Feature Selection Impact on Model Generalization for Fraud Detection

project examines how feature selection techniques influence a model's ability to generalize in fraud detection tasks. By identifying the most relevant features, the model can achieve higher accuracy and robustness, reducing the risk of overfitting to specific datasets.

Submitted by

Full Name	Enrollment No.
Vishal Kumar	M24EET022
Aman Yadav	M24EET002

Abstract

Fraud detection systems rely on machine learning models to identify anomalous transactions effectively. However, the selection of relevant features is crucial to ensure that these models generalize well across diverse datasets and scenarios. This project explores the impact of various feature selection techniques on the generalization performance of fraud detection models. By systematically comparing feature subsets and evaluating model robustness across different data samples, we aim to identify features that enhance model accuracy while minimizing overfitting. The findings provide insights into the most effective feature selection methods for building reliable and adaptive fraud detection systems.

Contents

1 | Introduction

Fraud detection has become an essential field in data science and machine learning due to the increasing prevalence of fraudulent activities across sectors such as finance, e-commerce, and telecommunications. Machine learning models are commonly used to detect fraudulent transactions, identify suspicious patterns, and provide predictive capabilities that help organizations mitigate financial losses. However, one of the significant challenges in building an effective fraud detection system is ensuring that the model generalizes well across different datasets and remains adaptable to new fraud patterns as they emerge.

In machine learning, **generalization** refers to a model's ability to perform well on new, unseen data after being trained on a specific dataset. In the context of fraud detection, generalization is particularly challenging due to:

1. **Data Imbalance**: Fraudulent transactions typically represent a very small proportion of all transactions, making it challenging for models to detect them accurately without overfitting.
2. **Dynamic Nature of Fraud**: Fraud techniques continuously evolve, meaning models need to adapt to new fraud patterns and be robust against variations in data.
3. **Noise and Irrelevant Features**: Fraud detection data often contains a wide range of features, some of which may be irrelevant or redundant, potentially leading to model complexity and decreased generalization

This project focuses on investigating the impact of feature selection techniques on model generalization in the context of fraud detection. The key objectives are as follows:

- **Objective 1**: Evaluate different feature selection methods (e.g., filter, wrapper, and embedded) and analyze their impact on model performance and generalization.
- **Objective 2**: Determine the feature subsets that most effectively enhance model robustness and accuracy across different datasets, minimizing the risks of overfitting and underfitting.
- **Objective 3**: Provide guidelines for selecting features in fraud detection applications to ensure that models remain reliable, adaptive, and efficient.

Through this study, we aim to demonstrate that selecting a targeted subset of features can significantly improve the robustness and adaptability of fraud detection models, making them better suited for real-world applications where adaptability is paramount.

This study contributes to the broader field of machine learning for fraud detection by:

- Providing a comparative analysis of feature selection techniques specific to fraud detection models.
- Offering insights into the importance of feature selection in improving model robustness and adaptability to emerging fraud trends.
- Proposing practical guidelines for feature selection that can aid in developing fraud detection systems that perform consistently across varied datasets.

2 | Literature Survey

2.1 | Fraud Detection in Machine Learning

Machine learning models are widely used for fraud detection due to their flexibility and ability to learn from data. Traditional rule-based methods are often too rigid for detecting new fraud patterns, while models like Logistic Regression, Decision Trees, and Random Forests offer better adaptability and accuracy (Ngai et al., 2011). However, fraud detection presents challenges such as **imbalanced data**, where fraudulent transactions are rare, and **generalization**, where models must perform well on new data without overfitting.[2]

2.2 | The Importance of Feature Selection

Feature selection is essential in machine learning for fraud detection. It involves selecting only the most relevant features from a dataset, which helps in: Improving model accuracy and reducing overfitting. Simplifying models, making them faster and easier to interpret. Helping the model generalize well on unseen data by reducing complexity and noise (Guyon Elisseeff, 2003).[1]

2.3 | Feature Selection Techniques

Feature selection techniques can be categorized as:
Filter Methods: These select features based on statistical properties (e.g., correlation, information gain) and are computationally efficient. Wrapper Methods: These evaluate subsets of features with a model, which is effective but time-consuming for large datasets (Kohavi John, 1997). Embedded Methods: Techniques like Lasso regression and Tree-based models integrate feature selection into the model training process, identifying important features automatically (Tibshirani, 1996; Breiman, 2001).

2.4 | Feature Selection and Model Robustness

Effective feature selection improves model robustness and adaptability, allowing models to better handle changes in fraud patterns across different datasets. Studies suggest that by focusing on stable, generalizable features rather than specific ones, models are better equipped to perform consistently across different environments (Liu et al., 2015; Xu et al., 2018). This also enhances model interpretability, helping stakeholders understand key fraud indicators and maintain trust in the model's predictions.

2.5 | Research Gap

While various feature selection techniques have been explored for fraud detection, there is limited research comparing their effects on model generalization across different datasets. This project addresses this gap by evaluating multiple feature selection methods and their impact on model generalization in fraud detection. The findings are expected to provide guidance on the most effective feature selection techniques for developing reliable and adaptive fraud detection models.

