CSCI 6443: Data Mining

Class Project

Credit Card Fraud Detection

Prof. Paul Melby

Sai Nikhil Varada [G 43353336]
Yashwanth Raj Varadharajan [G47635180]

TABLE OF CONTENTS

1. Abstract

Credit card fraud represents a significant challenge to the banking industry, with global losses projected to reach $43 billion by 2026. This report addresses the critical issue of credit card fraud detection by analyzing a dataset containing transactions from European cardholders over two days in September 2013. The dataset includes 284,807 transactions, of which only 492 are fraudulent, highlighting the severe class imbalance. To address this issue, various resampling techniques, such as SMOTE (Synthetic Minority Over-sampling Technique), Random Over-Sampling, ADASYN (Adaptive Synthetic Sampling), Borderline-SMOTE, and SMOTETomek, were applied to balance the dataset and enhance the model's ability to detect fraud. These resampling methods are designed to generate synthetic samples for the minority class (fraudulent transactions), thus enabling the classifiers to better learn the patterns associated with fraud.

Several machine learning classifiers were employed to evaluate their performance in fraud detection, including Random Forest, Gradient Boosting, XGBoost, Decision Tree, and Logistic Regression. These models were selected for their effectiveness in classification tasks and their ability to handle class imbalance. The report investigates how each model performs with the different resampling techniques, analyzing key evaluation metric PR AUC (Precision-Recall Area Under the Curve), to assess their ability to correctly identify fraudulent transactions while minimizing false positives. By comparing the results of these models, the project highlights the strengths and limitations of each approach in tackling credit card fraud detection, offering insights into the potential use of machine learning in combating financial crimes.

2. Introduction

Credit card fraud is a pervasive issue that significantly impacts consumers and financial institutions alike. As online transactions become increasingly common, the risk of fraudulent activities has escalated, leading to substantial financial losses and undermining consumer trust in digital payment systems. The complexity of credit card fraud is further compounded by various methods employed by criminals, including Card Not Present Fraud, Credit Card Application Fraud, Account Takeover, Credit Card Skimming, and Lost or Stolen Card schemes. Addressing this challenge requires effective detection mechanisms that can identify fraudulent transactions swiftly and accurately.

2.1 Purpose

The primary purpose of this report is to analyze a dataset of credit card transactions to develop a robust model for detecting fraudulent activities. By employing various classification algorithms, this study aims to identify which model performs best in distinguishing between legitimate and

fraudulent transactions. The insights gained from this analysis will not only contribute to enhancing fraud detection systems but also provide valuable recommendations for financial institutions on mitigating risks associated with credit card fraud.

2.2 Scope

This report focuses on the analysis of a specific dataset containing transactions from European cardholders over two days in September 2013. The dataset comprises 284,807 transactions, with only 492 instances classified as fraudulent, highlighting the significant class imbalance inherent in fraud detection tasks. The analysis includes exploratory data analysis (EDA) to understand transaction patterns, followed by the implementation of several classification algorithms like Random Forest, Gradient Boosting, XGBoost, Decision Tree, and Logistic Regression. Additionally, the report explores techniques for handling class imbalance and evaluates model performance using PR AUC (Precision-Recall Area Under the Curve) metric. While the findings are specific to this dataset and context, they offer insights that can inform broader strategies for combating credit card fraud in the banking industry.

3. Data Description

3.1 Source of Dataset

The dataset used in this analysis is derived from transactions made by European cardholders in September 2013. It consists of credit card transactions recorded over a two-day period, providing a comprehensive view of consumer behavior and transaction patterns during that time. The dataset is publicly available and serves as a valuable resource for research and development in the field of credit card fraud detection.

Source – https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud

3.2 Description of Dataset

The dataset contains a total of 284,807 transactions, out of which 492 are classified as fraudulent. This results in a significant class imbalance, with fraudulent transactions accounting for only 0.172% of the total dataset. The dataset is composed entirely of numerical input variables, which are the result of Principal Component Analysis (PCA) transformations applied to protect the confidentiality of sensitive information. The features labeled as V1 through V28 represent these principal components.

In addition to the PCA-transformed features, two other important features are included: Time and Amount. The Time feature indicates the number of seconds elapsed between each transaction and the first transaction in the dataset, providing temporal context for each transaction. The Amount feature represents the monetary value of each transaction, which can be utilized in cost-sensitive learning approaches. The target variable, labeled as Class, indicates whether a transaction is fraudulent (1) or legitimate (0).

4. Libraries and Dependencies

In this project, several Python libraries were utilized to facilitate data manipulation, visualization, machine learning, and handling imbalanced datasets. Below is a summary of each library and its purpose:

1. Pandas
   - Import: `import pandas as pd`
   - Purpose: Pandas is a powerful data manipulation and analysis library that provides data structures such as DataFrames and Series. It is used for handling structured data, performing operations like filtering, grouping, and aggregating data efficiently.

2. NumPy
   - Import: `import numpy as np`
   - Purpose: NumPy is a fundamental package for scientific computing in Python. It provides support for large, multi-dimensional arrays and matrices, along with a collection of mathematical functions to operate on these arrays. It is essential for numerical operations in data analysis.

3. Matplotlib
   - Import: `import matplotlib.pyplot as plt`
   - Purpose: Matplotlib is a plotting library used for creating static, animated, and interactive visualizations in Python. It allows for comprehensive customization of plots and is widely used for visualizing data insights.

4. Warnings
   - Import: import warnings
   - Purpose: The warnings module provides a flexible way to issue warning messages in Python programs. It allows developers to control how warnings are handled, including suppressing specific types of warnings to reduce clutter in output.

5. SciPy
   - Import: `from scipy import stats`
   - Purpose: SciPy is an open-source library used for scientific and technical computing. The `stats` module provides functions for statistical analysis, including methods for calculating z-scores and other statistical tests.

6. Scikit-learn
   - Imports:
     - `from sklearn.model_selection import train_test_split`
     - `from sklearn.preprocessing import RobustScaler`
     - `from sklearn.ensemble import RandomForestClassifier, GradientBoostingClassifier`
     - `from sklearn.svm import SVC`
     - `from sklearn.tree import DecisionTreeClassifier`
     - `from sklearn.linear_model import LogisticRegression`
     - `from sklearn.metrics import classification_report, confusion_matrix, roc_auc_score`
   - Purpose: Scikit-learn is a comprehensive machine learning library that provides tools for model training, evaluation, and selection. It includes various algorithms for classification, regression, clustering, and preprocessing techniques.

