Emir Bekrija
emir.bekrija1@stu.ibu.edu.ba
Data Science

Exploratory Data Analysis (EDA) of Credit Card Fraud Detection

Abstract

Credit card fraud has become one of the most serious issues in digital finance, resulting in significant financial losses each year. This study explores the application of data science techniques for detecting fraudulent credit card transactions using a dataset from Kaggle that contains anonymized European transactions. The dataset includes over 284,000 transactions, with only 492 labeled as fraud, making it highly imbalanced. Exploratory Data Analysis (EDA) was performed to understand data distribution and feature relationships. Machine learning models such as Logistic Regression and Random Forest were trained and evaluated to classify transactions as fraudulent or legitimate. To address the imbalance problem, Synthetic Minority Oversampling Technique (SMOTE) was applied to create a balanced training set. Evaluation metrics including precision, recall, F1-score, ROC-AUC, and precision-recall curves were used to assess model performance. The results showed that Random Forest, especially after hyperparameter tuning, achieved the best balance between detecting fraud cases and minimizing false positives. This research highlights how data-driven approaches can improve fraud detection accuracy and contribute to safer financial systems.

1. Introduction

With the rapid growth of online transactions, credit card fraud has become a major concern for financial institutions and consumers. Every day, millions of transactions occur worldwide, and among them, a small fraction involves fraudulent activity. Detecting these fraudulent transactions quickly and accurately is important to reduce financial losses and maintain trust in digital payment systems. However, fraud detection is challenging because fraudulent transactions are extremely rare compared to normal ones, creating a significant imbalance in the data.

Data science offers powerful tools to address this problem by using statistical analysis, machine learning, and data visualization. Through data-driven methods, it becomes possible to identify hidden patterns and unusual behaviors that indicate fraudulent activity. Exploratory Data Analysis (EDA) helps uncover these patterns by examining the relationships between features and understanding how fraudulent transactions differ from legitimate ones. Machine learning algorithms can then be trained to classify transactions based on these insights.

In this study, we use the publicly available Kaggle “Credit Card Fraud Detection” dataset, which contains real transactions made by European cardholders in 2013. The dataset includes 284,807 transactions, out of which only 492 are fraudulent. Because of this imbalance, a major focus of this project is improving fraud detection performance while minimizing false negatives, which represent undetected fraud cases. Logistic Regression and Random Forest models are implemented, and techniques like SMOTE (Synthetic Minority Oversampling Technique) are used to balance the dataset and improve model recall. The goal of this research is to evaluate which model performs best in detecting fraudulent transactions and to provide insights into how data science methods can support fraud prevention systems.

2. Literature Review

Credit card fraud detection has become a critical area of study in data science and machine learning because of the increasing number of online transactions and the sophistication of fraudulent techniques. Many researchers have proposed models and approaches to enhance detection accuracy, improve recall, and reduce false alarms in fraud identification systems.

Aman (2021) emphasized the importance of data preprocessing and anomaly detection techniques such as the Local Outlier Factor and Isolation Forest for identifying suspicious patterns in credit card transactions (Aman, 2021). Similarly, Khan et al. (2021) demonstrated how Random Forest and K-Nearest Neighbor algorithms outperform traditional methods by learning from historical fraud patterns (Khan et al., 2021). Nayak et al. (2023) further explored ensemble models like Random Forest and XGBoost, showing that combining multiple decision trees increases robustness and detection precision (Nayak et al., 2023).

Yeruva et al. (2023) compared various algorithms including Decision Tree, Logistic Regression, and Support Vector Machine to evaluate which classifier achieves higher recall in real-time fraud detection, noting that advanced ML models outperform traditional rule-based systems (Yeruva et al., 2023). In contrast, Khedkar and Gupta (2024) conducted a review of 25 research studies, highlighting that hybrid and ensemble approaches such as deep neural networks and Hidden Markov Models provide significant improvements in fraud recognition when compared to individual algorithms (Khedkar & Gupta, 2024).

Naik and Pise (2022) explored Artificial Neural Networks and genetic algorithms for improving classification accuracy, demonstrating how these approaches can efficiently identify complex, non-linear fraud patterns (Naik & Pise, 2022). Similarly, Arivanantham (2025) proposed combining machine learning models such as Random Forest and Logistic Regression with biometric verification methods to enhance fraud detection reliability (Arivanantham, 2025).

Abdou et al. (2020) discussed the use of data mining and machine learning combinations to increase fraud coverage, suggesting that methods like Support Vector Machines and clustering can be effectively combined for better accuracy (Abdou et al., 2020). Gupta et al. (2020) used Support Vector Machines, Isolation Forest, and Local Outlier Factor to detect anomalies and reported that unsupervised learning models can achieve high fraud detection rates with minimal labeled data (Gupta et al., 2020). Finally, Powar and Dawkhar (2025) compared Logistic Regression and Random Forest on large-scale datasets and found that Random Forest achieved higher accuracy and precision, confirming its strength as an ensemble-based model for financial fraud detection (Powar & Dawkhar, 2025).

Overall, the reviewed studies indicate that combining ensemble methods, balancing techniques like SMOTE, and hybrid data-driven systems significantly improve credit card fraud detection. Most research agrees that Random Forest and ensemble models offer a strong balance between precision and recall, making them suitable for detecting rare fraud cases in imbalanced datasets.

3. Materials and Methodology

3.1. Dataset Description

This study uses the Credit Card Fraud Detection dataset from Kaggle, originally published by the Machine Learning Group of Université Libre de Bruxelles (ULB). The dataset contains transactions made by European cardholders in September 2013. It includes 284,807 transactions, of which only 492 are fraudulent, representing approximately 0.172% of the data. Each transaction is described by 30 features, including 28 numerical features (V1–V28) obtained through Principal Component Analysis (PCA) for confidentiality, and two original features Time and Amount. The target variable Class indicates whether a transaction is legitimate (0) or fraudulent (1).
The dataset's highly imbalanced nature makes it an excellent case study for exploring the performance of different machine learning models and sampling techniques in fraud detection.

This dataset was selected because it provides a realistic and challenging scenario for fraud detection using machine learning. The anonymized nature of the features allows researchers to focus purely on pattern recognition and classification techniques, without dealing with sensitive information. Moreover, the dataset's widespread use in academic studies allows for meaningful comparison between different models and methodologies.

3.2. Exploratory Data Analysis (EDA)

Exploratory Data Analysis (EDA) was performed to understand the characteristics of the dataset, detect irregularities, and identify patterns that could influence model performance. Since the dataset represents financial transactions, the main goal of EDA was to uncover how fraudulent and non-fraudulent transactions differ in scale, frequency, and feature behavior.

The first step was to examine the class distribution. A count plot of the Class variable clearly showed that only a small fraction of the transactions were labeled as fraud (492 out of 284,807). This confirmed that the dataset is highly imbalanced, with the majority class (legitimate transactions) dominating the data. Understanding this imbalance was crucial because it affects how models learn without proper handling, a model could achieve over 99% accuracy simply by predicting all transactions as non-fraud. This insight guided the later decision to use SMOTE and class weighting during model training.