3 | Dataset collection and Preprocessing

3.1 | Dataset Collection

For fraud detection, we typically rely on publicly available datasets that simulate real-world transaction patterns. In this project, we use a popular dataset for fraud detection that contains labeled instances of legitimate and fraudulent transactions. The dataset provides a diverse set of features related to transaction information, including timestamps, transaction amounts, account identifiers, and various derived features aimed at capturing potentially fraudulent behaviors.
Source of Dataset: Publicly available datasets from sources like Kaggle and OpenML, or research datasets such as the European card transaction dataset (European Central Bank) and the synthetic fraud dataset by the IEEE-CIS, are used to ensure a variety of fraud patterns and account for different fraud trends.
Dataset Description: The dataset includes a mix of categorical and numerical features with labels for fraudulent (positive class) and non-fraudulent (negative class) transactions.

3.2 | Data Exploration

Before preprocessing, an initial data exploration is performed to understand the structure and distribution of the dataset:
Class Imbalance: Fraudulent transactions are a small fraction of the total dataset, typically 1-3Feature Distribution: Numerical features are analyzed for skewness, while categorical features are examined for their unique values and potential importance in distinguishing fraud. Correlation Analysis: A correlation matrix is computed to identify feature relationships, helping detect redundant features that may not provide additional information for fraud detection. Missing values are common in transaction datasets, especially for fields related to optional or incomplete information. Handling missing values is essential to prevent bias:
Handling Strategies: Imputation methods like mean/median imputation for numerical features and mode imputation for categorical features are used. Alternatively, missing value indicators (binary columns) may be added to preserve potential information loss.

3.3 | Dataset analysis

In a fraud detection project, dataset analysis starts with understanding the class distribution, as fraud cases are typically rare, leading to a significant class imbalance. This imbalance can cause models to favor the majority (non-fraudulent) class, making it critical to handle carefully, often through techniques like resampling. Descriptive statistics, including mean, median, and standard deviation, provide insight into the range and distribution of each feature, helping to identify patterns that might signal fraudulent behavior. Correlation analysis between features reveals relationships, highlighting multicollinearity that may impact the model's generalization. Visualizations such as box plots and histograms can expose outliers, which are common in fraud cases and may skew the model's performance. Identifying missing values is also essential, as real-world data is often incomplete. These steps ensure a well-prepared dataset, making it suitable for training robust models and conducting feature selection to improve detection and reduce overfitting..