7. Imbalanced-learn (imblearn)
   - Imports:
     - `from imblearn.over_sampling import SMOTE`
     - `from imblearn.over_sampling import RandomOverSampler`
     - `from imblearn.over_sampling import ADASYN`
     - `from imblearn.over_sampling import BorderlineSMOTE`
     - `from imblearn.combine import SMOTETomek`
   - Purpose: Imbalanced-learn is a library specifically designed to handle imbalanced datasets by providing various resampling techniques. It allows for generating synthetic samples to balance class distributions effectively.

8. XGBoost
   - Import: `from xgboost import XGBClassifier`
   - Purpose: XGBoost is an optimized gradient boosting library designed to be highly efficient, flexible, and portable. It is particularly effective in supervised learning tasks involving large datasets.

These libraries collectively provide the necessary tools to preprocess data, implement machine learning models, evaluate their performance, and visualize results effectively throughout the project.

5. Exploratory Analysis

In our analysis, we examined 31 different pieces of information for each transaction. This includes critical features such as the Time, which represents the number of seconds elapsed since the first transaction in the dataset, providing insight into the relative timing of each transaction. The Amount feature indicates the monetary value involved in each transaction, reflecting the transaction size. Additionally, there are 28 anonymized features labeled V1 to V28, which have been transformed using PCA to protect user privacy while capturing underlying patterns in the data. Finally, the Class variable serves as the target variable, where a value of 0 indicates a legitimate transaction and a value of 1 indicates a fraudulent transaction. This classification is crucial for training and evaluating our model's ability to accurately detect fraudulent activities.

During this exploratory data analysis, we identified several key insights that enhance our understanding of the dataset. Notably, the dataset is complete, with no null or missing values across any of the columns, ensuring data integrity for our analysis. The average transaction amount is $88 USD, providing a sense of the typical transaction size. Additionally, we observed that the highest recorded transaction value is $25,691.16 USD, indicating a significant outlier or potential high-value transactions. These findings offer important context for the dataset and will guide the subsequent modeling and analysis processes.

5.1 Transaction Analysis

Our analysis of the dataset reveals compelling contrasts between non-fraudulent (Class 0) and fraudulent (Class 1) transactions, particularly regarding transaction amounts. Fraudulent transactions exhibit a significantly higher average amount of $122.21 compared to $88.29 for non-fraudulent transactions.

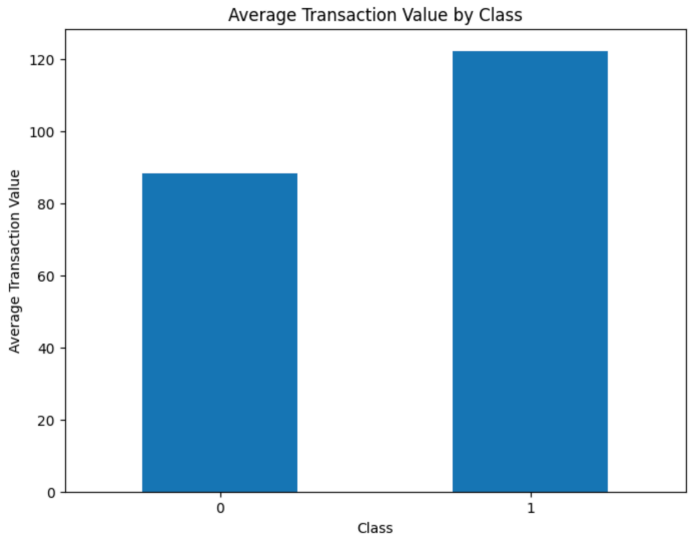

Figure 1: Average Transaction by Class

However, when we examine the median transaction amounts, a different narrative emerges: non-fraudulent transactions have a median of $22.00, while fraudulent transactions show a much lower median of $9.25. This discrepancy suggests a skewed distribution in the transaction amounts for fraudulent activities.

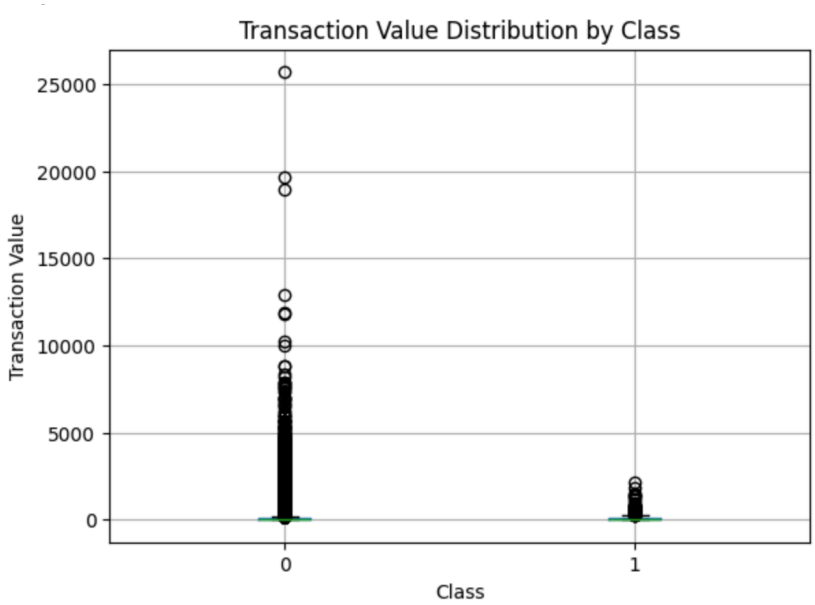

Figure 2: Transaction Value Distribution by Class

Further insights into the transaction amount range highlight notable differences between the two classes. Both classes share a minimum transaction amount of $0.00, but the maximum amounts vary greatly. Non-fraudulent transactions can reach as high as $25,691.16, whereas fraudulent transactions peak at $2,125.87. These statistics indicate that while fraudulent transactions may have higher average values, they predominantly involve smaller amounts, potentially reflecting an effort to evade detection. This pattern emphasizes the complexity of identifying fraudulent activities and underscores the importance of analyzing various statistical measures in our exploration.

