CREDIT CARD FRAUD – COMPARATIVE STUDY OF THE EFFECTIVENESS OF MACHINE LEARNING ALGORITHMS

by

Dibyendu Roy

DISSERTATION

Presented to the Swiss School of Business and Management Geneva

In Partial Fulfillment

Of the Requirements

For the Degree

DOCTOR OF BUSINESS ADMINISTRATION

SWISS SCHOOL OF BUSINESS AND MANAGEMENT GENEVA

JANUARY, 2024

CREDIT CARD FRAUD – COMPARATIVE STUDY OF THE EFFECTIVENESS OF MACHINE LEARNING ALGORITHMS

by

Dibyendu Roy

Supervised by

Derrald Stice

APPROVED BY

Ljiljana Kukec [illegible]

Dissertation chair

RECEIVED/APPROVED BY:

Admissions Director

Acknowledgements

I would like to acknowledge the paper titled “COMPARATIVE STUDY OF MACHINE LEARNING ALGORITHMS TO DETECT CREDIT CARD FRAUD”, Roy (2020) which served as the inspiration for this study. The current research represents a continuation and refinement of the work conducted in 2020, aiming to build upon the insights and findings presented in the original study.

ABSTRACT

CREDIT CARD FRAUD – COMPARATIVE STUDY OF THE EFFECTIVENESS OF MACHINE LEARNING ALGORITHMS

Dibyendu Roy
2024

Dissertation Chair: Ljiljana Kukec

Credit cards offer convenience and user-friendly options for everyday transactions, attracting a wide audience seeking hassle-free financial interactions. The continuous evolution of technology and expansive network coverage facilitates easier access to credit cards and encourages their frequent usage among citizens. However, alongside the growth of the credit card industry, the threat of fraud looms large. One prevalent form of fraud involves unauthorized use of credit card details for purchases without the card owner's consent. With the sheer volume of card transactions, financial institutions and credit card issuers face significant challenges in detecting these fraudulent activities. This research aims to investigate the potential of Machine Learning techniques in identifying fraudulent transactions within a given dataset of credit card transactions. Additionally, the study assesses various Machine Learning algorithms using a publicly available credit card transactions dataset, analyzing their effectiveness in distinguishing between fraudulent and genuine transactions. Machine Learning algorithms like “Logistic Regression”, “Gaussian Naïve Bayes”, “KNeighbors Classifier”, “Linear Support Vector Classifier”, “Random Forest Classifier”, “Isolation Forest”, “Bagging Classifier”, “Decision Tree Classifier”, “Keras Classifier”, “MLP Classifier”, “LightGBM Classifier”, “XGboost Classifier”, “Adaboost Classifier”, “Catboost Classifier”, “Dynamic Ensemble” and “Stacking” are applied on the available dataset and compared for various performance metrics.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER I:
INTRODUCTION

1.1 Introduction

The latest technological advancements have undoubtedly simplified our daily routines, offering unparalleled convenience in our day-to-day tasks. Yet, this very flexibility has also introduced novel threats previously unfamiliar to us. Embracing the prowess of technology exposes us to new vulnerabilities. Credit cards and online transactions stand out as prime examples of modern marvels, revolutionizing our financial dealings with unparalleled ease. No longer do we need to fret over exact change; a simple swipe of our credit card suffices to complete transactions, marking a significant departure from the days of haggling over loose coins.

In today's world, credit cards have seamlessly integrated into our daily lives, offering unparalleled ease of use and a straightforward transaction process. People experience a sense of empowerment through their use. Moreover, with the explosive growth of the e-commerce industry, online payments have become enticing for individuals across all financial spectrums. What was once considered a luxury, credit card usage has now transformed into a necessity in modern society.

Credit Cards usage has increased considerably across the globe, making this a lucrative financial instrument for fraudsters to target. According to the report "India: Number of Credit Cards in Use 2014-2029" by Degenhard (2024), between 2024 and 2029, there is a projected continuous surge in the utilization of credit cards in India, with an anticipated addition of 2.3 million cards, marking a growth rate of 5.29 percent. This trend signifies a consistent expansion, extending over a span of fifteen years. It is anticipated that by the end of this period, in 2029, the total number of credit cards in circulation is expected to reach 45.81 million, marking a notable milestone as it attains a new peak.

As per this same report, in United States of America, 25 million cards are expected to be added reflecting a growth rate of 2.25 percent.

However, this convenience comes at its own price. Credit Card fraud is resulting in loss of billions of dollars globally. According to the report "Card Fraud Value Worldwide" by Raynor de Best (2024), there has been a rise in card fraud losses globally, with a notable contribution from the United States. Card fraud losses across the world increased by more than 10 percent between 2020 and 2021, the largest increase since 2018. It was estimated that merchants and card acquirers lost more than 30 billion U.S. dollars during this period.