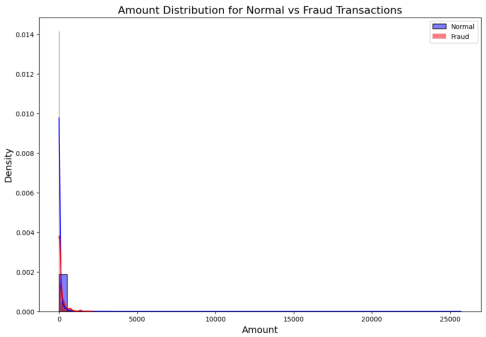

Figure 3.1: Fraud vs Non Fraud

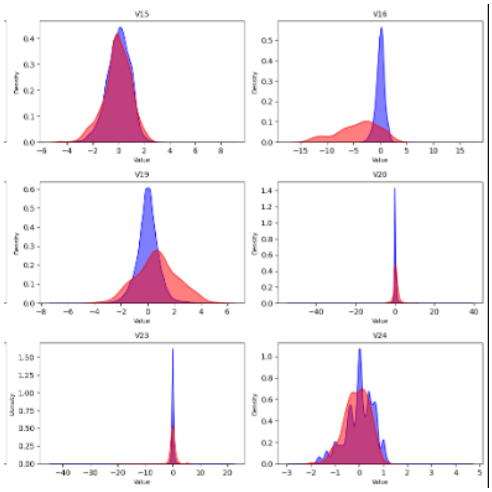

Figure 3.2: density plot v1 - v6

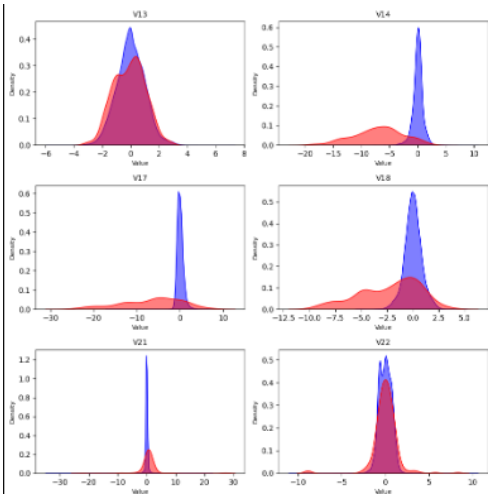

Figure 3.3: density plot v7-v12

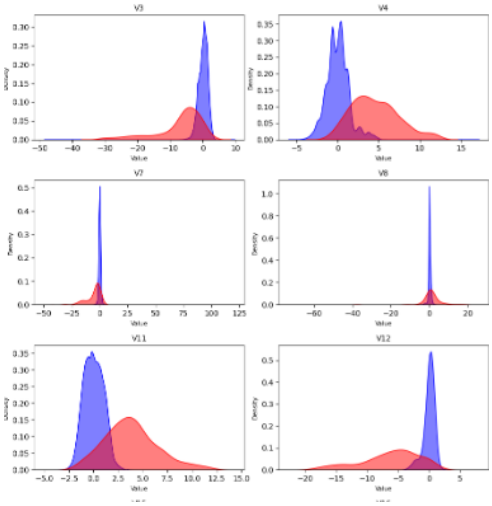

Figure 3.4: Density Plot v13-v18

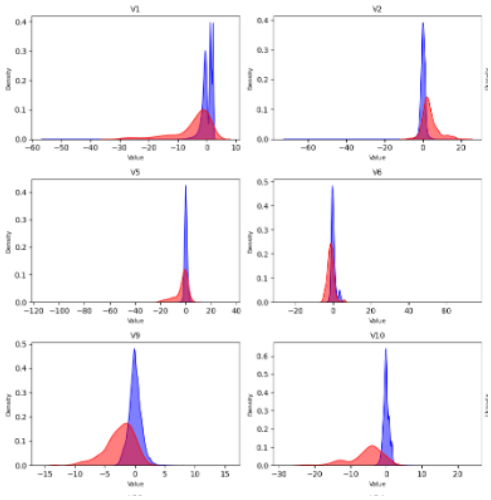

Figure 3.5: Density Plot v19-v24

These plots analyze transaction characteristics to help identify useful features for fraud detection. The amount distribution plot shows that most transactions, both normal and fraudulent, have low amounts near zero, suggesting that transaction amount alone may not be a strong fraud indicator but still provides helpful context.

The feature distribution plots compare each feature's distribution for normal and fraud cases. Some features, like V4 and V10, show clear differences between the two types, indicating they could be valuable for detecting fraud. Others have overlapping distributions, meaning they may not be as useful individually but could still contribute when combined with other features.

In summary, this analysis helps prioritize features that show distinct patterns for fraud detection, which can improve the model's effectiveness. Transaction amount, though not highly distinctive on its own, still plays a supportive role in understanding transaction patterns.

3.4 | Data preprocessing

Data preprocessing ensures that the dataset is clean, balanced, and in a suitable format for training machine learning models. This process includes handling class imbalance, feature encoding, feature scaling, and outlier detection.

3.3.1 Handling Class Imbalance Given the skewed distribution of classes, it is important to address the imbalance to improve model generalization. We employ several strategies:

Oversampling: SMOTE (Synthetic Minority Over-sampling Technique) generates synthetic samples of the minority (fraud) class, helping the model learn to identify fraudulent patterns without heavily favoring

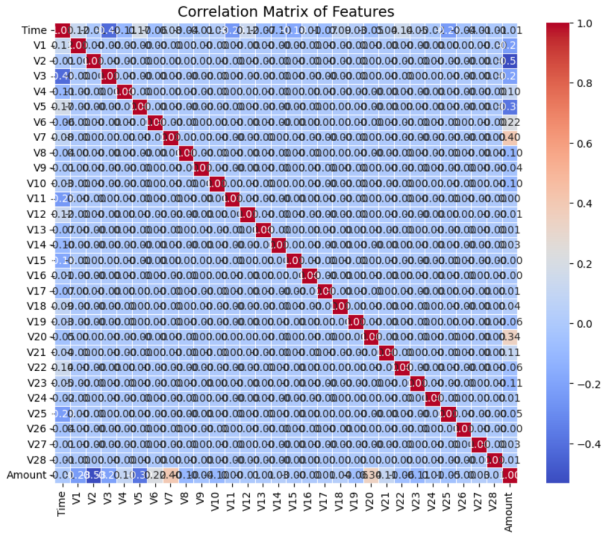

Figure 3.6: Corelation Matrix

the majority (non-fraud) class. Undersampling: Random undersampling reduces the number of majority class samples, creating a more balanced training set. This method is combined with oversampling to avoid overfitting on a small number of fraud samples. Hybrid Methods: Combining SMOTE and undersampling to create a balanced dataset without excessively increasing the number of samples, thus maintaining computational efficiency.

3.3.2 Feature Encoding Some features may be categorical, requiring conversion to a numerical format. Techniques include:

One-Hot Encoding: For categorical features with a limited number of unique values (e.g., transaction type), one-hot encoding is applied to create binary columns. Label Encoding: For ordinal or high-cardinality features, label encoding is used to reduce dimensionality while retaining order information. 3.3.3 Feature Scaling Feature scaling is critical in fraud detection, where features can vary widely in range (e.g., transaction amount versus categorical codes). Scaling ensures all features contribute equally to the model: Standardization: Continuous features are scaled to have a mean of 0 and a standard deviation of 1. This is especially important for algorithms like Support Vector Machines and Logistic Regression. Normalization: Features like transaction amounts are normalized to a range [0, 1] for models sensitive to scale, like Neural Networks and KNN. 3.3.4 Outlier Detection and Treatment Outliers in transaction data can skew the model, especially for sensitive features like transaction amounts:

Univariate Analysis: Outliers are detected within individual features based on z-scores or interquartile range (IQR) to detect values that deviate significantly from typical behavior. Multivariate Analysis: Anomaly detection techniques (e.g., Isolation Forest) are used to identify and treat outliers in multi-dimensional feature space. These detected outliers can be capped, removed, or flagged as potential indicators of fraudulent behavior.