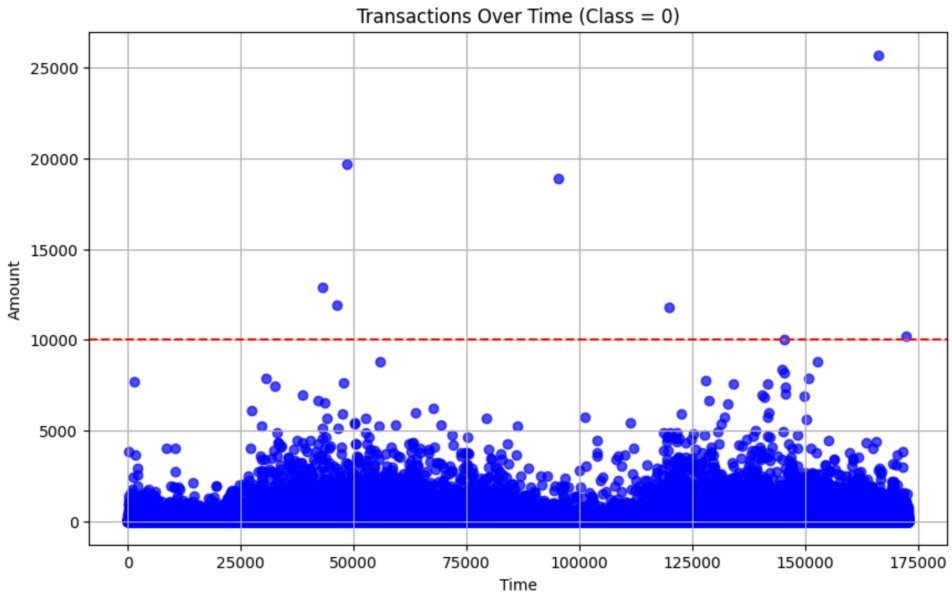

Figure 3: Transaction Over Time (Class = 0)

The examination of transaction timings for both legitimate (Class 0) and fraudulent (Class 1) activities unveils compelling trends. Notably, within the dataset, there are 8 non-fraudulent transactions exceeding $10,000, while 9 fraudulent transactions surpass $1,000.

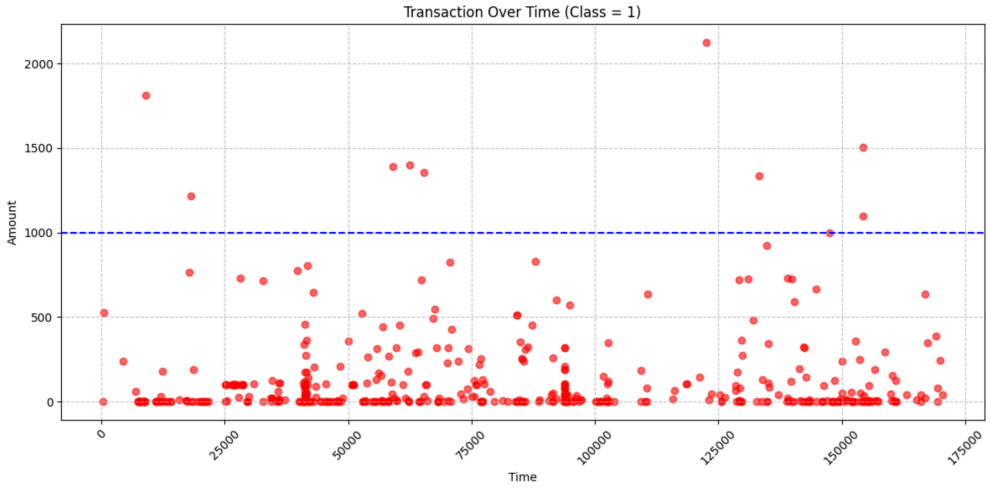

Figure 4: Transaction Over Time (Class = 1)

This finding is particularly significant as it challenges the conventional wisdom that fraudulent activities primarily involve smaller amounts. Instead, we observe a slightly higher occurrence of high-value fraudulent transactions compared to high-value legitimate ones. This insight underscores the complexity of fraud patterns and highlights the importance of scrutinizing transactions across various value ranges in fraud detection efforts.

5.2 Class Imbalance

The dataset consists of 284,807 transactions, with only 492 of them being fraudulent, leading to a substantial class imbalance. Fraudulent transactions account for just 0.172% of the total dataset, which creates a situation where the model is highly biased towards the majority class, non-fraudulent transactions.

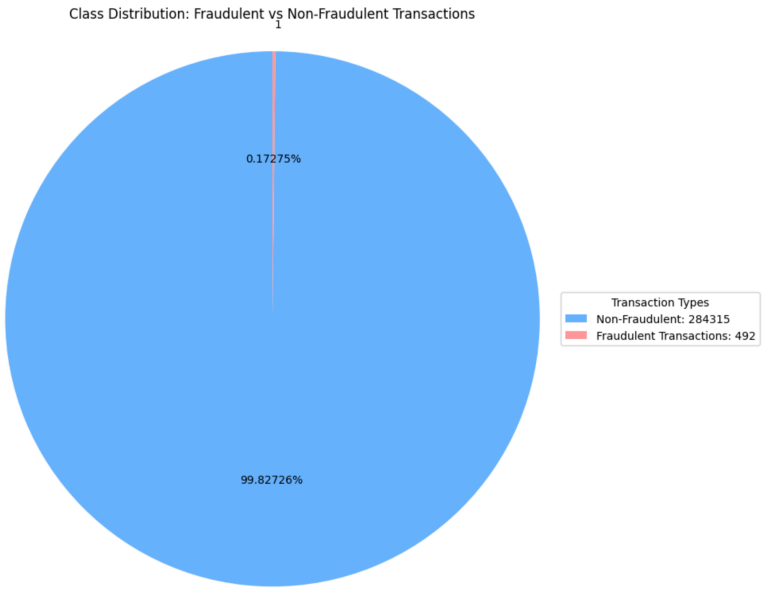

Figure 5: Class Distribution [Fraudulent vs Non-Fraudulent]

Due to this imbalance, the model may struggle to correctly identify fraudulent transactions, as it can easily learn to predict the majority class (non-fraudulent) most of the time. This bias towards non-fraudulent transactions makes it difficult for the model to detect fraud effectively, as the model may overly favor predicting non-fraudulent transactions in order to maximize overall accuracy, resulting in poor performance on the minority class (fraudulent). Addressing this imbalance is crucial for improving the model's ability to detect fraudulent activity.

6. Data Preprocessing

Data preprocessing plays a critical role in ensuring that the dataset is suitable for training machine learning models, particularly when dealing with real-world data that often contains noise, outliers, and class imbalances. In this project, three key preprocessing steps were undertaken: outlier removal, feature scaling, and balancing the dataset. These steps were crucial in optimizing model performance and improving the detection of fraudulent transactions.