According to the article "Card Industry Faces $400B in Fraud Losses over next Decade, Nilson Says" by Caitlin Mullen (2021), projections suggest significant challenges ahead for the card industry due to fraud losses. As per this article, over the forthcoming decade, the card industry is projected to face a staggering collective loss of $408.50 billion globally due to card fraud, as indicated by the annual report from the industry research firm Nilson Report. As total payment card volume is anticipated to surge to $79.14 trillion by 2030, the industry is estimated to incur $49.32 billion in losses attributed to fraud. This article reports, in the previous year, the United States once again held a larger portion of global card fraud compared to its share of total card volume. Although representing only 22% of the total card volume worldwide, the U.S. accounted for 36% of card fraud in the same period. In 2019, these percentages stood at 22% and 34%, respectively. It is projected that by 2030, U.S. fraud losses will escalate to $17 billion, coinciding with total card volume reaching nearly $19 trillion.

Hindustan Times (2020) reported that in over 10 years period (April 2009 to September 2019), fraudsters siphoned INR 615.39 Crore ($75 million) in more than 117 thousand cases of Credit and Debit card frauds.

The article titled "5 Most Popular Credit Card Frauds in India 2024," authored by Asefa Hafeez (2024) and published on Kuvera, highlights the persistent threat of credit card frauds in India. It brings to light data from the National Crime Records Bureau (NCRB), revealing a concerning trend. In 2021, a total of 3,342 fraud cases were reported, marking a substantial increase of approximately 20% compared to the preceding year. This surge follows a notable uptick of over 70% in such fraudulent activities during 2020.

According to the "Consumer Sentinel Network Annual Data Book 2021" by the Federal Trade Commission (FTC, 2021), for five consecutive years from 2017 to 2021, Credit Card fraud was the biggest identity theft reported. As per this report, the number of Credit Card frauds reported in US in the year 2017, 2018, 2019, 2020 and 2021 are 133,107, 157,745, 271,938, 393,378 and 389,737 respectively.

According to the article "Debit, Credit Card Frauds on Rise, ATM Scams down: NCRB" by Shuja Asrar (2022), 3,432 cases of credit and debit card frauds were filed from across India in 2021, up nearly 20% from the year-earlier. In 2020, such frauds increased by over 70%. In just two years, credit and debit card-related frauds nearly doubled, it showed.

As per the research report from Security.org team, 65% of US Credit Card holders have been fraud victims at some point in their lives. Same research found that in 2022, 44 percent of credit card users reported having two or more fraudulent charges, compared to 35 percent in 2021. Below image from the Security.org indicates percentage of card holders impacted by Credit Card fraud.

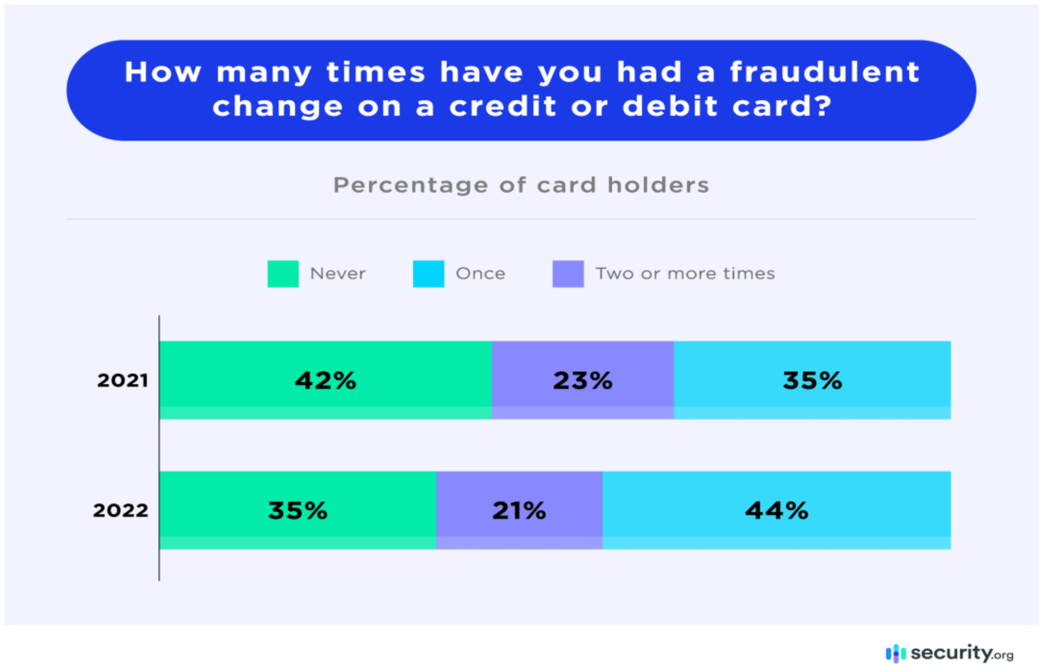

Figure 1.1
Percentage of card holders impacted by Credit Card fraud (security.org team, 2023)

