Credit Card Fraud Detection

Team Members

- Dhathri Bathini
- Vijaya Deepika Buddhiraju
- Rehonoma Hasan Jahin
- Sathvik Maridasana Nagaraj

Table of Content

1. Summary

The report presents a comprehensive analysis of a credit card fraud detection system using a dataset of transactions from European cardholders in September 2013. The dataset, highly imbalanced, contains 492 frauds out of 284,807 transactions. Various machine learning models were trained and evaluated, including Support Vector Machine (SVM), Decision Tree, Logistic Regression, and K-Nearest Neighbors (KNN), with and without the use of Synthetic Minority Over-sampling Technique (SMOTE) to address the class imbalance.

1.1. Purpose

The primary purpose of this project is to develop a reliable predictive model that can efficiently identify fraudulent transactions in real-time. This is crucial for preventing financial losses for both the cardholders and the issuing institution, and for maintaining consumer trust. Additionally, this study aims to evaluate the effectiveness of various classification algorithms and the impact of SMOTE on model performance, providing insights into the best practices for handling imbalanced datasets in fraud detection.

1.2. Key Finding

During the data preprocessing phase, it became apparent that there was a significant imbalance in the dataset, presenting a risk of overfitting and bias favoring the majority class. To counteract this, we employed the Synthetic Minority Over-sampling Technique (SMOTE) to balance the data. However, upon application of SMOTE, we observed that the four machine learning models—SVM, Decision Tree, Logistic Regression, and KNN—actually demonstrated improved precision, recall, and F1 scores when SMOTE was not used. These outcomes underscore the complexity of dealing with imbalanced data and suggest that alternative strategies may be necessary to enhance model performance in this context.

2. Introduction

2.1. Background

Credit card fraud is a pressing issue in the financial sector, leading to significant losses annually for both consumers and financial institutions. With the rise of digital transactions, the volume and complexity of fraudulent activities have also increased. The need for robust, scalable, and efficient fraud detection systems is more critical than ever. Machine learning offers promising solutions by automating the detection process and identifying fraudulent transactions with high accuracy.

2.2. Problem statement

Despite advancements in fraud detection technologies, the dynamic nature of fraud, coupled with the imbalanced nature of transaction datasets where fraudulent transactions are significantly outnumbered by legitimate ones, poses a substantial challenge. Traditional fraud detection systems often suffer from high rates of false positives and false negatives, which can lead to customer dissatisfaction and operational inefficiencies.

2.3. Objectives

The primary goal is to construct and appraise a range of machine learning models capable of detecting fraudulent transactions within a highly imbalanced dataset. A key aspect of this endeavor involves optimizing model performance by applying and evaluating the Synthetic Minority Over-sampling Technique (SMOTE) as a strategy to mitigate the challenges of imbalance and enhance model sensitivity. Additionally, there's a focus on a thorough comparison of these models' performances, with a special emphasis on metrics such as accuracy, precision, recall, and F1-score, to identify the most suitable model for real-time fraud detection applications.

2.4. Limitations

This project encounters a constraint due to the imbalance present in the dataset, which raises the potential for overfitting and inherent biases. Moreover, there is some uncertainty regarding the efficacy of SMOTE in achieving data balance, adding to the challenges faced in the analysis.

3. Data Description

3.1. Source

The dataset comprises transactions from European cardholders in September 2013, provided by a collaboration between Worldline and the Machine Learning Group at Université Libre de Bruxelles. It includes 284,807 transactions over two days, with 492 frauds, highlighting a real-world, highly imbalanced scenario as can be seen in the Fig-1.

The dataset has V1 to V28 variables which are generated after PCA and are not disclosed for the sake of data confidentiality. Fig-2 shows the correlation between the variables.

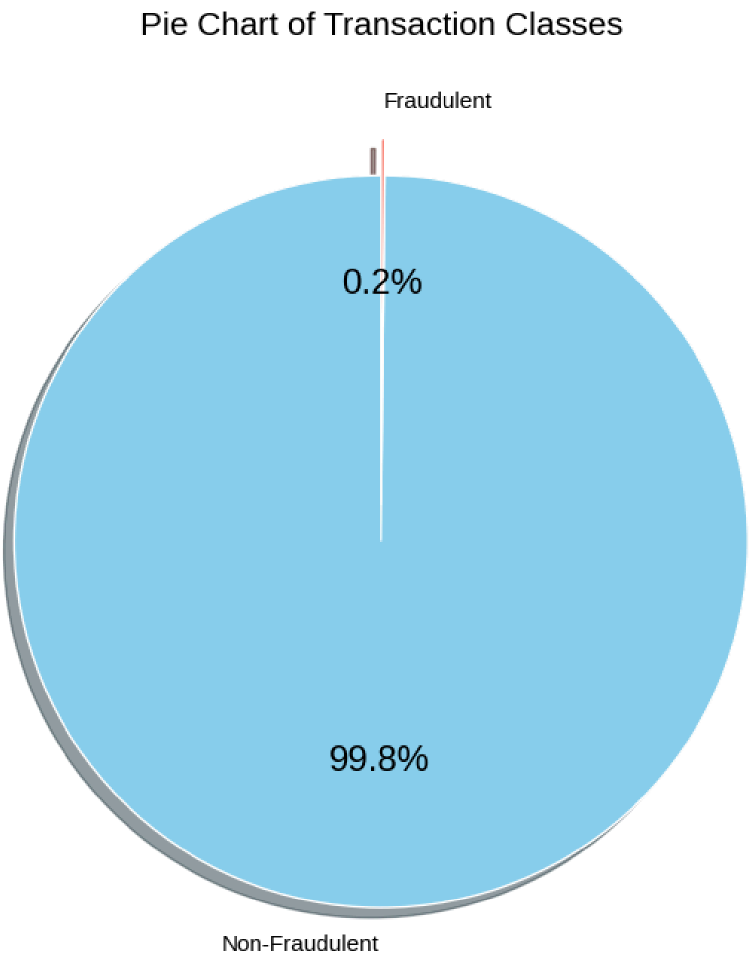

Fig-1: Transaction Class Distribution

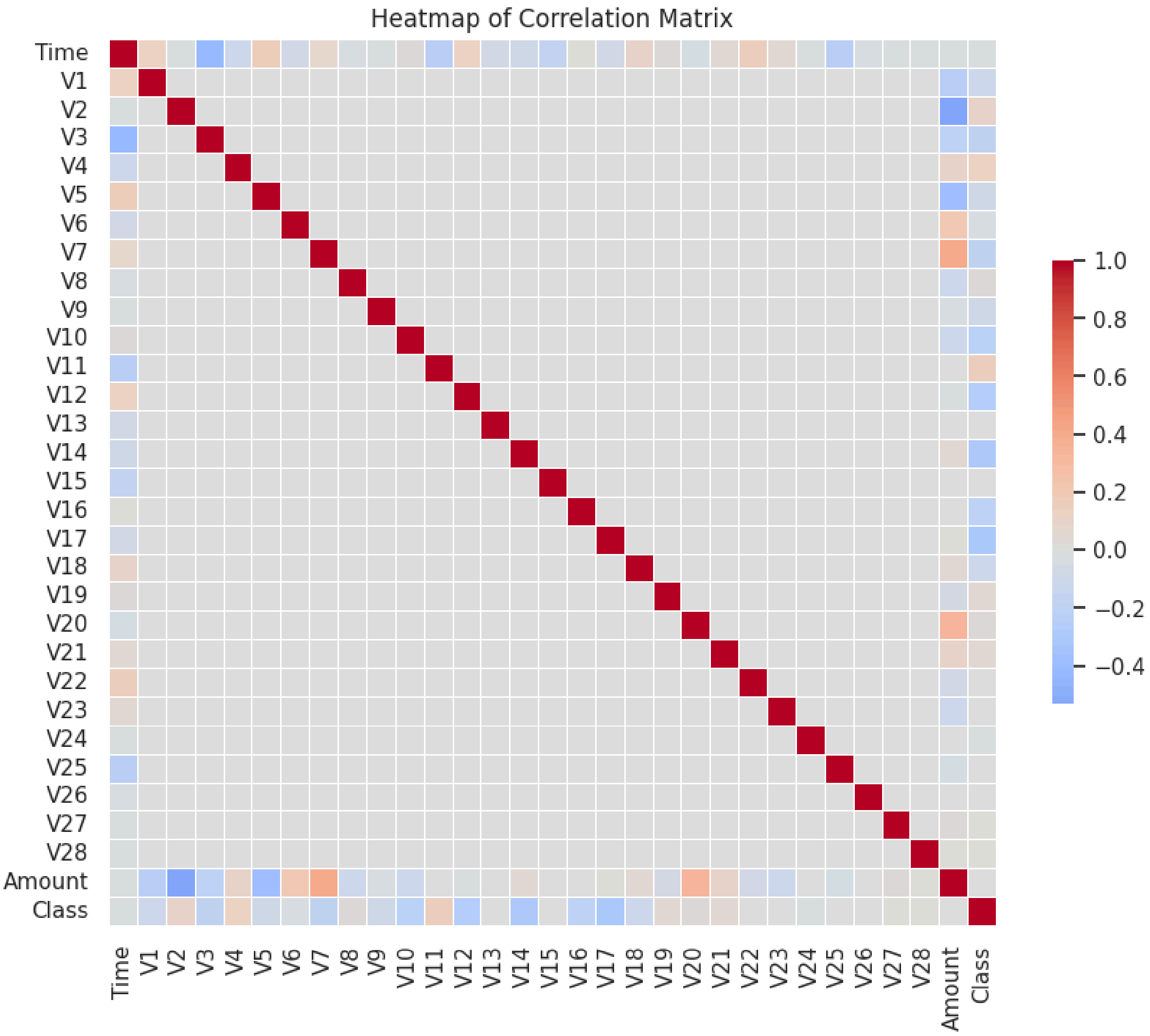

Fig-2: Correlation Heatmap of Variables

3.2. Preprocessing

- **Feature Transformation:** Excluding 'Time' and 'Amount', all features were anonymized through Principal Component Analysis, yielding 28 principal components to maintain data confidentiality (Availed via original dataset).
- **Normalization:** The 'Amount' feature was standardized using the StandardScaler to normalize the data distribution, aiming to enhance the models' predictive capabilities.
- **Handling Imbalanced Data:** Given the dataset's imbalance, with a disproportionate number of legitimate transactions to frauds, SMOTE was applied to the training data to oversample the minority class, thus creating a more balanced dataset for model training.

