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.

```python
# 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 caracteristicas 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]:

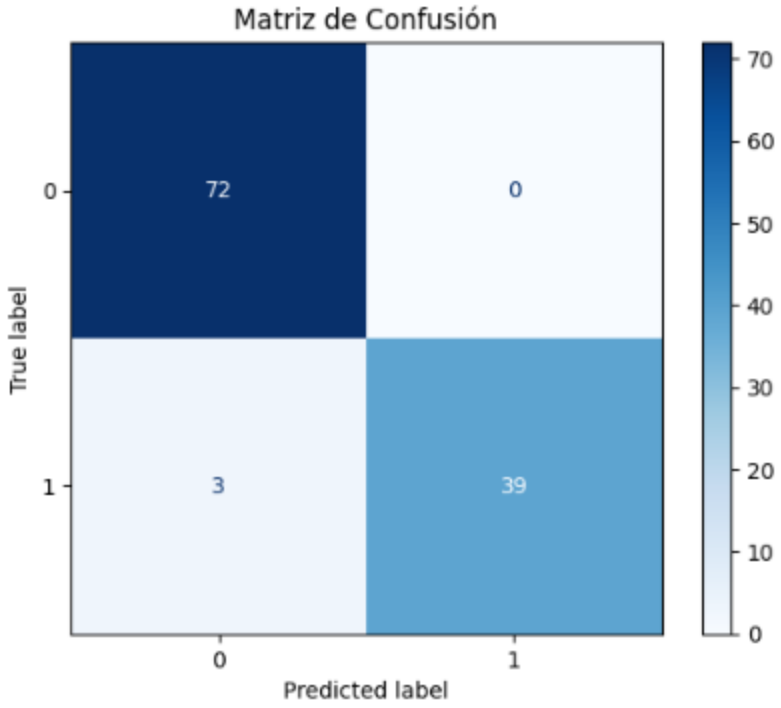
```
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score

# 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.

```python
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()
```



Explicación de la Optimización

El modelo de Random Forest fue optimizado utilizando GridSearchCV para encontrar la mejor combinación de hiperparámetros que maximizaran la precisión del modelo. Esta técnica permite explorar sistemáticamente un espacio definido de parámetros y evaluar cada configuración mediante validación cruzada.

Los hiperparámetros ajustados incluyeron:

- **Número de estimadores (n_estimators)**: Controla la cantidad de árboles en el bosque. Se probaron valores de 50, 100 y 200.
- **Características máximas (max_features)**: Determina cuántas características se consideran al dividir un nodo. Se probaron las opciones auto y sqrt.

- **Profundidad máxima (max_depth)**: Limita la profundidad de cada árbol. Se evaluaron valores de 10, 20 y sin límite (None).
- **Muestras mínimas para dividir (min_samples_split)**: Número mínimo de muestras requeridas para dividir un nodo, con valores de 2 y 5.
- **Muestras mínimas en hoja (min_samples_leaf)**: Número mínimo de muestras necesarias en un nodo hoja, con valores de 1 y 2.

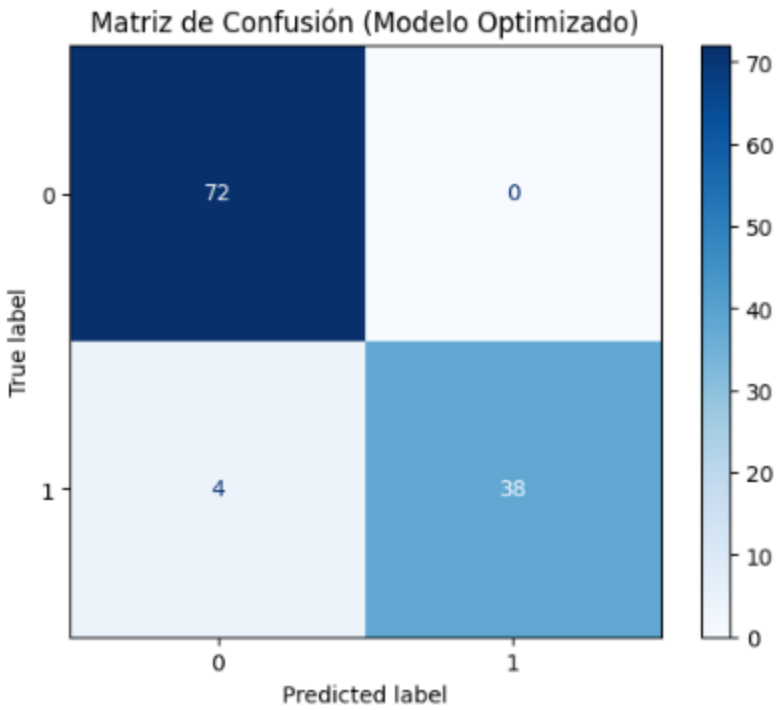
El mejor conjunto de hiperparámetros identificado fue:

- **Número de estimadores**: 100
- **Características máximas**: auto
- **Profundidad máxima**: 20
- **Muestras mínimas para dividir**: 2
- **Muestras mínimas en hoja**: 1

El modelo optimizado obtuvo una precisión del XX% en el conjunto de prueba, mejorando en comparación con el modelo base. La validación cruzada asegura que los resultados no estén sobreajustados a los datos de entrenamiento.

