2_ML_Flow_Optuna_Bankrupt

December 1, 2024

1 Carga de Datos

[1]:
```
# Montar Google Drive
# from google.colab import drive
# drive.mount('/content/drive')

# Importar librerías necesarias
import pandas as pd
import warnings

# Ignorar warnings
warnings.filterwarnings("ignore")

# Leer el dataset
file_path = "bankrupt.csv"
df = pd.read_csv(file_path)

# Verificar los datos
df.head()
```

[1]:

	Bankrupt?	ROA(C) before interest and depreciation before interest	ROA(A) before interest and % after tax	ROA(B) before interest and depreciation after tax
0	1	0.370594	0.424389	0.405750
1	1	0.464291	0.538214	0.516730
2	1	0.426071	0.499019	
3	1	0.399844	0.451265	
4	1	0.465022	0.538432	

```
2                                          0.472295
3                                          0.457733
4                                          0.522298

    Operating Gross Margin   Realized Sales Gross Margin  \
0                 0.601457                      0.601457
1                 0.610235                      0.610235
2                 0.601450                      0.601364
3                 0.583541                      0.583541
4                 0.598783                      0.598783

    Operating Profit Rate   Pre-tax net Interest Rate  \
0                0.998969                    0.796887
1                0.998946                    0.797380
2                0.998857                    0.796403
3                0.998700                    0.796967
4                0.998973                    0.797366

    After-tax net Interest Rate   Non-industry income and expenditure/revenue  \
0                      0.808809                                       0.302646
1                      0.809301                                       0.303556
2                      0.808388                                       0.302035
3                      0.808966                                       0.303350
4                      0.809304                                       0.303475

   …   Net Income to Total Assets   Total assets to GNP price  \
0  …                     0.716845                   0.009219
1  …                     0.795297                   0.008323
2  …                     0.774670                   0.040003
3  …                     0.739555                   0.003252
4  …                     0.795016                   0.003878

    No-credit Interval   Gross Profit to Sales  \
0             0.622879                0.601453
1             0.623652                0.610237
2             0.623841                0.601449
3             0.622929                0.583538
4             0.623521                0.598782

    Net Income to Stockholder's Equity   Liability to Equity  \
0                             0.827890              0.290202
1                             0.839969              0.283846
2                             0.836774              0.290189
3                             0.834697              0.281721
4                             0.839973              0.278514

    Degree of Financial Leverage (DFL)  \
```

```
0                                0.026601
1                                0.264577
2                                0.026555
3                                0.026697
4                                0.024752

    Interest Coverage Ratio (Interest expense to EBIT)   Net Income Flag  \
0                                             0.564050                  1
1                                             0.570175                  1
2                                             0.563706                  1
3                                             0.564663                  1
4                                             0.575617                  1

    Equity to Liability
0              0.016469
1              0.020794
2              0.016474
3              0.023982
4              0.035490

[5 rows x 96 columns]
```

[2]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 6819 entries, 0 to 6818
Data columns (total 96 columns):
 #   Column                                                  Non-Null Count
Dtype
---  ------                                                  --------------
-----
 0   Bankrupt?                                               6819 non-null
int64
 1    ROA(C) before interest and depreciation before interest  6819 non-null
float64
 2    ROA(A) before interest and % after tax                  6819 non-null
float64
 3    ROA(B) before interest and depreciation after tax       6819 non-null
float64
 4    Operating Gross Margin                                  6819 non-null
float64
 5    Realized Sales Gross Margin                             6819 non-null
float64
 6    Operating Profit Rate                                   6819 non-null
float64
 7    Pre-tax net Interest Rate                               6819 non-null
float64
 8    After-tax net Interest Rate                             6819 non-null
```