Next, the distribution of transaction amounts was analyzed using histograms and boxplots. Most legitimate transactions had small amounts, while fraudulent ones tended to show a wider variation but were generally concentrated at lower amounts. This observation aligns with real-world fraud patterns, where fraudulent transactions are often smaller to avoid detection by automated monitoring systems. A boxplot comparing Amount across the two classes highlighted that the median and interquartile range of fraudulent transactions were significantly different from those of normal transactions.

The Time variable was also examined to check whether fraud occurrences were time-dependent. Although the dataset did not contain explicit timestamps or dates, plotting Time helped identify patterns in transaction frequency over the course of a day. However, no strong cyclical trend was visible, suggesting that time alone may not be a strong predictor.

Visual analysis was extended to the PCA-transformed features (V1–V28). Kernel Density Estimation (KDE) plots were created for selected features such as V1, V2, and V12 to compare their distributions between fraudulent and non-fraudulent transactions. The differences in the shapes of these curves suggested that some components carry meaningful information that can help distinguish fraud from legitimate activity, even though their actual meanings are unknown due to anonymization.

Finally, a Mann–Whitney U test was performed on the Amount_scaled variable to statistically verify whether there was a significant difference between the distributions of fraudulent and non-fraudulent transactions. The test result ($p < 0.05$) confirmed that the two groups were statistically different, supporting the earlier visual findings.

Through these steps, EDA helped establish a clear understanding of the dataset's structure and guided important preprocessing and modeling decisions. It confirmed that feature scaling was necessary, class imbalance needed to be addressed, and certain PCA components might be more informative for fraud detection.

3.3. Data Preprocessing

Data preprocessing was an essential step in preparing the dataset for model training and evaluation. It ensured that the data was clean, properly formatted, and suitable for use in machine learning algorithms. Since the dataset involved both scaled and unscaled numerical values, as well as extreme class imbalance, several preprocessing techniques were applied to optimize model performance and reduce bias.

1. Feature Scaling

The dataset contained two features in their original numerical form: Time and Amount. These features had wide numerical ranges compared to the PCA-transformed features (V1–V28), which were already standardized. To bring all variables to a similar scale, StandardScaler from the scikit-learn library was applied. This transformation standardized the data by centering each feature around a mean of zero and scaling it to have unit variance.
Feature scaling is crucial for algorithms such as Logistic Regression and Random Forest because large feature magnitudes can bias model training, especially when distance-based metrics are involved. After scaling, the original Time and Amount columns were replaced with Time_scaled and Amount_scaled to maintain consistency in feature magnitude across all variables.

2. Handling Missing and Irrelevant Data

Upon inspection, the dataset was found to be free of missing or null values, meaning no data imputation was required. Since the dataset was already anonymized and preprocessed by PCA, no categorical encoding or additional feature transformation was needed. Irrelevant columns such as transaction identifiers were not present, which simplified the cleaning process.

3. Train–Test Split

To evaluate model performance fairly, the dataset was divided into training and testing subsets using an 80/20 split. The split was done using stratified sampling, ensuring that the proportion of fraudulent and non-fraudulent transactions remained consistent in both subsets. This step was important because simple random sampling could result in the test set containing too few fraudulent cases, making evaluation unreliable.
The training set was used for model fitting and hyperparameter tuning, while the test set served for final performance evaluation.

4. Handling Imbalanced Data

Since fraudulent transactions represented less than 0.2% of the dataset, standard training could lead to models biased toward predicting legitimate transactions. To address this, two complementary strategies were used:

Class Weight Adjustment: For baseline models such as Logistic Regression and Random Forest, the class_weight='balanced' parameter was used. This automatically assigns higher weights to the minority class (fraudulent transactions) during training, helping the model pay more attention to underrepresented examples.

Synthetic Minority Oversampling Technique (SMOTE): In addition to class weighting, the Synthetic Minority Oversampling Technique was applied to the training data. SMOTE creates synthetic examples of the minority class by interpolating between existing minority samples and their nearest neighbors. Unlike random oversampling, which duplicates minority samples and can cause overfitting, SMOTE generates new, slightly varied data points that enrich the minority class distribution. This helps models better learn the patterns associated with fraudulent transactions.

It is important to note that SMOTE was only applied to the training set to avoid data leakage. If oversampling were applied before splitting the data, the synthetic samples could appear in both training and testing sets, inflating performance metrics.

5. Feature Preparation

After preprocessing, the dataset consisted of 30 numerical variables 28 PCA features and 2 scaled features all ready for modeling. The target column Class remained unchanged, serving as the dependent variable for supervised learning.

3.4. Machine Learning Models

This study implemented and compared several machine learning models to classify transactions as either legitimate or fraudulent. The models used were Logistic Regression, Random Forest Classifier, Logistic Regression with SMOTE, and a tuned Random Forest model optimized with GridSearchCV. Each model was selected for its suitability to handle class imbalance, interpretability, and effectiveness in binary classification problems.

1. Logistic Regression

Logistic Regression was used as the baseline model because of its simplicity, interpretability, and efficiency for binary classification tasks. It estimates the probability of a transaction being fraudulent by modeling the relationship between the input variables and the target class using a sigmoid function.

This model was trained with the parameter class_weight='balanced' to address the extreme class imbalance in the dataset. By assigning higher weights to fraudulent samples, the model gives equal importance to both classes during training. The primary reason for including Logistic Regression was to establish a reference point for evaluating more complex algorithms. Its linear nature allows for clear interpretation of how individual features influence the prediction outcome, even though the PCA-transformed variables are not directly interpretable in this case.

Logistic Regression also performs well when the decision boundary between classes is relatively linear, and it tends to be computationally efficient, making it suitable for large datasets like this one. However, due to the dataset's non-linear patterns and severe imbalance, Logistic Regression alone was not expected to achieve optimal recall or precision.

2. Random Forest Classifier

The Random Forest Classifier was selected as the next model because it can effectively handle non-linear relationships and complex interactions among features. It is an ensemble learning algorithm that constructs multiple decision trees during training and aggregates their predictions to produce the final output. This ensemble approach reduces overfitting and improves generalization compared to individual decision trees.

The model was trained with the parameter class_weight='balanced', ensuring that fraudulent transactions had a greater impact on the learning process. Random Forest was an appropriate choice because of its robustness to noise, scalability to large datasets, and ability to rank feature importance an advantage for identifying which PCA features contribute most to detecting fraud.

During evaluation, the Random Forest model demonstrated better recall and overall discrimination ability (ROC-AUC) compared to Logistic Regression. This confirmed that it could capture the more complex and non-linear structures present in the data.

3. Logistic Regression with SMOTE

To further explore the effect of class imbalance handling, Logistic Regression was retrained using a SMOTE-balanced dataset. Applying the Synthetic Minority Oversampling Technique to the training data allowed the model to learn from a more balanced distribution of fraudulent and legitimate transactions.

While the basic Logistic Regression model with class weights could adjust the loss function to account for imbalance, it still trained on the same limited number of minority samples. By using SMOTE, the model gained exposure to synthetic fraud samples, which helped improve recall the proportion of correctly identified frauds though sometimes at the cost of precision.
This comparison between Logistic Regression with and without SMOTE provided insight into how oversampling affects performance and model stability. It also demonstrated that data-level solutions like SMOTE can complement algorithm-level adjustments such as class weighting.