The webpage "Credit Card Fraud Soars to 10-Year High" by Experian Plc (2023) reveals that Credit Card fraud rate rose by 18% in the last three months of 2022 which means the overall fraud rate for 2022 was the highest yearly rate recorded by Experian in the last 10 years.

1.2 Research Problem

Credit card fraud poses significant challenges and impacts various stakeholders in the financial ecosystem.

At its core, credit card fraud occurs when someone unlawfully uses a card or its associated account for transactions, unbeknownst to the legitimate cardholder. This illicit activity not only affects the cardholder, who faces potential financial losses, but also impacts merchants who may suffer revenue losses or reputational damage due to fraudulent transactions. Additionally, banks and financial institutions bear the brunt of fraud through reputational harm and financial liabilities.

As technology continues to advance, credit card fraud has become increasingly sophisticated, presenting greater hurdles for cardholders, financial institutions, and card issuers. The evolving landscape of fraud necessitates constant vigilance and adaptation to new tactics and technologies. Furthermore, regulatory and compliance pressures add another layer of complexity, requiring banks and financial institutions to invest heavily in fraud prevention measures while navigating stringent regulations to protect both themselves and their customers.

Credit card fraud encompasses various types, each presenting distinct risks and methods of exploitation. Here are the primary categories:

1) Card-not-present fraud: Occurs when someone illicitly utilizes vital credit card data, including name, card number, CVV, expiry date, to conduct unauthorized transactions online.

2) Lost or stolen Credit Card: If a credit card is misplaced or stolen, unauthorized individuals can exploit it for fraudulent transactions.

3) Phishing: Fraudsters masquerade as trusted entities, often via emails or text messages, to deceive individuals into divulging their credit card account details. With this information, they can perpetrate fraudulent transactions.

4) Skimming: Criminals employ specialized devices attached to point-of-sale terminals or ATMs to clandestinely capture credit card data from the magnetic strip. This stolen data is then used to create counterfeit cards for fraudulent transactions.

5) Identity theft: Perpetrators unlawfully obtain and misuse personal data, including credit card details, to engage in illicit financial activities without the victim's consent.

1.3 Purpose of Research

In the ever-evolving landscape of financial transactions, card frauds loom as a persistent menace. To combat this threat, the industry must embrace cutting-edge technological solutions. Financial institutions play a crucial role in this fight by offering sophisticated and automated fraud detection capabilities.

However, as technology is advancing, fraudsters have become craftier, blurring the lines between genuine and fraudulent transactions. Traditional pattern matching techniques—whether manual or automated—fall short in identifying these increasingly sophisticated scams.

Another hurdle is the access to credit card transaction datasets. Privacy concerns often lock them away from public view, making research outcomes elusive. Financial institutions conduct most of the research internally, leaving limited information accessible to the wider world.

As we navigate these challenges, setting performance standards for fraud detection models remains a puzzle. But with determination and innovation, we can stay one step ahead in this high-stakes game.

In a given dataset of credit card transactions, fraud detection involves accurately classifying new transactions as either genuine or fraudulent. An effective fraud detection system not only identifies fraudulent transactions but also minimizes the misclassification of genuine transactions as fraudulent.

1.4 Research Purpose and Questions

This paper intends to explore following research questions:

1) Is it possible to utilize Machine Learning methodologies for the identification and mitigation of fraudulent activities within credit card transactions? Machine Learning algorithms have demonstrated efficacy in discerning potentially fraudulent transactions from a pool of credit card activities. These algorithms operate by assessing various transactional attributes to determine the probability of fraudulent behavior. By analyzing a multitude of features associated with each transaction, such as

transaction amount, frequency, location, and timing, these models are adept at uncovering intricate patterns within complex datasets.

2) If Machine Learning techniques are to be employed for the detection and analysis of fraudulent transactions, the pivotal question emerges: which algorithm or model exhibits the highest efficacy in discerning fraudulent activities from legitimate ones? To elucidate this, an exhaustive comparative study was undertaken, meticulously scrutinizing the performance of an array of algorithms and models.