```
float64
 9    Non-industry income and expenditure/revenue             6819 non-null
float64
 10   Continuous interest rate (after tax)                   6819 non-null
float64
 11   Operating Expense Rate                                 6819 non-null
float64
 12   Research and development expense rate                  6819 non-null
float64
 13   Cash flow rate                                         6819 non-null
float64
 14   Interest-bearing debt interest rate                    6819 non-null
float64
 15   Tax rate (A)                                           6819 non-null
float64
 16   Net Value Per Share (B)                                6819 non-null
float64
 17   Net Value Per Share (A)                                6819 non-null
float64
 18   Net Value Per Share (C)                                6819 non-null
float64
 19   Persistent EPS in the Last Four Seasons                6819 non-null
float64
 20   Cash Flow Per Share                                    6819 non-null
float64
 21   Revenue Per Share (Yuan ¥)                             6819 non-null
float64
 22   Operating Profit Per Share (Yuan ¥)                    6819 non-null
float64
 23   Per Share Net profit before tax (Yuan ¥)               6819 non-null
float64
 24   Realized Sales Gross Profit Growth Rate                6819 non-null
float64
 25   Operating Profit Growth Rate                           6819 non-null
float64
 26   After-tax Net Profit Growth Rate                       6819 non-null
float64
 27   Regular Net Profit Growth Rate                         6819 non-null
float64
 28   Continuous Net Profit Growth Rate                      6819 non-null
float64
 29   Total Asset Growth Rate                                6819 non-null
float64
 30   Net Value Growth Rate                                  6819 non-null
float64
 31   Total Asset Return Growth Rate Ratio                   6819 non-null
float64
 32   Cash Reinvestment %                                    6819 non-null
```

```
float64
 33   Current Ratio                                          6819 non-null
float64
 34   Quick Ratio                                            6819 non-null
float64
 35   Interest Expense Ratio                                 6819 non-null
float64
 36   Total debt/Total net worth                             6819 non-null
float64
 37   Debt ratio %                                           6819 non-null
float64
 38   Net worth/Assets                                       6819 non-null
float64
 39   Long-term fund suitability ratio (A)                   6819 non-null
float64
 40   Borrowing dependency                                   6819 non-null
float64
 41   Contingent liabilities/Net worth                       6819 non-null
float64
 42   Operating profit/Paid-in capital                       6819 non-null
float64
 43   Net profit before tax/Paid-in capital                  6819 non-null
float64
 44   Inventory and accounts receivable/Net value            6819 non-null
float64
 45   Total Asset Turnover                                   6819 non-null
float64
 46   Accounts Receivable Turnover                           6819 non-null
float64
 47   Average Collection Days                                6819 non-null
float64
 48   Inventory Turnover Rate (times)                        6819 non-null
float64
 49   Fixed Assets Turnover Frequency                        6819 non-null
float64
 50   Net Worth Turnover Rate (times)                        6819 non-null
float64
 51   Revenue per person                                     6819 non-null
float64
 52   Operating profit per person                            6819 non-null
float64
 53   Allocation rate per person                             6819 non-null
float64
 54   Working Capital to Total Assets                        6819 non-null
float64
 55   Quick Assets/Total Assets                              6819 non-null
float64
 56   Current Assets/Total Assets                            6819 non-null
```

```
float64
 57   Cash/Total Assets                                  6819 non-null
float64
 58   Quick Assets/Current Liability                     6819 non-null
float64
 59   Cash/Current Liability                             6819 non-null
float64
 60   Current Liability to Assets                        6819 non-null
float64
 61   Operating Funds to Liability                       6819 non-null
float64
 62   Inventory/Working Capital                          6819 non-null
float64
 63   Inventory/Current Liability                        6819 non-null
float64
 64   Current Liabilities/Liability                      6819 non-null
float64
 65   Working Capital/Equity                             6819 non-null
float64
 66   Current Liabilities/Equity                         6819 non-null
float64
 67   Long-term Liability to Current Assets              6819 non-null
float64
 68   Retained Earnings to Total Assets                  6819 non-null
float64
 69   Total income/Total expense                         6819 non-null
float64
 70   Total expense/Assets                               6819 non-null
float64
 71   Current Asset Turnover Rate                        6819 non-null
float64
 72   Quick Asset Turnover Rate                          6819 non-null
float64
 73   Working capitcal Turnover Rate                     6819 non-null
float64
 74   Cash Turnover Rate                                 6819 non-null
float64
 75   Cash Flow to Sales                                 6819 non-null
float64
 76   Fixed Assets to Assets                             6819 non-null
float64
 77   Current Liability to Liability                     6819 non-null
float64
 78   Current Liability to Equity                        6819 non-null
float64
 79   Equity to Long-term Liability                      6819 non-null
float64
 80   Cash Flow to Total Assets                          6819 non-null
```

