

1. Descripción del funcionamiento de la aplicación:

La aplicación del Laboratorio 1 es un modelo de regresión lineal que se encarga primero de generar datos aleatorios que representan la relación entre los metros cuadrados de una vivienda y su precio. Además, demuestra cómo construir, entrenar y evaluar un modelo de regresión lineal simple en Python utilizando la biblioteca scikit-learn, además de visualizar los resultados para comprender mejor la relación entre las variables y la capacidad predictiva del modelo.

2. Explicación del código:

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

➤ Se importan las bibliotecas necesarias para el análisis y construcción del modelo de regresión lineal. Estas bibliotecas incluyen:

- Numpy: para operaciones numéricas.
- Pandas: para manipulación de datos.
- Sklearn: para el modelo de regresión lineal y métricas de evaluación.
- Matplotlib: para visualización de datos.

➤ División de los datos en conjuntos de entrenamiento y prueba:

```
X = datos['metros_cuadrados'].values.reshape(-1, 1)
y = datos['precio_vivienda'].values
```

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42)
```

Se divide el DataFrame 'datos' en dos conjuntos: 'X' para las características (metros cuadrados) y 'y' para las etiquetas (precio de vivienda).

Luego, se divide estos conjuntos en conjuntos de entrenamiento (80%) y prueba (20%) utilizando la función `train_test_split()` de sklearn. El parámetro `random_state=42` se utiliza para garantizar reproducibilidad en la división de datos.

➤ Creación y entrenamiento del modelo de regresión lineal:

```
modelo = LinearRegression()
modelo.fit(X_train, y_train)
```

Se crea un objeto de modelo de regresión lineal utilizando la clase `LinearRegression()` de sklearn.

Luego, se entrena el modelo utilizando los datos de entrenamiento (`X_train, y_train`) mediante el método `fit()`.

➤ Realización de predicciones y evaluación del modelo:

```
predicciones = modelo.predict(X_test)
error = mean_squared_error(y_test, predicciones)
print('Error cuadrático medio:', error)
```

Se utilizan los datos de prueba (X_{test}) para realizar predicciones utilizando el método `predict()` del modelo entrenado.

Se calcula el error cuadrático medio (MSE) entre las predicciones y los valores reales utilizando la función `mean_squared_error()` de `sklearn`.

Se imprime el error cuadrático medio para evaluar el rendimiento del modelo.

➤ Visualización de los resultados:

```
plt.scatter(X_test, y_test, color='blue')
plt.plot(X_test, predicciones, color='red', linewidth=3)
plt.xlabel('Metros cuadrados')
plt.ylabel('Precio vivienda')
plt.title('Modelo de regresión lineal')
plt.show()
```

Se visualizan los resultados del modelo de regresión lineal utilizando un gráfico de dispersión (scatter plot) para los datos de prueba (X_{test} , y_{test}) y la línea de regresión generada por el modelo.

Se etiquetan los ejes y se añade un título al gráfico para una mejor comprensión.

● En cuanto a la interpretación de la gráfica generada por el código:

- ➔ En el eje X (horizontal) del gráfico, tienes los metros cuadrados, que representan las características de tu modelo.
- ➔ En el eje Y (vertical) del gráfico, tienes el precio de la vivienda, que es la variable dependiente que estás tratando de predecir.

La gráfica incluye dos elementos principales:

Puntos azules: Estos puntos representan los datos reales de metros cuadrados y precios de vivienda en el conjunto de prueba. Cada punto azul representa un par de valores (metros cuadrados, precio de la vivienda) del conjunto de prueba.

Línea roja: Esta línea representa la línea de regresión generada por el modelo de regresión lineal. La línea de regresión muestra la relación estimada entre los metros cuadrados y el precio de la vivienda según el modelo. La dirección y la pendiente de la línea indican la tendencia de la relación estimada por el modelo.