Credit Card Fraud Detection

Contents

1 Introduction 3

2 Data Exploration and Preprocessing 4

2.1 Loading the Data 4

2.2 Checking for Missing Values 4

2.3 Handling Duplicate Rows 4

2.4 Analyzing Class Distribution 4

2.5 Data Splitting and Resampling 5

2.5.1 Stratified K-Fold Cross-Validation 5

2.5.2 SMOTETomek Resampling 5

3 Model Development and Hyperparameter Tuning 7

3.1 XGBoost Classifier 7

3.1.1 Hyperparameters Tuned 7

3.1.2 Best Parameters Found 7

3.1.3 Model Evaluation 7

3.2 Decision Tree Classifier 8

3.2.1 Hyperparameters Tuned 8

3.2.2 Best Parameters Found 8

3.2.3 Model Evaluation 8

3.3 Random Forest Classifier 9

3.3.1 Hyperparameters Tuned 9

3.3.2 Best Parameters Found 9

3.3.3 Model Evaluation 9

4 Results Interpretation and Visualization 10

4.1 Model Comparison 10

4.1.1 XGBoost Classifier 10

4.1.2 Decision Tree Classifier 10

4.1.3 Random Forest Classifier 10

4.2 Feature Importance Analysis 10

5 Challenges and Considerations **12**
5.1 Class Imbalance 12
5.2 Overfitting 12
5.3 Metric Selection 12

6 Conclusion **13**

7 Recommendations **14**

8 References **15**

Chapter 1

Introduction

Credit card fraud detection is a critical issue in the financial sector, where fraudulent transactions can lead to significant monetary losses and undermine customer trust. The challenge lies in accurately identifying fraudulent transactions from a vast number of legitimate ones, especially when the dataset is highly imbalanced, with fraudulent cases being a tiny fraction of the total transactions.

The objective of this report is to develop and evaluate machine learning models that can effectively detect fraudulent credit card transactions. This involves data exploration and preprocessing, model development, and results interpretation with visualization.

Chapter 2

Data Exploration and Preprocessing

2.1 Loading the Data

The dataset used for this analysis contains credit card transactions made by European cardholders in September 2013. It comprises transactions over two days, totaling 284,807 entries, with only 492 (0.172%) being fraudulent. All features are numerical, resulting from a Principal Component Analysis (PCA) transformation, except for 'Time' and 'Amount'. The 'Class' variable indicates whether a transaction is fraudulent (1) or legitimate (0).

The dataset is loaded into a pandas DataFrame, and the first few rows are inspected to understand the data structure.

2.2 Checking for Missing Values

We verify that there are no missing values in the dataset using the command `data.isnull().sum()`, ensuring data completeness.

2.3 Handling Duplicate Rows

The dataset contains 1,081 duplicate rows, which are removed to prevent bias and redundancy in the model training.

2.4 Analyzing Class Distribution

The dataset is highly imbalanced:

- Legitimate transactions (Class 0): 283,253 instances.
- Fraudulent transactions (Class 1): 473 instances.

A count plot visualizes this imbalance, highlighting the need for techniques to handle skewed class distributions.