```
float64
 81   Cash Flow to Liability                                  6819 non-null
float64
 82   CFO to Assets                                           6819 non-null
float64
 83   Cash Flow to Equity                                     6819 non-null
float64
 84   Current Liability to Current Assets                     6819 non-null
float64
 85   Liability-Assets Flag                                   6819 non-null
int64
 86   Net Income to Total Assets                              6819 non-null
float64
 87   Total assets to GNP price                               6819 non-null
float64
 88   No-credit Interval                                      6819 non-null
float64
 89   Gross Profit to Sales                                   6819 non-null
float64
 90   Net Income to Stockholder's Equity                      6819 non-null
float64
 91   Liability to Equity                                     6819 non-null
float64
 92   Degree of Financial Leverage (DFL)                      6819 non-null
float64
 93   Interest Coverage Ratio (Interest expense to EBIT)      6819 non-null
float64
 94   Net Income Flag                                         6819 non-null
int64
 95   Equity to Liability                                     6819 non-null
float64
dtypes: float64(93), int64(3)
memory usage: 5.0 MB
```

[3]:
```
df.columns
```

[3]:
```
Index(['Bankrupt?', ' ROA(C) before interest and depreciation before interest',
       ' ROA(A) before interest and % after tax',
       ' ROA(B) before interest and depreciation after tax',
       ' Operating Gross Margin', ' Realized Sales Gross Margin',
       ' Operating Profit Rate', ' Pre-tax net Interest Rate',
       ' After-tax net Interest Rate',
       ' Non-industry income and expenditure/revenue',
       ' Continuous interest rate (after tax)', ' Operating Expense Rate',
       ' Research and development expense rate', ' Cash flow rate',
       ' Interest-bearing debt interest rate', ' Tax rate (A)',
       ' Net Value Per Share (B)', ' Net Value Per Share (A)',
```

```
       ' Net Value Per Share (C)', ' Persistent EPS in the Last Four Seasons',
       ' Cash Flow Per Share', ' Revenue Per Share (Yuan ¥)',
       ' Operating Profit Per Share (Yuan ¥)',
       ' Per Share Net profit before tax (Yuan ¥)',
       ' Realized Sales Gross Profit Growth Rate',
       ' Operating Profit Growth Rate', ' After-tax Net Profit Growth Rate',
       ' Regular Net Profit Growth Rate', ' Continuous Net Profit Growth Rate',
       ' Total Asset Growth Rate', ' Net Value Growth Rate',
       ' Total Asset Return Growth Rate Ratio', ' Cash Reinvestment %',
       ' Current Ratio', ' Quick Ratio', ' Interest Expense Ratio',
       ' Total debt/Total net worth', ' Debt ratio %', ' Net worth/Assets',
       ' Long-term fund suitability ratio (A)', ' Borrowing dependency',
       ' Contingent liabilities/Net worth',
       ' Operating profit/Paid-in capital',
       ' Net profit before tax/Paid-in capital',
       ' Inventory and accounts receivable/Net value', ' Total Asset Turnover',
       ' Accounts Receivable Turnover', ' Average Collection Days',
       ' Inventory Turnover Rate (times)', ' Fixed Assets Turnover Frequency',
       ' Net Worth Turnover Rate (times)', ' Revenue per person',
       ' Operating profit per person', ' Allocation rate per person',
       ' Working Capital to Total Assets', ' Quick Assets/Total Assets',
       ' Current Assets/Total Assets', ' Cash/Total Assets',
       ' Quick Assets/Current Liability', ' Cash/Current Liability',
       ' Current Liability to Assets', ' Operating Funds to Liability',
       ' Inventory/Working Capital', ' Inventory/Current Liability',
       ' Current Liabilities/Liability', ' Working Capital/Equity',
       ' Current Liabilities/Equity', ' Long-term Liability to Current Assets',
       ' Retained Earnings to Total Assets', ' Total income/Total expense',
       ' Total expense/Assets', ' Current Asset Turnover Rate',
       ' Quick Asset Turnover Rate', ' Working capitcal Turnover Rate',
       ' Cash Turnover Rate', ' Cash Flow to Sales', ' Fixed Assets to Assets',
       ' Current Liability to Liability', ' Current Liability to Equity',
       ' Equity to Long-term Liability', ' Cash Flow to Total Assets',
       ' Cash Flow to Liability', ' CFO to Assets', ' Cash Flow to Equity',
       ' Current Liability to Current Assets', ' Liability-Assets Flag',
       ' Net Income to Total Assets', ' Total assets to GNP price',
       ' No-credit Interval', ' Gross Profit to Sales',
       ' Net Income to Stockholder's Equity', ' Liability to Equity',
       ' Degree of Financial Leverage (DFL)',
       ' Interest Coverage Ratio (Interest expense to EBIT)',
       ' Net Income Flag', ' Equity to Liability'],
      dtype='object')
```

[4]:
```
# Evaluar si existen valores nulos
nulos = df.isnull().sum().sum()
print(f"Total de valores nulos: {nulos}")
```

