Laboratorio 1

Regresión Lineal

Geraldi Mejía Segura B84772

Preparación de datos

Importar bibliotecas

```
# Importar las bibliotecas necesarias
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_squared_error, r2_score
```

- numpy: permite el manejo de utilidades para números.
- matplotlib: permite realizar diferentes tipos de gráficos.
- sklearn: permite generar un modelo, entrenarlo y calcular su exactitud.

Dataset

- Se generan los datos para el entrenamiento y prueba de forma aleatoria. El tamaño en metros cuadrados y el precio en millones de colones.
- Nota: el precio se calcula en relación con el tamaño de la vivienda por eso lo incluye al momento de ser calculado.

```
# Generar datos de ejemplo
np.random.seed(0)
num_observaciones = 100
size = np.random.uniform(40, 200, (num_observaciones, 1))  # Tamaño de vivie
price = 100 + 0.6 * size + np.random.normal(0, 20, (num_observaciones, 1)) #
```

Preprocesamiento de datos

División de datos

- Se dividen los datos generados en 2 grupos:
  - 1. Grupo de entrenamiento para el modelo.
  - 2. Grupo de pruebas para ejecutar el modelo.

```
# Dividir los datos en conjuntos de entrenamiento y prueba
size_train, size_test, price_train, price_test = train_test_split(size, price, test_size=0.2, random_state=0)
```

Construcción del modelo

Entrenamiento

- Se genera un nuevo modelo de regresión lineal y se entrena con los datos de prueba generados anteriormente.

```
# Crear el modelo de regresión lineal
modelo = LinearRegression()

# Entrenar el modelo con los datos de entrenamiento
modelo.fit(size_train, price_train)
```

Evaluación del modelo

Predicción

- Se realiza una predicción basado en el tamaño de la vivienda de los datos de prueba esperando una predicción del precio ajustada a lo que realmente se esperaría.

```
# Realizar predicciones utilizando el conjunto de prueba
price_pred = modelo.predict(size_test)
```

Métricas de rendimiento

- Se contrasta el precio esperado con el precio predicho.
- Se obtiene:
  - Error cuadrático medio (Indica la magnitud promedio de los errores cuadrados entre los valores predichos por el modelo y los valores reales observados.)
  - Coeficiente de determinación (indica qué tan bien se ajustan los datos a la línea de regresión ajustada)

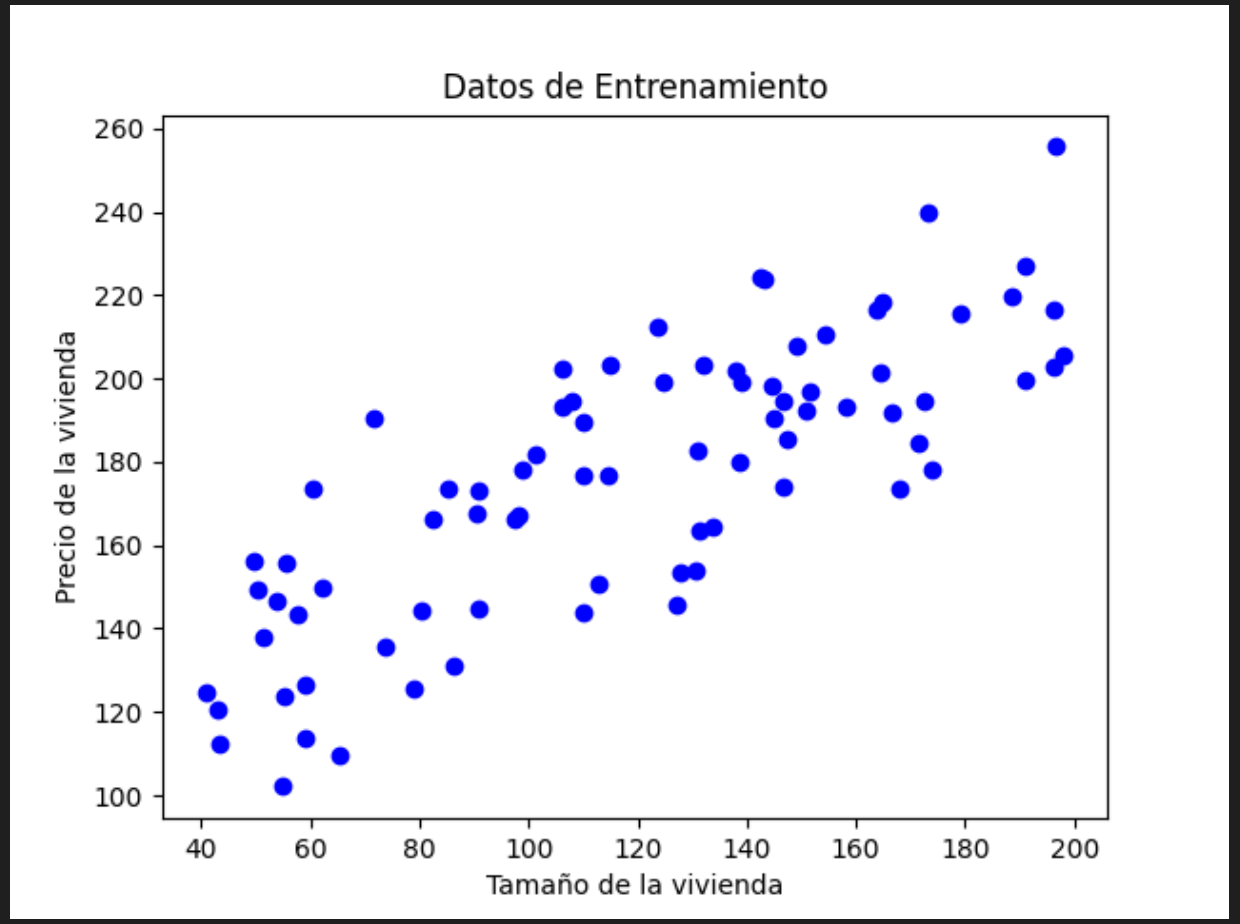
```
# Calcular el MSE y R^2
mse = mean_squared_error(price_test, price_pred)
r2 = r2_score(price_test, price_pred)
```