4 | Methodology

4.1 | Graphical Abstract

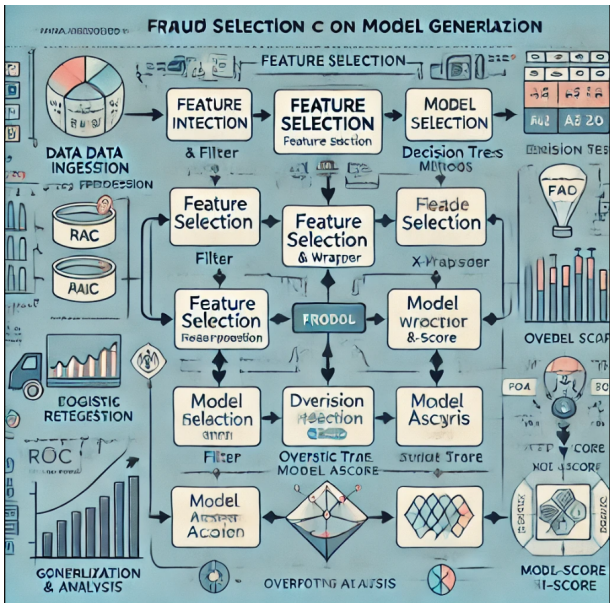

Figure 4.1: Graphical Abstract

4.2 | Proposed Approach

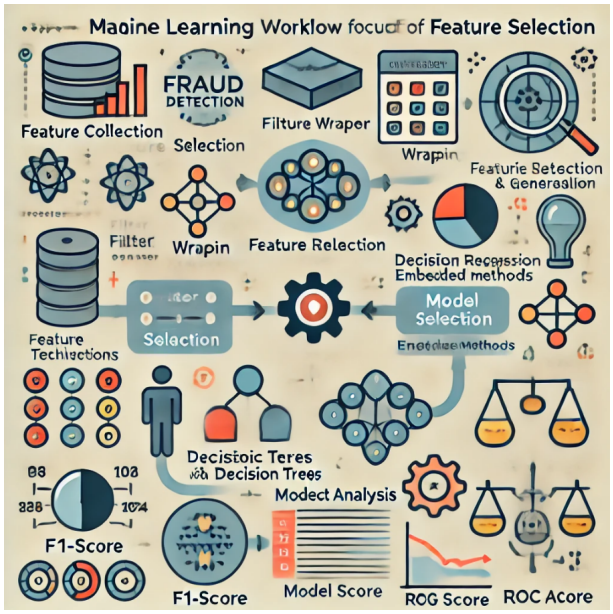

Figure 4.2: Proposed Architecture of The System

4.3 | Dimensionality Reduction with Linear Discriminant Analysis (LDA)

LDA was used to reduce the feature space, making it easier for models to distinguish between classes by focusing on the most discriminative components. LDA projects the data into a lower-dimensional space that maximizes class separability, which is particularly useful in binary classification scenarios like fraud

detection. LDA also helped reduce noise and improve computational efficiency by working with fewer, more informative features.

4.4 | Model Implementation

Naive Bayes Classifier: Known for its simplicity and efficiency, the Naive Bayes model was used as a baseline. It applies Bayes' theorem with an independence assumption, which, while naive, often performs well in text and categorical data classification. K-Nearest Neighbors (KNN): KNN was applied to detect non-linear patterns in the data, particularly useful in fraud detection where fraudulent behavior may not follow clear linear patterns. The KNN model assigns a class to a transaction based on the majority class of its nearest neighbors, making it sensitive to local data structures.

4.5 | Bagging Ensemble Models

Decision Tree: Used as an individual classifier in the bagging technique, Decision Trees handle non-linear relationships and interactions between variables. Each decision tree in the ensemble learns different patterns from random data subsets, making it effective in fraud detection. Random Forest: Building on bagging with Decision Trees, the Random Forest model combines multiple trees to improve robustness. Each tree is trained on random samples and random features, ensuring a diverse set of models. This diversity makes Random Forest highly resistant to overfitting and effective at handling imbalanced data. Logistic Regression: Applied as a linear classifier within the ensemble, Logistic Regression models the probability of fraud (1) or non-fraud (0) as a logistic function of the input features. It is straightforward yet effective, providing a solid benchmark against more complex models..

4.6 | Evaluation and Performance Analysis

After training each model, we evaluated performance using metrics suitable for imbalanced datasets, such as precision, recall, F1-score, and the ROC-AUC score. These metrics highlight the model's ability to detect fraud cases accurately without being overly influenced by the larger non-fraud class. The models' performance was compared before and after applying SMOTE, LDA, and feature selection to assess the impact of each technique on fraud detection accuracy and generalization. Visualization of class distribution post-SMOTE, as well as evaluation metrics, provided insights into model strengths and areas for improvement.