```
# Evaluar si existen duplicados
duplicados = df.duplicated().sum()
print(f"Total de filas duplicadas: {duplicados}")
```

```
Total de valores nulos: 0
Total de filas duplicadas: 0
```

2 Preparacion de datos

[5]:
```
# Separar características (X) y variable objetivo (y)
X = df.drop("Bankrupt?", axis=1)  # Eliminar la columna objetivo de las␣
 ↪características
y = df["Bankrupt?"]  # Variable objetivo
```

[6]:
```
# Convertir columnas enteras a flotantes
X = X.astype({col: 'float64' for col in X.select_dtypes('int').columns})
X.dtypes
```

[6]:
```
ROA(C) before interest and depreciation before interest    float64
ROA(A) before interest and % after tax                     float64
ROA(B) before interest and depreciation after tax          float64
Operating Gross Margin                                     float64
Realized Sales Gross Margin                                float64
                                                            …
Liability to Equity                                        float64
Degree of Financial Leverage (DFL)                         float64
Interest Coverage Ratio (Interest expense to EBIT)         float64
Net Income Flag                                            float64
Equity to Liability                                        float64
Length: 95, dtype: object
```

2.1 Dividir los Datos

[7]:
```
from sklearn.model_selection import train_test_split

# Dividir los datos en entrenamiento y prueba
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=42
)

print("Tamaño del conjunto de entrenamiento:", X_train.shape)
print("Tamaño del conjunto de prueba:", X_test.shape)
```

```
Tamaño del conjunto de entrenamiento: (5455, 95)
Tamaño del conjunto de prueba: (1364, 95)
```

3 Configurar MLFlow

[8]:
```
# Instalar MLFlow si no está ya instalado
# !pip install mlflow

# Importar MLFlow
import mlflow
import mlflow.sklearn

# Configurar el experimento en MLFlow
mlflow.set_experiment("Bankruptcy Prediction")
```

[8]:
```
<Experiment: artifact_location='file:///c:/Users/heldi/PycharmProjects/MLOps-
Taller3-EIA/mlruns/899372182312550061', creation_time=1733113178450,
experiment_id='899372182312550061', last_update_time=1733113178450,
lifecycle_stage='active', name='Bankruptcy Prediction', tags={}>
```

4 Entrenar el Modelo y Registrar Métricas

[9]:
```
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, recall_score, f1_score

# Entrenar el Modelo Inicial y Registrar Métricas
with mlflow.start_run(run_name="Random Forest Initial"):
    # Entrenar el modelo
    model_initial = RandomForestClassifier(
        max_depth=20, min_samples_split=5, n_estimators=50, random_state=42
    )
    model_initial.fit(X_train, y_train)

    # Realizar predicciones
    y_pred_initial = model_initial.predict(X_test)

    # Calcular métricas
    acc_initial = accuracy_score(y_test, y_pred_initial)
    recall_initial = recall_score(y_test, y_pred_initial)
    f1_initial = f1_score(y_test, y_pred_initial)

    # Registrar métricas en MLFlow
    mlflow.log_metric("accuracy", acc_initial)
    mlflow.log_metric("recall", recall_initial)
    mlflow.log_metric("f1_score", f1_initial)

    # Guardar el modelo en MLFlow
    mlflow.sklearn.log_model(model_initial, "random_forest_model_initial")
```

```
# Mostrar los resultados en la consola
print(f"Initial Model - Accuracy: {acc_initial:.2f}")
print(f"Initial Model - Recall: {recall_initial:.2f}")
print(f"Initial Model - F1 Score: {f1_initial:.2f}")
```

```
2024/12/01 23:21:33 WARNING mlflow.models.model: Model logged without a
signature and input example. Please set `input_example` parameter when logging
the model to auto infer the model signature.
```