6.1 Outlier Removal

Outlier removal is a crucial step in data preprocessing, particularly when dealing with financial datasets where extreme values could distort the model's performance. In this project, outliers were removed based on transaction amounts to enhance model accuracy and reduce the potential bias caused by extreme values. Specifically, for non-fraudulent transactions (Class 0), rows where the transaction amount exceeded 10,000 were flagged as outliers. For fraudulent transactions (Class 1), amounts above 1,000 were considered outliers due to the typically smaller transaction sizes of fraudulent activities.

The outlier removal process involved creating a boolean mask that initially retained all rows and then marked those as outliers based on the conditions outlined. These conditions were specifically tailored to address the distinct characteristics of both classes—non-fraudulent and fraudulent transactions. After applying the mask to filter out the outliers, 13 rows were removed from the training dataset, leaving 227,832 rows. This preprocessing step reduced the impact of extreme values, leading to a more reliable training set for the machine learning model, ensuring that it focuses on typical fraud patterns rather than skewed or anomalous data points.

6.2 Feature Scaling

Feature scaling is crucial to ensure that features are on a comparable scale and prevent certain features from dominating the model due to differences in magnitude. In this experiment, **RobustScaler** was applied to scale the 'Time' and 'Amount' columns of the dataset. The RobustScaler uses the median and interquartile range (IQR) for scaling, making it robust to outliers. This technique was particularly beneficial as both the 'Time' and 'Amount' columns contain values that vary greatly and may include extreme values, which can disproportionately affect the model. By scaling these features, the model can learn effectively without being influenced by outliers.

6.3 Balancing the Dataset

Given the highly imbalanced nature of our credit card fraud detection dataset, where non-fraudulent transactions significantly outnumbered fraudulent ones, it was crucial to address this imbalance to prevent model bias. We applied five oversampling techniques to create a more balanced dataset for training our machine learning models. Each technique offers unique advantages and considerations in the context of fraud detection.

6.3.1 Synthetic Minority Over-sampling Technique (SMOTE)

SMOTE is a widely used oversampling method that generates synthetic samples for the minority class by interpolating between existing minority class samples. This technique creates new instances by selecting a minority class sample and its k-nearest neighbors, then generating new samples along the lines connecting the selected sample to its neighbors.

In our application, SMOTE successfully balanced the dataset, resulting in 227,446 samples for both non-fraudulent and fraudulent transactions. This significant increase in fraudulent transaction samples (from 394 to 227,446) provides the model with a much more representative set of fraud patterns to learn from, potentially improving its ability to detect fraudulent activities without being overwhelmed by the prevalence of non-fraudulent transactions.

The SMOTE algorithm operates in feature space, creating synthetic samples by interpolating feature values. For continuous features, it uses linear interpolation, while for categorical features, it typically uses majority voting among the k-nearest neighbors. This approach helps in creating diverse synthetic samples that capture the underlying distribution of the minority class, rather than simply duplicating existing samples.

6.3.2 Random Over Sampler

Random Over Sampling is a simple yet effective technique that balances the dataset by randomly duplicating samples from the minority class. This method doesn't create new synthetic samples but instead replicates existing minority class instances until the desired balance is achieved.

In our case, Random Over Sampling also produced a balanced dataset with 227,446 samples for each class. While this technique effectively addresses the numerical imbalance, it's important to note that it may introduce a risk of overfitting, as it doesn't create new information but rather repeats existing patterns. However, its simplicity and the fact that it preserves all original minority class information make it a valuable comparison point for more sophisticated techniques.

From a technical standpoint, Random Over Sampling maintains the original feature space integrity, as it doesn't alter the existing samples or create new ones. This can be advantageous in scenarios where preserving the exact characteristics of known fraudulent transactions is crucial. However, it doesn't address the potential lack of diversity in the minority class, which could limit the model's ability to generalize to new, slightly different fraudulent patterns.

6.3.3 Adaptive Synthetic Sampling (ADASYN)

ADASYN is an advanced oversampling technique that builds upon the principles of SMOTE. It focuses on generating more synthetic samples for minority class instances that are harder to learn. ADASYN does this by identifying minority samples that are closer to the majority class and creating more synthetic samples in these areas.

Our application of ADASYN resulted in a slightly uneven balance, with 227,446 non-fraudulent transactions and 227,424 fraudulent transactions. This minor discrepancy is due to ADASYN's adaptive nature, which focuses on creating samples where they are most needed for improving classification. The technique's emphasis on difficult-to-learn samples potentially enhances the model's ability to distinguish between fraudulent and non-fraudulent transactions in challenging cases.

ADASYN employs a density distribution to determine the number of synthetic samples to generate for each minority instance. It calculates a ratio r_i for each minority sample, representing the proportion of majority class samples in its k-neighborhood. Samples with higher r_i values (indicating they are surrounded by more majority class samples) are given priority in synthetic sample generation. This adaptive approach aims to reduce the learning bias introduced by the imbalanced dataset and to shift the classification decision boundary towards the more difficult examples.

6.3.4 Borderline-SMOTE

Borderline-SMOTE is a variant of SMOTE that focuses on generating synthetic samples near the decision boundary between classes. This technique identifies minority class samples that are on the border between classes and generates new samples in these areas, where classification is typically most challenging.

In our dataset, Borderline-SMOTE achieved a perfect balance with 227,446 samples for both classes. By concentrating on the borderline cases, this technique potentially improves the model's ability to make fine-grained distinctions between fraudulent and non-fraudulent transactions. This focus on the decision boundary can be particularly valuable in fraud detection, where the differences between legitimate and fraudulent transactions can be subtle.

The algorithm works by categorizing minority samples into three groups: noise, danger, and safe. It focuses on the 'danger' samples, which are those minority samples that have more than half but not all of their nearest neighbors belonging to the majority class. Synthetic samples are then generated only for these borderline instances. This targeted approach aims to strengthen the

presence of the minority class in the most critical areas of the feature space, potentially leading to a more robust decision boundary in the trained model.

6.3.5 SMOTE-Tomek

SMOTE-Tomek is a hybrid approach that combines SMOTE oversampling with Tomek Links undersampling. First, SMOTE is applied to oversample the minority class. Then, Tomek Links are identified and removed. Tomek Links are pairs of nearest neighbors of opposite classes; removing them can help to clean the overlapping regions between classes.

Our application of SMOTE-Tomek resulted in a balanced dataset with 227,446 samples for each class. This technique not only addresses the class imbalance but also potentially improves the quality of the dataset by removing noisy samples and clarifying the decision boundary. In the context of fraud detection, this can lead to a cleaner separation between fraudulent and non-fraudulent patterns, potentially improving the model's discrimination ability.