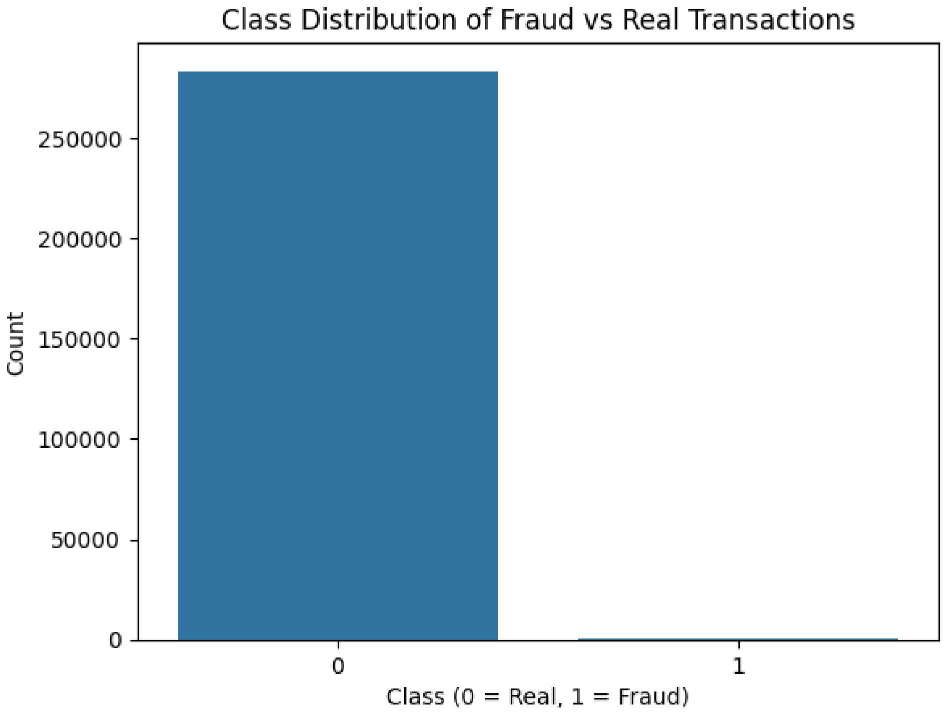

Figure 2.1: Class Distribution

2.5 Data Splitting and Resampling

2.5.1 Stratified K-Fold Cross-Validation

To ensure that the training and testing sets have the same class distribution, we use stratified k-fold cross-validation. This technique maintains the proportion of classes in each fold, which is crucial for imbalanced datasets. It helps prevent the model from being biased towards the majority class and ensures that minority class instances are adequately represented during training and evaluation.

2.5.2 SMOTETomek Resampling

Given the class imbalance, we apply the SMOTETomek technique on the training data to balance the classes.

- **SMOTE (Synthetic Minority Over-sampling Technique)**: Generates synthetic examples of the minority class to increase its representation.
- **Tomek Links**: Removes instances of the majority class that are borderline with the minority class, helping to clean overlapping data.

The combination helps in creating a more balanced and cleaner dataset, improving the model's ability to learn patterns associated with fraudulent transactions.

Chapter 3

Model Development and Hyperparameter Tuning

We experiment with three classifiers: XGBoost, Decision Tree, and Random Forest. For each model, we perform hyperparameter tuning using `GridSearchCV` to find the optimal parameters that yield the best performance.

3.1 XGBoost Classifier

3.1.1 Hyperparameters Tuned

- `n_estimators`: [100, 200]
- `learning_rate`: [0.01, 0.1]
- `max_depth`: [3, 5]

3.1.2 Best Parameters Found

- `learning_rate`: 0.01
- `max_depth`: 3
- `n_estimators`: 200

3.1.3 Model Evaluation

The XGBoost classifier shows a good balance between precision and recall for the minority class.

Table 3.1: Confusion Matrix for XGBoost Classifier

	Predicted Legitimate	Predicted Fraud
Actual Legitimate	56,036	614
Actual Fraud	16	79

Classification Report for Class 1 (Fraud):

- Precision: 0.11
- Recall: 0.83
- F1-Score: 0.20

3.2 Decision Tree Classifier

3.2.1 Hyperparameters Tuned

- `max_depth`: [None, 10]
- `min_samples_split`: [2, 5]

3.2.2 Best Parameters Found

- `max_depth`: None
- `min_samples_split`: 2

3.2.3 Model Evaluation

The Decision Tree classifier performs poorly in detecting fraudulent transactions.

Table 3.2: Confusion Matrix for Decision Tree Classifier

	Predicted Legitimate	Predicted Fraud
Actual Legitimate	56,626	24
Actual Fraud	95	0

Classification Report for Class 1 (Fraud):

- Precision: 0.00
- Recall: 0.00
- F1-Score: 0.00

3.3 Random Forest Classifier

3.3.1 Hyperparameters Tuned

- `n_estimators`: [100, 200]
- `max_depth`: [None, 10]
- `min_samples_split`: [2, 5]

3.3.2 Best Parameters Found

- `n_estimators`: 100
- `max_depth`: 10
- `min_samples_split`: 5

3.3.3 Model Evaluation

The Random Forest classifier shows improvement over the Decision Tree.

Table 3.3: Confusion Matrix for Random Forest Classifier

	Predicted Legitimate	Predicted Fraud
Actual Legitimate	56,638	12
Actual Fraud	23	72

Classification Report for Class 1 (Fraud):

- Precision: 0.86
- Recall: 0.76
- F1-Score: 0.80

Chapter 4

Results Interpretation and Visualization

4.1 Model Comparison

4.1.1 XGBoost Classifier

- Provides a good balance between precision (0.11) and high recall (0.83) for the minority class.
- However, the precision is low, indicating a high number of false positives.

4.1.2 Decision Tree Classifier

- Fails to detect any fraudulent transactions, with zero recall and precision for the minority class.

4.1.3 Random Forest Classifier

- Achieves high precision (0.86) and good recall (0.76) for fraud detection.
- The F1-score of 0.80 suggests a strong balance between precision and recall.

4.2 Feature Importance Analysis

An analysis of feature importance from the Random Forest model helps identify which features contribute most to fraud detection. The features with the highest importance scores are:

Feature	**Importance Score**
V14	0.2237

V10	0.1385
V12	0.1120
V4	0.1013
V17	0.1004
V3	0.0564
V16	0.0549
V11	0.0524
V2	0.0068

A bar plot visualizes these importance scores, emphasizing the most influential features in predicting fraud.