```
Initial Model - Accuracy: 0.97
Initial Model - Recall: 0.20
Initial Model - F1 Score: 0.31
```

5 Optimización de Hiperparámetros con Optuna

[10]:
```
# Instalar Optuna si no está ya instalado
# !pip install optuna

import optuna
from sklearn.model_selection import cross_val_score

# Definir la función objetivo para Optuna
def objective(trial):
    # Sugerir valores para los hiperparámetros
    max_depth = trial.suggest_int("max_depth", 10, 30)
    min_samples_split = trial.suggest_int("min_samples_split", 2, 10)
    n_estimators = trial.suggest_int("n_estimators", 50, 200)

    # Modelo con los hiperparámetros sugeridos
    model = RandomForestClassifier(
        max_depth=max_depth,
        min_samples_split=min_samples_split,
        n_estimators=n_estimators,
        random_state=42
    )

    # Validación cruzada
    return cross_val_score(model, X_train, y_train, cv=3, scoring="accuracy").
 ↪mean()

# Iniciar la optimización
study = optuna.create_study(direction="maximize")
study.optimize(objective, n_trials=50)

# Imprimir los mejores hiperparámetros
print("Mejores parámetros:", study.best_params)
```

```
[I 2024-12-01 23:21:33,733] A new study created in memory with name: no-
```

```
name-746038be-9f70-4606-95ce-3079357cda30
[I 2024-12-01 23:21:38,744] Trial 0 finished with value: 0.9704858244666322 and
parameters: {'max_depth': 19, 'min_samples_split': 8, 'n_estimators': 115}. Best
is trial 0 with value: 0.9704858244666322.
[I 2024-12-01 23:21:46,323] Trial 1 finished with value: 0.9703023720000733 and
parameters: {'max_depth': 26, 'min_samples_split': 6, 'n_estimators': 174}. Best
is trial 0 with value: 0.9704858244666322.
[I 2024-12-01 23:21:53,605] Trial 2 finished with value: 0.9708525278036325 and
parameters: {'max_depth': 17, 'min_samples_split': 5, 'n_estimators': 164}. Best
is trial 2 with value: 0.9708525278036325.
[I 2024-12-01 23:22:00,168] Trial 3 finished with value: 0.9712191303425742 and
parameters: {'max_depth': 10, 'min_samples_split': 8, 'n_estimators': 156}. Best
is trial 3 with value: 0.9712191303425742.
[I 2024-12-01 23:22:04,334] Trial 4 finished with value: 0.9706691761351323 and
parameters: {'max_depth': 25, 'min_samples_split': 3, 'n_estimators': 96}. Best
is trial 3 with value: 0.9712191303425742.
[I 2024-12-01 23:22:10,874] Trial 5 finished with value: 0.9706691761351323 and
parameters: {'max_depth': 13, 'min_samples_split': 8, 'n_estimators': 151}. Best
is trial 3 with value: 0.9712191303425742.
[I 2024-12-01 23:22:16,016] Trial 6 finished with value: 0.9708521246113984 and
parameters: {'max_depth': 14, 'min_samples_split': 9, 'n_estimators': 118}. Best
is trial 3 with value: 0.9712191303425742.
[I 2024-12-01 23:22:18,766] Trial 7 finished with value: 0.9701191211296317 and
parameters: {'max_depth': 16, 'min_samples_split': 9, 'n_estimators': 61}. Best
is trial 3 with value: 0.9712191303425742.
[I 2024-12-01 23:22:27,041] Trial 8 finished with value: 0.9701191211296317 and
parameters: {'max_depth': 18, 'min_samples_split': 7, 'n_estimators': 190}. Best
is trial 3 with value: 0.9712191303425742.
[I 2024-12-01 23:22:31,318] Trial 9 finished with value: 0.970852628601691 and
parameters: {'max_depth': 18, 'min_samples_split': 2, 'n_estimators': 97}. Best
is trial 3 with value: 0.9712191303425742.
[I 2024-12-01 23:22:37,755] Trial 10 finished with value: 0.9697525185906898 and
parameters: {'max_depth': 30, 'min_samples_split': 10, 'n_estimators': 148}.
Best is trial 3 with value: 0.9712191303425742.
[I 2024-12-01 23:22:40,954] Trial 11 finished with value: 0.96975251859069 and
parameters: {'max_depth': 10, 'min_samples_split': 2, 'n_estimators': 74}. Best
is trial 3 with value: 0.9712191303425742.
[I 2024-12-01 23:22:45,092] Trial 12 finished with value: 0.9706691761351323 and
parameters: {'max_depth': 22, 'min_samples_split': 4, 'n_estimators': 93}. Best
is trial 3 with value: 0.9712191303425742.
[I 2024-12-01 23:22:50,799] Trial 13 finished with value: 0.9708526286016911 and
parameters: {'max_depth': 10, 'min_samples_split': 6, 'n_estimators': 133}. Best
is trial 3 with value: 0.9712191303425742.
[I 2024-12-01 23:22:56,563] Trial 14 finished with value: 0.9706692769331907 and
parameters: {'max_depth': 10, 'min_samples_split': 6, 'n_estimators': 136}. Best
is trial 3 with value: 0.9712191303425742.
[I 2024-12-01 23:23:05,317] Trial 15 finished with value: 0.970852427005574 and
parameters: {'max_depth': 13, 'min_samples_split': 7, 'n_estimators': 200}. Best
```

