



**UANL**

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN

**UNIVERSIDAD AUTONOMA DE NUEVO LEON  
FACULTAD DE CIENCIAS FISICO MATEMATICO**



**FCFM**

FACULTAD DE CIENCIAS FÍSICO MATEMÁTICAS

## **Inteligencia Artificial**

### **Act 13: Programando Random Forest en Python**

**Docente:** Luis Ángel Gutiérrez Rodríguez

**Alumno:** Jhoana Esmeralda Escobar Barron. 1950748.

**Gpo:**031

# Informe sobre Análisis de Fraude en Tarjetas de Crédito utilizando Random Forest

March 31, 2025

## 1 Introducción

El algoritmo Random Forest es un clasificador supervisado basado en un conjunto de árboles de decisión. Este modelo pertenece a la categoría de métodos de ensamble, donde varios clasificadores (en este caso, árboles de decisión) trabajan juntos para tomar decisiones más precisas que un único clasificador. En particular, Random Forest es ampliamente utilizado debido a su capacidad para manejar grandes volúmenes de datos y su eficacia para prevenir el sobreajuste (overfitting), especialmente en escenarios de clasificación.

En este informe, se utilizará Random Forest para clasificar transacciones bancarias como normales o fraudulentas, basándose en el conjunto de datos de transacciones de tarjetas de crédito. Este es un problema clásico de detección de fraudes, donde el modelo debe ser capaz de identificar correctamente las transacciones fraudulentas (Clase 1) frente a las normales (Clase 0).

## 2 Metodología

El proceso seguido en esta actividad incluyó los siguientes pasos:

1. **Carga y preprocesamiento de datos:** Se cargó el conjunto de datos de transacciones de tarjetas de crédito, que contiene información sobre 31 variables numéricas y una variable de clase ("Class") que indica si la transacción es normal (0) o fraudulenta (1). A continuación, se separaron las transacciones normales y fraudulentas.
2. **División de datos:** Se dividió el conjunto de datos en conjuntos de entrenamiento (70%) y prueba (30%), utilizando la función `train_test_split` de la librería `sklearn`.
3. **Modelo Random Forest:** Se entrenó un modelo Random Forest con 100 árboles de decisión, utilizando los datos de entrenamiento. La función `RandomForestClassifier` de la librería `sklearn.ensemble` fue utilizada para crear el modelo.

4. **Evaluación del modelo:** Una vez entrenado el modelo, se hizo una predicción con el conjunto de prueba. Los resultados se evaluaron utilizando una matriz de confusión y un reporte de clasificación.

El código utilizado para el preprocesamiento y entrenamiento del modelo es el siguiente:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import confusion_matrix, classification_report

# Cargar el conjunto de datos
df = pd.read_csv("creditcard.csv")

# Divisi n de los datos
y = df['Class']
X = df.drop('Class', axis=1)
X_train, X_test, y_train, y_test = train_test_split(X, y, train_size=0.7)

# Crear el modelo Random Forest
model = RandomForestClassifier(n_estimators=100, bootstrap=True, max_features='s
model.fit(X_train, y_train)

# Realizar predicciones
pred_y = model.predict(X_test)

# Funci n para mostrar los resultados
def mostrar_resultados(y_test, pred_y):
    conf_matrix = confusion_matrix(y_test, pred_y)
    plt.figure(figsize=(8, 8))
    sns.heatmap(conf_matrix, annot=True, fmt="d", cmap="Blues", xticklabels=["No
    plt.title("Matriz de Confusi n")
    plt.ylabel('Clase Real')
    plt.xlabel('Clase Predicha')
    plt.show()
    print(classification_report(y_test, pred_y))

# Mostrar los resultados
mostrar_resultados(y_test, pred_y)
```

### 3 Resultados

Tras entrenar el modelo Random Forest y realizar la predicción con el conjunto de prueba, se presentó una matriz de confusión y el reporte de clasificación. La matriz de confusión muestra cuántas transacciones fueron correctamente clasificadas como normales o fraudulentas, así como cuántas fueron clasificadas erróneamente. El reporte de clasificación incluye métricas como precisión, recall, F1-score y la exactitud (accuracy), que son cruciales para evaluar el rendimiento del modelo, especialmente en el caso de clases desbalanceadas como en la detección de fraudes.

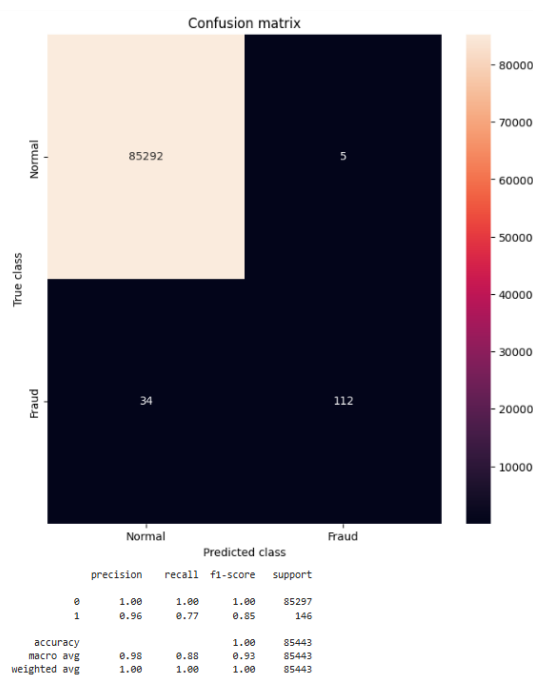


Figure 1: Matriz de Confusión del Modelo Random Forest.

El reporte de clasificación para el modelo muestra la capacidad del modelo para detectar correctamente las transacciones fraudulentas, además de su habilidad para evitar falsos positivos.

### 4 Conclusión

El modelo Random Forest mostró buenos resultados en la clasificación de las transacciones de tarjetas de crédito. La capacidad del modelo para manejar grandes volúmenes de datos y la robustez contra el sobreajuste fueron factores clave para su rendimiento. Sin embargo, se debe considerar la utilización de

técnicas de balanceo de clases, como el sobremuestreo o submuestreo, para mejorar la detección de fraudes, ya que las clases en el conjunto de datos están desbalanceadas.

En futuras actividades, sería interesante explorar la optimización de los hiperparámetros del modelo y evaluar el rendimiento utilizando otras métricas, como la curva ROC o el área bajo la curva (AUC), para obtener una evaluación más detallada del modelo.