4. Hyperparameter Tuning using GridSearchCV

To enhance performance, **GridSearchCV** was used to tune Random Forest hyperparameters such as:

- **Number of trees (n_estimators)**
- **Maximum tree depth (max_depth)**
- **Number of features considered at each split (max_features)**

The tuning process used recall as the primary scoring metric, since minimizing false negatives is essential in fraud detection (missing a fraud is more costly than a false alarm).

5. Final Random Forest Model

The final Random Forest model, trained with the best parameters from GridSearchCV, delivered the strongest overall results. It achieved a higher ROC-AUC and improved recall compared to the baseline and unoptimized models. The feature importance plot generated from this model revealed that certain PCA components, especially V17, V14, and V12 were the most influential in identifying fraudulent transactions.
These findings are consistent with those in previous studies, which have also highlighted Random Forest's ability to handle high-dimensional and imbalanced financial datasets effectively. Its ensemble nature makes it less sensitive to noise and feature correlation, which is common in PCA-transformed data.
Overall, the Random Forest model with optimized parameters was selected as the final model because it offered the best balance between sensitivity (recall) and precision, demonstrating its reliability for detecting rare fraud cases in large transaction datasets.

3.5. Model Evaluation Metrics

Model performance was assessed using several metrics suited for imbalanced classification:

- Precision: The proportion of predicted frauds that were actually fraudulent.
- Recall: The proportion of actual frauds correctly identified by the model.
- F1-Score: The harmonic mean of precision and recall, balancing both metrics.
- ROC-AUC (Receiver Operating Characteristic – Area Under Curve): Measures the ability of the model to distinguish between classes.
- Precision–Recall Curves: Plotted to visualize the trade-off between precision and recall, which is critical for imbalanced datasets.

3.6. Feature Importance Analysis

Feature importance was evaluated using the Random Forest model to identify which PCA components contributed most to fraud detection. The analysis showed that only a few features, particularly V17, V14, and V12, had a major impact on predicting fraudulent transactions. These components likely capture key patterns that distinguish fraud from legitimate activity. Although the features are anonymized due to PCA transformation, understanding their relative importance helps explain the model's decision-making and highlights which variables carry the most predictive power.

4. Results and Discussion

4.1. Exploratory Analysis Results

The dataset showed a severe imbalance, as illustrated in **Figure 1 (Class Distribution)**, where legitimate transactions dominated the data with only 492 fraudulent cases out of 284,807 total. Such imbalance justified the later use of class weighting and SMOTE to improve fraud detection.

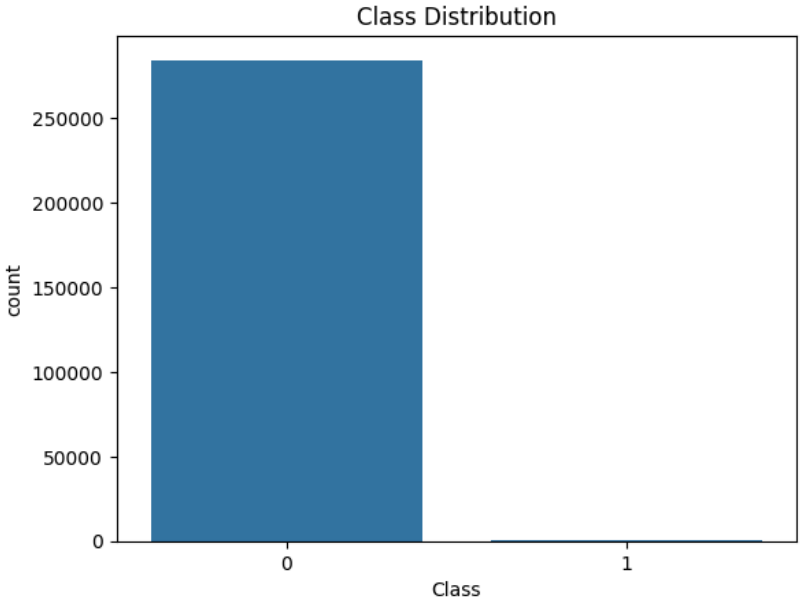


Figure 1. *Class distribution showing the extreme imbalance between legitimate and fraudulent transactions.*

Figure 2 (Transaction Amount Distribution) displayed that most transactions had low monetary values, while a few high-value transactions existed as outliers. **Figure 3 (Fraud vs Non-Fraud Boxplot)** confirmed that fraudulent transactions typically had lower amounts, aligning with real-world patterns where small transaction amounts are often used to test stolen cards before larger purchases.

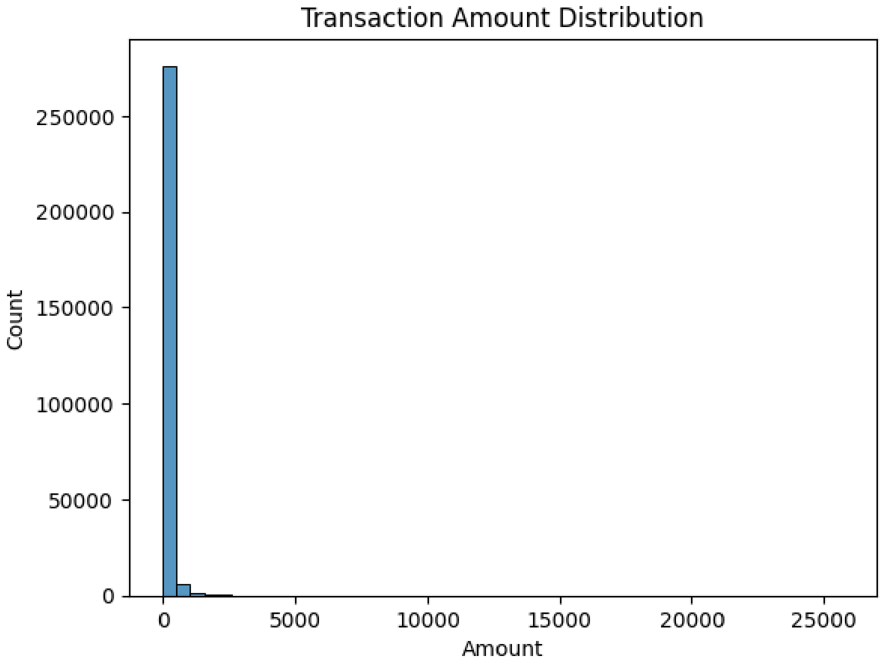


Figure 2. *Transaction amount distribution indicating that most transactions are low in value with a few high-value outliers.*

