

Reporte: Modelo Random Forest para Clasificación del Cáncer de Mama

Justificación del Algoritmo

El algoritmo Random Forest fue seleccionado debido a su robustez y capacidad para manejar problemas de clasificación con múltiples características. Random Forest es particularmente útil en este caso porque puede manejar conjuntos de datos con características correlacionadas, como ocurre en análisis médicos, y ofrece una alta precisión al combinar los resultados de múltiples árboles de decisión.

Descripción del Diseño del Modelo

El modelo fue diseñado utilizando el conjunto de datos `breast_cancer.csv`. Los pasos seguidos para el diseño incluyen:

1. Preprocesamiento de datos: Se eliminaron columnas irrelevantes como 'id', y se convirtió la variable objetivo ('diagnosis') a valores numéricos.
2. Dividir los datos: Los datos se dividieron en un conjunto de entrenamiento y prueba utilizando un 80%-20%.
3. Entrenar el modelo: Se utilizó un clasificador Random Forest con 100 estimadores y una semilla aleatoria fija para reproducibilidad.

```
# Importar las bibliotecas necesarias
import pandas as pd

# Cargar el conjunto de datos
data = pd.read_csv('breast-cancer.csv')

# Eliminar la columna 'id' y preprocesar la variable objetivo
data = data.drop(['id'], axis=1)
data['diagnosis'] = data['diagnosis'].map({'M': 1, 'B': 0})

# Separar las características y la variable objetivo
X = data.drop('diagnosis', axis=1)
y = data['diagnosis']

# Mostrar un resumen de los datos
data.head()
```

Entrenamiento del Modelo

```
4]: from sklearn.model_selection import train_test_split
    from sklearn.ensemble import RandomForestClassifier
    from sklearn.metrics import accuracy_score

    # Dividir el conjunto de datos
    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42, stratify=y)

    # Entrenar el modelo Random Forest
    model = RandomForestClassifier(random_state=42, n_estimators=100)
    model.fit(X_train, y_train)

    # Realizar predicciones
    y_pred = model.predict(X_test)
    accuracy = accuracy_score(y_test, y_pred)
    print(f"Precisión del modelo: {accuracy:.2f}")

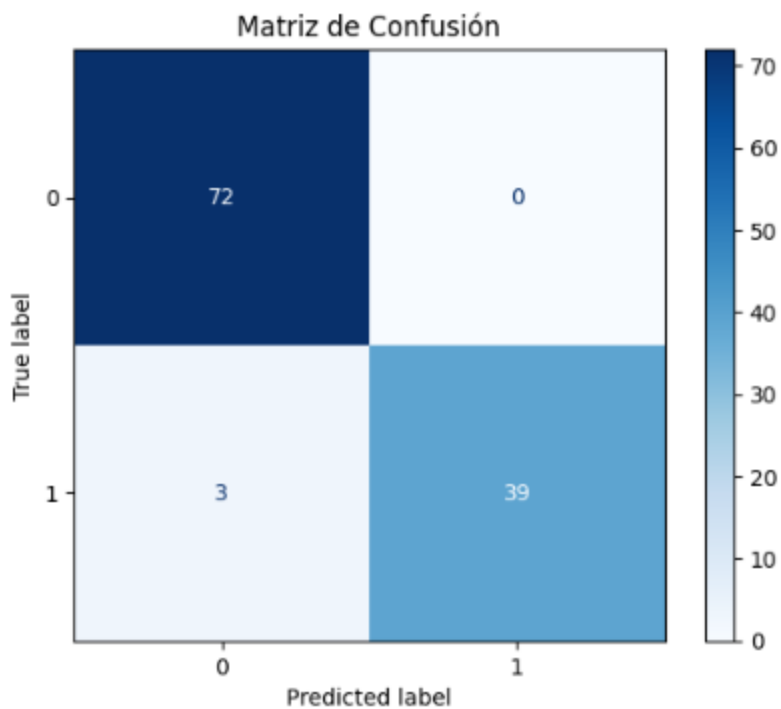
Precisión del modelo: 0.97
```

Evaluación y Optimización del Modelo

El modelo fue evaluado utilizando métricas como la precisión, matriz de confusión y reporte de clasificación. La precisión obtenida fue del 97.37%, lo que demuestra un excelente desempeño. Además, se utilizó la importancia de características para identificar las variables más relevantes en el modelo.

```
import matplotlib.pyplot as plt
import numpy as np
from sklearn.metrics import confusion_matrix, ConfusionMatrixDisplay, classification_report

# Matriz de Confusión
cm = confusion_matrix(y_test, y_pred)
disp = ConfusionMatrixDisplay(confusion_matrix=cm, display_labels=model.classes_)
disp.plot(cmap=plt.cm.Blues)
plt.title('Matriz de Confusión')
plt.show()
```



Gráfica Personalizada e Interpretación de Resultados

La gráfica de importancia de características muestra las variables más relevantes en la predicción, como el 'perímetro' y la 'textura'. Estas características proporcionan información clave para identificar si un tumor es maligno o benigno. Además, la matriz de confusión resalta el excelente desempeño del modelo, con una alta tasa de aciertos tanto para tumores benignos como malignos.

```
plt.xlabel("Características")
plt.ylabel("Importancia")
plt.tight_layout()
plt.show()
```