- **Data Splitting:** The dataset was divided in a 70/30 train-test ratio, ensuring a stratified split that maintains a consistent proportion of fraud cases in both subsets.
- **Data Integrity Check:** The absence of null values in the dataset was confirmed, indicating no requirement for missing data handling.

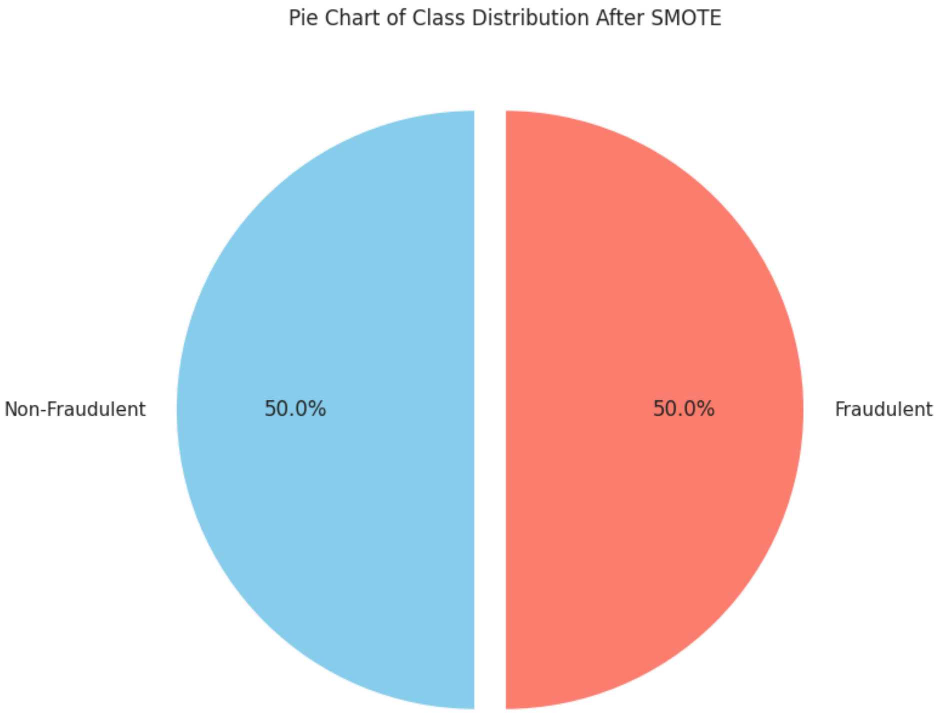

Fig-3: Transaction Class Distribution After SMOTE

4. Methodology

4.1. Model selection

The project evaluates several machine learning models to determine their efficacy in detecting fraudulent transactions:

- **Support Vector Machine (SVM):** Chosen for its effectiveness in high-dimensional spaces and its ability to handle non-linear data separations using kernel tricks.
- **Decision Tree Classifier:** Selected for its interpretability and ease of use in binary classification tasks.
- **Logistic Regression:** Utilized for its efficiency and simplicity in binary classification problems.
- **K-Nearest Neighbors (KNN):** Considered due to its straightforward implementation and strong performance in classification by examining the closest neighboring points.

4.2. Training process

Each model underwent the following training process:

- **Data Splitting:** The dataset was split into training (70%) and testing (30%) sets, ensuring stratification to preserve the imbalance ratio.
- **Preprocessing:** Features were scaled using StandardScaler to normalize the data, particularly the 'Amount' feature.
- **Handling Imbalanced Data:** SMOTE was applied to the training data to enhance the representation of the minority class (frauds), aiming to improve model sensitivity to fraudulent transactions.
- **Model Fitting:** Models were trained on both the original and SMOTE-enhanced datasets to compare their performance under different conditions.
- **Parameter Tuning:** For models like SVM and KNN, parameters such as kernel type and the number of neighbors were tuned to optimize performance.

4.3. Tools used

- **Python Libraries**: Pandas for data manipulation, numpy for numerical operations, matplotlib and seaborn for visualizations.
- **Scikit-learn:** Utilized for model building, preprocessing, and evaluation tools.
- **SMOTE from imbalanced-learn**: Used to address the class imbalance problem by oversampling the minority class in the training set

5. Model Description

5.1. Support Vector Machine (SVM)

5.1.1. SVM with SMOTE:

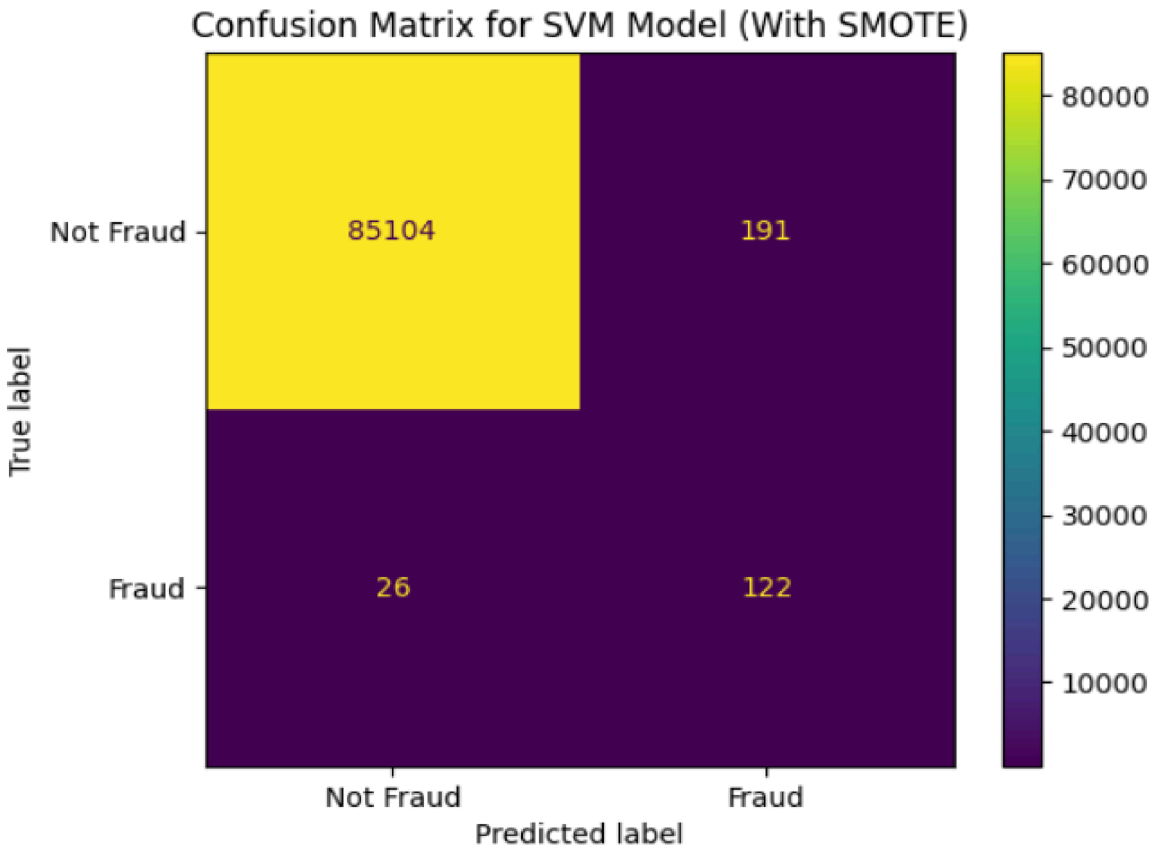

Fig-4: Confusion Matrix for SVM model(with SMOTE)

The SVM model with a third-degree polynomial kernel, when enhanced with SMOTE, illustrates high classification accuracy at approximately 99.76% in identifying fraudulent transactions. However, the Confusion Matrix for this model presents a nuanced picture: while it correctly identified 85,104 non-fraudulent transactions, it also incorrectly flagged 191 legitimate transactions as fraudulent (false positives). On the other hand, it successfully detected 122 out of 148 fraudulent transactions (true positives), missing 26 fraud cases (false negatives). This performance results in a modest precision of about 0.39, indicating a somewhat high rate of false positives, which is balanced against a robust recall of approximately 0.82, reflecting the model's ability to catch a high number of fraud cases. The F1-score of 0.53 suggests there's room for improvement in achieving a better balance between precision and recall. The Confusion Matrix underscores the trade-offs made and points towards potential refinement areas, particularly in minimizing the false positives without significantly sacrificing the model's ability to detect fraud.