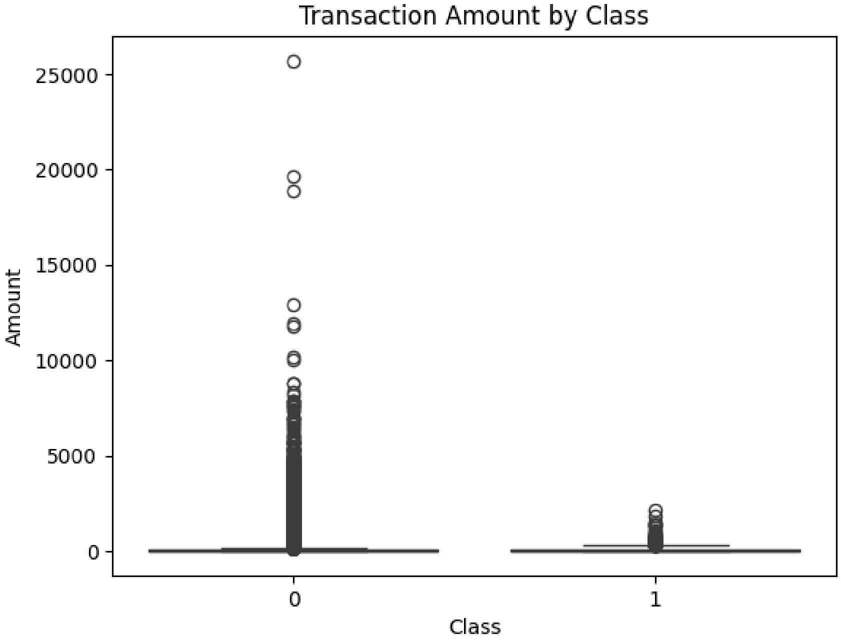
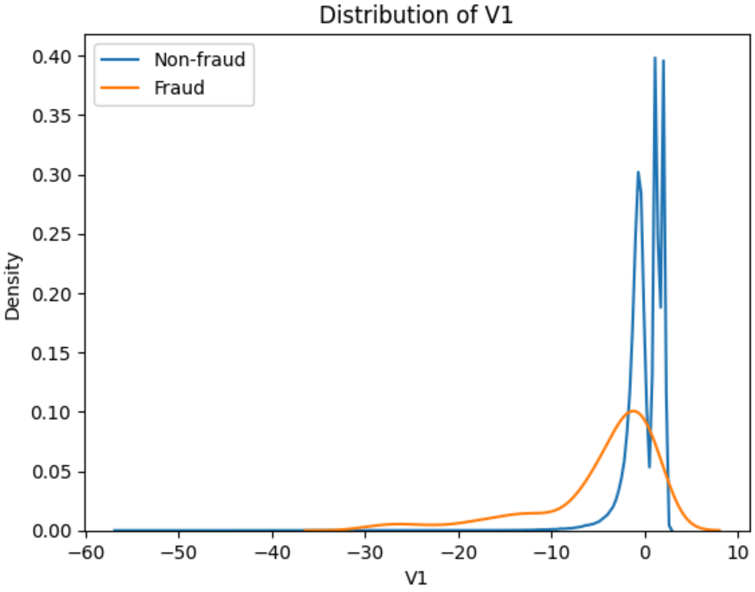
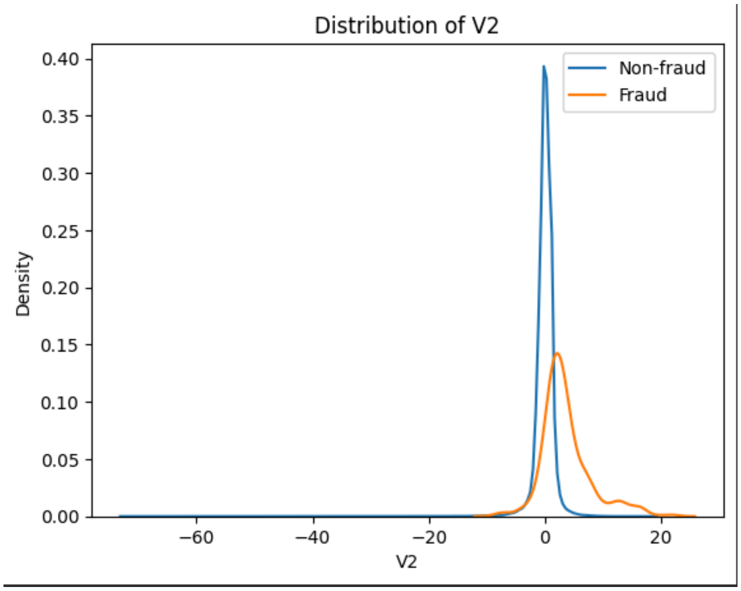


Figure 3. *Boxplot comparing transaction amounts for fraud and non-fraud classes, showing that fraudulent transactions typically involve smaller amounts.*

Further inspection of selected PCA-transformed features, shown in **Figure 4 (PCA Feature Comparison for V1, V2, and V12)**, revealed noticeable distribution differences between fraudulent and non-fraudulent transactions. Fraudulent samples displayed distinct peaks or spreads across these components, suggesting that PCA successfully captured underlying patterns related to fraudulent behavior.





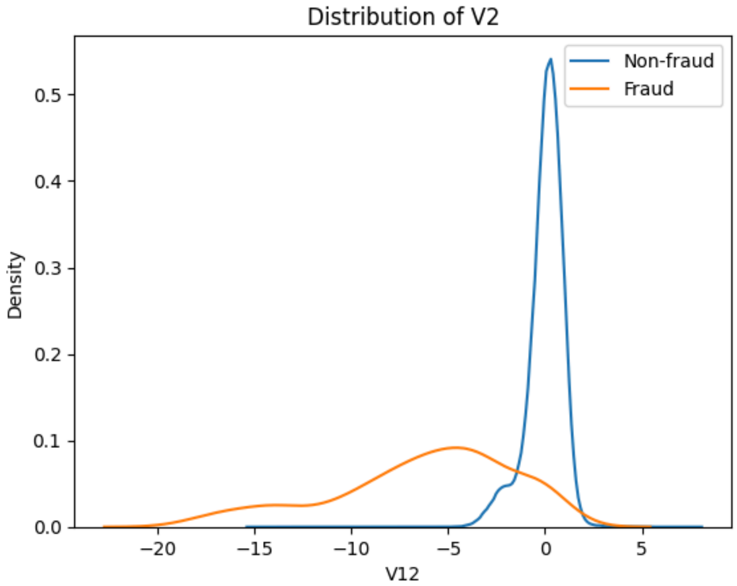


Figure 4. *Distribution plots of PCA features V1, V2, and V12 comparing fraudulent and non-fraudulent transactions.*

4.2. Model Performance and Evaluation

Four models were trained and evaluated: Logistic Regression, Random Forest, Logistic Regression with SMOTE, and the GridSearchCV-optimized Random Forest. Each model's performance was assessed using precision, recall, F1-score, ROC-AUC, and average precision metrics.

a. Logistic Regression

The **Logistic Regression model** served as a baseline. Its **confusion matrix (Figure 5)** and **ROC curve (Figure 6)** showed that the model correctly identified most legitimate transactions but missed several fraud cases. The classification report indicated a high precision but a moderate recall, confirming that it was conservative in predicting fraud. While this reduced false positives, it also meant missing some true fraud cases, an undesirable trade-off in fraud detection.