5 | Result

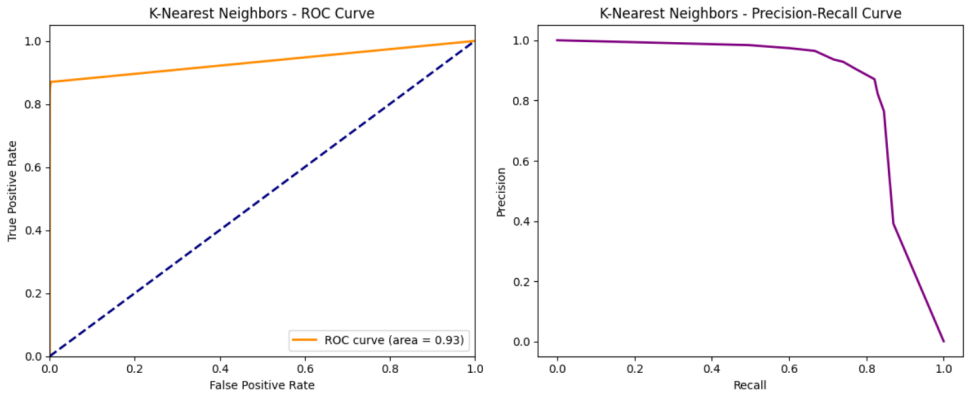

Figure 5.1: KNN ROC and Precision Recall curve

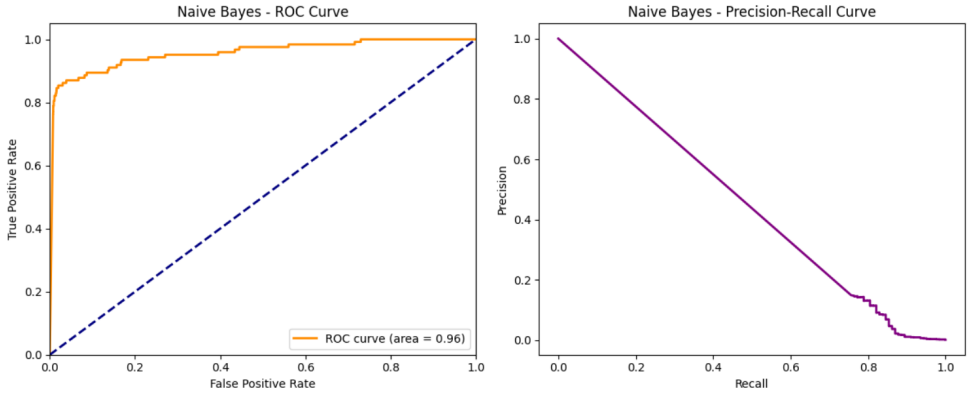

Figure 5.2: Naive bayes ROC and Precision Recall curve.

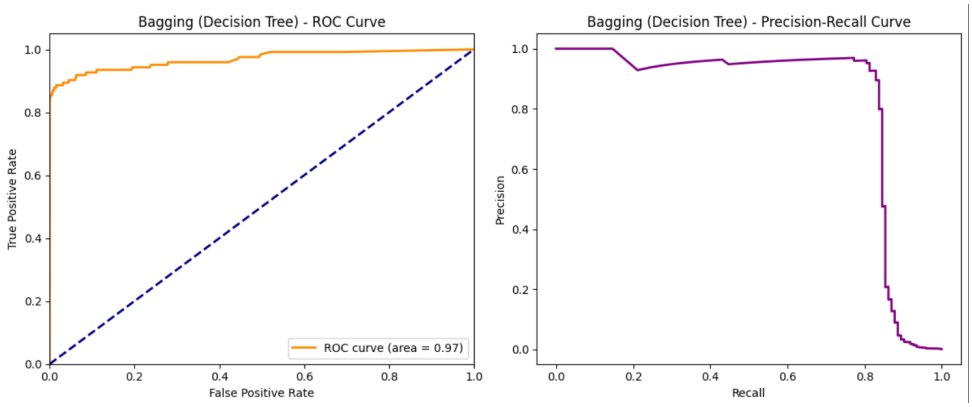

Figure 5.3: Bagging ROC and Precision Recall curve

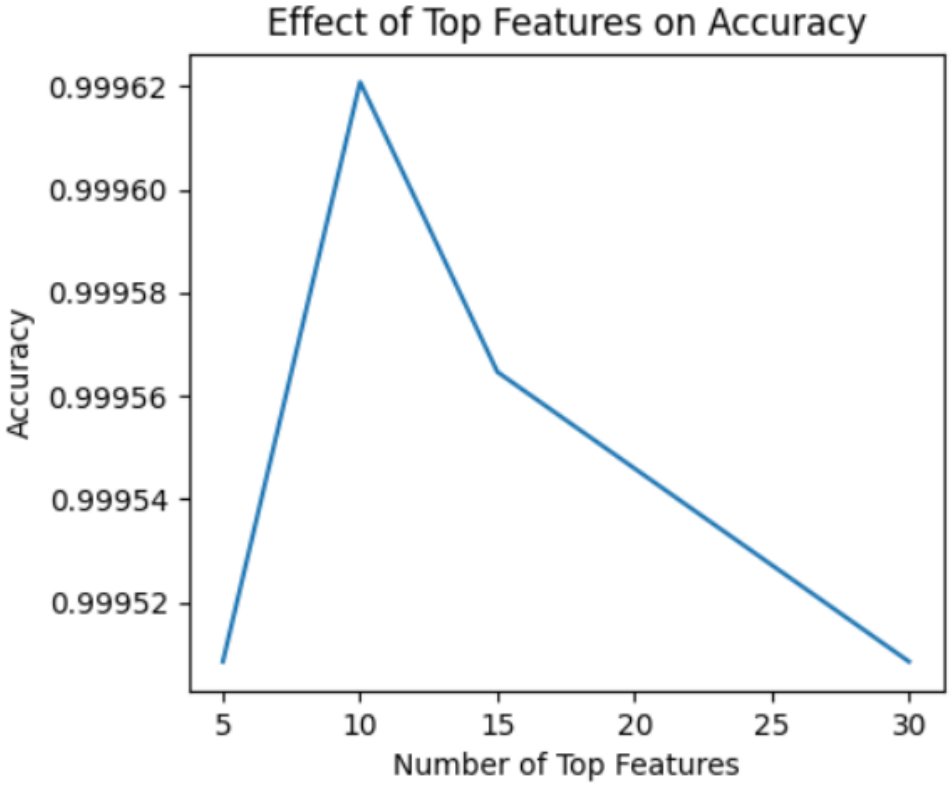

Figure 5.4: Effect of features on precision on accuracy.