```
is trial 3 with value: 0.9712191303425742.
[I 2024-12-01 23:23:10,932] Trial 16 finished with value: 0.9703028759903661 and
parameters: {'max_depth': 10, 'min_samples_split': 5, 'n_estimators': 133}. Best
is trial 3 with value: 0.9712191303425742.
[I 2024-12-01 23:23:17,895] Trial 17 finished with value: 0.9708525278036325 and
parameters: {'max_depth': 14, 'min_samples_split': 7, 'n_estimators': 161}. Best
is trial 3 with value: 0.9712191303425742.
[I 2024-12-01 23:23:25,693] Trial 18 finished with value: 0.9701190203315733 and
parameters: {'max_depth': 22, 'min_samples_split': 10, 'n_estimators': 183}.
Best is trial 3 with value: 0.9712191303425742.
[I 2024-12-01 23:23:31,914] Trial 19 finished with value: 0.9701192219276904 and
parameters: {'max_depth': 12, 'min_samples_split': 5, 'n_estimators': 143}. Best
is trial 3 with value: 0.9712191303425742.
[I 2024-12-01 23:23:37,112] Trial 20 finished with value: 0.9703025735961904 and
parameters: {'max_depth': 15, 'min_samples_split': 8, 'n_estimators': 120}. Best
is trial 3 with value: 0.9712191303425742.
[I 2024-12-01 23:23:41,573] Trial 21 finished with value: 0.9704860260627491 and
parameters: {'max_depth': 12, 'min_samples_split': 2, 'n_estimators': 102}. Best
is trial 3 with value: 0.9712191303425742.
[I 2024-12-01 23:23:45,004] Trial 22 finished with value: 0.9704858244666322 and
parameters: {'max_depth': 21, 'min_samples_split': 3, 'n_estimators': 80}. Best
is trial 3 with value: 0.9712191303425742.
[I 2024-12-01 23:23:49,791] Trial 23 finished with value: 0.9704859252646907 and
parameters: {'max_depth': 11, 'min_samples_split': 4, 'n_estimators': 110}. Best
is trial 3 with value: 0.9712191303425742.
[I 2024-12-01 23:23:52,131] Trial 24 finished with value: 0.9699357694611316 and
parameters: {'max_depth': 24, 'min_samples_split': 6, 'n_estimators': 52}. Best
is trial 3 with value: 0.9712191303425742.
[I 2024-12-01 23:23:57,908] Trial 25 finished with value: 0.971035577077957 and
parameters: {'max_depth': 16, 'min_samples_split': 9, 'n_estimators': 132}. Best
is trial 3 with value: 0.9712191303425742.
[I 2024-12-01 23:24:03,546] Trial 26 finished with value: 0.971035577077957 and
parameters: {'max_depth': 15, 'min_samples_split': 9, 'n_estimators': 129}. Best
is trial 3 with value: 0.9712191303425742.
[I 2024-12-01 23:24:10,741] Trial 27 finished with value: 0.9704858244666322 and
parameters: {'max_depth': 16, 'min_samples_split': 9, 'n_estimators': 166}. Best
is trial 3 with value: 0.9712191303425742.
[I 2024-12-01 23:24:16,241] Trial 28 finished with value: 0.96975251859069 and
parameters: {'max_depth': 15, 'min_samples_split': 10, 'n_estimators': 128}.
Best is trial 3 with value: 0.9712191303425742.
[I 2024-12-01 23:24:22,942] Trial 29 finished with value: 0.9704857236685736 and
parameters: {'max_depth': 19, 'min_samples_split': 9, 'n_estimators': 154}. Best
is trial 3 with value: 0.9712191303425742.
[I 2024-12-01 23:24:29,200] Trial 30 finished with value: 0.9710357786740742 and
parameters: {'max_depth': 20, 'min_samples_split': 8, 'n_estimators': 143}. Best
is trial 3 with value: 0.9712191303425742.
[I 2024-12-01 23:24:35,474] Trial 31 finished with value: 0.9710357786740742 and
parameters: {'max_depth': 20, 'min_samples_split': 8, 'n_estimators': 143}. Best
```