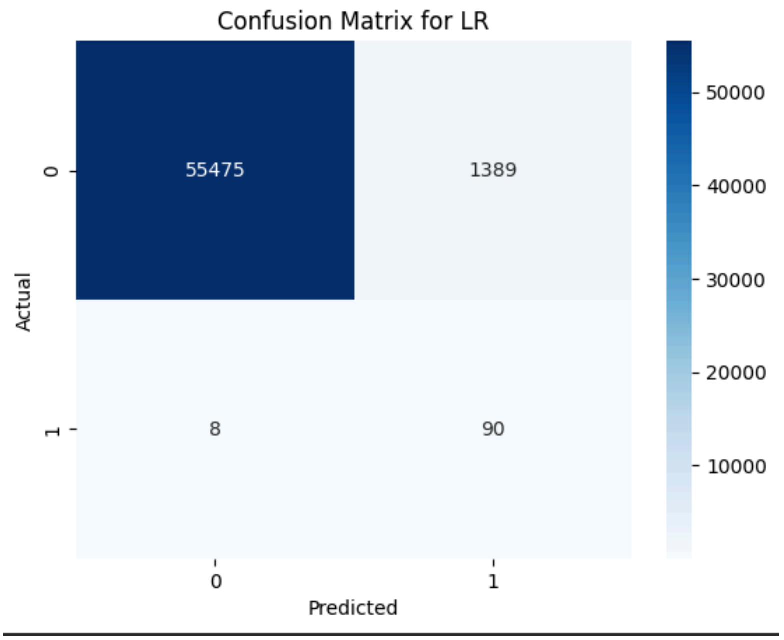


Figure 5. Confusion matrix for Logistic Regression showing correct and incorrect classifications on the test set.

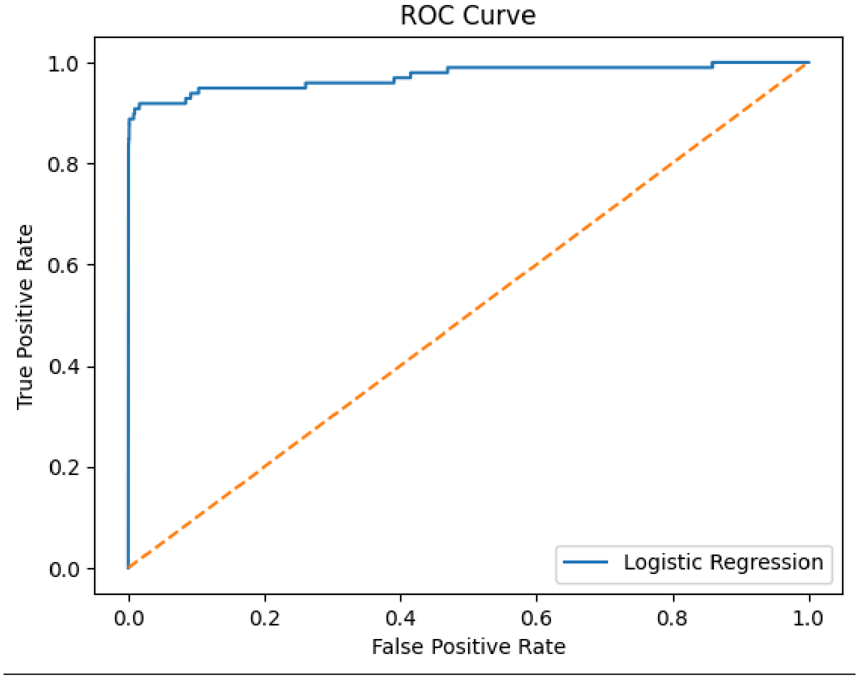


Figure 6. ROC curve for Logistic Regression demonstrating model performance in distinguishing fraud from legitimate transactions.

b. Random Forest Classifier

The **Random Forest model** produced stronger results. The **confusion matrix (Figure 7)** showed fewer false negatives compared to Logistic Regression, and the **ROC curve (Figure 8)** indicated improved discrimination between classes with a higher AUC score. Random Forest effectively captured complex patterns in the PCA features and demonstrated higher recall without significantly reducing precision.

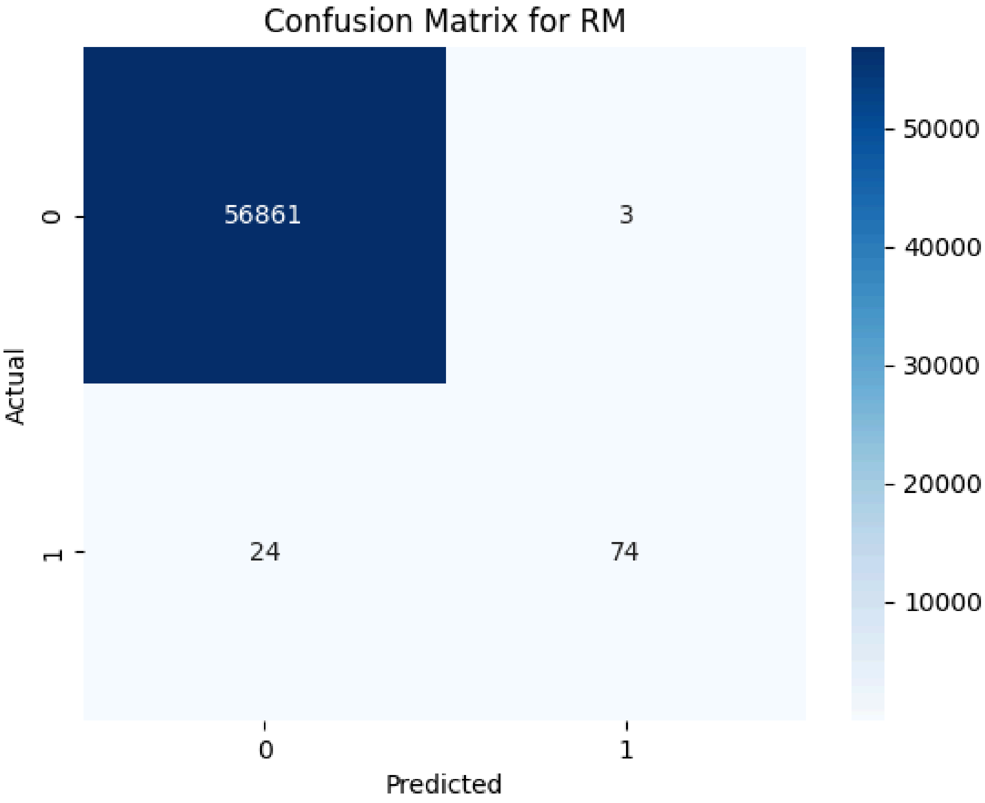


Figure 7. *Confusion matrix for the baseline Random Forest model displaying improved fraud detection compared to Logistic Regression.*

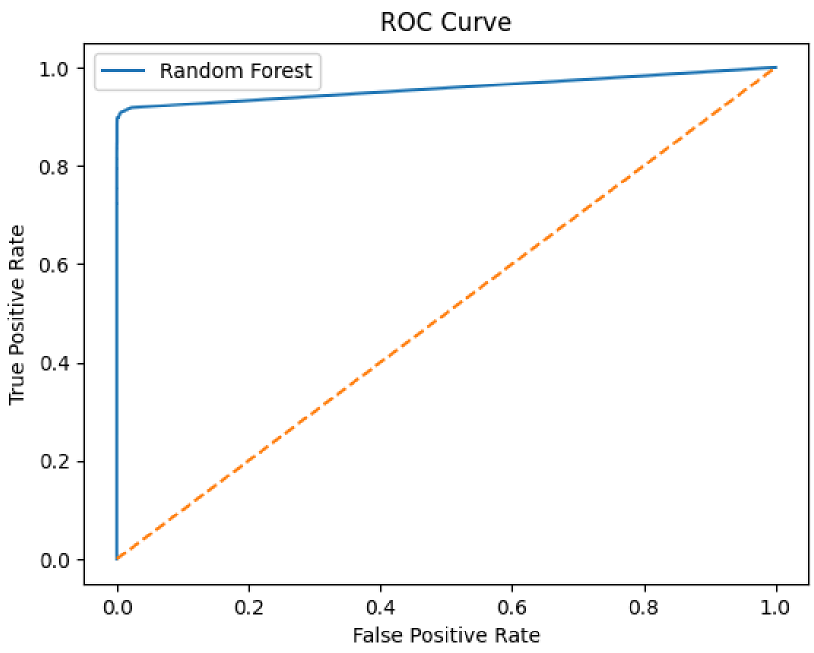


Figure 8. *ROC curve for the baseline Random Forest model showing higher AUC compared to Logistic Regression.*