5.1.2. SVM without SMOTE

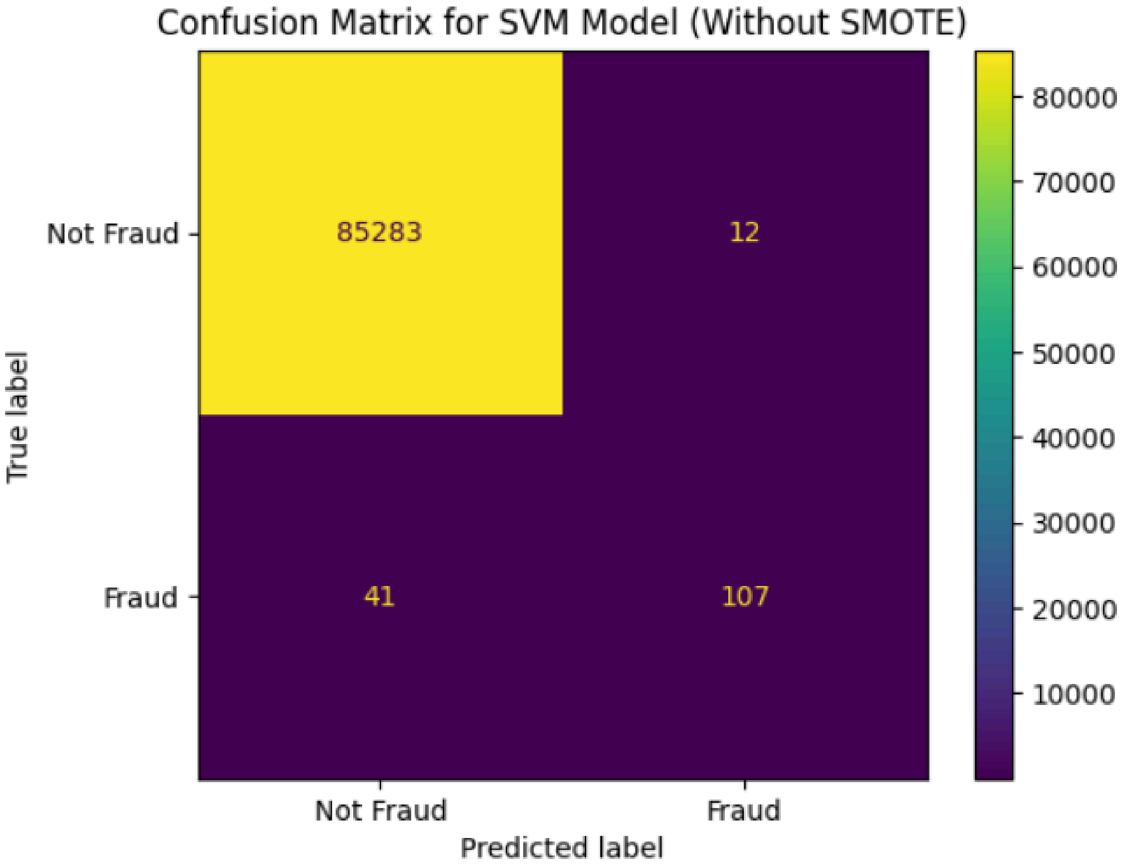

Fig-5: Confusion Matrix for SVM model(without SMOTE)

The SVM model crafted with a degree-3 polynomial kernel, in the absence of SMOTE to adjust for data imbalances, has achieved an impressive accuracy rate of nearly 99.94% in detecting fraudulent transactions. The model boasts a high precision of about 0.90, markedly reducing the misclassification of legitimate transactions as fraudulent when compared to its SMOTE-enhanced counterpart. The recall, at roughly 0.72, indicates the model's slightly reduced capacity to catch all fraudulent cases, but it remains effective. The F1-score stands at 0.80, denoting a well-calibrated balance between precision and recall and suggesting the model's adeptness at accurately flagging fraud. The Confusion Matrix reveals a compelling narrative: the model precisely identifies the vast majority of non-fraudulent transactions, evidenced by the high true negative count, while maintaining a commendable true positive rate. Despite fewer false positives, the slightly higher false negatives highlight a trade-off made by not using SMOTE. Overall, the SVM without SMOTE showcases robust performance metrics, affirming its utility in practical scenarios where the preservation of natural class distributions is crucial.

5.1.3. Comparison of SVM with & without SMOTE

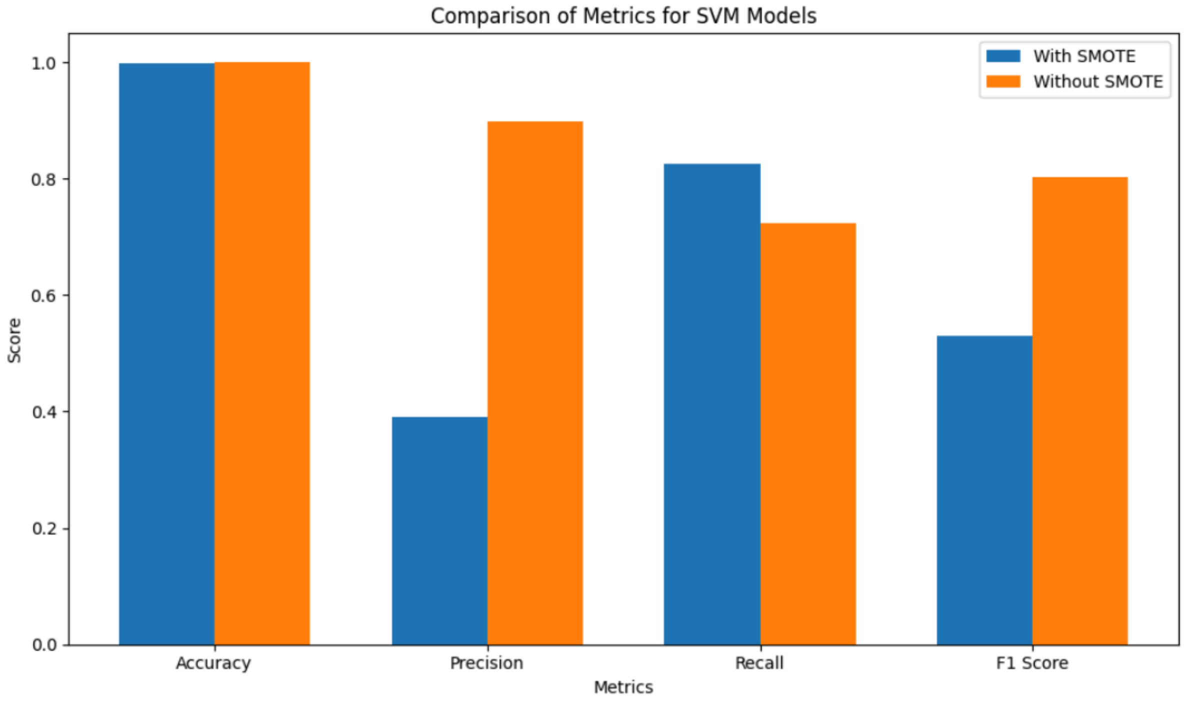

Fig-6: Comparison of Metrics for SVM models

When comparing the performance of SVM models, the one utilizing SMOTE exhibits an accuracy of about 99.76%, a relatively modest precision of 0.39 indicative of more frequent false positives, and a strong recall of 0.82, denoting its effectiveness in identifying fraudulent transactions. Its F1-score, at 0.53, highlights a need for a better balance between precision and recall. On the other hand, the SVM model without SMOTE boasts a higher accuracy of nearly 99.94%, significantly improved precision at 0.90, which suggests fewer legitimate transactions are misclassified as fraudulent, but a lower recall of 0.72, pointing to potentially more missed fraud cases. The non-SMOTE model's F1-score of 0.80 indicates a more effective balance of precision and recall, implying a more reliable performance in scenarios where false positives are a greater concern. The bar chart visual comparison of these models clearly reflects these trade-offs, with the non-SMOTE model showing advantages in precision and F1-score, while the SMOTE model is superior in terms of recall, underscoring the importance of context in selecting the appropriate model for fraud detection tasks.