This study encompassed a diverse spectrum of methodologies, spanning traditional techniques such as Logistic Regression and Gaussian Naïve Bayes, alongside more sophisticated models including KNeighbors Classifier, Linear Support Vector Classifier, Random Forest Classifier, Isolation Forest, Bagging Classifier, Decision Tree Classifier, Keras Classifier, MLP Classifier, LightGBM Classifier, XGBoost Classifier, Adaboost Classifier, Catboost Classifier, Dynamic Ensemble, and Stacking.

Each of these Machine Learning algorithms possesses its own distinct strengths and weaknesses. Logistic Regression and Gaussian Naïve Bayes are adept at binary classification tasks, with Logistic Regression modelling event probabilities based on input features, and Gaussian Naïve Bayes assuming conditional feature independence given class labels. While interpretable and computationally efficient, these models struggle in capturing intricate data relationships.

KNeighbors Classifier categorizes data points by the majority class of their nearest neighbors, potentially effective for detecting local patterns but susceptible to computational overhead. Linear Support Vector Classifier (SVC) constructs hyperplanes for class separation, proficient in linearly separable scenarios yet encountering challenges with non-linear boundaries.

The Random Forest Classifier combines multiple decision trees to enhance accuracy and feature importance interpretation, however, this model is susceptible to overfitting. Isolation Forest isolates anomalies within a tree structure, efficient for high-dimensional data but potentially less effective in lower dimensions.

Bagging Classifier employs bootstrap aggregation to mitigate variance among multiple models, beneficial for unstable models but potentially lacking robustness against outliers. Decision Tree Classifier, while interpretable, is prone to overfitting due to its inherent nature of data partitioning.

Ensemble Models such as Stacking and Dynamic Ensemble combine predictions from multiple models, enhancing robustness albeit requiring meticulous tuning. Deep Learning Models like Keras Classifier and MLP Classifier learn intricate data representations, excelling in capturing complex patterns but demanding substantial data volumes.

Gradient Boosting Models including LightGBM, XGBoost, Catboost, and Adaboost sequentially construct weak learners to refine accuracy, proficient in handling non-linearity and feature interactions albeit at a computational cost.

Each algorithm underwent rigorous evaluation, scrutinizing its capacity to accurately discern fraudulent transactions while minimizing false positives. Metrics encompassing Accuracy, Precision, Recall, and Receiver Operating Characteristic (ROC) curves were meticulously analysed to gauge performance across diverse dimensions.

The insights gained from this comprehensive comparative study shed light on the nuanced strengths and weaknesses of various Machine Learning approaches in combating fraudulent activities within credit card transactions, offering invaluable guidance for both practitioners and researchers alike.

In this paper, we utilized a dataset comprising credit card transactions that took place in September 2013 involving European cardholders. The dataset captures transactions occurring over a span of two days, during which 492 fraudulent transactions were identified out of a total of 284,807 transactions. This dataset poses a significant challenge due to its highly imbalanced nature, with fraudulent transactions accounting for a mere 0.172% of the entire dataset.

Comprising 31 numerical input variables, the dataset has been anonymized and features are denoted by names such as V1, V2, V3 up to V28. For confidentiality reasons, specific details about the cardholders and merchants have been obfuscated. Additionally, most of the feature data have been scaled to maintain consistency, except for three variables: 'Time', 'Amount', and 'Class'.

The 'Time' feature represents the elapsed time in seconds between each transaction and the first transaction recorded in the dataset. This provides insights into the temporal aspect of the transactions, aiding in the analysis of transaction patterns over time. Meanwhile, the 'Amount' feature signifies the monetary value of each transaction, enabling examination of transaction sizes and potential correlations with fraudulent activity.

'Class' is the target variable for our analysis. It assigns a value of 0 to genuine transactions and 1 to fraudulent transactions, facilitating the classification task of distinguishing between legitimate and fraudulent activities.

This dataset serves as a valuable resource for studying fraud detection techniques in real-world financial transactions. Despite its anonymized nature, the dataset offers rich insights into transaction patterns, temporal dynamics, and transaction amounts, laying the foundation for robust fraud detection methodologies."

CHAPTER II:
REVIEW OF LITERATURE

Credit card fraud inflicts substantial financial losses on consumers, eroding trust in the credit card ecosystem. An effective fraud detection system is crucial for timely intervention to restore consumer confidence. Consequently, numerous researchers have explored various solutions to detect and prevent fraud. The following section reviews some prominent work in this field:

Alam et al. (2021) evaluated machine learning classifiers such as Random Forest (RF), AdaBoost, and CatBoost for detecting credit card fraud, highlighting their high accuracy rates. They discussed the significance of feature selection in improving fraud detection performance, with RF and CatBoost showing the best results. The research used a highly imbalanced dataset from European cardholders in September 2013, emphasizing the need for careful handling of such data. The study concluded that RF and CatBoost classifiers achieved the highest accuracy, suggesting further research on different datasets to validate these findings.

