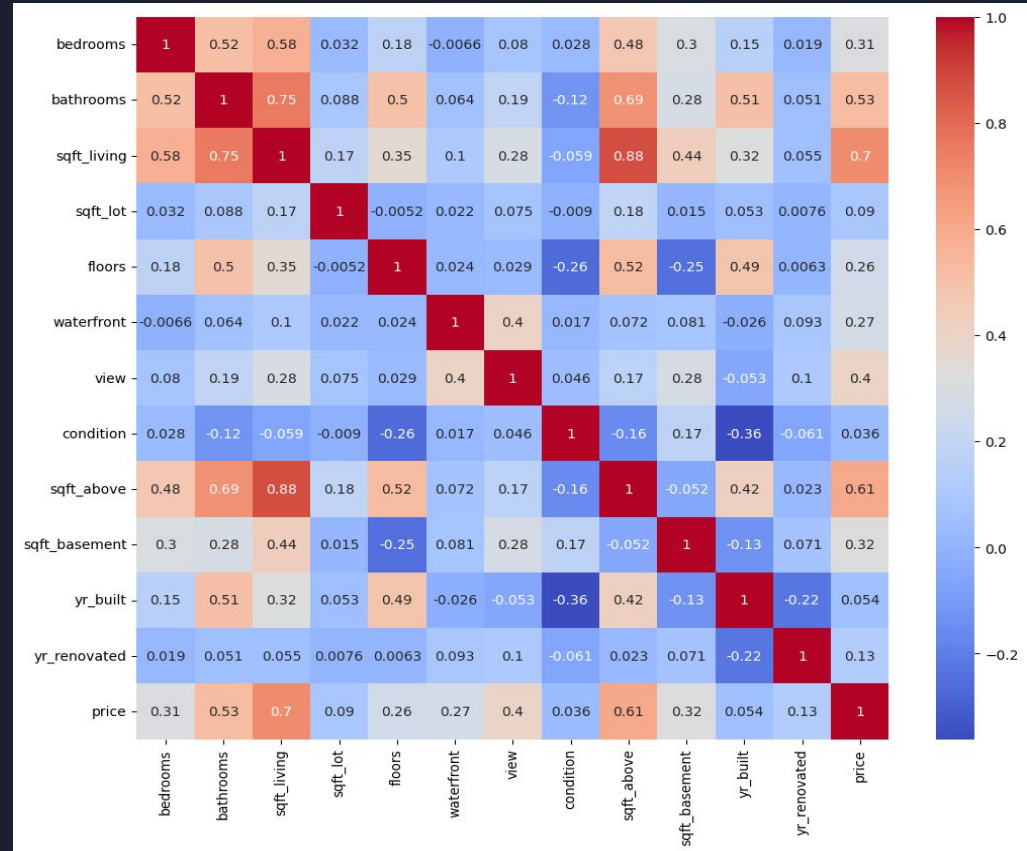
se realizó un predictor de precios de viviendas, este proyecto propone el desarrollo de un modelo de machine learning específicamente de regresión lineal múltiple que implementa algoritmos para analizar datos y proporcionar un predictor de precios según características de las viviendas

```
[ ] import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import statsmodels.api as sm
from sklearn.impute import SimpleImputer

[ ] from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score
```

Elección de variables

Luego para decidir las variables que seleccionaremos usamos La matriz de correlación permite visualizar las relaciones entre cada par de variables para decidir cuáles son más relevantes en el modelo. La relación entre variables se mide con coeficientes de correlación que varían entre -1 y 1.



La selección de variables predictoras

Elegimos la mas relevantes según el análisis anterior

- **bedrooms:** Habitaciones
- **bathrooms:** Baños
- **sqft_basement:** Tamaño del sótano en pies cuadrados
- **sqft_above:** Espacio interior de la vivienda sobre el nivel del suelo en pies cuadrados
- **sqft_living:** Espacio habitable en pies cuadrados

