

Laboratorio de Machine Learning con Scikit-Learn y Matriz de Confusión

Juan David Tamayo Quintero

Mayo 2024

1. Introducción

En este laboratorio, usaremos la biblioteca **Scikit-Learn** para entrenar un modelo de clasificación y evaluar su desempeño utilizando una matriz de confusión.

1. Cargar el conjunto de datos 'Breast Cancer' de Scikit-Learn.
2. Entrenar un modelo de clasificación.
3. Evaluar el modelo con una matriz de confusión.

2. Configuración del Entorno

Para instalar Scikit-Learn, usa el siguiente comando:

```
1 # Instalar Scikit-Learn si no esta preinstalado
2 !pip install scikit-learn
```

3. Carga y Preprocesamiento del Conjunto de Datos

Vamos a utilizar el conjunto de datos 'Breast Cancer' para construir un modelo.

```
1 # Importar bibliotecas necesarias
2 import numpy as np
3 import pandas as pd
4 from sklearn.datasets import load_breast_cancer
5 from sklearn.model_selection import train_test_split
6 from sklearn.preprocessing import StandardScaler
7
8 # Cargar el conjunto de datos Breast Cancer
```

```

9 cancer = load_breast_cancer()
10 X = pd.DataFrame(cancer.data, columns=cancer.feature_names)
11 y = pd.Series(cancer.target)
12
13 # Dividir los datos en entrenamiento (80%) y prueba (20%)
14 X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
15
16 # Normalizar las características para que estén en la misma escala
17 scaler = StandardScaler()
18 X_train = scaler.fit_transform(X_train)
19 X_test = scaler.transform(X_test)

```

4. Entrenamiento del Modelo

Vamos a entrenar un modelo de clasificación, en este caso un ‘RandomForestClassifier’.

```

1 from sklearn.ensemble import RandomForestClassifier
2
3 # Crear el modelo de Random Forest
4 model = RandomForestClassifier(random_state=42)
5
6 # Entrenar el modelo con los datos de entrenamiento
7 model.fit(X_train, y_train)

```

5. Evaluación del Modelo con Matriz de Confusión

Para evaluar el modelo, se usa una matriz de confusión que permite visualizar el rendimiento del clasificador.

```

1 from sklearn.metrics import confusion_matrix, classification_report
2 import seaborn as sns
3 import matplotlib.pyplot as plt
4
5 # Realizar predicciones con los datos de prueba
6 y_pred = model.predict(X_test)
7
8 # Generar la matriz de confusion
9 matriz_confusion = confusion_matrix(y_test, y_pred)
10
11 # Visualizar la matriz de confusion usando seaborn
12 plt.figure(figsize=(8, 6))
13 sns.heatmap(matriz_confusion, annot=True, fmt="d", cmap="Blues")
14 plt.xlabel("Prediccion")
15 plt.ylabel("Realidad")

```

```
16 plt.title("Matriz de Confusino - Clasificacion")
17 plt.show()
18
19 # Generar un informe de clasificacion
20 informe = classification_report(y_test, y_pred)
21 print("\nInforme de Clasificacion:\n", informe)
```

6. Conclusiones

En este laboratorio, entrenamos un modelo de clasificación usando el conjunto de datos ‘Breast Cancer’. Luego evaluamos el modelo con una matriz de confusión, proporcionando una representación visual clara de su rendimiento. Además, se generó un informe de clasificación con métricas como precisión y sensibilidad.