Visualización

Entrenamiento

- Se muestran los datos de entrenamiento del modelo.

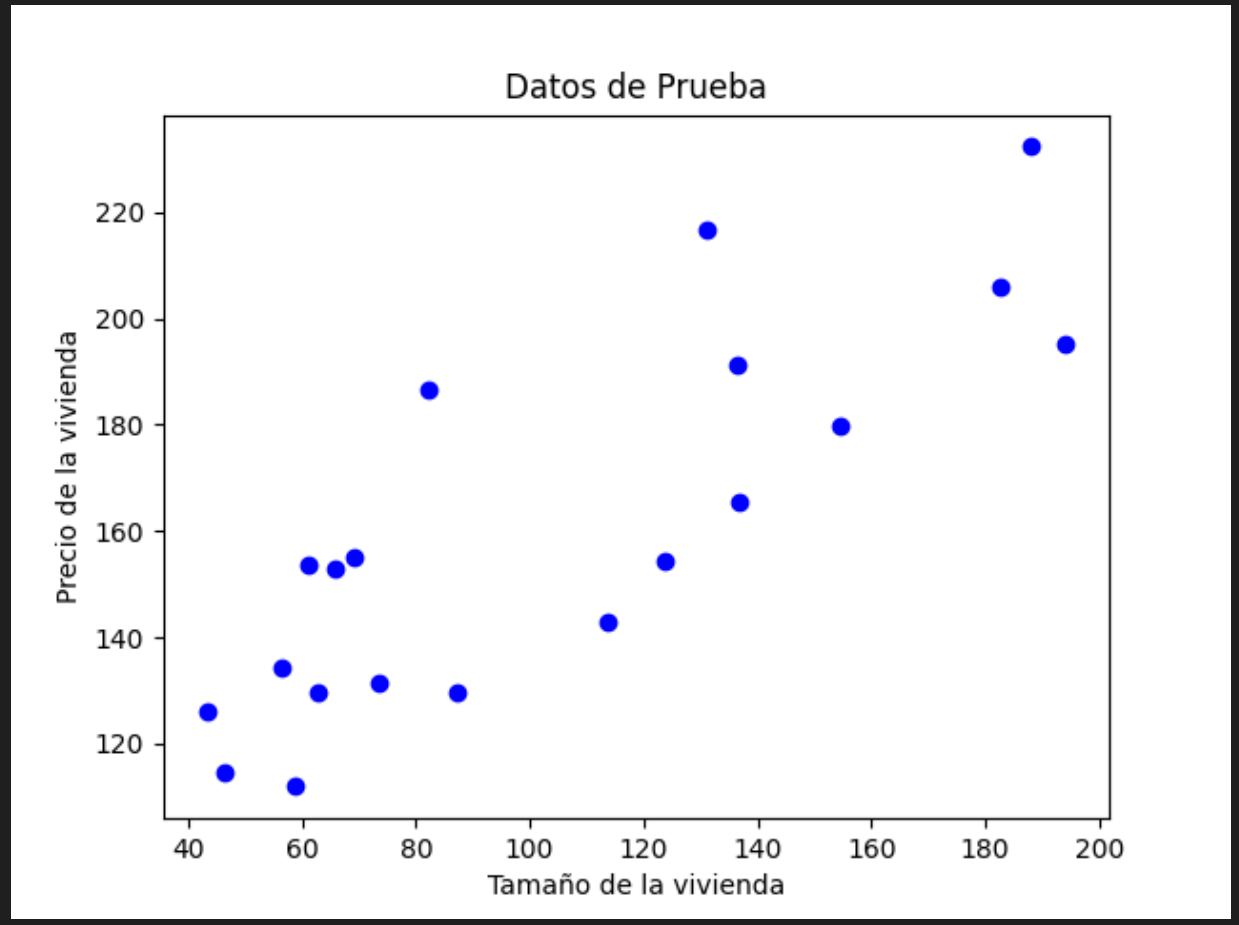
```
# Mostrar los datos de entrenamiento en un gráfico
plt.scatter(size_train, price_train, color='blue')
plt.title('Datos de Entrenamiento')
plt.xlabel('Tamaño de la vivienda')
plt.ylabel('Precio de la vivienda')
plt.show()
```