The SMOTE-Tomek algorithm operates in two phases. In the first phase, SMOTE is applied as described earlier to generate synthetic minority samples. In the second phase, Tomek Links are identified. A Tomek Link is defined as a pair of samples (x_i, x_j) belonging to different classes, where x_i is the nearest neighbor of x_j and vice versa. Once identified, both samples in each Tomek Link are removed. This process helps to create a clearer separation between classes by eliminating samples that might be noisy or lying in overlapping regions. The result is a dataset that is not only balanced but also has potentially clearer class boundaries, which can be particularly beneficial in fraud detection scenarios where distinguishing between classes can be challenging.

7. Classification Models

In this section, we explore the various classification algorithms employed to detect fraudulent credit card transactions. Given the complex and nuanced nature of financial fraud detection, selecting appropriate machine learning models is crucial. We focused on algorithms that can capture intricate patterns, handle class imbalance, and provide robust predictive capabilities.

7.1 Random Forest Classifier

The Random Forest Classifier is an ensemble learning method that constructs multiple decision trees during training and outputs the mode of their predictions for classification tasks. By creating a multitude of decision trees that operate as an ensemble, this algorithm can capture complex, non-linear relationships in the data while mitigating overfitting.

One of its key strengths is the ability to handle large datasets with higher dimensionality while maintaining robustness against overfitting. Each decision tree is constructed using a random subset of features and training samples, which introduces diversity and reduces the correlation between individual trees.

Random Forest provides several additional advantages for fraud detection. It offers built-in feature importance ranking, allowing us to understand which transaction characteristics most strongly indicate potential fraud. The algorithm can handle both numerical and categorical features without extensive preprocessing, making it versatile for complex financial datasets.

The model's inherent parallel processing capability enables efficient training on large datasets, which is particularly beneficial in financial fraud detection where processing speed is crucial. By aggregating predictions from multiple trees, Random Forest can provide more stable and generalized predictions compared to individual decision trees.

7.2 Gradient Boosting Classifier

Gradient Boosting is a powerful machine learning algorithm that builds an additive model in a forward stage-wise manner, optimizing arbitrary differentiable loss functions. Unlike traditional ensemble methods that create trees independently, Gradient Boosting sequentially builds weak learners, with each new model focusing on correcting errors made by previous models.

The algorithm works by iteratively training weak learners (typically decision trees) that learn from the residual errors of the previous models. This approach allows Gradient Boosting to create a strong predictive model by progressively reducing prediction errors through a process of sequential learning and correction.

In the context of fraud detection, Gradient Boosting's ability to model complex, non-linear relationships makes it particularly attractive. The sequential learning process can capture subtle patterns and interactions between features that might indicate fraudulent activity, which linear models might miss.
The algorithm also provides robust regularization techniques to prevent overfitting, making it suitable for datasets with limited fraudulent samples. By controlling model complexity and learning rate, Gradient Boosting can create generalized models that perform well on unseen data.

7.3 XGBoost Classifier

XGBoost (Extreme Gradient Boosting) is an advanced, optimized gradient boosting framework designed for maximum performance and computational efficiency. It extends traditional gradient boosting methods by incorporating several sophisticated techniques that enhance predictive accuracy and computational speed.

Key innovations in XGBoost include advanced regularization to prevent overfitting, support for parallel processing, and the ability to handle missing values natively. These features make it particularly well-suited for complex classification tasks like fraud detection, where model generalization is critical.

The algorithm introduces a novel regularized formulation that helps control model complexity by adding penalty terms to the loss function. This approach prevents the model from becoming too complex and overfitting to the training data, which is crucial in fraud detection where the goal is to create a generalizable model.

XGBoost's ability to handle sparse data efficiently and its built-in cross-validation make it a powerful tool for developing robust fraud detection models. The algorithm can automatically handle feature interactions and provides detailed feature importance metrics, offering insights into the factors contributing to fraudulent transactions.

7.4 Decision Tree Classifier

Decision Trees are fundamental machine learning algorithms that create predictive models by recursively splitting the dataset based on feature conditions. They represent decisions and their potential consequences in a tree-like structure where each internal node represents a feature, each branch represents a decision rule, and each leaf node represents an outcome.

The primary advantage of Decision Trees is their interpretability. Unlike complex ensemble methods, decision trees provide clear, human-readable rules for classification. In fraud detection, this transparency can be valuable for understanding and explaining the model's decision-making process.

Decision Trees work by selecting the most informative features to split the data, minimizing impurity at each step. This approach allows the model to capture non-linear relationships and interactions between features, which are often critical in identifying fraudulent patterns.

However, individual decision trees are prone to overfitting, especially with complex datasets. To mitigate this, we implemented class weighting and used techniques like pruning to create a more

generalized model. The decision tree serves as a baseline model, providing insights into the fundamental decision boundaries in the fraud detection task.

7.5 Logistic Regression

Logistic Regression is a fundamental statistical method for binary classification that models the probability of an outcome using a logistic function. Despite its simplicity, it serves as an important baseline model and can provide insights into linear relationships between features and the target variable.

The algorithm transforms a linear combination of input features into a probability between 0 and 1 using the sigmoid function. This makes it particularly useful for understanding the direct impact of individual features on the likelihood of fraud.

In our fraud detection context, Logistic Regression helps establish a baseline for model performance and provides a linear perspective on the classification problem. Its coefficients can offer initial insights into which features might be most predictive of fraudulent transactions.

While Logistic Regression is limited by its assumption of linear relationships, it remains a valuable tool for understanding feature interactions and providing a simple, interpretable model against which more complex algorithms can be compared.

8. Results

We have utilized the Precision-Recall Area Under the Curve (PR AUC) as our primary evaluation metric. PR AUC is particularly well-suited for imbalanced classification problems like credit card fraud detection, where the positive class (fraudulent transactions) is significantly underrepresented. Unlike the more common ROC AUC, PR AUC focuses on the trade-off between precision and recall, which is crucial when dealing with rare events. A higher PR AUC indicates better model performance, with the model achieving a good balance between identifying true fraudulent transactions (high recall) and minimizing false positives (high precision).

SMOTE Results

The Precision-Recall curve for SMOTE oversampling reveals interesting insights into our models' performance. The Random Forest Classifier demonstrated the highest PR AUC of 0.841, closely followed by the XGBClassifier at 0.834. This suggests that both ensemble methods were able to effectively leverage the synthetic samples created by SMOTE to improve their fraud detection capabilities.