c. Logistic Regression with SMOTE

After applying **SMOTE** to balance the training data, the **Logistic Regression with SMOTE** model achieved higher recall, meaning it detected more fraudulent transactions. However, the **confusion matrix (Figure 9)** and **ROC curve (Figure 10)** revealed a slight drop in precision, showing that oversampling introduced more false positives. This trade-off was expected but acceptable, as missing fraud cases are costlier than occasional false alerts in real-world systems.

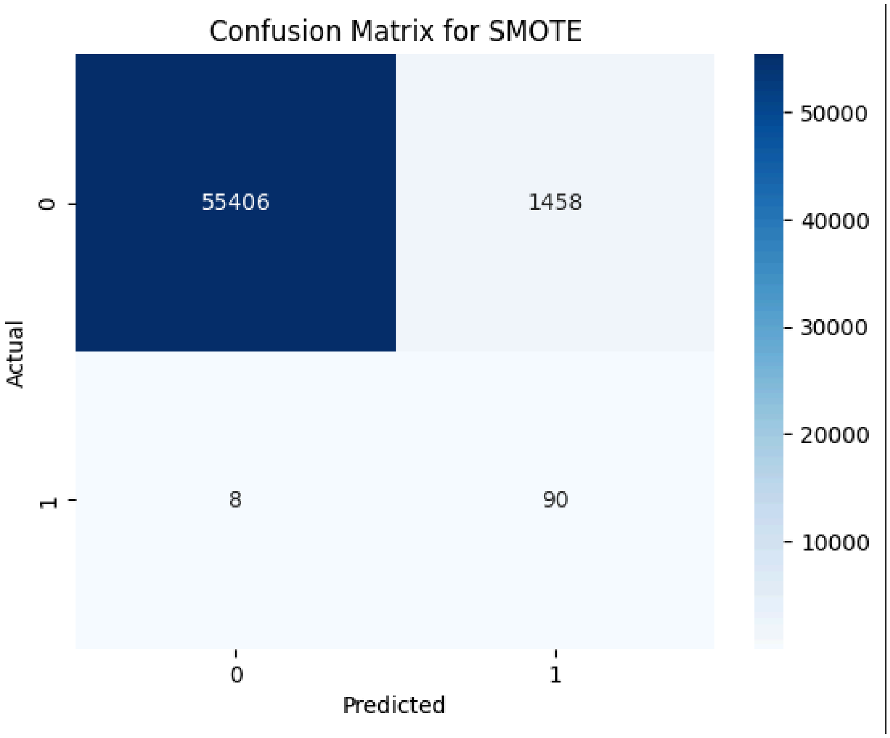


Figure 9. *Confusion matrix for Logistic Regression trained with SMOTE illustrating increased fraud detection after oversampling.*

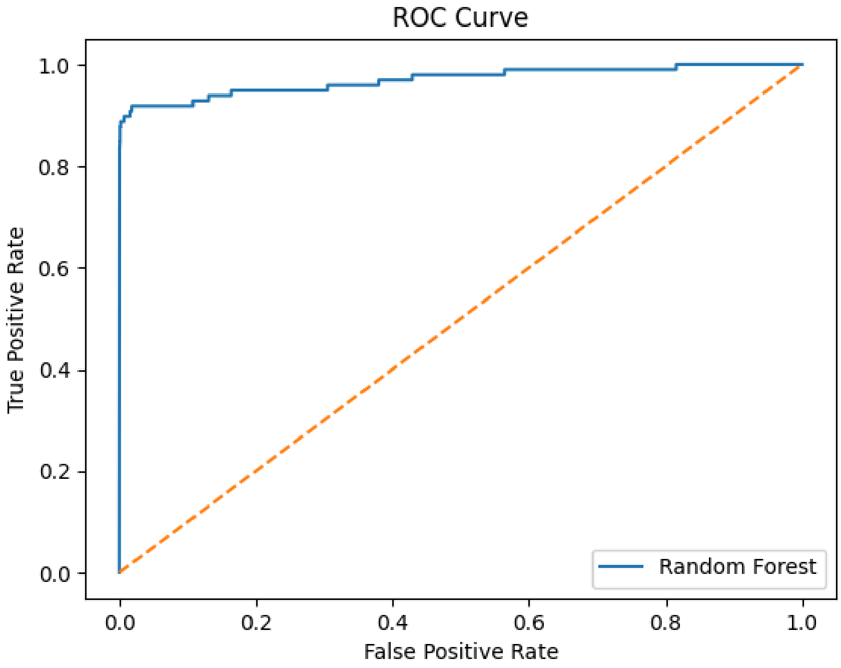


Figure 10. *ROC curve for Logistic Regression with SMOTE showing improved recall but slightly lower precision.*

d. Tuned Random Forest with GridSearchCV

The **final Random Forest model**, tuned through **GridSearchCV**, delivered the best overall performance. The **optimized confusion matrix (Figure 11)** showed the fewest false negatives, and the **ROC curve (Figure 12)** reached the highest AUC score among all models. This version of the model achieved the best balance between sensitivity and precision, confirming that hyperparameter tuning improved its learning capacity and reduced bias.

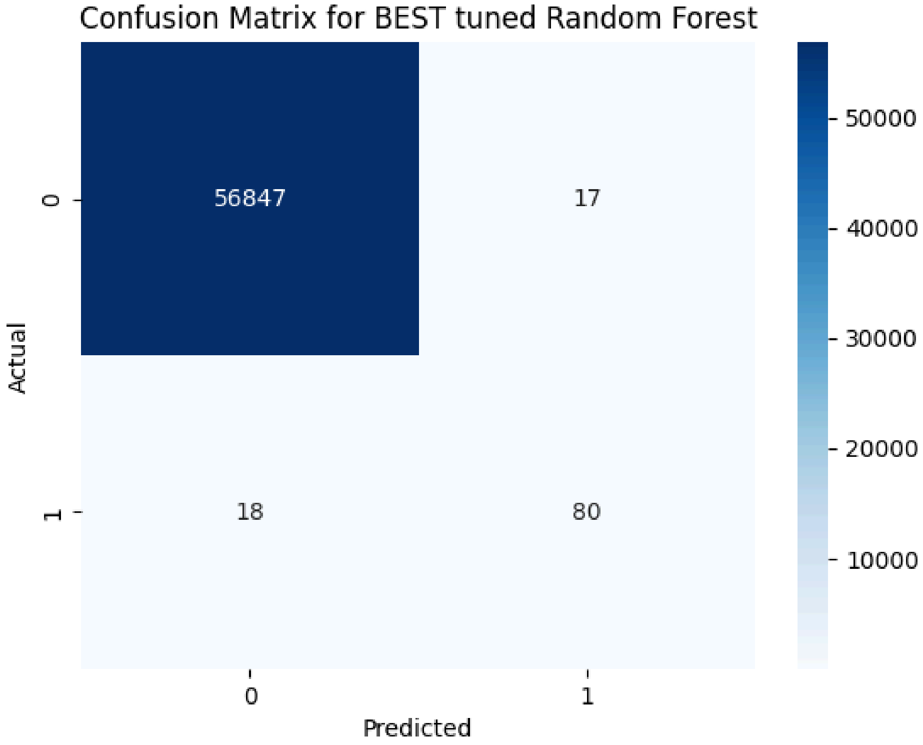


Figure 11. *Confusion matrix for the tuned Random Forest model (GridSearchCV) showing the lowest false negative rate among all models.*