Prueba

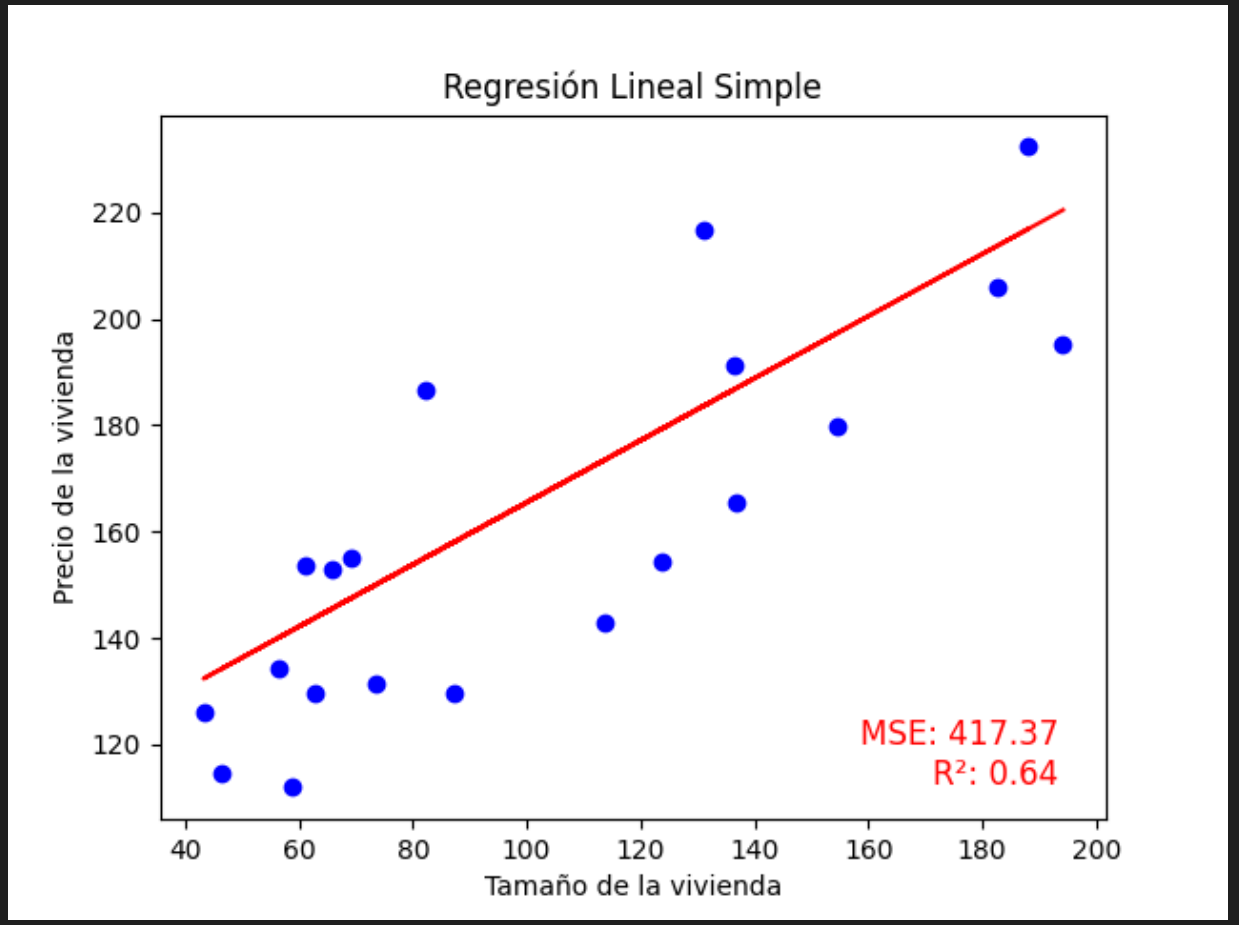
- Se muestran los datos de prueba seleccionados para poner a prueba el modelo.

```
# Mostrar los datos de prueba en un gráfico de dispersión
plt.scatter(size_test, price_test, color='blue')
plt.title('Datos de Prueba')
plt.xlabel('Tamaño de la vivienda')
plt.ylabel('Precio de la vivienda')
plt.show()
```