Azhan & Meraj (2020) from the Jamia Millia Islamia University, India used Multiple Linear Regression, Logistic Regression, K-Nearest Neighbors, Gaussian Naïve Bayes, Random Forest and Neural Network to determine potential fraudsters using the previous mistakes and details of previous fraudsters. Model evaluation was done using precision and F1 scores. This research found K-Nearest Neighbors to be most effective in detecting Credit Card frauds. The study concluded that ML techniques were more effective in handling class imbalance compared to shallow Neural Networks.

Shirgave, Awati, More, and Patil (2019) reviewed various machine learning techniques for detecting credit card fraud and compared their performance. They discussed supervised and unsupervised algorithms, including Random Forest, Logistic Regression, KNN, SVM, Decision Tree, and Naive Bayes. A new system was proposed using the Random Forest algorithm, addressing issues like concept drift and class imbalance. According to the study's comparison, the Random Forest algorithm outperformed others in accuracy, precision, and specificity. This document provided a comprehensive analysis of different approaches to improving credit card fraud detection using machine learning.

Four researchers from University of Luxembourg proposed to create a new set of features based on analyzing the periodic behavior of the time of a transaction using the von Mises distribution. Using a real credit card fraud dataset provided by a large European card processing company, authors compared state-of-the-art credit card fraud detection models, and evaluated how the different sets of features had an impact on the results. By including the proposed periodic features into the methods, the results showed an average increase in savings of 13% (Bahnsen et al., 2015)

Another research examined and compared the performance of four traditional supervised classifiers viz. Logistic Regression, Gaussian Naïve Bayes, k-Nearest Neighbor and Decision Tree. Authors concluded that Logistic Regression and Decision

Tree performed better than other classifiers. They also found that ensemble methods performed better in case of highly imbalanced dataset. (C., 2020)

In their paper titled “Supervised Machine Learning Algorithms for Credit Card Fraudulent Transaction Detection: A Comparative Study,” Dhankhad et al. (2018) conducted a thorough investigation into the efficacy of various Machine Learning algorithms in detecting fraudulent transactions using the Kaggle Credit Card fraud dataset. The study, authored by three researchers from Lakehead University, Canada, aimed to identify classifiers with superior accuracy in predicting fraud. The findings revealed that ensemble learning exhibited marginally superior performance compared to individual Machine Learning algorithms such as Logistic Regression or Support Vector Machines.

In their research paper, Dighe et al. (2018) conducted a comparative analysis of machine learning algorithms including Naive Bayes, KNN, Decision Trees, Logistic Regression, and Neural Network Algorithms such as Multi-Layer Perceptron and Chebyshev Functional Link Artificial Neural Network. The study involved training four classifiers based on different machine learning techniques and evaluating their performance using accuracy metrics. The results showed that the performance of KNN surpassed that of other machine learning algorithms, as concluded by the authors.

Dornadula & Geetha (2019) proposed a novel fraud detection method for Streaming Transaction Data, aiming to analyze customers' past transaction details and extract behavioral patterns. The research involved clustering cardholders into different groups based on transaction amounts and extracting behavioral patterns using the Sliding

Window Strategy. Subsequently, various Machine Learning classifiers were applied to these groups. The study concluded that Logistic Regression, Decision Tree, and Random Forest yielded superior results.

In the research paper titled “Predicting Credit Card Transaction Fraud Using Machine Learning Algorithms”, Gao et. al. (2019) explored the application of linear and nonlinear statistical modeling and machine learning models on real credit card transaction data. Machine Learning models used in this research were Logistic Regression, Neural Networks, Random Forest, Boosted Tree and Support Vector Machines.

Kazemi & Zarrabi (2017) proposed a deep autoencoder to extract the best features from the information of the credit card transactions and then appended a softmax network to determine the class labels. This paper delved deep to evaluate a couple of variants of autoencoders and compared their results.

In their research paper “Credit Card Fraud Detection Using Machine Learning Models and Collating Machine Learning Models”, Khare & Sait (2018) compared four classifier models based on Logistic Regression, SVM, Decision Tree and Random Forest. Authors concluded that accuracy of Random Forest was best followed by the results of Logistic Regression, SVM and Decision Tree.