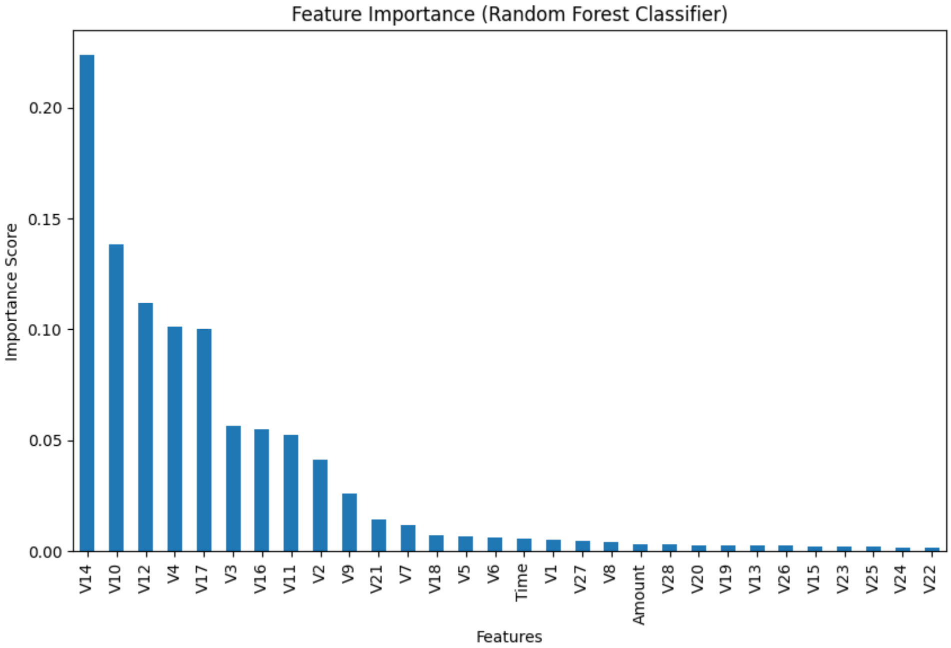

Figure 4.1: Feature Importance from Random Forest Classifier

Chapter 5

Challenges and Considerations

5.1 Class Imbalance

The extreme imbalance between legitimate and fraudulent transactions makes it challenging for models to learn the patterns of the minority class. Without handling the imbalance, models tend to predict the majority class, leading to poor recall for fraud detection.

5.2 Overfitting

Synthetic data generated by SMOTE can cause models to overfit. Using stratified cross-validation and limiting the complexity of models (e.g., setting `max_depth` in Decision Trees) helps mitigate overfitting.

5.3 Metric Selection

Accuracy is not a reliable metric in imbalanced datasets. Focusing on precision, recall, and F1-score for the minority class provides a better understanding of model performance.

Chapter 6

Conclusion

The Random Forest classifier, after hyperparameter tuning and data balancing with SMOTETomek, performs best in detecting fraudulent transactions, with a precision of 86% and recall of 76% for the minority class. The approach of using stratified k-fold cross-validation ensures that the model generalizes well and is not biased due to class imbalance. SMOTETomek effectively balances the dataset by generating synthetic minority class instances and cleaning overlapping majority class instances. Feature importance analysis provides insights into which features are most indicative of fraud, which can be valuable for further investigation and understanding of fraudulent behavior patterns.

Future work could involve exploring other resampling techniques, ensemble methods, or anomaly detection algorithms to further improve fraud detection performance.

Chapter 7

Recommendations

- **Threshold Adjustment:** Adjusting the classification threshold may improve the balance between precision and recall, depending on the specific costs associated with false positives and false negatives.
- **Cost-Sensitive Learning:** Incorporate cost-sensitive learning approaches that assign higher misclassification costs to fraudulent transactions.
- **Anomaly Detection Techniques:** Explore unsupervised learning methods or anomaly detection techniques that may capture fraud patterns not evident in supervised learning.
- **Real-Time Detection:** Implement the model in a real-time detection system, considering computational efficiency and response time.

Chapter 8

References

1. **Dataset Source:** Credit Card Fraud Detection Dataset

2. **SMOTETomek Reference:** Batista, G.E.A.P.A., Prati, R.C., & Monard, M.C. (2004). A study of the behavior of several methods for balancing machine learning training data. *ACM SIGKDD Explorations Newsletter*, 6(1), 20-29.

3. **Evaluation Metrics for Imbalanced Datasets:** Saito, T., & Rehmsmeier, M. (2015). The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets. *PLoS One*, 10(3), e0118432.