5.2. Decision Tree

5.2.1. Decision Tree with SMOTE

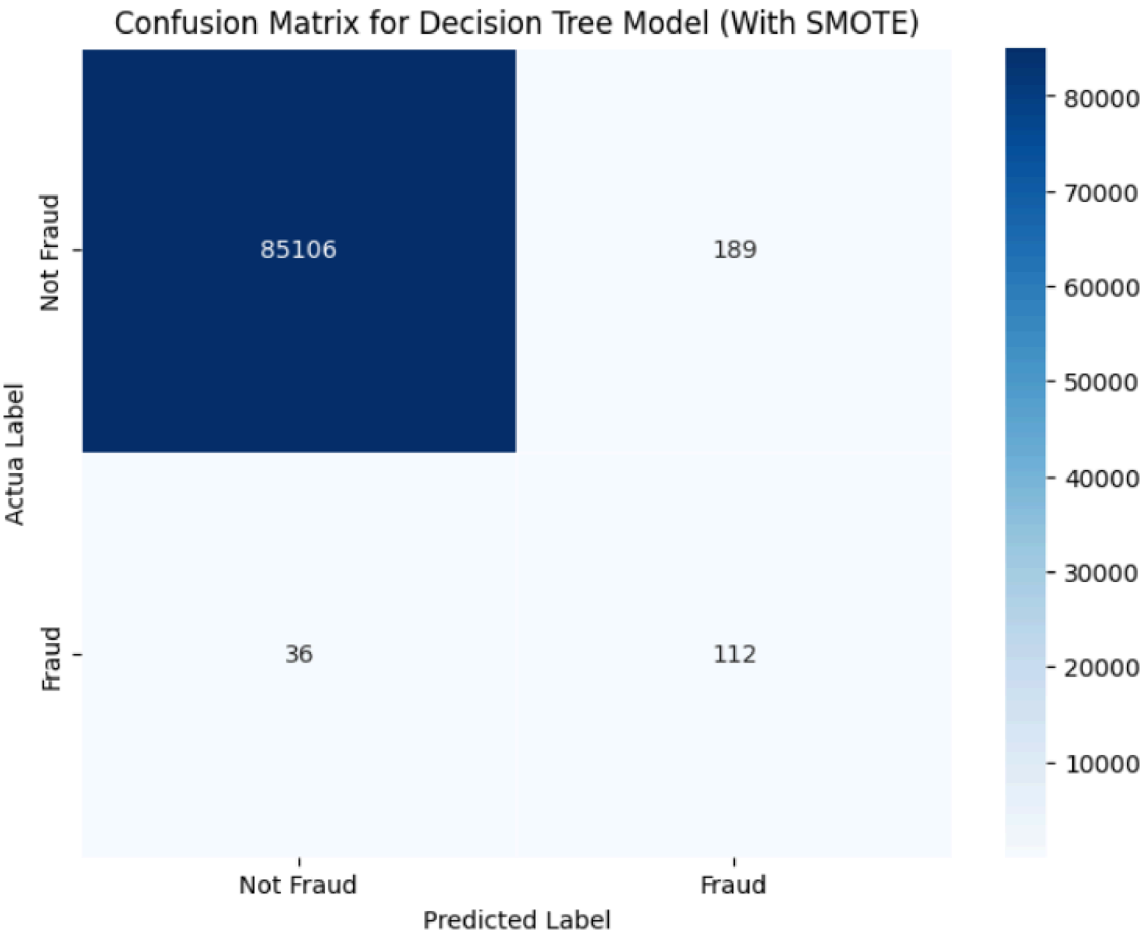

Fig-7: Confusion Matrix for Decision Tree model(with SMOTE)

The Decision Tree model trained with SMOTE to mitigate class imbalance demonstrates a commendable accuracy of around 99.73%, with a recall of 0.76, signifying its efficacy in identifying a considerable number of fraudulent transactions. However, the model's precision at 0.37 suggests a tendency to misclassify legitimate transactions as fraudulent, a limitation reflected in an F1-score of 0.50 that indicates room for improvement in balancing precision and recall. The Confusion Matrix further elucidates the model's performance trade-offs, revealing that while it accurately predicts 85,106 true negatives, it also produces 189 false positives and fails to catch 36 fraud cases, showing its capability to detect fraud effectively but with a notable rate of false alarms—a critical factor to weigh for its practical deployment in fraud detection systems.

5.2.2. Decision Tree without SMOTE

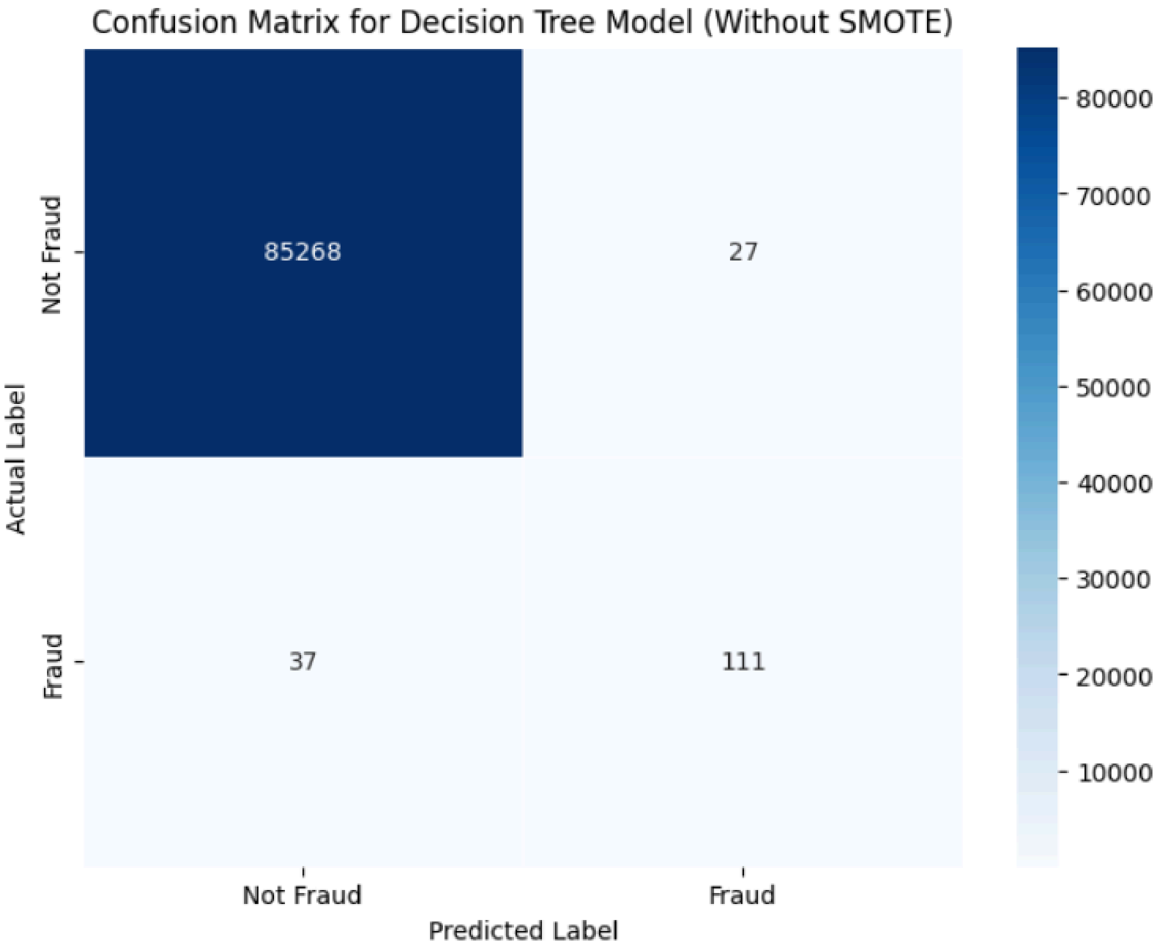

Fig-8: Confusion Matrix for Decision Tree model(without SMOTE)

The non-SMOTE Decision Tree model achieves an impressive 99.92% accuracy, with a precision rate of 0.80 indicating fewer instances of false positives, and a recall rate of 0.75, showing it captures a substantial portion of fraudulent transactions. With an F1-score of 0.76, this model demonstrates a strong balance between precision and recall. The Confusion Matrix reveals that it correctly predicted 85,268 true negatives and accurately identified 111 out of 148 fraud cases, with only 27 false positives and 37 missed fraud cases, which underscores its effectiveness in discerning fraudulent transactions while maintaining a lower rate of incorrectly flagging legitimate activity—a vital aspect for real-world applications.

5.2.3. Comparison of Decision Tree with & without SMOTE

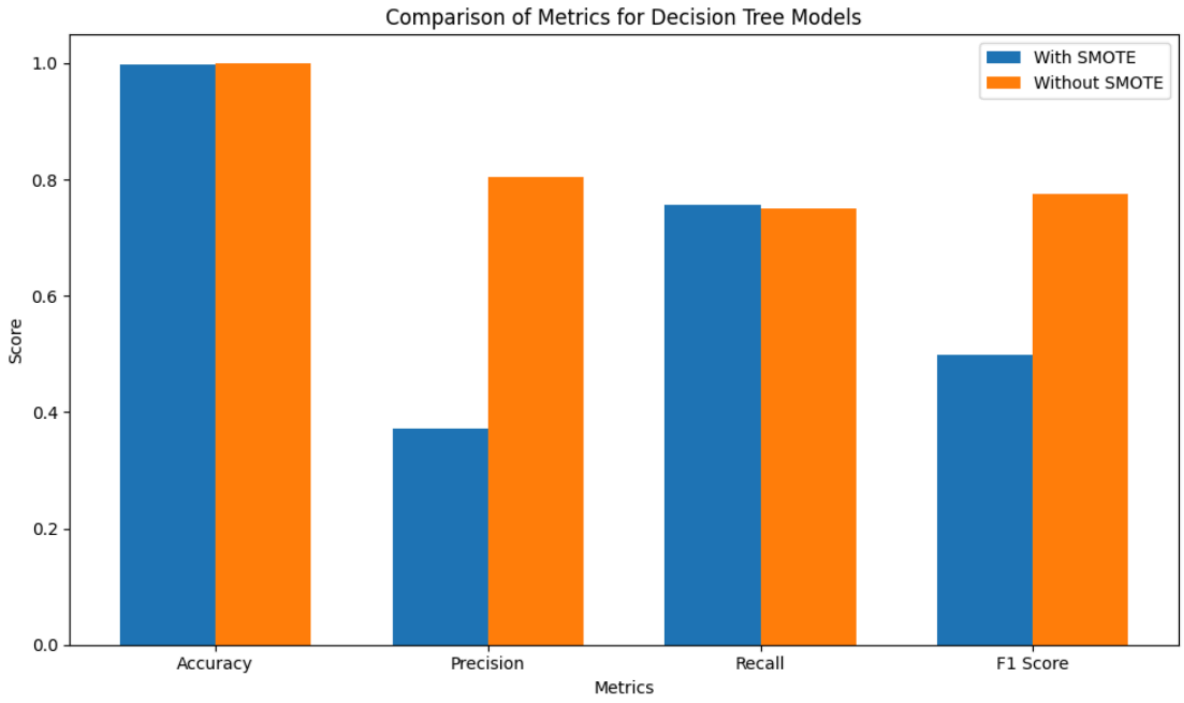

Fig-9: Comparison of Metrics for Decision Tree models

The Decision Tree models, when analyzed for their performance with and without SMOTE, present a contrasting picture. The model with SMOTE achieves an accuracy of 99.73% with a recall of 0.76, indicating its strong detection capabilities for fraudulent transactions, but has a lower precision of 0.37, suggesting a higher rate of false positives. The F1-score of 0.49 reflects the need for a balance between precision and recall. In contrast, the Decision Tree without SMOTE registers an impressive accuracy of 99.92%, with a higher precision of 0.80, denoting fewer false positives, and a recall of 0.75. The F1-score improves to 0.77, indicating a more balanced model. The comparative bar chart underscores these differences, highlighting the significant increase in precision and F1-score without SMOTE, despite a slight drop in recall, showcasing the non-SMOTE model's capacity for greater overall accuracy in fraud detection while minimizing incorrect fraud alerts.