In one comparative study (Mittal & Tyagi, 2019), authors evaluated the suitability of machine learning algorithms in detecting credit card frauds. Several popular algorithms in supervised, ensemble and unsupervised categories were evaluated on different metrics. It was concluded that unsupervised algorithms handle the dataset skewness in better ways and hence perform well over all metrics absolutely and relative to other techniques.

Nadim et. al. (2020) used six Machine Learning algorithms viz. Logistic Regression, Linear Discriminant Analysis, K Nearest Neighbors (KNN), Classification Trees, Support Vector Classifier, Random Forest Classifier and XGBoost Classifier on the European Credit Card dataset. These algorithms were evaluated based on Accuracy, Sensitivity, Specificity and Precision. The research found Random Forest and XGBoost to be more apt for fraud detection.

In the research paper "Credit Card Fraud Detection Framework – A Machine Learning Perspective", Parmar et. al. (2020) explored the presentation of K-Nearest Neighbor, Decision Trees, Support Vector Machine (SVM), Logistic Regression, Random Forest, and XGBoost for credit card fraud detection. This paper concluded that K-Nearest Neighbors performed best, and Logistic Regression performed worst basis precision and F1 score.

Usage of Genetic Algorithm and Neural Network (GANN) was advocated in a research paper. (Patidar & Sharma, 2011). Back Propagation Network (BPN) was used to train the neural network and genetic algorithm was used to choose the most effective parameters for the neural network.

One study (Puh & Brkic, 2019) implemented and measured performance of three selected ML algorithms: Random Forest, Support Vector Machine and Logistic Regression. Measures used to evaluate the algorithms were area under the ROC curve

(AUC) and average precision (AP). Random Forest performed slightly better than Logistic Regression. SVM performed poorest among the group.

Pumsirirat & Yan (2018) suggested to use unsupervised learning to detect Credit Card frauds. In their research paper, they focused on fraud cases that could not be detected based on supervised learning. They suggested a model of deep Auto-encoder and Restricted Boltzmann Machine (RBM) to use normal transactions to detect fraudulent transactions in real time.

A study like the one proposed here was conducted by Rajora et al. (2018). The primary aim of the paper was to find the classifiers that have better accuracy prediction for the fraud detection. The paper found that the ensemble learning had slightly better performance than individual Machine Learning algorithms like Logistic Regression or Support Vector Machines.

S P Maniraj et al. (2019) illustrated the modelling of a dataset using Machine Learning. Available credit card transaction data was modelled using the data of the ones that turned out to be fraud. This model was then used to recognize whether a new transaction was fraudulent.

Saloni & Rout (2021) compared several machine learning models like Random Forest, Logistic Regression, Naïve Bayes and XGBoost on the European cardholder data. SMOTE was used to tackle the class imbalance problem in the dataset. Soft Voting and AdaBoost were used to compare the performance of various models. They concluded that Random Forest applied with AdaBoost along with SMOTE technique gave the best result.

Babu et al. (2020) discussed the use of machine learning algorithms such as Naïve Bayes, Logistic Regression, and AdaBoost for detecting credit card fraud. They highlighted the importance of exploratory data analysis and the role of non-anonymized predictors like time and amount in fraud detection. The paper investigated the performance of Logistic Regression, Decision Tree, and Random Forest algorithms using a dataset from Kaggle. It mentioned the use of t-SNE for visualization and the challenges of handling high-dimensional data in fraud detection. The document provided insights into various techniques and methodologies for improving the predictive power of fraud detection systems.

In one research paper (Singh et al., 2012), SVM (support vector machine) based method with multiple kernel involvement was proposed including several fields of user profile instead of only spending profile. The simulation result showed improvement in TP (True Positive) & TN (True Negative) rate and decrease in the FP (False Positive) & FN (False Negative) rate.

Researchers from Sri Lanka Institute of Information Technology applied Logistic Regression, Naïve Bayes, K-Nearest Neighbors and Support Vector Machines on a real-world credit card transactions dataset to check their effectiveness in detecting fraudulent transactions. Accuracy was used as the performance metrics and Support Vector Machine was found to be best performing model. (Thennakoon et. al., 2019)

Trivedi et al. (2020) from Institute of Engineering and Technology, Chitkara University, India tried Random Forest, Naïve Bayes Classifier, Logistic Regression, Support Vector Classifier, K-Nearest Neighbors, Decision Trees, and Gradient Boosting on the Kaggle Credit Card dataset. They measured the performance of the models using the Accuracy, Precision, Recall, F1 and False Positive Rate. The research concluded Random Forest as the best performing model among the evaluated models.

In the research paper “Credit Card Fraud Detection Using Bayesian and Neural Networks”, Tuyls et. al. (2015) applied Artificial Neural Networks and Bayesian Belief Networks on the real-world financial data. As per this comparative study, Bayesian Networks yielded better results concerning fraud detection and they took less time to train but fraud detection process was faster with Artificial Neural Networks.