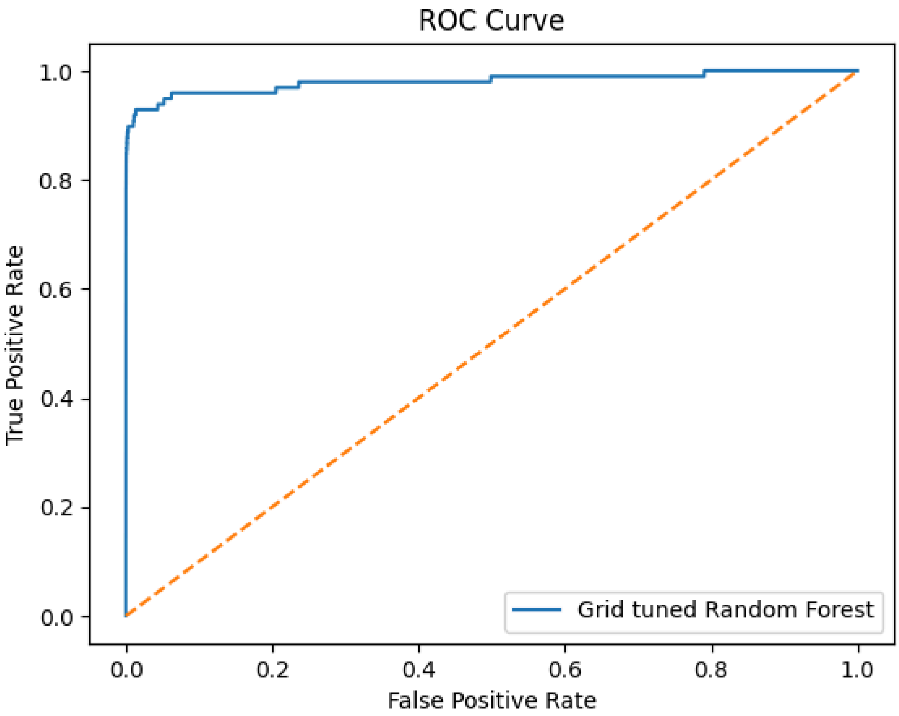


Figure 12. *ROC curve for the tuned Random Forest model with the highest area under the curve (AUC), indicating superior classification performance.*

4.3. Model Comparison

To summarize model performance, evaluation metrics were compiled into a comparative table generated using the custom evaluation functions.

The results DataFrame reported precision, recall, F1-score, and ROC-AUC for each model, while the avg_results DataFrame showed the average precision scores. The tuned Random Forest achieved the highest recall and ROC-AUC, followed closely by the Random Forest baseline. Logistic Regression with SMOTE showed improved recall compared to the standard Logistic Regression, demonstrating the effectiveness of oversampling in handling imbalance.

4.4. Precision–Recall Analysis

Precision–Recall curves were plotted for each model (**Figures 13–16**) to visualize the trade-off between detecting frauds and avoiding false alarms. The tuned Random Forest displayed the best curve, maintaining high precision even at increasing recall levels. This confirmed that the optimized model was not only accurate but also reliable when identifying rare fraud cases.

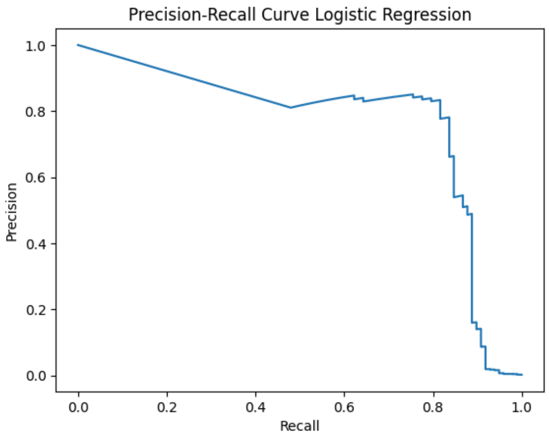
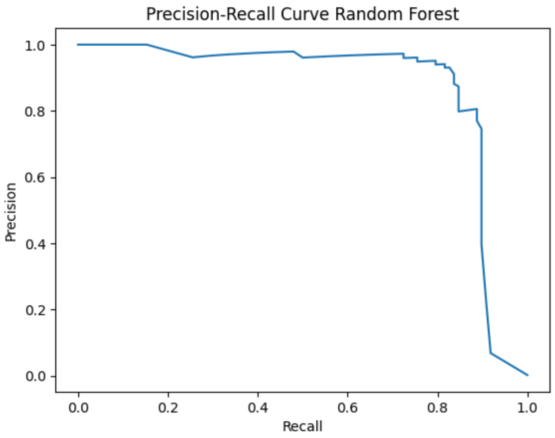


Figure 13. *Precision–Recall curve for Logistic Regression showing trade-offs between identifying frauds and false alarms*



.

Figure 14. *Precision–Recall curve for the baseline Random Forest model demonstrating higher recall than Logistic Regression.*

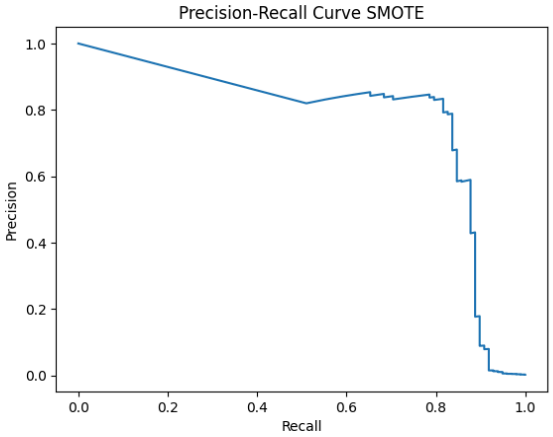


Figure 15. *Precision–Recall curve for Logistic Regression with SMOTE showing improvement in recall due to oversampling.*

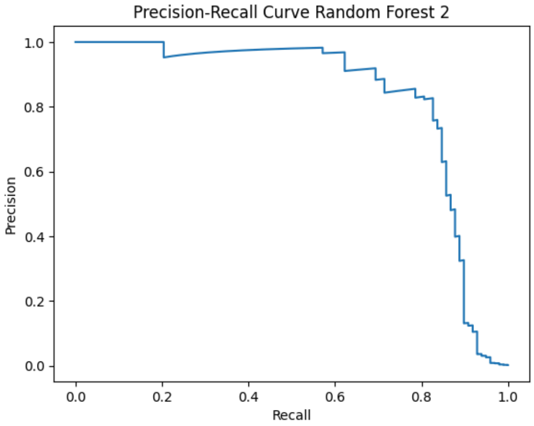


Figure 16. *Precision–Recall curve for the tuned Random Forest model showing the strongest balance between precision and recall.*

4.5. Feature Importance

Feature importance visualizations (Figures 17–18) from both Random Forest models revealed that certain PCA components particularly V14, V10, and V4 contributed most to fraud detection. These features carried strong predictive power, indicating that the PCA transformation successfully preserved meaningful information even without original feature names.