5.3. Logistic Regression

5.3.1. Logistic Regression with SMOTE

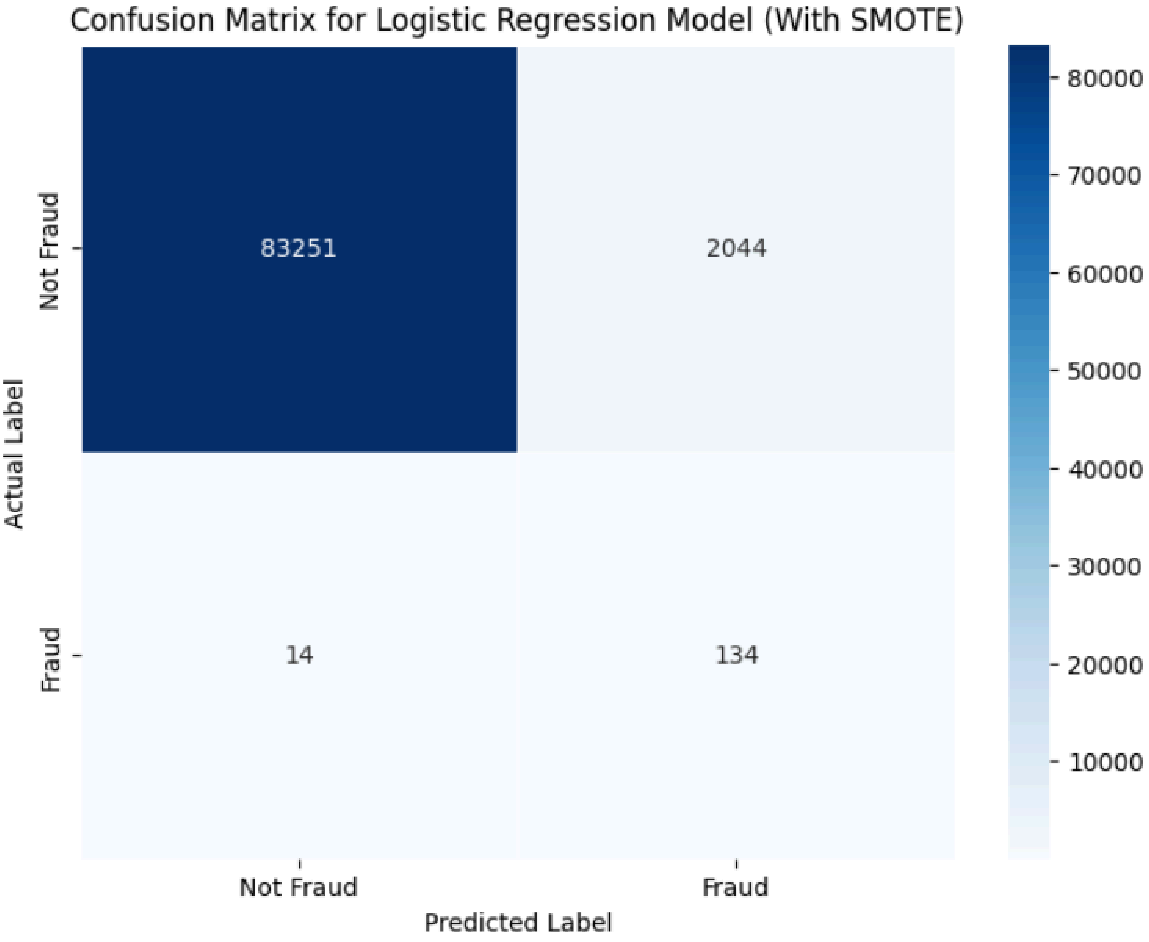

Fig-10: Confusion Matrix for Logistic Regression model(with SMOTE)

The Logistic Regression model with SMOTE demonstrates an accuracy of roughly 79.51%, indicating a strong capability to correctly classify transactions. It has a high recall of 0.90, showing effectiveness in identifying the majority of fraudulent transactions. However, its precision is low at 0.065, suggesting a high rate of false positives, where many legitimate transactions are incorrectly labeled as fraud. The F1 Score at 0.12 further reflects this imbalance between precision and recall. The confusion matrix visualizes these outcomes, showing the model correctly identified 83,251 non-fraudulent transactions but also incorrectly flagged 2,044 as fraud. For actual fraudulent transactions, it correctly identified 134, missing only 14. This performance points to a model with a strong ability to detect fraud but with a significant cost of misclassifying many legitimate transactions, which could lead to operational inefficiencies if used in a practical setting without further adjustment to its predictive thresholds.

5.3.2. Logistic Regression without SMOTE

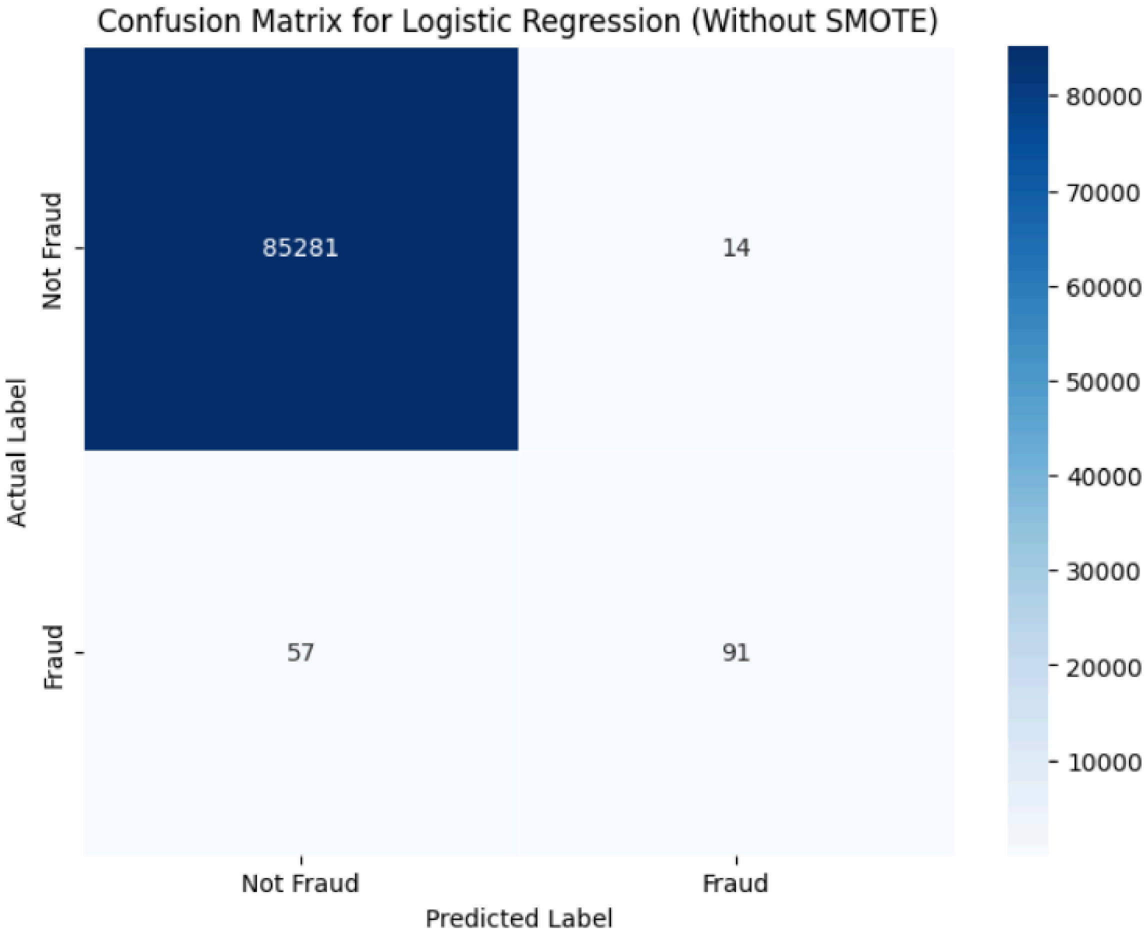

Fig-11: Confusion Matrix for Logistic Regression model(without SMOTE)

The Logistic Regression model without SMOTE achieves a high accuracy of approximately 99.92%, with a precision of 0.87, indicating it is less likely to falsely label legitimate transactions as fraud. The recall, however, is lower at 0.61, suggesting it may miss some fraudulent transactions. The F1-score of 0.72 shows a decent balance between precision and recall, signifying a well-rounded approach to classification. The Confusion Matrix reveals that the model correctly identified 85,281 true negatives but mistakenly flagged 14 legitimate transactions as fraud. It detected 91 out of 148 fraud cases, leaving 57 undetected. Despite the lower recall, the model's high precision may make it a more suitable choice in scenarios where the cost of false positives is high, such as in financial applications where customer trust is paramount.