```
is trial 3 with value: 0.9712191303425742.
[I 2024-12-01 23:24:41,590] Trial 32 finished with value: 0.9708524270055738 and
parameters: {'max_depth': 20, 'min_samples_split': 8, 'n_estimators': 140}. Best
is trial 3 with value: 0.9712191303425742.
[I 2024-12-01 23:24:49,186] Trial 33 finished with value: 0.9706689745390152 and
parameters: {'max_depth': 23, 'min_samples_split': 7, 'n_estimators': 174}. Best
is trial 3 with value: 0.9712191303425742.
[I 2024-12-01 23:24:56,169] Trial 34 finished with value: 0.9706691761351323 and
parameters: {'max_depth': 20, 'min_samples_split': 8, 'n_estimators': 160}. Best
is trial 3 with value: 0.9712191303425742.
[I 2024-12-01 23:25:03,644] Trial 35 finished with value: 0.9704857236685736 and
parameters: {'max_depth': 27, 'min_samples_split': 8, 'n_estimators': 172}. Best
is trial 3 with value: 0.9712191303425742.
[I 2024-12-01 23:25:10,137] Trial 36 finished with value: 0.9704857236685736 and
parameters: {'max_depth': 18, 'min_samples_split': 7, 'n_estimators': 150}. Best
is trial 3 with value: 0.9712191303425742.
[I 2024-12-01 23:25:15,050] Trial 37 finished with value: 0.9712188279483988 and
parameters: {'max_depth': 26, 'min_samples_split': 9, 'n_estimators': 112}. Best
is trial 3 with value: 0.9712191303425742.
[I 2024-12-01 23:25:20,176] Trial 38 finished with value: 0.9703025735961904 and
parameters: {'max_depth': 28, 'min_samples_split': 8, 'n_estimators': 119}. Best
is trial 3 with value: 0.9712191303425742.
[I 2024-12-01 23:25:24,970] Trial 39 finished with value: 0.9706690753370738 and
parameters: {'max_depth': 26, 'min_samples_split': 8, 'n_estimators': 110}. Best
is trial 3 with value: 0.9712191303425742.
[I 2024-12-01 23:25:31,165] Trial 40 finished with value: 0.9699357694611316 and
parameters: {'max_depth': 25, 'min_samples_split': 10, 'n_estimators': 143}.
Best is trial 3 with value: 0.9712191303425742.
[I 2024-12-01 23:25:37,886] Trial 41 finished with value: 0.9704857236685736 and
parameters: {'max_depth': 17, 'min_samples_split': 9, 'n_estimators': 154}. Best
is trial 3 with value: 0.9712191303425742.
[I 2024-12-01 23:25:43,200] Trial 42 finished with value: 0.9712188279483988 and
parameters: {'max_depth': 21, 'min_samples_split': 9, 'n_estimators': 123}. Best
is trial 3 with value: 0.9712191303425742.
[I 2024-12-01 23:25:47,977] Trial 43 finished with value: 0.9710354762798984 and
parameters: {'max_depth': 30, 'min_samples_split': 9, 'n_estimators': 111}. Best
is trial 3 with value: 0.9712191303425742.
[I 2024-12-01 23:25:53,272] Trial 44 finished with value: 0.9699358702591901 and
parameters: {'max_depth': 23, 'min_samples_split': 10, 'n_estimators': 124}.
Best is trial 3 with value: 0.9712191303425742.
[I 2024-12-01 23:25:57,692] Trial 45 finished with value: 0.9699357694611317 and
parameters: {'max_depth': 29, 'min_samples_split': 7, 'n_estimators': 103}. Best
is trial 3 with value: 0.9712191303425742.
[I 2024-12-01 23:26:03,868] Trial 46 finished with value: 0.9704855220724564 and
parameters: {'max_depth': 21, 'min_samples_split': 9, 'n_estimators': 145}. Best
is trial 3 with value: 0.9712191303425742.
[I 2024-12-01 23:26:09,160] Trial 47 finished with value: 0.9703025735961904 and
parameters: {'max_depth': 19, 'min_samples_split': 8, 'n_estimators': 126}. Best
```