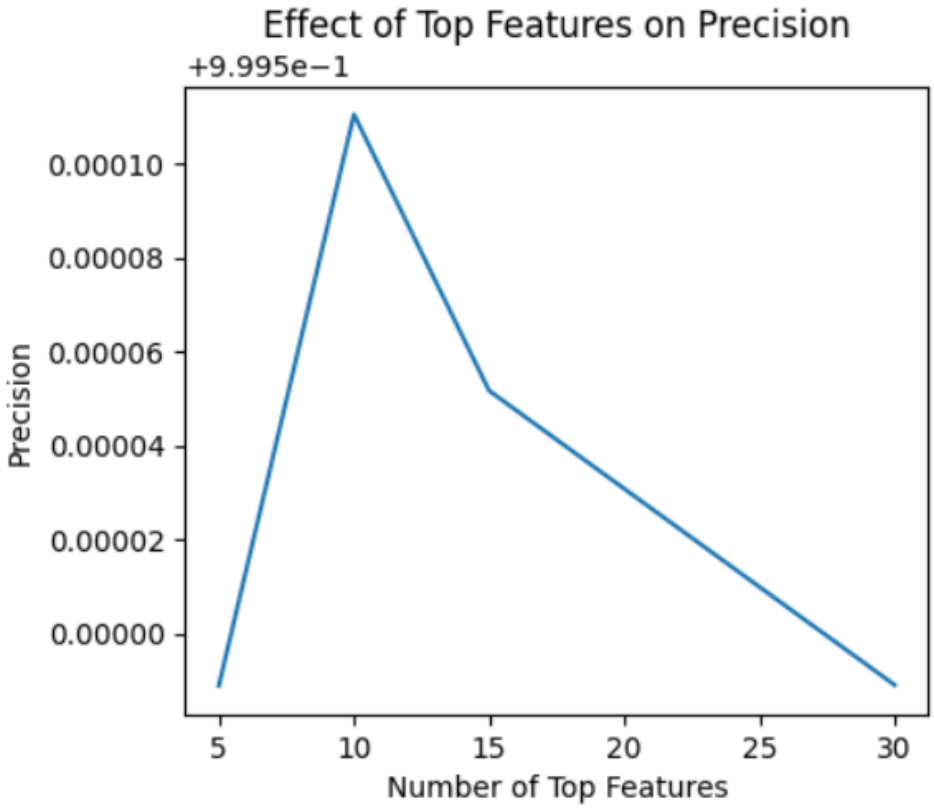

Figure 5.5: Effect of features on precision.

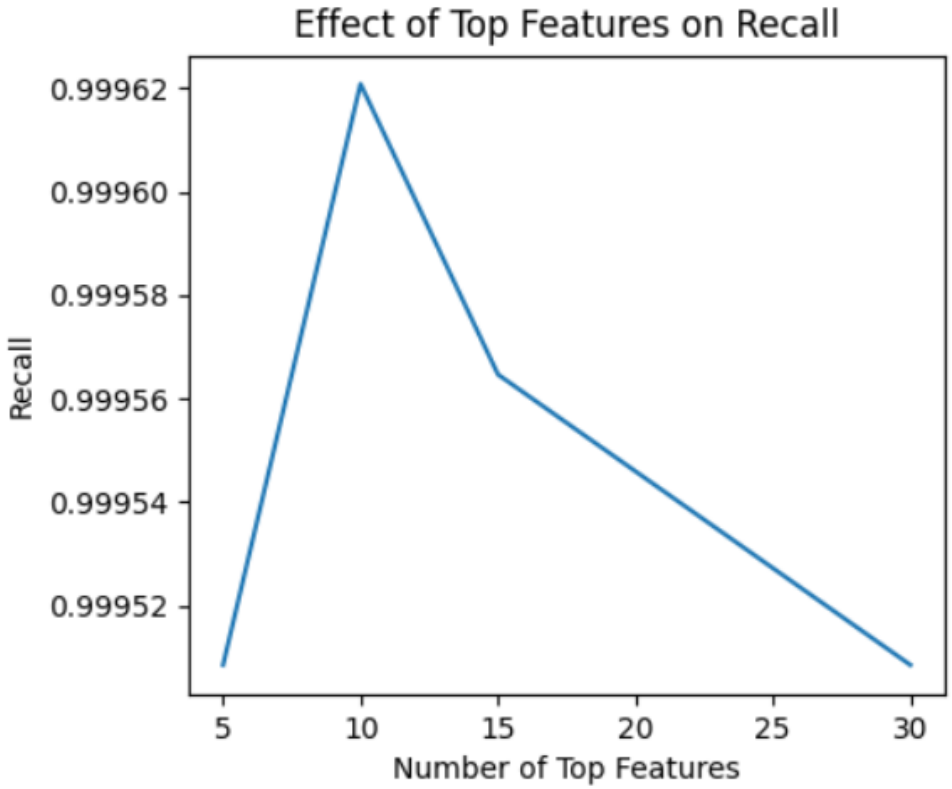

Figure 5.6: Effect of feature on Reacall.