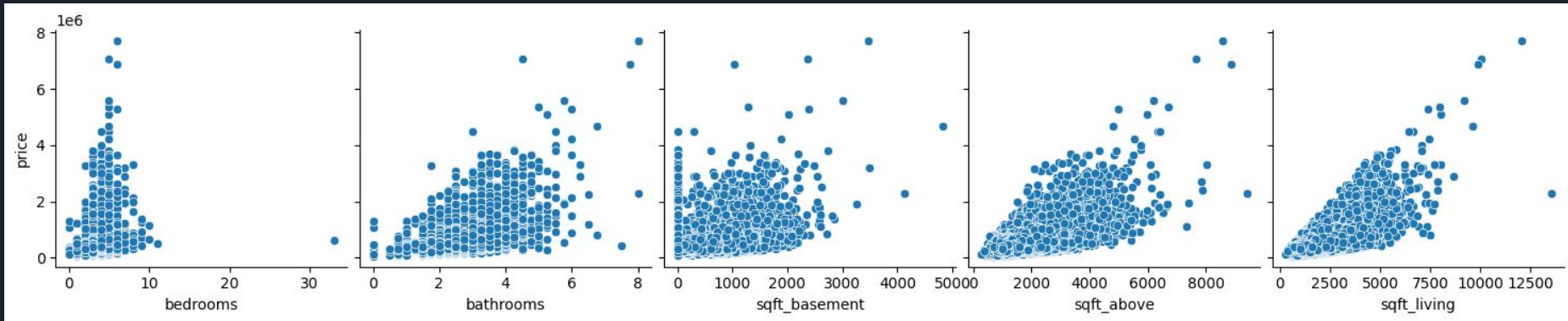
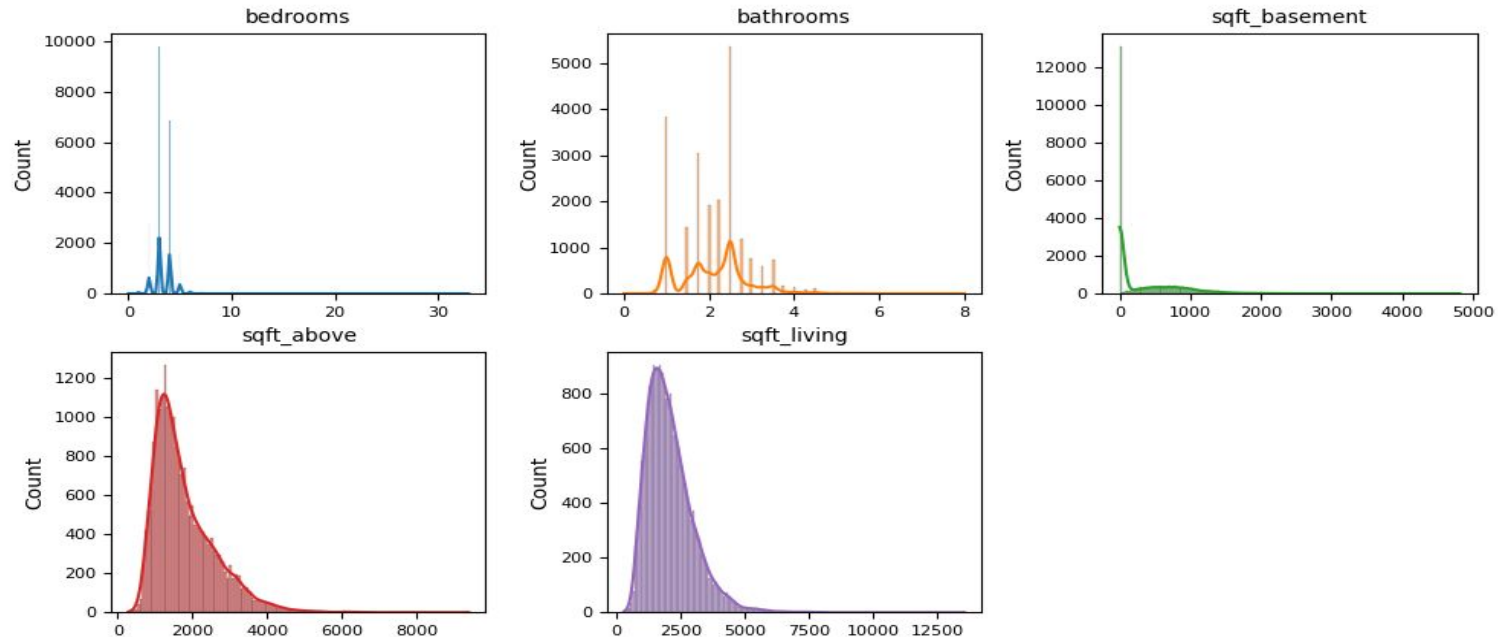


Grafico creado con matplotlib este gráfico te permite ver cómo se distribuyen los valores de cada una de estas variables en tu conjunto de datos, lo que puede ser útil para identificar patrones, tendencias y posibles anomalías

Distribución variables numéricas





Proceso y ajuste del modelo

```
] # Separar las características (X) y la variable objetivo (y)

X = data[['bedrooms', 'bathrooms', 'sqft_basement', 'sqft_above', 'sqft_living']]
y = data['price']
```

Se separan las variables y, x y luego dividen los datos de entrenamiento y de prueba siendo usualmente 80% de los datos se utiliza para entrenar el modelo y el 20% para probarlo. para garantizar que los coeficientes obtenidos no dependen exclusivamente del conjunto de entrenamiento.

