

Introducción al Laboratorio de Machine Learning con Python y Scikit-Learn

Juan David Tamayo Quintero

May 2024

1. Configuración del Entorno

Para instalar y cargar las bibliotecas necesarias, asegúrate de que tienes acceso a Google Colab o a un entorno Python con *pip*. A continuación, el código para instalar las bibliotecas necesarias:

```
1 # Instala bibliotecas si no estan preinstaladas (generalmente ya lo estan)
2 !pip install scikit-learn
3 !pip install matplotlib pandas numpy seaborn
4
5 # Importa las bibliotecas comunes
6 import numpy as np
7 import pandas as pd
8 import matplotlib.pyplot as plt
9 import seaborn as sns
10 from sklearn import datasets, model_selection, metrics, preprocessing, linear_model
```

2. Cargar el Conjunto de Datos

Vamos a utilizar el conjunto de datos California Housing, disponible en la biblioteca **Scikit-Learn**. El código para cargarlo es el siguiente:

```
1 # Cargar un conjunto de datos de ejemplo
2 from sklearn.datasets import fetch_california_housing
3 california = fetch_california_housing()
4 X = pd.DataFrame(california.data, columns=california.feature_names)
5 y = pd.Series(california.target, name='Target')
```

3. Preprocesamiento de Datos

En esta etapa, vamos a normalizar las características para que tengan una escala consistente.

```

1 # Normalizar las características
2 scaler = preprocessing.StandardScaler()
3 X_scaled = scaler.fit_transform(X)

```

4. División del Conjunto de Datos

Vamos a dividir el conjunto de datos en un 80 % para entrenamiento y un 20 % para pruebas.

```

1 # Dividir los datos en conjunto de entrenamiento y prueba
2 X_train, X_test, y_train, y_test = model_selection.train_test_split(X_scaled, y, test

```

5. Entrenamiento del Modelo

En esta sección, entrenamos un modelo de regresión lineal con los datos de entrenamiento.

```

1 # Entrenar un modelo de regresion lineal
2 model = linear_model.LinearRegression()
3 model.fit(X_train, y_train)

```

6. Evaluación del Modelo

Para evaluar el rendimiento del modelo, calculamos el RMSE (Raíz Cuadrada del Error Cuadrático Medio) y el coeficiente R^2 .

```

1 # Realizar predicciones y evaluar el modelo
2 y_pred = model.predict(X_test)
3 rmse = np.sqrt(metrics.mean_squared_error(y_test, y_pred))
4 r2 = metrics.r2_score(y_test, y_pred)
5
6 print(f"RMSE: {rmse}")
7 print(f"R^2: {r2}")

```

7. Conclusiones

Los resultados del laboratorio muestran que el modelo de regresión lineal tiene un RMSE de `rmse` y un coeficiente R^2 de `r2`. Los valores pueden variar dependiendo del conjunto de datos utilizado, y el modelo puede mejorarse utilizando métodos de regularización o modelos más complejos.