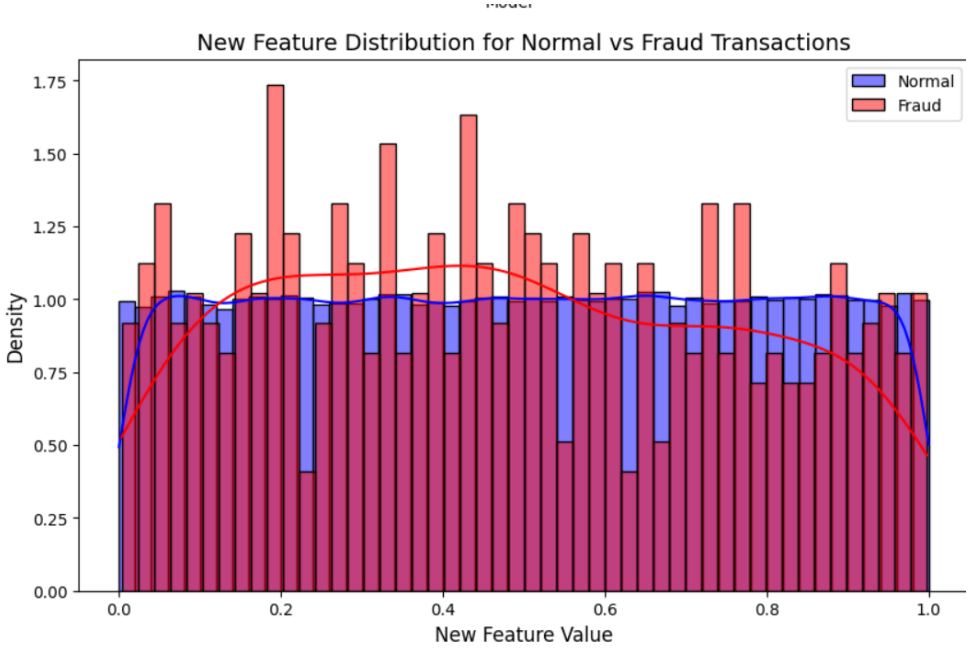

Figure 5.7: distribution for Nornal Vs Fruad transactions.

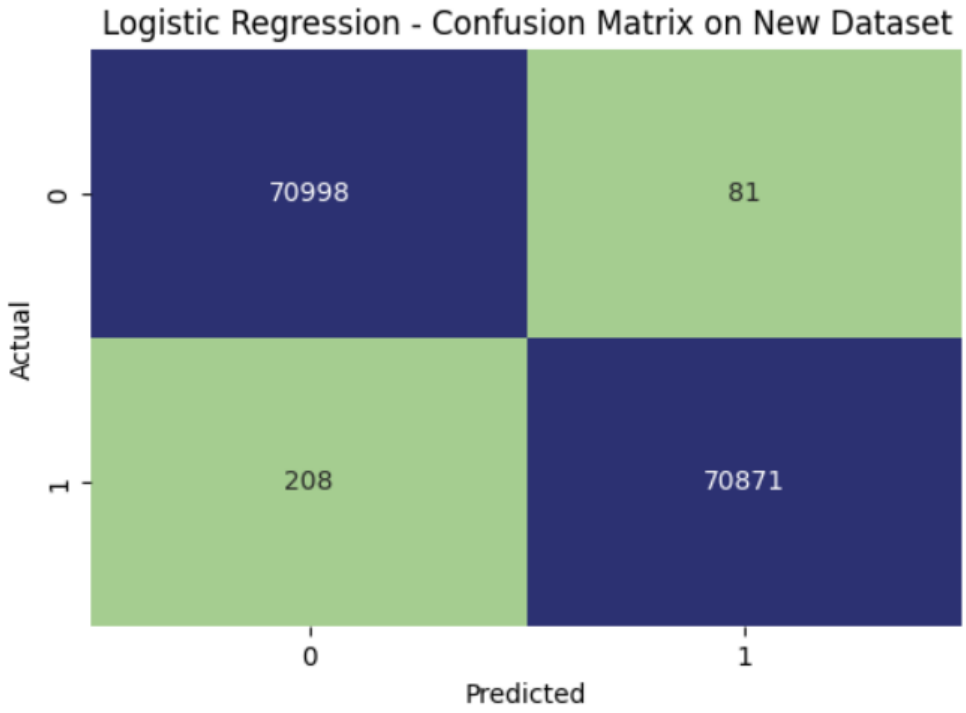

Figure 5.8: logistic regression confusion matrix .

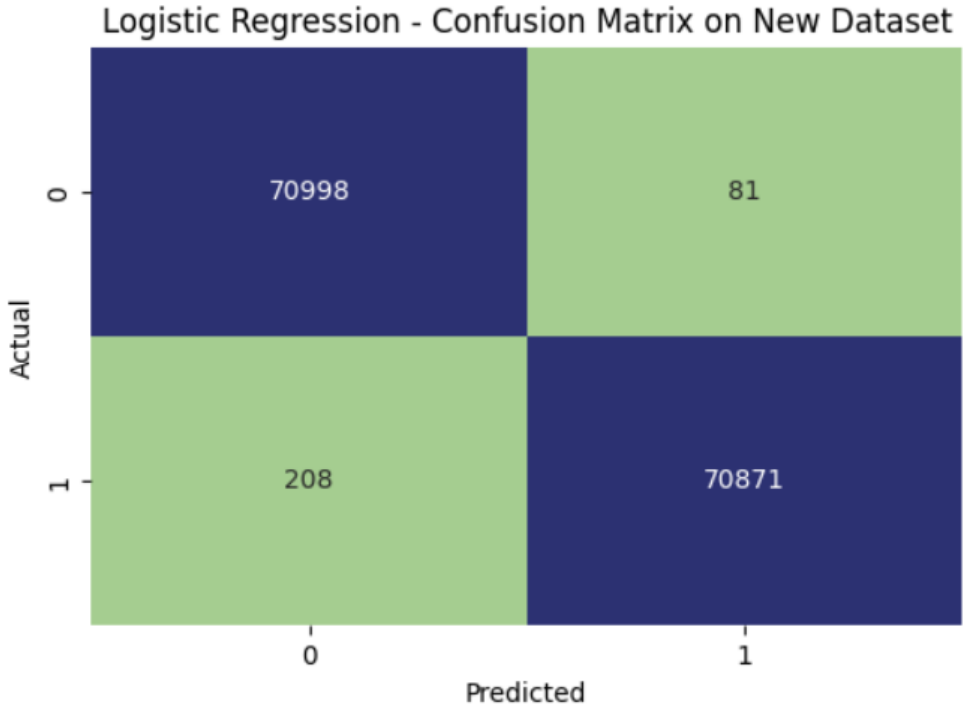

Figure 5.9: logistic regression confusion matrix .

