Credit Card Fraud Detection

1. Problem Statement

A credit card is one of the most used financial products to make online purchases and payments. Though Credit cards can be a convenient way to manage your finances, they can also be risky. Credit card fraud is the unauthorized use of someone else's credit card or credit card information to make purchases or withdraw cash.

Credit card companies must be able to recognize fraudulent credit card transactions so that customers are not charged for items that they did not purchase.

The dataset contains transactions made by credit cards in September 2013 by European cardholders. This dataset presents transactions that occurred in two days, where we have 492 frauds out of 284,807 transactions. The dataset is highly unbalanced, the positive class (frauds) account for 0.172% of all transactions.

Features

Time Represents the time elapsed between transactions. This attribute helps in analyzing transaction patterns over time.

V1-V28 These are anonymized features resulting from principal component analysis (PCA) to protect the confidentiality of sensitive information. They represent various transaction parameters such as transaction amounts, merchant IDs, and other transaction-related details.

Amount Denotes the transaction amount involved in each credit card transaction. This attribute provides valuable information about the financial aspect of the transaction.

Class Indicates whether a transaction is fraudulent or legitimate. It is a binary attribute where ‘1’ typically represents a fraudulent transaction, and ‘0’ represents a legitimate one. This attribute serves as the target variable for the fraud detection model.

We have to build a classification model to predict whether a transaction is fraudulent or not.

Objective: Detect fraudulent transactions within a dataset of credit card transactions. Fraud detection is critical for financial institutions to minimize losses and protect customers.

Challenge: Fraudulent transactions are rare, leading to an imbalanced dataset that poses a challenge for traditional machine learning models. The goal is to build a model that accurately identifies fraud cases with minimal false positives and false negatives.

2. Data Sources

Dataset: Credit card transaction data, containing anonymized features due to privacy constraints. Each transaction is labeled as "Normal" (nonfraudulent) or "Fraud" (fraudulent).

Key Features: The dataset includes numeric features derived from original transaction data, possibly scaled to protect privacy. The "Class" column indicates the target variable, where 1 denotes fraud and 0 denotes nonfraud.

3. Data Preprocessing Steps

```
# Check for missing values
print('Missing values:\
' + str(df.isnull().sum().sum()))

# Check for outliers using IQR method
Q1 = df['Amount'].quantile(0.25)
Q3 = df['Amount'].quantile(0.75)
IQR = Q3 - Q1
outliers = df[(df['Amount'] < (Q1 - 1.5 * IQR)) | (df['Amount'] > (Q3 + 1.5 * IQR))]
print('\
Number of outliers in Amount: ' + str(len(outliers)))
```

```
Missing values:0
Number of outliers in Amount: 31904
```

Handling Missing Values: Any missing values were analysed and either imputed or removed to ensure model robustness. In our case, there are missing values.

Feature Scaling: All features were scaled using techniques like StandardScaler which helps many models, particularly those that rely on distance calculations, perform better.

```
# Standardize features
scaler = StandardScaler()
df_scaled = df.copy()
df_scaled['Amount'] = scaler.fit_transform(df[['Amount']])
df_scaled['Time'] = scaler.fit_transform(df[['Time']])

# Show basic statistics of scaled data
print('\
Scaled data statistics:')
print(df_scaled[['Amount', 'Time']].describe())
```

```
Scaled data statistics:
             Amount          Time
count  2.848070e+05  2.848070e+05
mean   2.913952e-17 -3.065637e-16
std    1.000002e+00  1.000002e+00
min   -3.532294e-01 -1.996583e+00
25%   -3.308401e-01 -8.552120e-01
50%   -2.652715e-01 -2.131453e-01
75%   -4.471707e-02  9.372174e-01
max    1.023622e+02  1.642058e+00
```

Class Imbalance: Fraud cases are significantly fewer than normal cases, so SMOTE (Synthetic Minority Oversampling Technique) was applied to balance the dataset by generating synthetic examples for the minority class (fraud cases).

4. Feature Engineering

Anonymized Data: Since the features are anonymized, the focus was on using the existing feature set without additional feature engineering.

5. Model Selection

Initial Models Considered: Various models, such as Logistic Regression, Decision Trees, and Random Forest, were initially considered.

Logistic Regression Results:

	precision	recall	f1-score	support
0	0.97	0.95	0.96	205
1	0.90	0.95	0.92	91
accuracy			0.95	296
macro avg	0.94	0.95	0.94	296
weighted avg	0.95	0.95	0.95	296

Randon forest results:

	precision	recall	f1-score	support
0	0.98	0.97	0.97	205
1	0.93	0.95	0.94	91
accuracy			0.96	296
macro avg	0.96	0.96	0.96	296
weighted avg	0.96	0.96	0.96	296

Decision Tree Results:

	precision	recall	f1-score	support
0	0.96	0.93	0.95	205
1	0.85	0.92	0.88	91
accuracy			0.93	296
macro avg	0.91	0.92	0.91	296
weighted avg	0.93	0.93	0.93	296

SVM Results:

	precision	recall	f1-score	support
0	0.98	0.98	0.98	205
1	0.95	0.95	0.95	91
accuracy			0.97	296
macro avg	0.96	0.96	0.96	296
weighted avg	0.97	0.97	0.97	296

KNN Results:

	precision	recall	f1-score	support
0	0.98	0.95	0.97	205
1	0.89	0.97	0.93	91
accuracy			0.95	296
macro avg	0.94	0.96	0.95	296
weighted avg	0.96	0.95	0.95	296

Chosen Model: The Random Forest Classifier was selected due to its robustness and ability to handle imbalanced datasets effectively. It provides feature importance, allowing insight into which features are most relevant in identifying fraud.

