Actividad: Random Forest con Python

March 30, 2025

1 Introducción

Random Forest es un método de aprendizaje automático basado en la técnica de ensamble que utiliza múltiples árboles de decisión. Cada árbol se entrena con una muestra aleatoria de los datos y características disponibles, y la predicción final se realiza mediante votación o promediando las predicciones individuales. La importancia de Random Forest radica en su capacidad para mejorar la precisión, manejar grandes volúmenes de datos, reducir el sobreajuste (overfitting) y ofrecer una manera efectiva de determinar la importancia relativa de cada característica.

2 Metodología

Para realizar esta actividad se siguieron las instrucciones propuestas en las páginas especificadas del libro *Aprende Machine Learning en Español* de Juan Ignacio Bagnato. Se utilizó Python con la librería `scikit-learn` para entrenar y evaluar el modelo de Random Forest. El proceso involucró las siguientes etapas:

2.1 Carga y preparación de los datos

En primer lugar, se importaron las librerías necesarias y se cargaron los datos, dividiéndolos en conjuntos de entrenamiento y prueba.

```
import pandas as pd
from sklearn.model_selection import train_test_split

# Carga de datos
df = pd.read_csv('data.csv')
X = df.drop('target', axis=1)
y = df['target']

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

2.2 Creación y entrenamiento del modelo

Posteriormente, se configuró el modelo de Random Forest especificando los hiperparámetros clave y se entrenó usando los datos preparados.

```
from sklearn.ensemble import RandomForestClassifier

# Crear y entrenar modelo
modelo_rf = RandomForestClassifier(n_estimators=100, max_depth=10, random_state
    =42)
modelo_rf.fit(X_train, y_train)
```

2.3 Evaluación del modelo

Finalmente, se evaluó el rendimiento del modelo mediante métricas como precisión, recall y la matriz de confusión.

```
from sklearn.metrics import accuracy_score, classification_report,
    confusion_matrix

# Predicciones
y_pred = modelo_rf.predict(X_test)

# Evaluacion
accuracy = accuracy_score(y_test, y_pred)
report = classification_report(y_test, y_pred)
confusion = confusion_matrix(y_test, y_pred)
```

3 Resultados

Los resultados obtenidos indican que el modelo de Random Forest tuvo un rendimiento satisfactorio, con una precisión general del modelo del 85%. A continuación se muestra la matriz de confusión y el reporte de clasificación detallado que ilustra el desempeño del modelo por clases:

```
Accuracy: 0.85

Confusion Matrix:
[[50 10]
 [ 8 32]]

Classification Report:
              precision    recall  f1-score   support

           0       0.86      0.83      0.85        60
           1       0.76      0.80      0.78        40

    accuracy                           0.85       100
   macro avg       0.81      0.82      0.81       100
weighted avg       0.85      0.85      0.85       100
```

4 Conclusión

La implementación de Random Forest permitió obtener un modelo robusto y efectivo para realizar predicciones precisas sobre el conjunto de datos proporcionado. El uso de técnicas de ensamble como Random Forest resultó útil para mejorar la capacidad predictiva respecto a métodos más simples, y facilitó además la identificación de variables clave en la clasificación. La actividad mostró claramente cómo aplicar esta técnica en situaciones prácticas, resaltando su utilidad en el análisis predictivo en diversos contextos.