Lakshmi & Kavila (2018) from Anil Neerukonda Institute of Technology and Sciences, India used European Credit Card transactions dataset to check the effectiveness of Logistic Regression, Decision Tree and Random Forest. They used oversampling to tackle the issue of data imbalance. These three Machine Learning models were evaluated using the performance metrics Accuracy, Specificity, Sensitivity and Error Rate. The comparative results showed that the Random Forest performed better than the Logistic Regression and Decision Tree techniques.

Varmedja et al. (2019) from University of Novi Sad, Serbia used Machine Learning algorithms Logistic Regression, Naïve Bayes, Random Forest, Multilayer Perceptron and Artificial Neural Network on the European card-holders data set which was downloaded from Kaggle. To manage the data imbalance, SMOTE technique was used. This paper compared the results of these Machine Learning algorithms and concluded that Random Forest was best to classify whether transactions were fraud or not.

A group of researchers from School of Computer Science, Hubei University of Technology Wuhan, China proposed a credit card fraud detection technology based on Whale Optimization Algorithm (WOA) for Back Propagation (BP) neural network aiming at solving the problems of slow convergence rate, easy to fall into local optimum, network defects and poor system stability derived from BP neural network. Using whale swarm optimization algorithm to optimize the weight of BP network, authors first used WOA algorithm to get an optimal initial value, and then used BP network algorithm to correct the error value, so as to obtain the optimal value (Wang et al., 2018)

Yu & Wang (2009) proposed a credit card fraud detection model using outlier detection based on distance sum according to the infrequency and unconventionality of fraud in credit card transaction data, applying outlier mining into credit card fraud detection. This paper concluded that such a model was feasible and accurate in detecting credit card fraud.

Zareapoor & Shamsolmoali (2015) trained various data mining techniques used in credit card fraud detection and evaluated each methodology based on certain design criteria. This study concluded that the bagging classifier based on decision tree was the best classifier to construct the fraud detection model.

Ileberi et al. (2022) from University of Johannesburg discussed the use of Machine Learning (ML) algorithms to detect credit card fraud, focusing on a system that employs the genetic algorithm (GA) for feature selection. The GA is used to select the most effective features for ML classifiers like Decision Tree, Random Forest, Logistic Regression, Artificial Neural Network, and Naïve Bayes. European cardholders data was used to demonstrate that the proposed approach outperformed existing systems in terms of accuracy. The GA-based feature selection combined with ML classifiers achieved high accuracy rates, with the GA-RF model using the selected features scored an optimal accuracy of 99.98%.

In the research titled “Application of Machine Learning and Resampling Techniques to Credit Card Fraud Detection”, Udeze et al. (2022) compared three machine learning algorithms viz. Random Forest, XGBoost, and TensorFlow Deep Neural Network (DNN) for credit card fraud detection. It explored four resampling techniques to handle imbalanced datasets: baseline train test split, class weighted hyperparameter approach, undersampling, and oversampling. The performance of these algorithms was evaluated using metrics like accuracy, precision, recall, and F1-score. The study found that TensorFlow DNN was more efficient in modeling the under-sampled dataset, while Random Forest performed better in the baseline approach. XGBoost had better performance with the oversampling technique.

The research paper titled “Credit card fraud detection using artificial neural network”, delves into the application of various machine learning algorithms, such as Support Vector Machine (SVM), K-Nearest Neighbor (KNN), and Artificial Neural Network (ANN), in forecasting credit card fraud occurrences. It juxtaposes traditional supervised machine learning approaches against deep learning methodologies to discern between fraudulent and legitimate transactions effectively. Notably, the investigation unveils that the Artificial Neural Network (ANN) exhibits superior performance compared to its counterparts, demonstrating exceptional precision in fraud prediction.

Employing a dataset sourced from European bank transactions, the study rigorously evaluates the models based on accuracy, precision, and recall metrics, offering valuable insights into the efficacy of different methodologies in combating credit card fraud. (RB and Suresh Kumar, 2021)

Tiwari et al. (2021) conducted a thorough investigation into the efficacy of various machine learning techniques in detecting credit card fraud. Their study meticulously examined approaches such as Hidden Markov Model, Decision Trees, and Neural Networks, among others, assessing their strengths and weaknesses. With a keen eye on the growing significance of fraud detection in the context of increasing e-commerce and digital transactions, the paper highlights the pressing need for robust fraud detection systems. Moreover, the research explored a range of machine learning methods including Random Forests, Bayesian Belief Networks, and Support Vector Machines for fraud detection. The study concludes by noting the absence of a universally effective technique across all scenarios, emphasizing the necessity for precision-driven technologies adaptable to diverse datasets. Although Neural Networks demonstrated high precision, their costly training requirements pose a notable challenge.