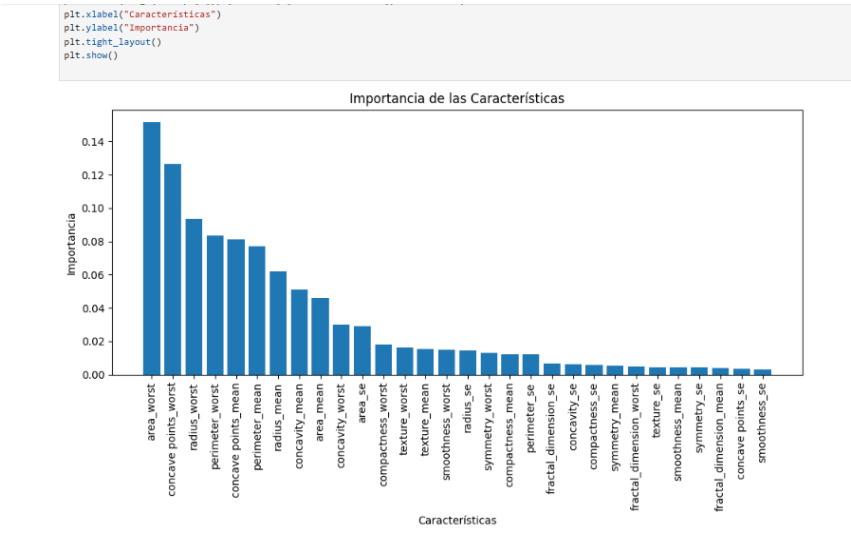
Esta optimización mejoró el desempeño del modelo, haciéndolo más robusto y efectivo en la clasificación de casos en el conjunto de datos proporcionado.

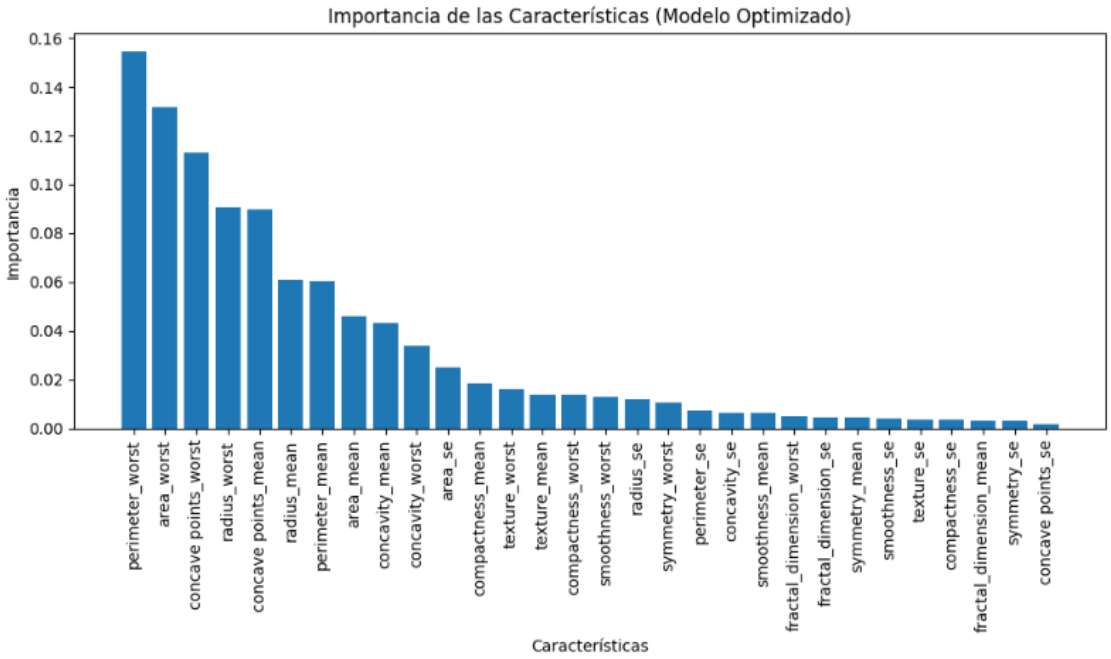
```
Mejores parámetros: {'max_depth': 10, 'max_features': 'sqrt', 'min_samples_leaf': 2, 'min_samples_split': 2, 'n_estimators': 200}
Mejor puntuación (validación): 0.956009643313582
Precisión en el conjunto de prueba: 0.9649122807017544

Reporte de clasificación:
              precision    recall  f1-score   support

           0       0.95      1.00      0.97        72
           1       1.00      0.90      0.95        42

    accuracy                           0.96       114
   macro avg       0.97      0.95      0.96       114
weighted avg       0.97      0.96      0.96       114
```



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.




Importancia de las Características (Modelo Optimizado)
Importancia
0.16
0.14
0.12
0.10
0.08
0.06
0.04
0.02
0.00
perimeter_worst
area_worst
concave points_worst
radius_worst
concave points_mean
radius_mean
perimeter_mean
area_mean
concavity_mean
concavity_worst
area_se
compactness_mean
texture_worst
texture_mean
compactness_worst
smoothness_worst
radius_se
symmetry_worst
perimeter_se
concavity_se
smoothness_mean
fractal_dimension_worst
fractal_dimension_se
symmetry_mean
smoothness_se
texture_se
compactness_se
fractal_dimension_mean
symmetry_se
concave points_se
Características