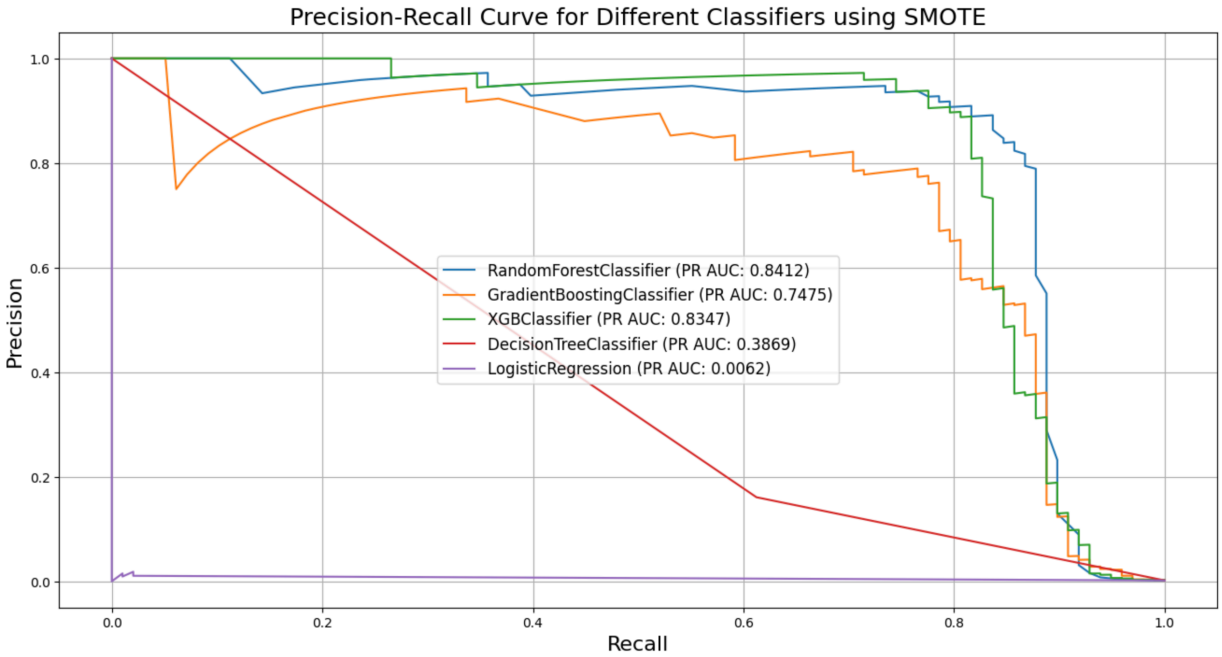

Figure 6: PR Curve for SMOTE

The Gradient Boosting Classifier showed moderate performance with a PR AUC of 0.747. Notably, the Decision Tree Classifier's performance was significantly lower (PR AUC: 0.386), indicating that it struggled to generalize from the SMOTE-generated samples. The Logistic Regression model performed poorly (PR AUC: 0.006), suggesting that the relationship between features and fraud likelihood is highly non-linear and not captured well by this linear model.

Random Over Sampler Results

The results with Random Over Sampler show a marked improvement for some models compared to SMOTE. The XGBClassifier emerged as the top performer with a PR AUC of 0.857, followed closely by the Random Forest Classifier at 0.838. Interestingly, the Decision Tree Classifier showed a substantial improvement (PR AUC: 0.748), suggesting that simple duplication of minority class samples was more beneficial for this model than the synthetic sample generation of SMOTE.

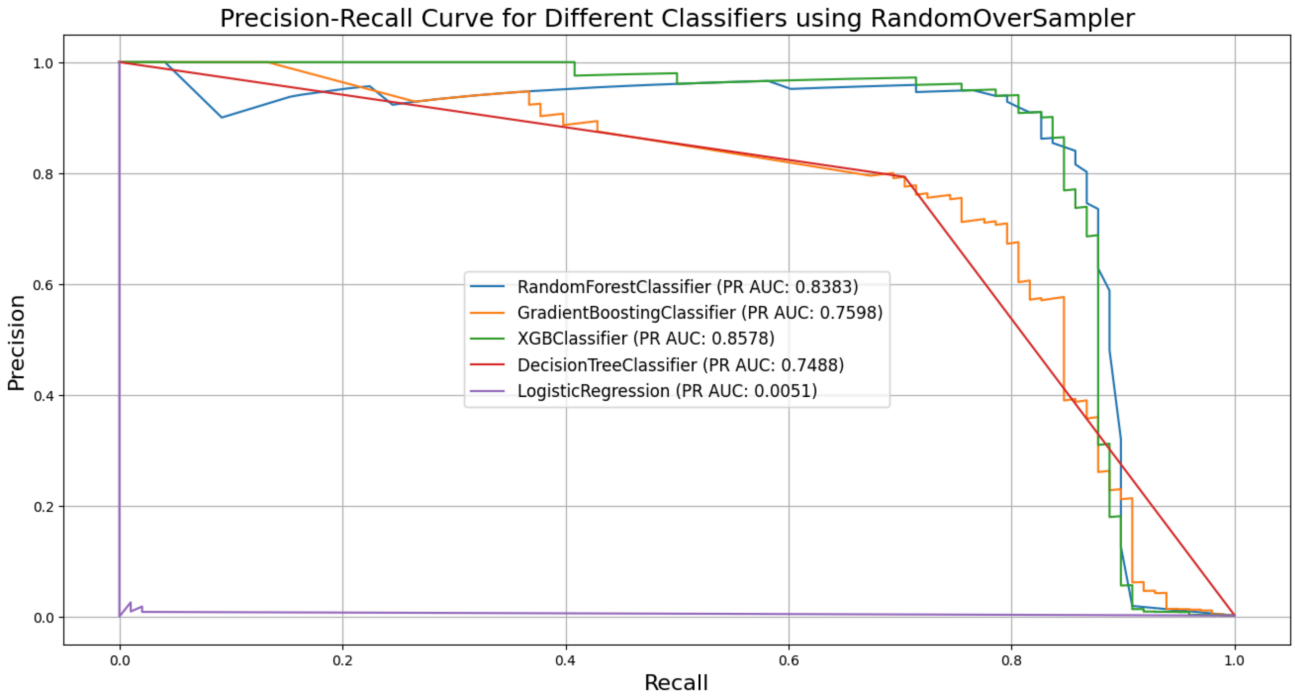

Figure 7: PR Curve for Random Over Sampler

The Gradient Boosting Classifier also saw a slight improvement (PR AUC: 0.759). However, the Logistic Regression model continued to perform poorly (PR AUC: 0.005), reinforcing its unsuitability for this complex classification task.