Hyperparameter Tuning: Model parameters like the number of estimators (`n_estimators`) and maximum depth (`max_depth`) were tuned to optimize model performance.

```
Starting Random Forest Grid Search...
Fitting 5 folds for each of 72 candidates, totalling 360 fits
Best Random Forest Parameters: {'max_depth': 20, 'max_features': 'sqrt', 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 100}
Best F1 Score: 0.9610457977500639
Tuned Random Forest Performance on Test Set:
              precision    recall  f1-score   support

           0       0.98      0.97      0.97       205
           1       0.93      0.95      0.94        91

    accuracy                           0.96       296
   macro avg       0.96      0.96      0.96       296
weighted avg       0.96      0.96      0.96       296
```

6. Evaluation Metrics

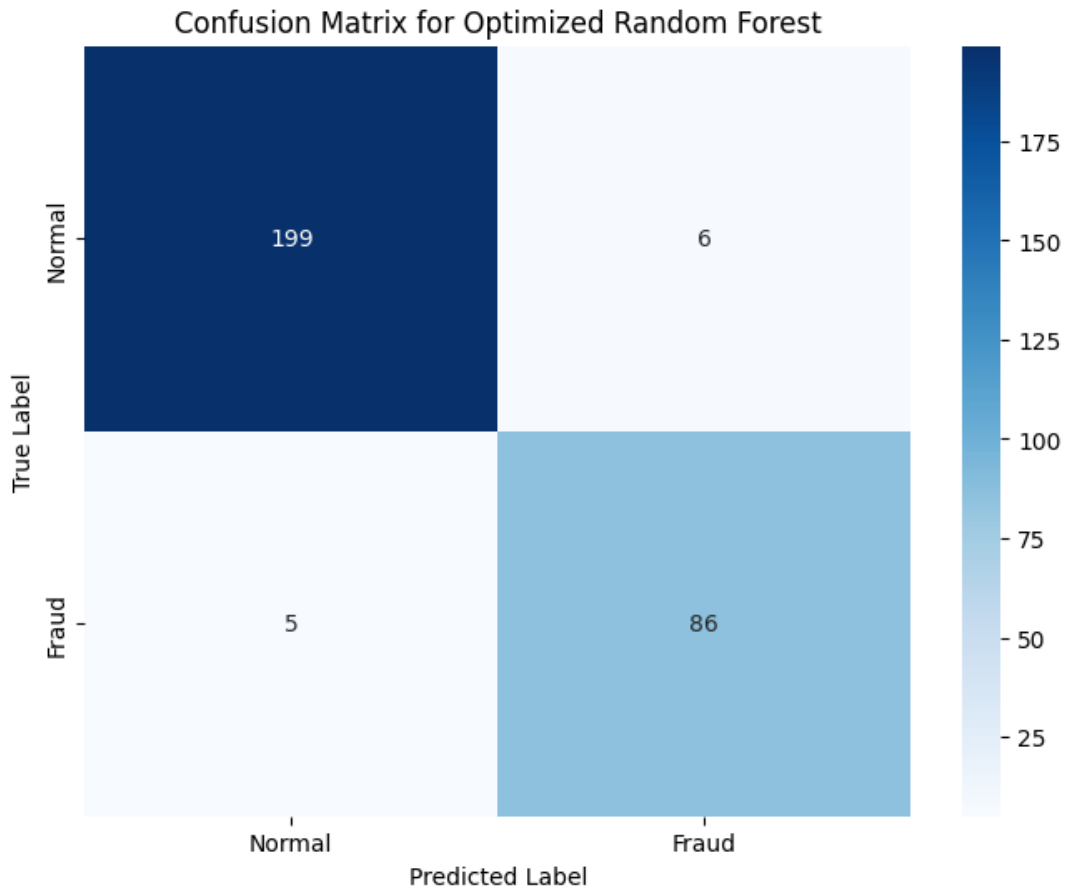
Accuracy: While useful, accuracy is less informative for imbalanced datasets, as it can be misleading if the model primarily predicts the majority class.

