Clasificación Binaria con Random Forest: Aplicación al Diagnóstico de Cáncer de Mama

Yury Patricia Basto Figueroa
Universidad El Bosque
Curso: Machine Learning 2

2025-I

Proyecto Machine Learning 2
Profesor: Jorge Enrique García Farieta
Semestre: 2025-2

1. Introducción

En la actualidad, el análisis de datos permite extraer patrones relevantes para la toma de decisiones en distintas áreas como la medicina, la biología, las finanzas y la ingeniería. Una de las tareas más comunes en el campo del aprendizaje automático (*Machine Learning*) es la **clasificación binaria**, que consiste en asignar una etiqueta entre dos posibles categorías a un conjunto de observaciones.

El presente proyecto tiene como objetivo aplicar el algoritmo **Random Forest** a un problema de clasificación binaria, evaluando su desempeño mediante la métrica de **accuracy (precisión global)**.

2. Idea Principal

El proyecto busca construir un modelo de clasificación binaria utilizando **Random Forest** para predecir correctamente la clase de nuevas observaciones.

Se tomará como ejemplo el **dataset de cáncer de mama** de *Scikit-learn*, que contiene información sobre características de tumores y su diagnóstico (maligno o benigno). La idea es comprobar la efectividad de Random Forest en un problema real de la salud, mostrando paso a paso la preparación de los datos, el entrenamiento del modelo y la evaluación de los resultados.

3. Problema a Solucionar

El diagnóstico temprano del cáncer de mama es esencial para la supervivencia de los pacientes. Los médicos cuentan con múltiples pruebas y características (tamaño del tumor, textura, suavidad, concavidad, entre otras), pero el análisis manual puede ser lento y propenso a errores.

El problema que se busca resolver es:

> ¿Es posible construir un modelo automático que clasifique un tumor como benigno o maligno a partir de sus características medibles, con alta precisión?

4. Descripción de Random Forest

El algoritmo **Random Forest** es un método de *ensamble* basado en la construcción de múltiples árboles de decisión.

4.1. Cómo funciona

1. A partir del conjunto de datos original, se generan múltiples subconjuntos mediante muestreo aleatorio con reemplazo (*bootstrap*).
2. Para cada subconjunto, se entrena un árbol de decisión.
3. En cada nodo del árbol, la división se realiza seleccionando aleatoriamente un subconjunto de variables, lo que introduce mayor diversidad entre los árboles.
4. Para predecir una nueva muestra, cada árbol emite un voto y la clase final es la más votada (clasificación).

4.2. Ventajas

- Es robusto frente al sobreajuste en comparación con un árbol de decisión único.
- Maneja bien datos con muchas variables.
- No requiere un fuerte preprocesamiento de los datos.

4.3. Aplicaciones

- Diagnóstico médico (como en el caso del cáncer de mama).
- Detección de fraudes en transacciones bancarias.
- Clasificación de texto e imágenes.

5. Metodología

5.1. Selección del dataset

Se utilizará el dataset *Breast Cancer Wisconsin* incluido en *Scikit-learn*.

- Etiquetas: 0 = maligno, 1 = benigno.
- Número de muestras: 569.
- Número de variables: 30 características numéricas.

5.2. Preprocesamiento de datos

1. Cargar y organizar el dataset en `pandas`.
2. Dividir en entrenamiento (70 %) y prueba (30 %).

5.3. Modelo de Clasificación

- Algoritmo: Random Forest Classifier.
- Parámetros principales: número de árboles (`n_estimators = 100`), semilla aleatoria (`random_state = 42`).

5.4. Entrenamiento y Predicción

1. Ajuste del modelo con los datos de entrenamiento.
2. Predicción sobre el conjunto de prueba.

5.5. Métrica de Evaluación

Se utilizará **accuracy**, que mide la proporción de casos correctamente clasificados.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Donde:

- TP: verdaderos positivos
- TN: verdaderos negativos
- FP: falsos positivos
- FN: falsos negativos

6. Resultados Esperados

- Se espera obtener un accuracy superior al 95 %, dado que Random Forest es altamente efectivo en este dataset.
- La matriz de confusión permitirá analizar si el modelo tiende a cometer más errores en los casos malignos o benignos.
- Se generará un reporte de clasificación con precisión, recall y F1-score para complementar el análisis.

7. Conclusiones

El proyecto mostrará cómo los métodos de *Machine Learning* pueden aplicarse a problemas reales de clasificación binaria, en este caso en el área médica.

El algoritmo Random Forest ofrece un balance entre precisión y robustez, permitiendo manejar datasets con múltiples características sin necesidad de un preprocesamiento complejo.

La métrica de accuracy permitirá verificar de manera sencilla la capacidad del modelo para clasificar correctamente nuevos casos, lo cual resulta fundamental en aplicaciones sensibles como la detección de enfermedades.

Anexo: Código Python Explicado

Listing 1: Entrenamiento y evaluación de Random Forest en Python

```
# 1. Importar librer as necesarias
import pandas as pd
from sklearn.datasets import load_breast_cancer
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, confusion_matrix,
    classification_report

# 2. Cargar el dataset de c ncer de mama
```

```
data = load_breast_cancer()
X = pd.DataFrame(data.data, columns=data.feature_names)
y = pd.Series(data.target)  # 0 = maligno, 1 = benigno

# 3. Dividir en conjunto de entrenamiento (70%) y prueba (30%)
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.3, random_state=42
)

# 4. Crear y entrenar el modelo Random Forest
modelo = RandomForestClassifier(n_estimators=100, random_state=42)
modelo.fit(X_train, y_train)

# 5. Hacer predicciones
y_pred = modelo.predict(X_test)

# 6. Evaluar con Accuracy
accuracy = accuracy_score(y_test, y_pred)
print(f"Accuracy del modelo: {accuracy:.2f}")

# 7. Mostrar resultados adicionales
print("Matriz de confusi n:\n", confusion_matrix(y_test, y_pred))
print("\nReporte de clasificaci n:\n", classification_report(y_test,
   y_pred))
```