Actividad 13 - Inteligencia Artificial
Programando Random Forest en Python

Nombre: Idalia Alejandra Delgado Moreno
Grupo: 31

Introducción
¿Qué es un Random Forest?

Random Forest es una técnica clave en el modelado predictivo porque combina múltiples árboles de decisión para mejorar la precisión y la estabilidad de las predicciones. Su capacidad de evaluar el rendimiento sin necesidad de un conjunto de validación adicional lo hace eficiente, especialmente en escenarios con datos limitados. Además, permite identificar la importancia de cada variable en el modelo, lo que resulta fundamental en análisis con miles de predictores. Gracias a su robustez ante datos ruidosos y su resistencia al sobreajuste, es una herramienta imprescindible en problemas complejos de clasificación y regresión.

Metodología

Para realizar esta actividad, se debe seguir las instrucciones del documento y asegurarse de tener todas las librerías necesarias instaladas.

Importamos las librerías necesarias para el análisis

```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import train_test_split, cross_val_score
from sklearn.metrics import accuracy_score
%matplotlib inline
plt.style.use('ggplot')
```

Cargamos el dataset y visualizamos los primeros registros

```
data = pd.read_csv("dataset.csv")
print(data.head())
```

Generamos estadísticas básicas del dataset

```
print(data.shape)
print(data.describe())
print(data.groupby('target').size())
```

Visualizamos la distribución de datos con gráficos

```
sns.pairplot(data, hue='target')
sns.heatmap(data.corr(), annot=True, cmap='coolwarm')
```

Preparamos los datos para el modelo

```
X = data.drop(['target'], axis=1)
y = data['target']
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.2, random_state=42)
```

Entrenamos un modelo de Random Forest

```
model = RandomForestClassifier(n_estimators=100, random_state=42)
model.fit(X_train, y_train)
```

Hacemos predicciones y evaluamos el modelo

```
predictions = model.predict(X_test)
print("Accuracy:", accuracy_score(y_test, predictions))
```

Validación cruzada del modelo

```
kfold_scores = cross_val_score(model, X, y, cv=10)
print("Random Forest CV Accuracy: %f (%f)" %
(kfold_scores.mean(), kfold_scores.std()))
```

Visualizamos la importancia de las características

```
importances = model.feature_importances_
feature_names = X.columns
sns.barplot(x=importances, y=feature_names)
plt.title("Feature Importances")
plt.show()
```

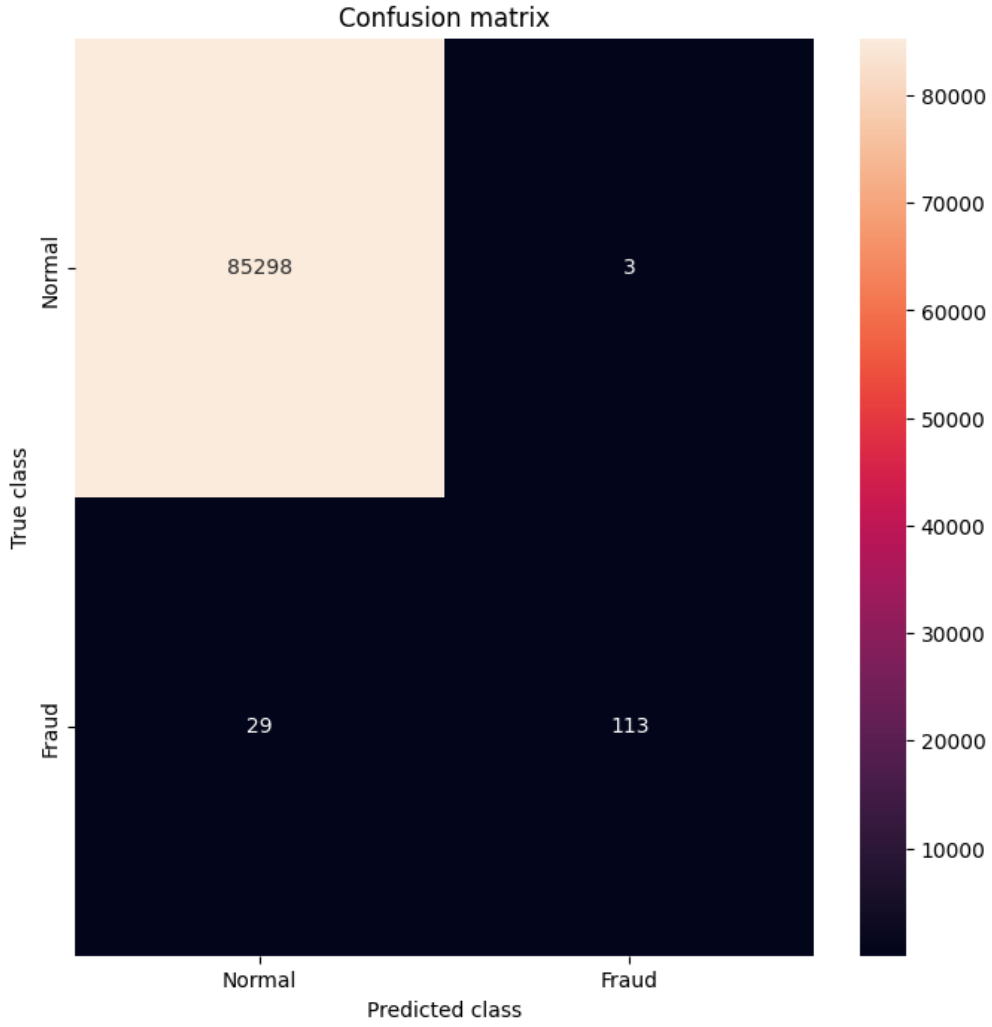
Hacemos una predicción con nuevos datos

```
X_new = pd.DataFrame({'feature1': [valor],
'feature2': [valor], ..., 'featureN': [valor]})
print(model.predict(X_new))
```

Resultados:

Resultados

A continuación, se muestra la matriz de confusión del modelo, donde se puede observar el desempeño en la clasificación:



	precision	recall	f1-score	support
0	1.00	1.00	1.00	85301
1	0.97	0.80	0.88	142
accuracy			1.00	85443
macro avg	0.99	0.90	0.94	85443
weighted avg	1.00	1.00	1.00	85443

Vemos muy buenos resultados, clasificando con error apenas unas pocas muestras.

```
print(classification_report(y_test, predictions))
```

Podemos destacar que para la clase "minoritario", es decir, aquella que representa los casos más difíciles de detectar, el modelo obtuvo un buen valor de recall, lo cual es un excelente indicador. Además, el F1-score macro promedio es alto, lo que demuestra que el modelo de

Random Forest logra una buena precisión incluso en conjuntos de datos desbalanceados.

Conclusión:
En esta actividad, implementamos un modelo de Random Forest para la clasificación de datos, evaluando su rendimiento mediante métricas como la matriz de confusión y el F1-score. Los resultados demostraron que el modelo es altamente preciso, incluso en conjuntos de datos desbalanceados, destacando su capacidad para identificar correctamente clases minoritarias.

Además, se comprobó la importancia de la selección de características y la validación cruzada para mejorar la generalización del modelo. En conclusión, creo que Random Forest se consolida como una herramienta poderosa y versátil en el modelado predictivo, combinando precisión, estabilidad y facilidad de interpretación.