```
is trial 3 with value: 0.9712191303425742.
[I 2024-12-01 23:26:12,914] Trial 48 finished with value: 0.9701193227257487 and
parameters: {'max_depth': 25, 'min_samples_split': 10, 'n_estimators': 90}. Best
is trial 3 with value: 0.9712191303425742.
[I 2024-12-01 23:26:18,693] Trial 49 finished with value: 0.9704858244666322 and
parameters: {'max_depth': 21, 'min_samples_split': 7, 'n_estimators': 137}. Best
is trial 3 with value: 0.9712191303425742.

Mejores parámetros: {'max_depth': 10, 'min_samples_split': 8, 'n_estimators':
156}
```

6 Entrenar el Modelo Optimizado y Registrar Métricas

[11]:
```python
# Entrenar el Modelo Optimizado y Registrar Métricas
with mlflow.start_run(run_name="Random Forest Optimized"):
    # Entrenar el modelo con los mejores hiperparámetros
    best_params = study.best_params
    model_optimized = RandomForestClassifier(
        max_depth=best_params["max_depth"],
        min_samples_split=best_params["min_samples_split"],
        n_estimators=best_params["n_estimators"],
        random_state=42,
    )
    model_optimized.fit(X_train, y_train)

    # Realizar predicciones
    y_pred_optimized = model_optimized.predict(X_test)

    # Calcular métricas
    acc_optimized = accuracy_score(y_test, y_pred_optimized)
    recall_optimized = recall_score(y_test, y_pred_optimized)
    f1_optimized = f1_score(y_test, y_pred_optimized)

    # Registrar métricas en MLFlow
    mlflow.log_metric("accuracy", acc_optimized)
    mlflow.log_metric("recall", recall_optimized)
    mlflow.log_metric("f1_score", f1_optimized)

    # Guardar el modelo en MLFlow
    mlflow.sklearn.log_model(model_optimized, "random_forest_model_optimized")

# Mostrar los resultados en la consola
print(f"Optimized Model - Accuracy: {acc_optimized:.2f}")
print(f"Optimized Model - Recall: {recall_optimized:.2f}")
print(f"Optimized Model - F1 Score: {f1_optimized:.2f}")
```

```
2024/12/01 23:26:23 WARNING mlflow.models.model: Model logged without a
signature and input example. Please set `input_example` parameter when logging
```

```
the model to auto infer the model signature.

Optimized Model - Accuracy: 0.97
Optimized Model - Recall: 0.14
Optimized Model - F1 Score: 0.24
```

7 Iniciar el servidor de MLFlow UI

[ ]:
```
# Ejecutar en la terminal el siguiente comando: mlflow ui
!mlflow ui
```

```
^C
```

7.1 Despliegue del ML Flow UI en Colab

[ ]:
```
# Instalar ngrok si no está instalado
!pip install pyngrok
```

[ ]:
```
!ngrok authtoken 2pcb0NRrykNCTdohIids2WMqu5L_5ML23hn9GHoN9xSYYEn8s
```

[ ]:
```
from pyngrok import ngrok

# Exponer el puerto 5000 con un túnel HTTP
mlflow_ui = ngrok.connect(5000, "http")
print(f"MLFlow UI disponible en: {mlflow_ui}")
!mlflow ui --port 5000
```

```
MLFlow UI disponible en: NgrokTunnel: "https://0574-35-245-56-243.ngrok-
free.app" -> "http://localhost:5000"
[2024-12-01 16:04:06 +0000] [12587] [INFO] Starting gunicorn 23.0.0
[2024-12-01 16:04:06 +0000] [12587] [INFO] Listening at: http://127.0.0.1:5000
(12587)
[2024-12-01 16:04:06 +0000] [12587] [INFO] Using worker: sync
[2024-12-01 16:04:06 +0000] [12588] [INFO] Booting worker with pid: 12588
[2024-12-01 16:04:06 +0000] [12589] [INFO] Booting worker with pid: 12589
[2024-12-01 16:04:06 +0000] [12594] [INFO] Booting worker with pid: 12594
[2024-12-01 16:04:06 +0000] [12595] [INFO] Booting worker with pid: 12595
```