5.3.3. Comparison of Logistic Regression with & without SMOTE

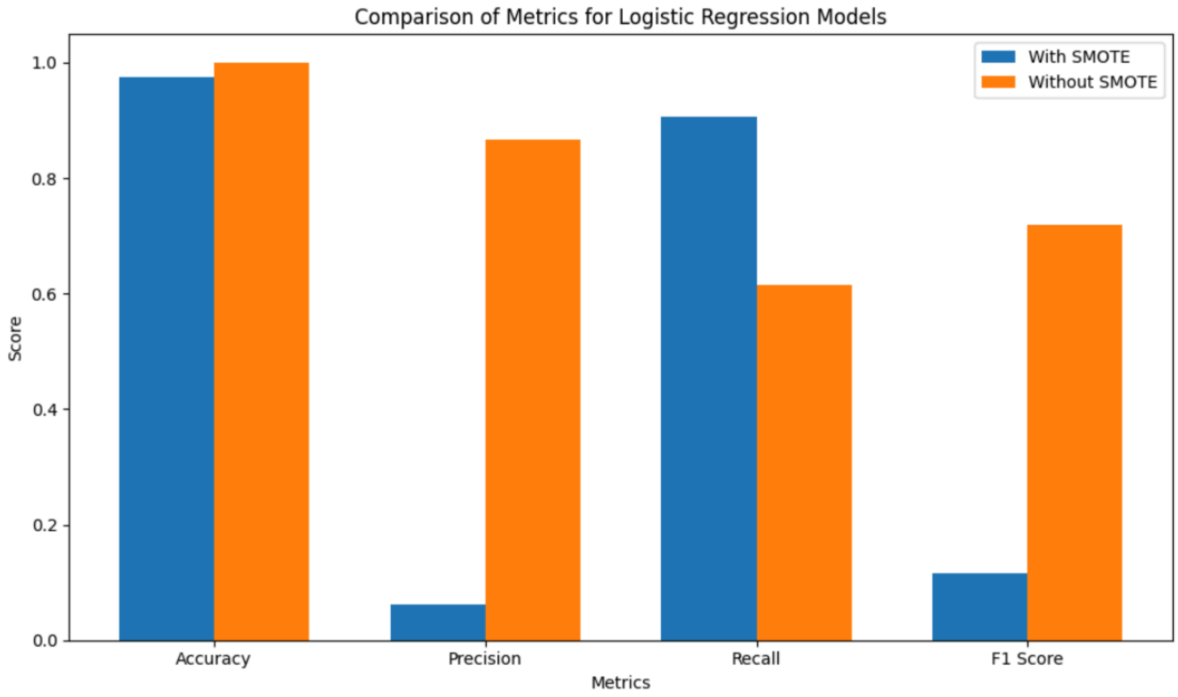

Fig-12: Comparison of Metrics for Logistic Regression models

When comparing the performance of Logistic Regression models with and without SMOTE, we see a notable difference in the precision-recall balance. The model with SMOTE exhibits lower precision, indicative of a tendency to misclassify legitimate transactions as fraud, but it compensates with a high recall, meaning it captures a larger number of actual fraudulent transactions. In contrast, the model without SMOTE achieves higher precision, reducing false positives, yet it does so at the expense of recall, missing more fraudulent transactions. The F1-scores of the two models reflect this trade-off, with the non-SMOTE model achieving a better balance between precision and recall, as evidenced by its higher score. These differences are critical to consider depending on the cost of false positives versus false negatives in a given application: the model without SMOTE may be preferable in scenarios where falsely flagging legitimate transactions is particularly undesirable.

5.4. K-Nearest Neighbour (KNN)

5.4.1. KNN with SMOTE

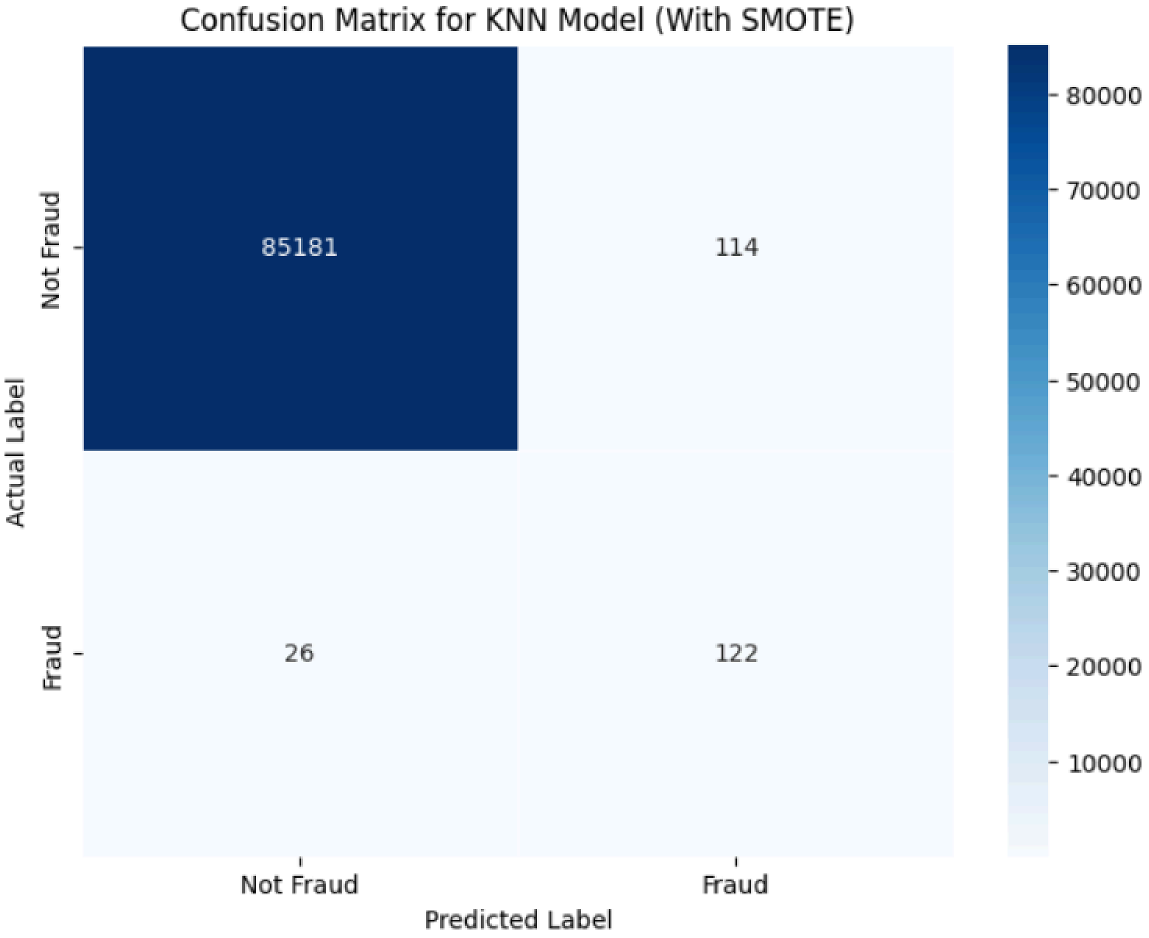

Fig-13: Confusion Matrix for KNN model(with SMOTE)

The KNN model with SMOTE boasts a high accuracy of 99.84%, signifying it correctly classifies transactions most of the time; however, the precision is moderately low at 51.69%, indicating that when fraud is predicted, it is accurate just over half the time. The model excels in recall at 82.43%, showcasing a strong capability to catch actual fraud instances, and an F1 score of 63.54% strikes a balance between precision and recall, reflecting a moderate trade-off. The accompanying confusion matrix further illuminates this performance: it records 85,181 true negatives, confirming its robustness in identifying non-fraudulent transactions, alongside 122 true positives, affirming its effectiveness at detecting fraud. On the downside, there are 114 false positives, where legitimate transactions are wrongly flagged as fraud, and 26 false negatives, where fraudulent transactions slip through undetected, underlining the challenges in precision that the model faces.