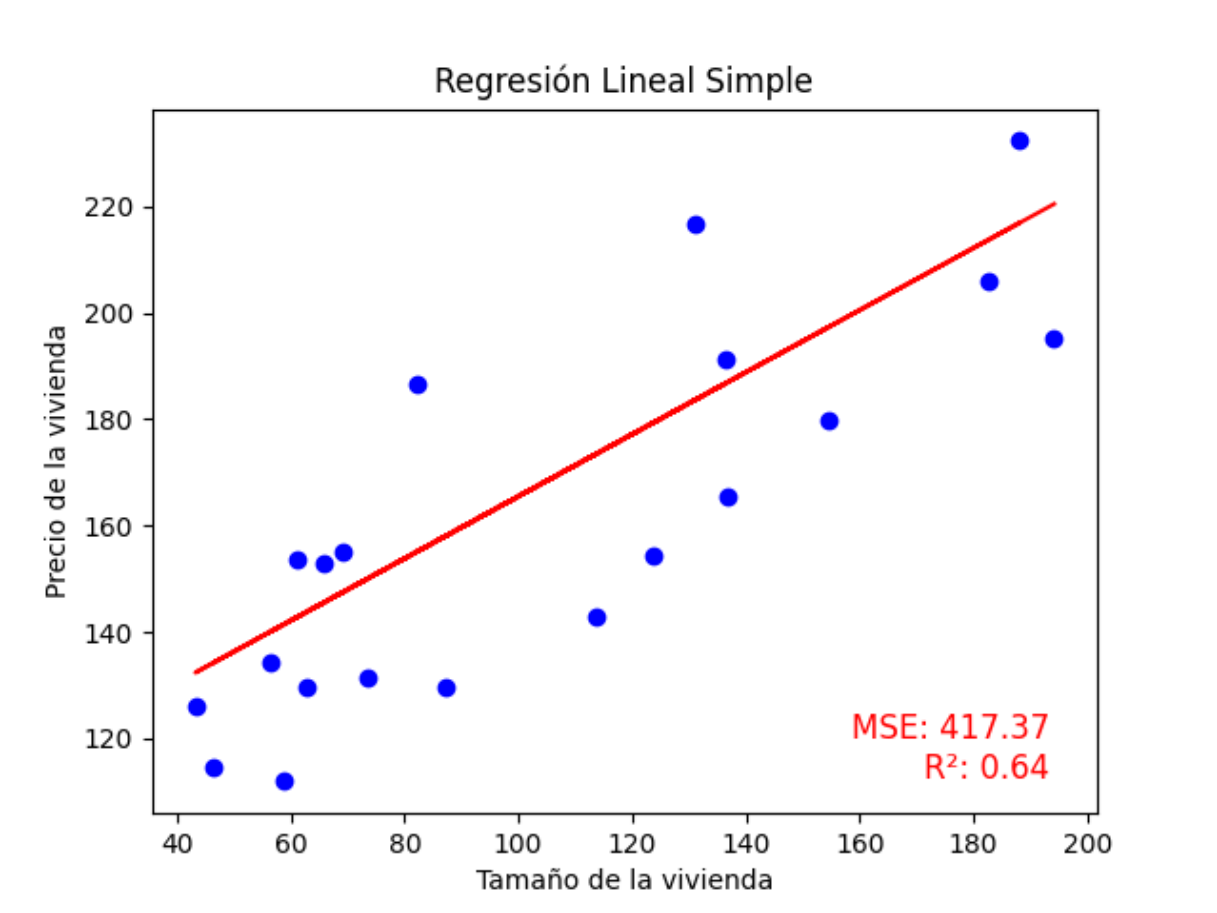
Modelo aplicado

- Se muestra la regresión lineal y los datos de prueba utilizados.