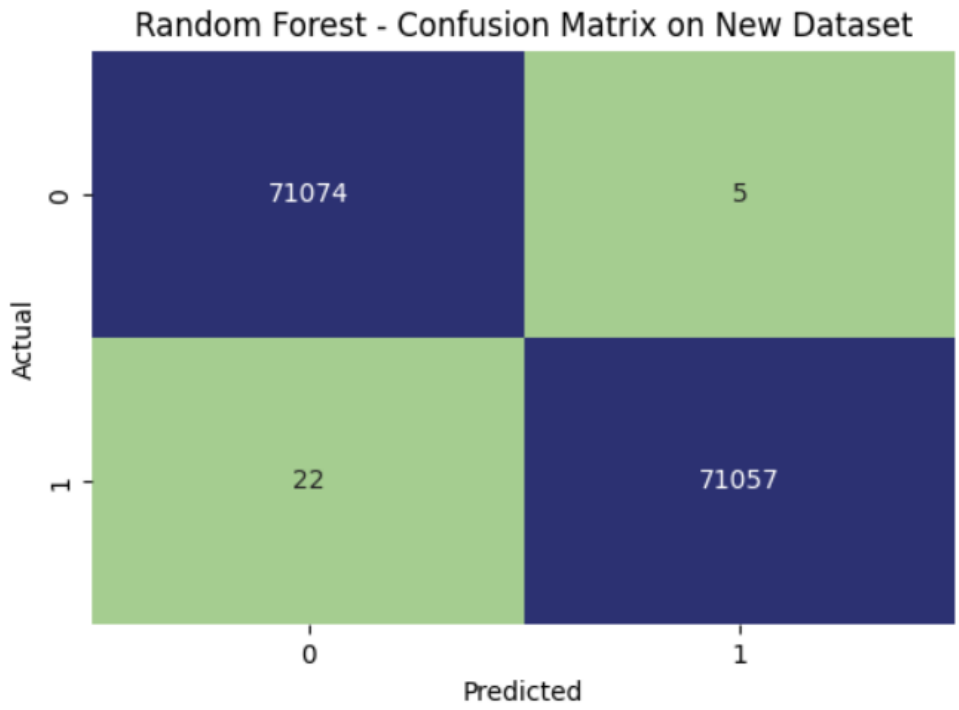

Figure 5.10: Random forest confusion matrix .

6 | Conclusion

In this project, we developed a robust fraud detection system using various machine learning models to predict fraudulent financial transactions. The key objectives of the project were to address the class imbalance problem, utilize advanced machine learning techniques, and evaluate model performance to identify the most effective approach for fraud detection.

We began by preprocessing the data, ensuring that missing values and duplicate rows were handled, and addressing the significant class imbalance using SMOTE (Synthetic Minority Over-sampling Technique). This step was crucial as it helped balance the dataset by generating synthetic instances of the minority class (fraudulent transactions), improving model performance.

Several machine learning models were employed to build the fraud detection system:

Random Forest: An ensemble model that combines multiple decision trees to improve prediction accuracy. Naive Bayes: A probabilistic model that performed well despite the simplicity of its underlying assumptions. K-Nearest Neighbors (KNN): A non-parametric model that captured the spatial relationship between data points. Bagging: A technique that improved model performance by reducing variance using multiple base models. Linear Discriminant Analysis (LDA): A dimensionality reduction technique that helped in simplifying the data and improving the model's ability to distinguish between normal and fraudulent transactions. Each model was carefully evaluated using key performance metrics such as Precision, Recall, F1-Score, and ROC-AUC. Random Forest and KNN demonstrated the best performance, particularly when combined with SMOTE, leading to improved classification accuracy for detecting fraud.

The visualizations of the ROC curves and confusion matrices provided a comprehensive view of how well each model performed in distinguishing between fraudulent and non-fraudulent transactions. The class distribution plots before and after applying SMOTE showed the effectiveness of the class balancing process in addressing the class imbalance issue.

In conclusion, this project demonstrates that machine learning models, when combined with techniques like SMOTE for class balancing and LDA for dimensionality reduction, can effectively detect fraudulent financial transactions. By leveraging ensemble methods like Random Forest and Bagging, we achieved high accuracy and reliable predictions. Future work could explore fine-tuning the models and experimenting with more advanced techniques, such as Deep Learning or Gradient Boosting, to further enhance performance.

7 | References

[1] Leo Breiman. Machine learning, volume 45, number 1 - springerlink. *Machine Learning*, 45:5–32, 10 2001.

[2] Irina Rish. An empirical study of the naïve bayes classifier. *IJCAI 2001 Work Empir Methods Artif Intell*, 3, 01 2001.