5.4.2. KNN without SMOTE

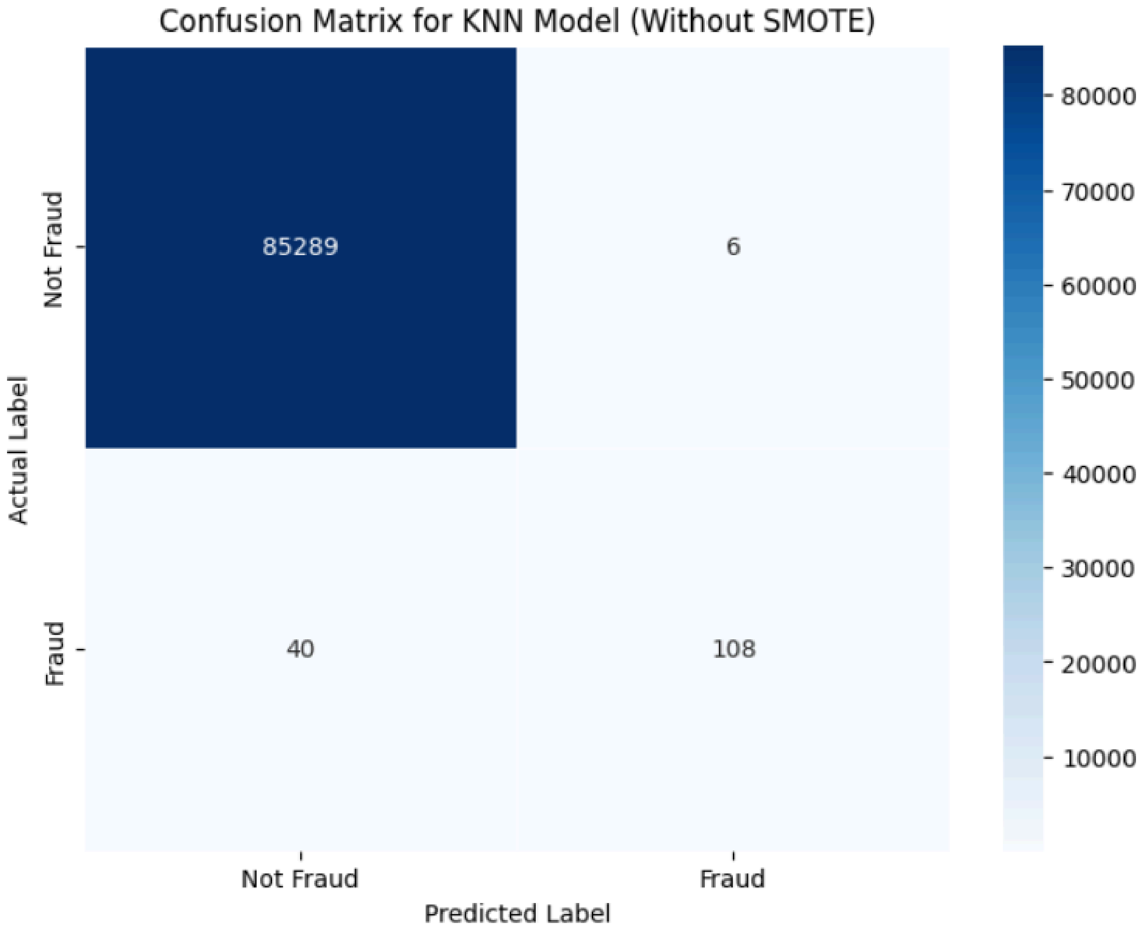

Fig-14: Confusion Matrix for KNN model(without SMOTE)

The KNN model without SMOTE exhibits exceptional accuracy at 99.95%, implying that it correctly identifies the vast majority of transactions. It shows outstanding precision of 94.74%, indicating that when it predicts fraud, it is correct most of the time. However, its recall is relatively lower at 72.97%, pointing out that it misses some fraudulent transactions. The F1 score, a measure that balances precision and recall, stands at 82.44%, suggesting a strong overall performance. The confusion matrix shows 85,289 true negatives, which underscores its effectiveness in recognizing non-fraudulent transactions. The model identifies 108 true positives, showing capability in detecting fraudulent transactions, but with 40 false negatives, it indicates missed detections of fraud. Interestingly, there are only 6 false positives, which means there are very few cases where normal transactions are mistakenly labeled as fraud, emphasizing the model's precision.

5.4.3. Comparison of KNN with & without SMOTE

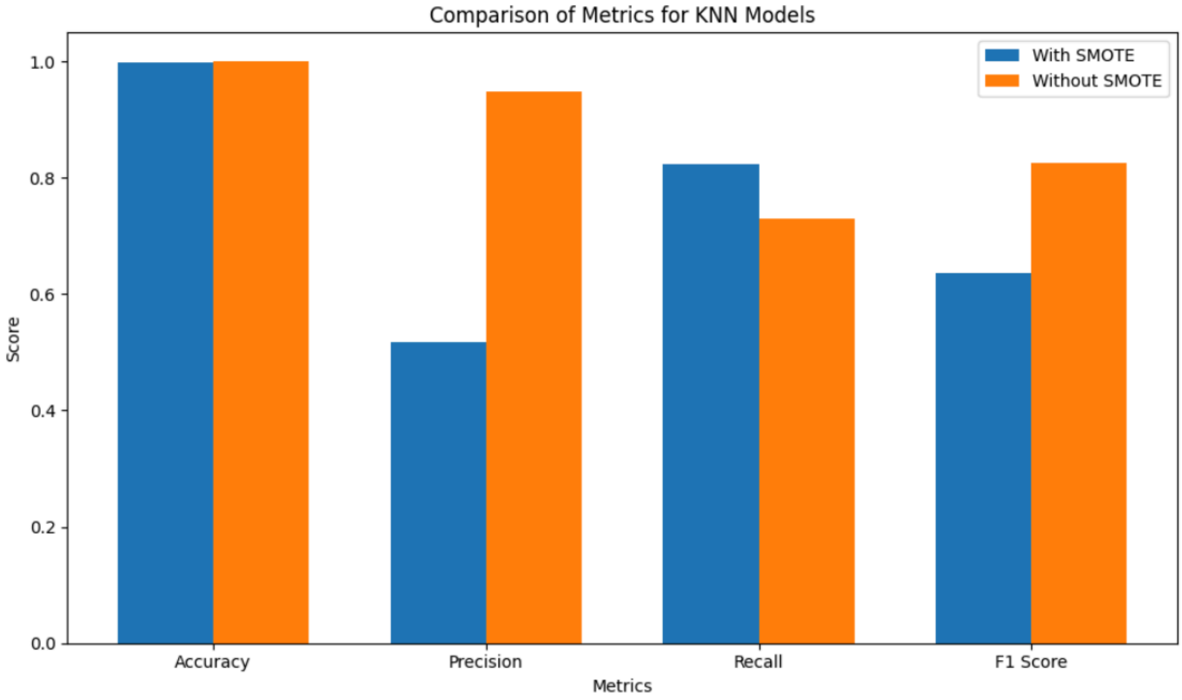

Fig-15: Comparison of Metrics for Logistic Regression models

Comparing the KNN model's performance with and without SMOTE reveals key differences in handling class imbalance for fraud detection. With SMOTE, accuracy is slightly lower at 99.84% compared to the 99.95% without SMOTE, reflecting a marginal decrease in overall correctness. Precision sees a significant drop from 94.74% without SMOTE to 51.69% with SMOTE, suggesting that the model with SMOTE is less accurate when predicting fraudulent transactions. However, recall increases from 72.97% without SMOTE to 82.43% with SMOTE, indicating that the model with SMOTE is better at identifying all positive instances, catching more actual frauds. The F1 score, which balances precision and recall, is lower with SMOTE at 63.54% compared to 82.44% without, pointing to a trade-off between the precision and recall when SMOTE is applied.

6. Results

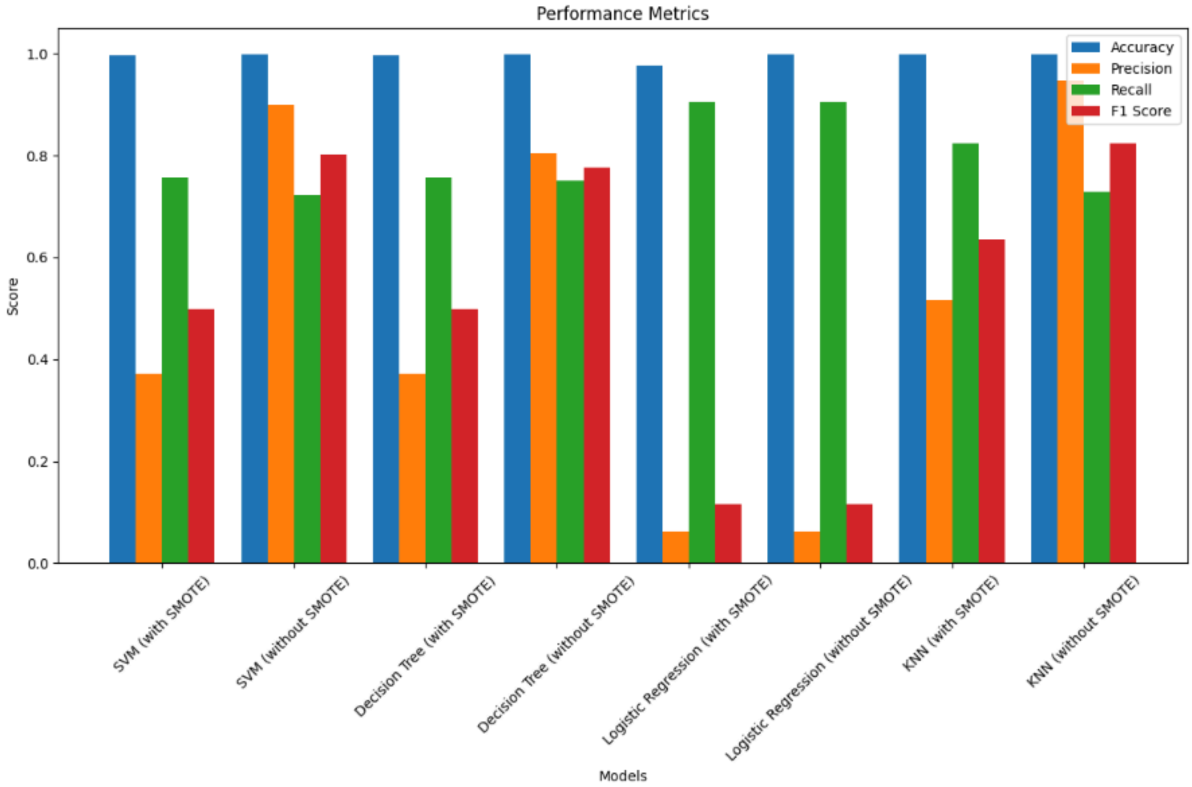

Fig-16: Comparison of Metrics for all the models

Analyzing the performance metrics for the various models reveals a consistent pattern: introducing SMOTE enhances the models' ability to identify fraudulent activities, as evidenced by the higher recall rates. This is beneficial in scenarios where failing to detect fraud has serious repercussions. Nonetheless, this advantage is offset by a decrease in precision, leading to more false positives where genuine transactions are incorrectly flagged as fraudulent. For example, while the KNN model with SMOTE significantly improves fraud detection (recall of 82.43%), it also misidentifies legitimate transactions as fraud more often, with precision dropping to 51.69% compared to 94.74% without SMOTE.