Luego se crea y entrena el Modelo con: LinearRegression() que crea una instancia del modelo de regresión lineal.

```
# dividir los datos ne entranmiento y prueba
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
imputer = SimpleImputer(strategy='mean')

# crear modelo, y entrenarlo con los datos de entrenamiento.
model = LinearRegression() #Esto genera una ecuación lineal que intenta predecir

# y (el precio) en función de las variables en X (Rooms, Bathroom, BuildingArea).
model.fit(X_train, y_train)
```



Ajuste del Modelo

Cuando se ejecuta `model.fit(X_train, y_train)`, el modelo calcula los coeficientes (β) y la intersección (β_0) que mejor se ajustan a los datos de entrenamiento.

- Fórmula: $\beta = (X^T X)^{-1} X^T y$

`model.fit(X_train, y_train)`: Entrena el modelo utilizando los datos de entrenamiento. Durante este proceso, el modelo ajusta los coeficientes para minimizar la suma de los errores al cuadrado entre los valores predichos y los valores reales.




Predicción:

Al ejecutar `model.predict(X_test)`, el modelo aplica la ecuación de regresión lineal múltiple a los datos de prueba (`X_test`) utilizando los coeficientes e intersección calculados.

- Fórmula: $\hat{y} = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_n x_n$

- **Y** es la variable dependiente que queremos predecir
- **b** es la intersección o término constante.
- **b_1, b_2...**, son los coeficientes de las variables independientes.
- **X_1, X_2**, son las variables independientes (por ejemplo, número de habitaciones, tamaño del sótano, etc.).



```
# Realizar predicciones en los datos de prueba
print(X_test)
y_pred = model.predict(X_test)
```

Cuando se ejecuta `model.predict(X_test)`, el modelo utiliza esta ecuación para calcular las predicciones (y^{\wedge}) para cada fila de `X_test`

En resumen, `model.predict` aplica la ecuación de regresión lineal múltiple utilizando los coeficientes e intersección calculados durante el ajuste del modelo para generar las predicciones.

Coeficiente de Determinación (R^2)

$$R^2 = 1 - \frac{\sum (y_i - \hat{y}_i)^2}{\sum (y_i - \bar{y})^2}$$

- y_i : Valor real de la variable dependiente.
- \hat{y}_i : Valor predicho por el modelo.
- \bar{y} : Media de los valores reales de la variable dependiente.

Error Absoluto Medio (MAE)

$$MAE = \frac{1}{n} \sum |y_i - \hat{y}_i|$$

- n : Número de observaciones.
- y_i : Valor real de la variable dependiente.
- \hat{y}_i : Valor predicho por el modelo.

Error Cuadrático Medio (MSE)

$$MSE = \frac{1}{n} \sum (y_i - \hat{y}_i)^2$$

- n : Número de observaciones.
- y_i : Valor real de la variable dependiente.
- \hat{y}_i : Valor predicho por el modelo.

Métricas de evaluación

Mide qué tan bien el modelo explica la variabilidad de los datos

observados. Un R^2 cercano a 1 significa que el modelo explica bien los datos; un valor cercano a 0 significa que no los explica bien.

El MAE mide la magnitud promedio de los errores en un conjunto de predicciones. Es una medida de la precisión del modelo. Cuanto más pequeño sea el MAE, más preciso será nuestro modelo."

Es similar al MAE, pero penaliza más los errores grandes. Es una medida de la calidad del modelo y se utiliza comúnmente para ajustar los parámetros de los modelos.

Resultados de las métricas

```
r2 = r2_score(y_test, y_pred)
mse = mean_squared_error(y_test, y_pred)
mae = mean_absolute_error(y_test, y_pred)
mae = mean_absolute_error(y_test, y_pred)
```