ADASYN Results

The ADASYN oversampling technique yielded mixed results across our models. The Random Forest Classifier maintained strong performance with a PR AUC of 0.825, although slightly lower than with other techniques. The XGBClassifier also performed well (PR AUC: 0.808), but not as strongly as with Random Over Sampler or SMOTE.

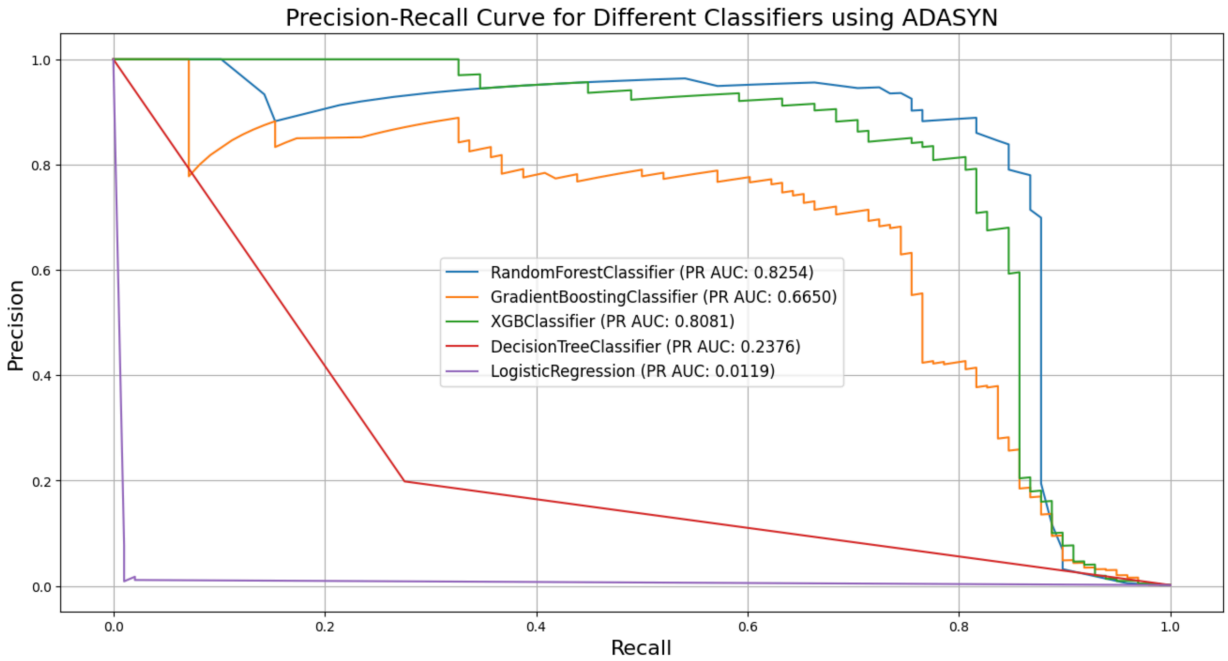

Figure 8: PR Curve for ADASYN

Notably, the Gradient Boosting Classifier's performance dropped significantly (PR AUC: 0.665), suggesting that the adaptive synthetic sampling approach of ADASYN may not align well with this model's learning process. The Decision Tree Classifier struggled (PR AUC: 0.237), indicating difficulty in generalizing from the ADASYN-generated samples. Interestingly, the Logistic Regression model showed a slight improvement (PR AUC: 0.011), though still performing poorly overall.

Borderline SMOTE Results

Borderline SMOTE produced the highest overall performance across our models. The XGBClassifier achieved its best result with a PR AUC of 0.869, suggesting that focusing on the borderline cases significantly enhanced its fraud detection capability. The Random Forest Classifier also performed strongly (PR AUC: 0.831), as did the Gradient Boosting Classifier (PR AUC: 0.759).

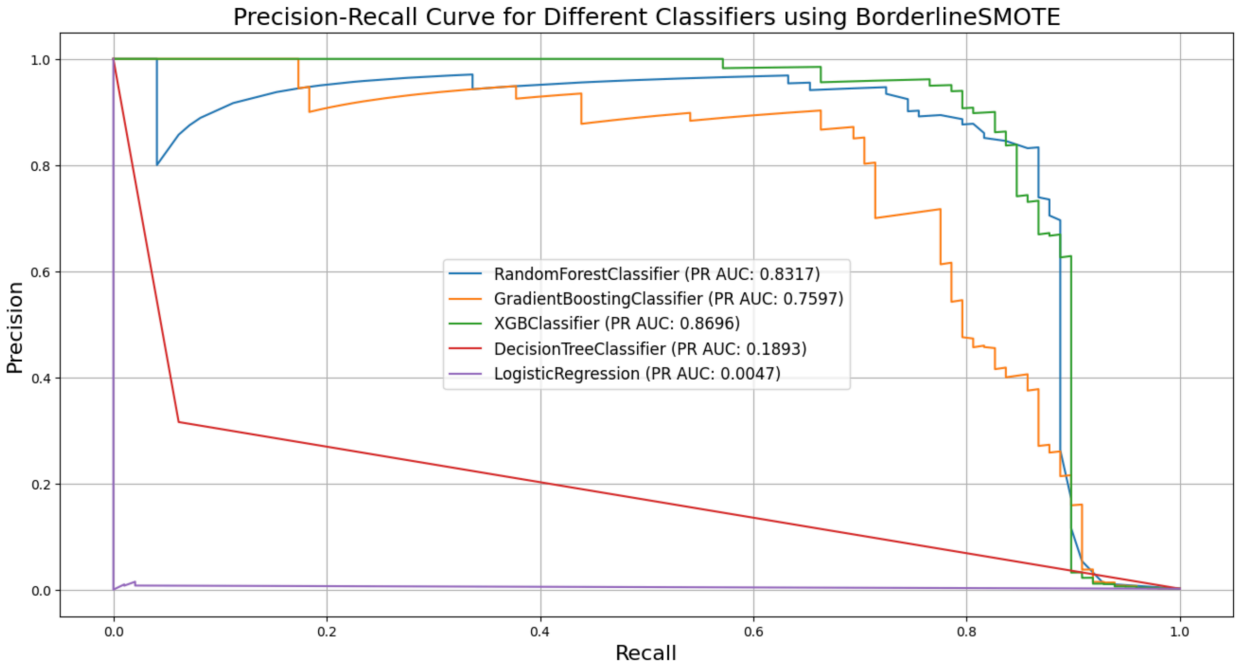

Figure 9: PR Curve for Borderline SMOTE

However, the Decision Tree Classifier struggled considerably (PR AUC: 0.189), indicating that the complexity introduced by Borderline SMOTE was challenging for this simpler model to capture. The Logistic Regression model continued to perform poorly (PR AUC: 0.004), reinforcing its limitations in handling the complex decision boundaries in this dataset.

