Sistemas Expertos
Laboratorio 1
Milena Rojas Leiva B76703

Documentación Laboratorio 1

```
7   # Importar las bibliotecas necesarias
8   import numpy as np
9   import pandas as pd
10  import matplotlib.pyplot as plt
11  from sklearn.model_selection import train_test_split
12  from sklearn.linear_model import LinearRegression
13  from sklearn.metrics import mean_squared_error, r2_score
14
```

Numpy se utiliza para trabajar con matrices y funciones matemáticas.

Matplotlib se usa para poder realizar la visualización de los datos.

Sklearn-learn importa una función (la primera) que divide el conjunto de datos en entrenamiento y prueba, una clase (la segunda) que sirve para hacer el modelo de regresión lineal y de tercero importa 2 funciones que evalúan el rendimiento del modelo.

```
16  # Generación de datos sintéticos
17  np.random.seed(0)  # Para reproducibilidad
18  tamaño = np.random.rand(100, 1) * 125  # Tamaño de las viviendas en metros cuadrados
19  precio = (tamaño * 4) + (np.random.rand(100, 1) * 100)  # Precio de las viviendas en miles de dólares
20
21
22  # Convertir los datos a un DataFrame de pandas
23  data = pd.DataFrame(data=np.hstack((tamaño, precio)), columns=['tamaño', 'precio'])
24
```

En las líneas 17,18 y 19 de código se generan los datos sintéticos aleatorios para el tamaño y precio de las viviendas.

En la línea 22 se estructuran los datos de forma tabular, en este caso los datos sintéticos de las líneas anteriores.

```
26    # Explorar y visualizar los datos
27    plt.scatter(data['tamaño'], data['precio'])
28    plt.title('Relación entre tamaño y precio de la vivienda')
29    plt.xlabel('Tamaño (m^2)')
30    plt.ylabel('Precio ($)')
31    plt.show()
32
```

Con estas líneas logramos crear un diagrama de dispersión que se rige con el tamaño de la vivienda en el eje x y el precio de las viviendas en el eje y, de igual forma el diagrama se visualiza con las respectivas etiquetas.

```
32
33    # Preprocesamiento de Datos
34    X = data['tamaño'].values.reshape(-1, 1)  # Característica (variable independiente)
35    y = data['precio'].values  # Variable dependiente
36
37    # Dividir el dataset en entrenamiento y prueba
38    X_train, X_test, y_train, y_test = train_test_split( *arrays: X, y, test_size=0.2, random_state=42)
39
```

En las líneas 34 y 35 extraemos los valores del dataFrme, devolviendo una matriz unidimensional que contiene los tamaños de las viviendas y los precios de las viviendas por separado.

En la línea 38 se dividen los datos en dos conjuntos uno de entrenamiento y otro conjunto de prueba, lo que permite poder entrenar el modelo con una parte de los datos y probar su rendimiento con la otra aparte. Para este caso se utiliza un 20% de los datos como conjunto de prueba.

```
40    # Construcción del Modelo
41    model = LinearRegression()
42    # Entrenar el modelo
43    model.fit(X_train, y_train)
44
45    # Evaluación del Modelo
46    y_pred = model.predict(X_test)
47    mse = mean_squared_error(y_test, y_pred)
48    r2 = r2_score(y_test, y_pred)
49    print(f"Error cuadrático medio (MSE):", mse)
50    print(f"Coeficiente de determinación (R²):", r2)
```

En la línea 41 se crea la instancia que representa el modelo de regresión lineal.

En la línea 42 se utiliza el método fit() del modelo para entrenarlos con los datos ya seleccionados

anteriormente.

De la línea 46 a la 48 se hacen las predicciones con los datos de prueba y se evalúa el rendimiento utilizando el error cuadrático (MSE) y el coeficiente de determinación (R^2). Esto nos indica que tan bien se ajusta el modelo que se construyó a los datos de prueba.

```
52    # Visualización
53    plt.scatter(X_test, y_test, color='black')
54    plt.plot( *args: X_test, y_pred, color='blue', linewidth=3)
55    plt.title('Regresión lineal simple')
56    plt.xlabel('Tamaño (m^2)')
57    plt.ylabel('Precio ($)')
58    plt.show()
59
```

En las últimas líneas del laboratorio generamos el diagrama de dispersión con los datos, para así comprender de forma visual la relación que existe entre el tamaño y el precio de las viviendas, además que se comprende la calidad del ajuste del modelo.