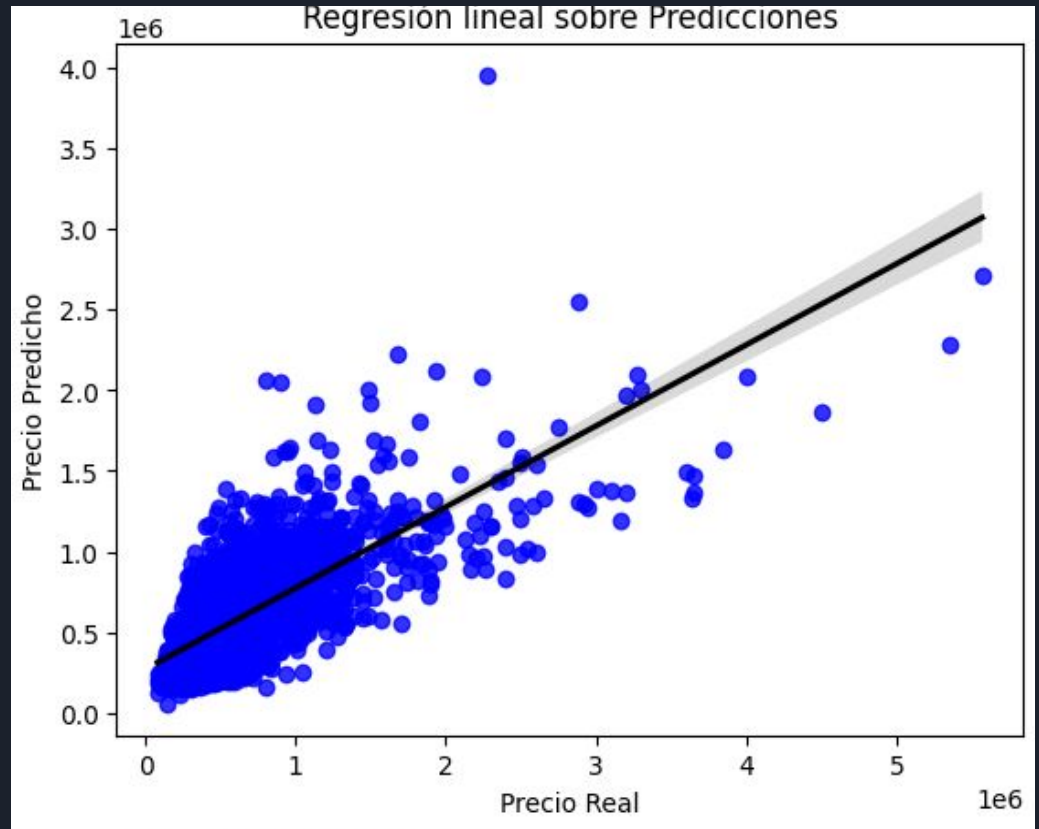
```
Coeficiente de determinación R^2: 0.51
Error cuadrático medio (MSE): 74198772469.47
Error absoluto medio: 174405.6777496375
```

- El coeficiente de determinación fue de 0.51 es decir aproximadamente el 51% de la variabilidad en los precios de las casas puede ser explicada por las variables independientes en el modelo. Esto indica que el modelo tiene una capacidad explicativa moderada, pero aún hay un 49% de la variabilidad que no está siendo capturada,
- El MAE (error absoluto medio) indica que, en promedio, las predicciones del modelo se desvían de los valores reales en aproximadamente 174,405.68 unidades monetarias (por ejemplo, dólares). Esto proporciona una medida de la magnitud promedio del error en las predicciones sin considerar la dirección del error.
- El MSE indica que, en promedio, las predicciones del modelo se desvían de los valores reales en una cantidad significativa. Un MSE alto sugiere que las predicciones del modelo no son muy precisas y que hay margen para mejorar la precisión del modelo.

Conclusión

El modelo de regresión lineal múltiple permite, mediante operaciones algebraicas, ajustar coeficientes que ayudan a aproximar el precio de una vivienda tiene una capacidad explicativa del: El R^2 de 0.51 sugiere que el modelo explica una proporción moderada de la variabilidad en los datos, es decir un 50%.

Precisión Limitada: Tanto el MSE como el MAE indican que hay una desviación significativa entre las predicciones y los valores reales, lo que sugiere que el modelo podría beneficiarse de mejoras adicionales. como mas variables o con técnicas más avanzadas como regularizacion u otros modelos de ml.



Grupo 12 | Entrega final Álgebra

Dirección General de
EDUCACIÓN TÉCNICA Y
FORMACIÓN PROFESIONALMinisterio de
EDUCACIÓN

▼ Modelo de regresión Lineal Múltiple

Problema: predecir el precio de una casa basado en los datos, usando regresión lineal.

Regresión Lineal Múltiple



Viabilidad del Modelo

El modelo puede ser útil como una primera aproximación para predecir los precios de las casas, pero hay margen para mejorar su precisión y capacidad explicativa. Para hacerlo más viable se podría considerar:

1. Incluir más variables relevantes: Agregar características adicionales que puedan influir en el precio de las casas.
2. Utilizar técnicas de regularización: Como Ridge o Lasso, para reducir el sobreajuste.
3. Probar con modelos más complejos: Como árboles de decisión, bosques aleatorios o redes neuronales.

En resumen, el modelo actual es un buen punto de partida, pero se beneficiaría de mejoras adicionales para ser más preciso y confiable en sus predicciones.