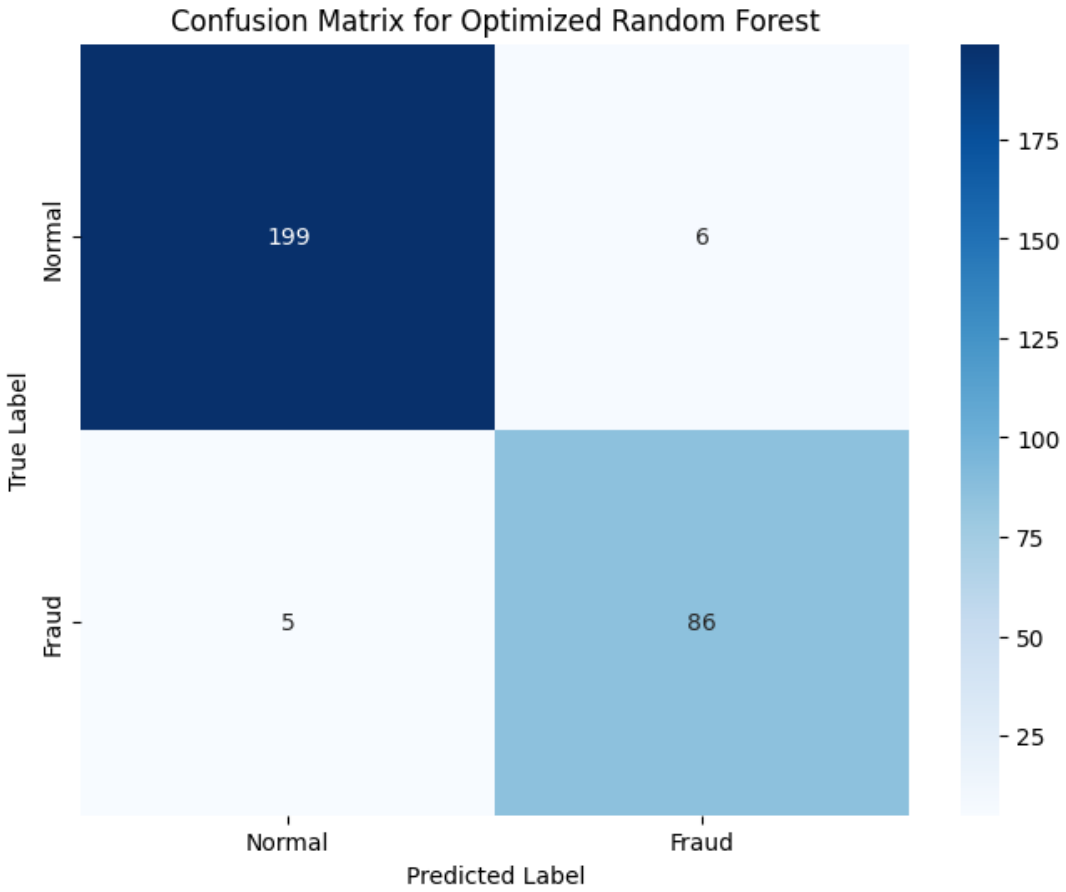
Precision: Measures the percentage of transactions classified as fraud that were indeed fraudulent, helping evaluate the false positive rate.

Recall: Measures the percentage of actual fraudulent transactions correctly identified by the model, which is critical to minimize undetected fraud.

F1Score: The harmonic mean of precision and recall, providing a balanced metric that considers both false positives and false negatives.

Confusion Matrix: Visualized as a normalized matrix to show the percentages of true positives, false positives, true negatives, and false negatives, offering an intuitive understanding of model performance.





Findings

1. Methodology Overview

Data Handling and Balancing: We addressed the data imbalance using SMOTE, which allowed the model to train on an equal number of fraudulent and nonfraudulent cases, improving the model's ability to detect fraud.

Model Selection and Tuning: After evaluating various models, the Random Forest Classifier was chosen. Its parameters were fine-tuned to maximize precision and recall, thus reducing the number of missed fraud cases.

2. Key Results

High Detection Rate of Fraud: The model achieved a recall rate of 94.51% for fraud cases, meaning it successfully identified the majority of fraudulent transactions.

Low False Positive Rate: Only 3.41% of normal transactions were misclassified as fraud, suggesting the model minimizes unnecessary alerts to legitimate users.

Balanced Performance: The model's high precision and recall make it effective in detecting fraud while keeping false alarms low, making it suitable for deployment in a real-world setting.

3. Conclusion

Combined with SMOTE for balancing, the Random Forest model demonstrates excellent potential for real-world fraud detection applications. The model’s high recall and precision mean it can detect fraudulent transactions accurately without overwhelming the system with false positives.

The model achieved high accuracy in detecting both fraudulent and non-fraudulent transactions, with the following outcomes:

High True Positive Rate (94.51%): Most fraudulent transactions are accurately detected.

Low False Negative Rate (5.49%): Few fraudulent transactions are missed, which is crucial in a fraud detection setting.

Acceptable False Positive Rate (3.41%): Some normal transactions are misclassified as fraud, but this rate is low enough to be acceptable in many real-world scenarios.

Future work could explore more complex ensemble techniques or real-time data streaming solutions to further enhance model performance and integrate with a live detection pipeline.

4. Suggestions for Non-Technical Stakeholders

Impact of the Model: The model provides a reliable way to detect fraudulent transactions, offering a balance between catching fraudulent activities and minimizing disruptions for legitimate users.