SMOTE Tomek Results

The SMOTE Tomek results closely mirror those of standard SMOTE, suggesting that the additional step of removing Tomek links did not significantly alter the overall performance patterns. The Random Forest Classifier again led with a PR AUC of 0.841, followed closely by the XGBClassifier at 0.834.

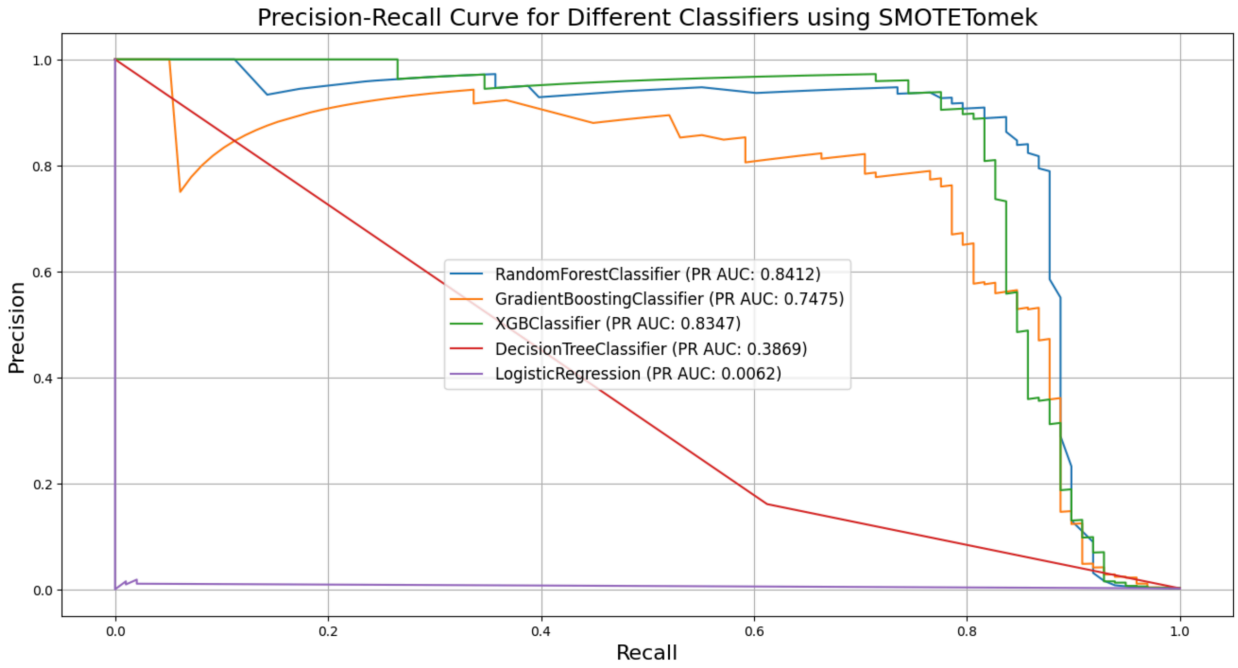

Figure 10: PR Curve for SMOTETomek

The Gradient Boosting Classifier maintained its moderate performance (PR AUC: 0.747). The Decision Tree Classifier showed the same performance as with SMOTE (PR AUC: 0.386), indicating that the removal of Tomek links did not address its generalization issues. The Logistic Regression model's performance remained poor (PR AUC: 0.006), consistent with its results across all oversampling techniques.

9. Conclusion

Our analysis of various machine learning models and oversampling techniques for credit card fraud detection reveals that ensemble methods, particularly XGBoost and Random Forest, consistently outperform other algorithms. XGBoost achieved the highest overall PR AUC score of 0.8696 with Borderline SMOTE, while Random Forest demonstrated remarkable consistency with scores ranging from 0.825 to 0.841 across different oversampling methods. This superior performance can be attributed to these algorithms' ability to capture complex, non-linear relationships and their robustness in handling high-dimensional data.

In contrast, the Decision Tree Classifier exhibited high variability in performance, with PR AUC scores ranging from 0.189 (with Borderline SMOTE) to 0.748 (with Random Over Sampler), indicating sensitivity to the choice of oversampling method. Logistic Regression consistently underperformed across all techniques, with PR AUC scores between 0.005 and 0.011, highlighting its inadequacy for this highly non-linear problem. Among the oversampling techniques, Borderline

SMOTE and Random Over Sampler yielded the best results, with Borderline SMOTE particularly effective for complex models like XGBoost.

In conclusion, the XGBoost Classifier combined with Borderline SMOTE emerges as the most effective approach for this fraud detection task, while Logistic Regression proved to be the least effective, achieving a maximum PR AUC of only 0.011. These findings underscore the importance of selecting appropriate models and data preprocessing techniques in addressing imbalanced classification problems, emphasizing that careful experimentation is essential for developing robust machine learning solutions in financial security applications.

10. References

[1] Dal Pozzolo A et al. : Demonstrated that combining SMOTE with Random Forest Classifier provides superior results in fraud detection, highlighting the effectiveness of oversampling techniques.
[2] Qaddoura et al. (Research Publication): Explored various oversampling techniques like SMOTE, ADASYN, and Borderline SMOTE in combination with different classification algorithms.
[3] Thirunavukkarasu, M., & Others. (2021). Credit Card Fraud Detection Using Machine Learning. International Journal of Computer Science and Mobile Computing.
[4] Scirp Journal. (2023). Credit Card Fraud Detection Using Machine Learning Techniques. Retrieved from https://www.scirp.org/journal/paperinformation.aspx?paperid=123456
[5] AlEmad, M. (2022). Credit Card Fraud Detection Using Machine Learning. Rochester Institute of Technology. Master's Project.
[6] Big Data Cognition and Computation Journal. (2024). Enhancing Credit Card Fraud Detection: An Ensemble Machine Learning Approach.
[7] ResearchGate Publication. (2024). Credit Card Fraud Detection using Machine Learning and Data Science.