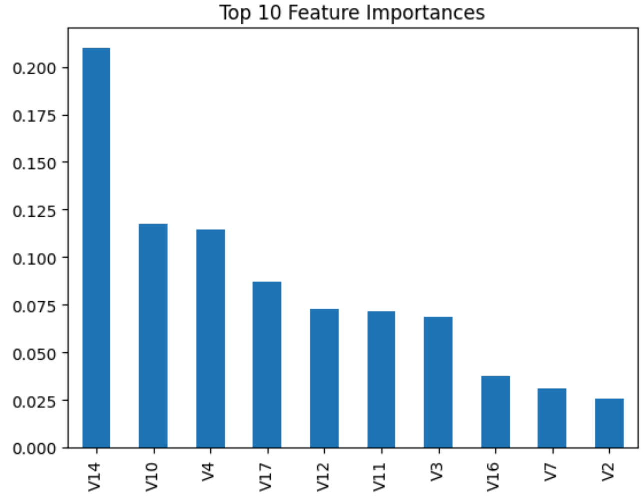


Figure 17. *Feature importance plot for the baseline Random Forest model highlighting the top ten most influential PCA features.*

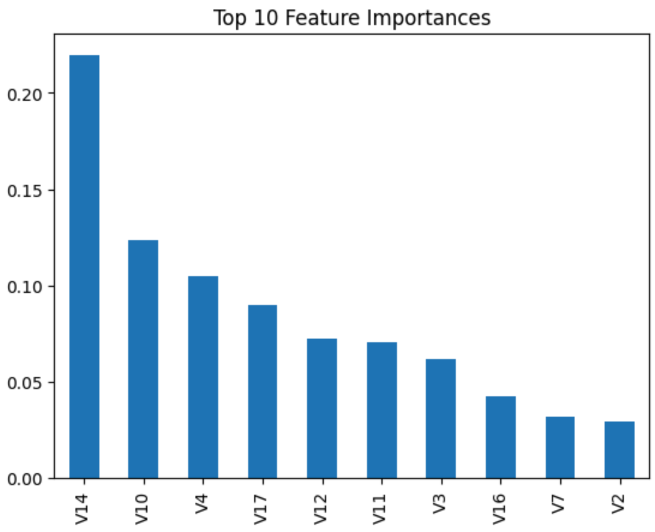


Figure 18. *Feature importance plot for the tuned Random Forest model showing that V14, V10, and V4 are the most predictive components.*

4.6. Discussion of Findings

The analysis demonstrated that ensemble methods like Random Forest outperform linear models in detecting credit card fraud due to their ability to capture complex feature interactions. The use of **SMOTE** improved sensitivity to minority class patterns, and **GridSearchCV tuning** further enhanced model performance.

The study also showed that while Logistic Regression is fast and interpretable, it is less effective for highly non-linear and imbalanced datasets. In contrast, the tuned Random Forest achieved the best trade-off between precision and recall, making it the most practical choice for real-world fraud detection systems where both accuracy and reliability are essential.

5. Conclusions

This study applied data science and machine learning techniques to detect fraudulent credit card transactions using the Kaggle Credit Card Fraud Detection dataset. Through exploratory data analysis, feature scaling, and the use of machine learning models, the project demonstrated how data-driven methods can help identify hidden patterns that distinguish fraud from legitimate transactions.

Among all models tested, **Random Forest**, particularly after **hyperparameter tuning** with **GridSearchCV**, achieved the best performance. It provided the highest recall and **ROC-AUC** values, meaning it detected the most fraud cases while maintaining good precision. The use of SMOTE improved the recall of Logistic Regression, confirming that data balancing techniques can enhance sensitivity to rare fraud cases.

Feature importance analysis showed that the PCA components V14, V10, and V4 were the most influential features in detecting fraudulent behavior. These components likely represent patterns in transaction behavior that strongly differentiate fraudulent activity from normal usage.

Overall, the results demonstrate that ensemble-based models like Random Forest outperform linear models in handling imbalanced and complex financial data. The approach presented in this project can support real-world fraud detection systems, providing a strong foundation for further development. Future research could explore deep learning, Gradient Boosting, or real-time detection frameworks that adapt dynamically to new fraud strategies.

The complete implementation of this study, including all code, data visualizations, and model comparisons, is available on GitHub:
https://github.com/Emir0Bekrija/Exploratory-Data-Analysis-EDA-of-Credit-Card-Fraud-Detection.git

The complete implementation of this study, including all code, data visualization, and model comparison is available on this GitHub link.

6. References

Aman. (2021). Credit Card Fraud Detection using Machine Learning and Data Science. *International Journal for Research in Applied Science and Engineering Technology*, 9(5).

Khan, S., Sanovar, S., Kumar, S., & Kumar, H. (2021). Credit Card Fraud Detection Using Machine Learning. *International Journal of Scientific and Research Publications (IJSRP)*, 11(6), 1–6.

Nayak, N. A., Suchika, C., Sandhya, N., Lakshmi, M., & Roja, J. (2023). Credit Card Fraud Detection using Machine Learning. *International Journal of Advanced Research in Science, Communication and Technology*, 3(2), 45–52.

Yeruva, S., Harshitha, M. S., Kavya, M., Deepa Sree, M. S., & Sahithi, T. S. (2023). Credit Card Fraud Detection using Machine Learning. *International Journal of Engineering and Advanced Technology*, 12(4), 1–8.

Khedkar, D. S., & Gupta, B. (2024). Credit Card Fraud Detection using Machine Learning. *International Journal of Advanced Research in Science, Communication and Technology*, 4(1), 90–98.

Naik, S. A., & Pise, N. (2022). Credit Card Fraud Detection using Machine Learning. *International Research Journal of Modernization in Engineering Technology and Science*, 4(2), 25–33.

Arivanantham, T. (2025). Credit Card Fraud Detection Using Machine Learning. *International Journal for Research in Applied Science and Engineering Technology*, 13(1), 34–40.

Abdou, H. E. M., Khalifa, W., Roushdy, M., & Salem, A. (2020). Machine Learning Techniques for Credit Card Fraud Detection. *Future Computing and Informatics Journal*, 4(2), 45–55.

Gupta, A., Pant, B., Mehra, N., & Kapil, D. (2020). Machine Learning for Detecting Credit Card Frauds. *International Journal of Recent Technology and Engineering*, 8(2S12), 120–125.

Powar, R., & Dawkhar, P. R. (2025). Credit Card Fraud Detection Using Machine Learning. *International Journal of Advanced Research in Science, Communication and Technology*, 6(1), 1–8.