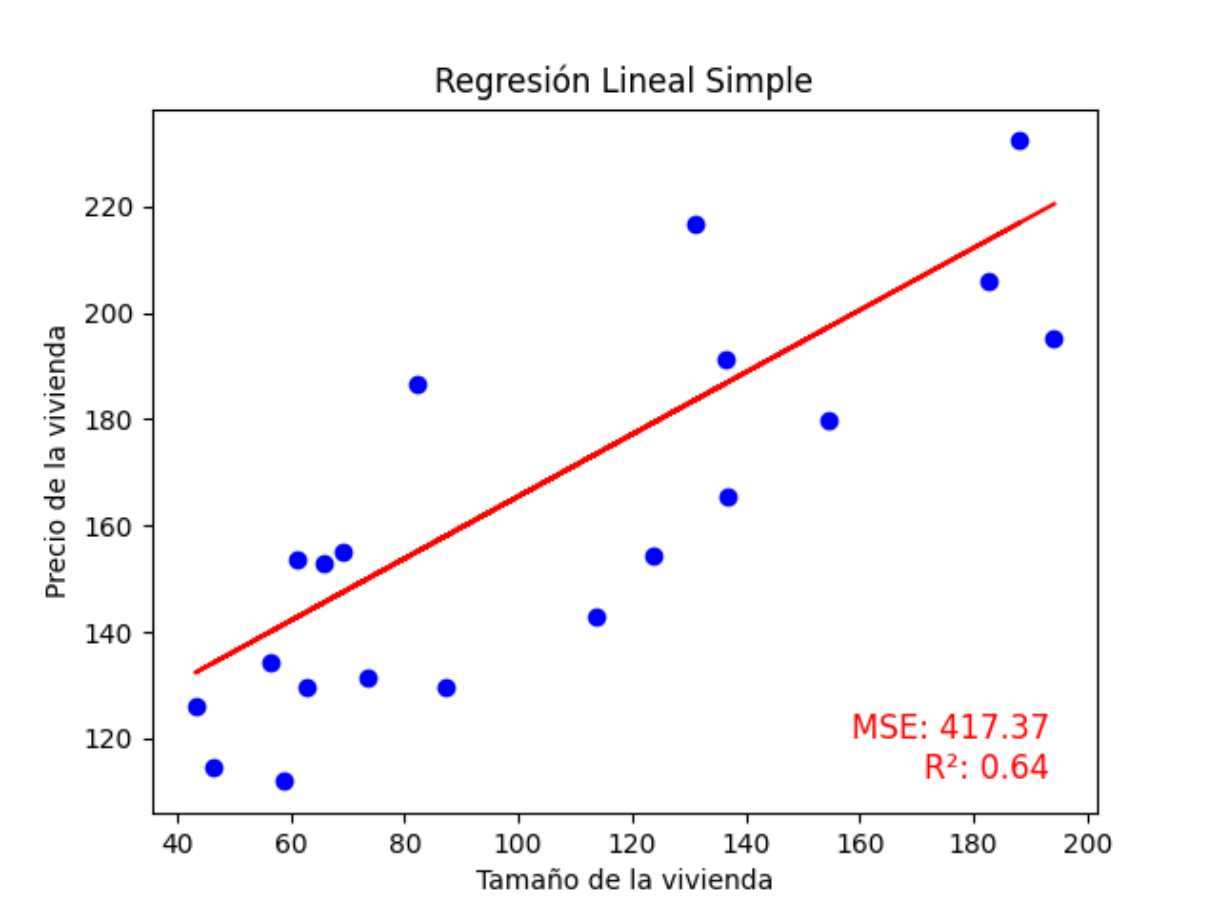

```
# Visualizar los resultados
plt.scatter(size_test, price_test, color='blue')
plt.plot(size_test, price_pred, color='red')
plt.title('Regresión Lineal Simple')
plt.xlabel('Tamaño de la vivienda')
plt.ylabel('Precio de la vivienda')
plt.text(0.95, 0.05, f"MSE: {mse:.2f}\nR²: {r2:.2f}", fontsize=12, color='red', transform=plt.gca().transAxes,
ha='right')

plt.show()
```

Resultados



- Se puede observar como la regresión lineal entrega valores cercanos a los esperados en el set de pruebas. En este caso se utilizaron 100 observaciones, para ajustar mejor el modelo se debe aumentar esta cantidad.

Resultados



- Aunque el R^2 de 0.64 indica que el modelo tiene un ajuste bueno a los datos, el MSE de 417 sugiere que todavía hay una cantidad significativa de error en las predicciones del modelo.

- En este caso se puede considerar una limpieza de datos extremos no representativos de la norma para ajustar mejor el modelo.