Without SMOTE, all models achieve a higher accuracy rate, exceeding 99.9%, and present better F1 scores, reflecting a more balanced performance between precision and missed fraud cases. This suggests that without SMOTE, models are more reliable in their fraud predictions but at the risk of missing some fraudulent transactions.

In essence, if the objective is to minimize the risk of fraud by catching as many cases as possible, using SMOTE may be the strategy of choice despite the higher rate of false positives. Conversely, if the priority is to maintain a high level of precision to avoid the repercussions of false alarms, then models without SMOTE offer a more balanced solution. The choice between these approaches depends on the specific needs and the acceptable risk levels of false positives in the operational context.

6.1. Final model selection

Among the evaluated models, KNN (without SMOTE) emerges as the leading choice due to its superior F1 score, indicating an effective balance between precision and recall. This model excels in minimizing false positives while reliably identifying a significant portion of fraudulent transactions. Other models like SVM (without SMOTE) and Decision Tree (without SMOTE) also demonstrate commendable performance but with slightly lower F1 scores compared to KNN.

In fraud detection systems, where false positives can cause significant disruptions and false negatives may allow fraudulent activities to go unnoticed, it's crucial to select a model that effectively manages both types of errors. The KNN (without SMOTE) model, with its high precision and robust recall, is the optimal choice based on the data analysis. This selection aims to reduce the incidence of legitimate transactions being incorrectly flagged as fraud, while still maintaining a strong detection capability for genuine fraud cases.

7. Conclusion

7.1. Summary of findings

Upon examining the effectiveness of various machine learning models in detecting credit card fraud both with and without the implementation of Synthetic Minority Over-sampling Technique (SMOTE), it was discovered that the absence of SMOTE generally yielded higher precision, suggesting that models without SMOTE are less prone to false positives, though they tend to miss a greater number of fraudulent transactions, indicated by a lower recall rate. Specifically, the KNN model without SMOTE surfaced as the most balanced model, offering a high accuracy of 99.95%, precision of 94.74%, and an F1 score of 82.44%, which are critical in a real-world setting where false positives can be costly. The Decision Tree and SVM models followed suit, also showing better performance without SMOTE. Conversely, Logistic Regression's performance dropped significantly in precision when SMOTE was applied, despite a considerable gain in recall.

Reason for drop in the Model Performance with SMOTE:

The application of SMOTE (Synthetic Minority Over-sampling Technique) to the dataset, which primarily consisted of PCA-derived variables (V1 to V28), has inadvertently led to a reduction in model performance metrics such as recall, precision, and F1 score, alongside an increase in collinearity among the features which can be seen in the Fig-17. This occurred because SMOTE generated new synthetic samples by interpolating between existing minority class instances, potentially exaggerating any slight correlations originally present among the PCA variables.

These variables are intended to be orthogonal and minimally correlated; however, the synthetic sample generation process introduced dependencies that were not originally present. Consequently, this increased collinearity distorted the statistical reliability of the model's predictors, leading to poorer generalization on new, unseen data and reduced effectiveness of the model overall. The models become overly fitted to the characteristics of the synthetic data, impairing their performance in practical scenarios and thus diminishing the utility of SMOTE in contexts where the integrity of PCA-transformed features is crucial.

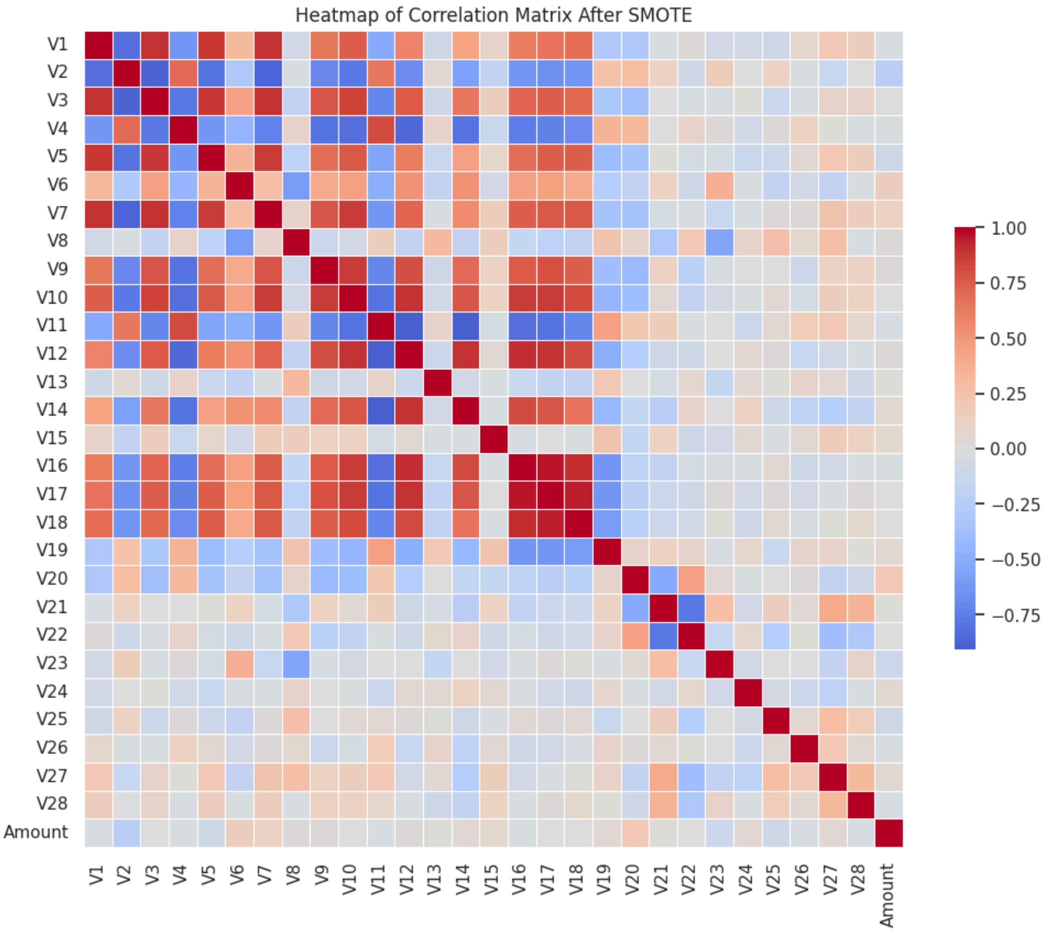

Fig-17: Correlation Heatmap after SMOTE

In summary, these findings highlight the nuanced impact of class imbalance on model performance. While SMOTE improves the detection rate of fraudulent transactions, it also increases the likelihood of falsely labeling legitimate transactions as fraudulent, as seen in the reduced precision scores across all models. Consequently, it is essential to weigh the benefits of increased sensitivity against the costs associated with false positives when choosing to implement SMOTE in a fraud detection system. This analysis underscores the importance of selecting the right model and data balancing technique based on the specific needs of the fraud detection task at hand.

7.2. Future plan

- **Detailed Features:** Consider including additional raw features like merchant details, transaction contexts, and user behavior metrics before applying PCA transformation. These features can offer deeper insights and enhance the predictive accuracy of your model.
- **Deep Learning:** Implement neural networks, such as Convolutional Neural Networks (CNNs) for feature extraction and Recurrent Neural Networks (RNNs) or Long Short-Term Memory networks (LSTMs) for detecting patterns in sequential transaction data.
- **Ensemble Methods:** Explore ensemble methods like Gradient Boosting Machines (GBMs) or Random Forests to leverage the strengths of multiple learning algorithms and improve prediction stability.
- **Anomaly Detection:** Use unsupervised learning models for anomaly detection, which could be especially useful for identifying new or uncommon types of fraud that may not be well-represented in the training dataset.
- **Advanced Over-sampling Techniques:** Beyond SMOTE, explore variants like Borderline-SMOTE or ADASYN, which focus on generating synthetic samples near the borderline of decision classes, potentially improving decision boundaries.
- **Under-sampling Methods:** Implement techniques such as Cluster Centroids or Tomek Links to reduce the number of instances from the majority class, making the dataset more manageable and less biased.
- **Cost-sensitive Learning:** Employ models that integrate the cost associated with misclassifications, particularly emphasizing the high cost of missing a fraudulent transaction.

8. References

1. https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud
2. https://journalofbigdata.springeropen.com/articles/10.1186/s40537-022-00573-8
3. https://www.sciencedirect.com/science/article/pii/S1877050923002314
4. https://www.kaggle.com/code/janiobachmann/credit-fraud-dealing-with-imbalanced-datasets
5. https://medium.com/@corymaklin/synthetic-minority-over-sampling-technique-smote-7d419696b88c