Lucas and Jurgovsky (2020) conducted a thorough investigation into the intricacies of data-driven credit card fraud detection and explored numerous machine learning techniques to tackle its complex challenges. Their aim was to detect fraudulent transactions issued illegitimately on behalf of genuine cardholders. The study addresses the formidable challenge that credit card fraud poses to electronic payment systems. It delves into data-driven detection methods designed to uncover unauthorized transactions made on behalf of legitimate cardholders. The survey commences by outlining the typical tasks involved in fraud detection, including dataset characterization, metric selection, and strategies for managing imbalanced data. Furthermore, it delves into the concept of dataset shift, where evolving transaction patterns and fraudster tactics over time present hurdles to the effectiveness of machine learning. Various methodologies are discussed for capturing the sequential nature of transactions, ranging from conventional feature engineering to sophisticated models like recurrent neural networks and hidden Markov models. The study emphasizes inherent challenges such as the infrequency of fraudulent transactions and the dynamic nature of purchase behaviors, stressing the significance of detailed sequential information often overlooked in transaction feature sets. Throughout the survey, a diverse array of strategies are explored to address these challenges, with the overarching goal of furnishing readers with a comprehensive grasp of credit card fraud detection methodologies and ongoing research advancements.

S, Varun Kumar et al. (2020) conducted an in-depth exploration into constructing a predictive model for distinguishing between fraudulent and legitimate credit card transactions, leveraging a variety of machine learning algorithms and neural networks. Their primary objective was to accurately anticipate instances of fraud by considering transaction attributes such as time and amount, employing statistical and mathematical techniques including calculus and linear algebra. Notably, their model yielded promising results, achieving an accuracy of 94.84% with logistic regression, 91.62% with naive Bayes, and 92.88% with decision tree algorithms. However, the Artificial Neural Network (ANN) model emerged as the top performer, boasting an impressive accuracy rate of 98.69%. The overarching goal of the study was to mitigate financial losses attributed to credit card fraud by effectively identifying and segregating fraudulent transactions based on the temporal and monetary characteristics embedded within the dataset.

Yee et al. (2018) examined the utilization of machine learning and data mining techniques in detecting fraudulent credit card transactions. Their study explored a range of Bayesian network classifiers, including K2, TAN, and Naïve Bayes, as well as logistics and J48 classifiers. The research emphasized the significance of data preprocessing methods such as normalization and Principal Component Analysis (PCA) in enhancing classification accuracy. Their findings demonstrated that post-preprocessing, classifiers consistently achieved accuracy levels exceeding 95%, thus highlighting the effectiveness of the proposed methodology in combating credit card fraud.

Shukur and Kurnaz (2019) explored the application of Machine Learning (ML) methodologies for detecting credit card fraud in their research paper titled "Credit Card Fraud Detection using Machine Learning Methodology." The study aimed to tackle the challenge posed by highly imbalanced datasets, where fraudulent transactions constituted a small fraction. Three ML models were deployed: Logistic Regression, K-means clustering, and Neural Network. Following preprocessing, Logistic Regression emerged as the top performer among the models. The research concluded by suggesting the exploration of more sophisticated techniques like Random Forest, Autoencoder, and Support Vector Machines for potential enhancements in fraud detection accuracy.

Bhanusri et al. (2020) delved into the utilization of machine learning algorithms for the detection of credit card fraud, centering their investigation on Logistic Regression, Naïve Bayes, and Random Forest with ensemble classifiers. Their study extensively reviewed multiple supervised learning techniques, assessing their efficacy in accurately predicting fraudulent transactions. Employing a dataset sourced from Kaggle, comprising European cardholder transactions, the authors meticulously evaluated the performance of various models based on metrics such as accuracy, precision, recall, f1 score, and support. Ultimately, their findings revealed that the Random Forest Classifier, implemented with boosting technique, surpassed other methods in effectively detecting instances of credit card fraud.

Alarfaj et al. (2022) centered their research on the detection of credit card fraud by leveraging advanced machine learning (ML) and deep learning (DL) algorithms. Addressing challenges such as public data accessibility, class imbalance, evolving fraud patterns, and high false alarm rates, the authors conducted empirical analysis utilizing the European card benchmark dataset. They initially applied ML algorithms and subsequently integrated convolutional neural network (CNN) architectures to enhance fraud detection accuracy. The proposed model demonstrated exceptional performance metrics, achieving optimized values of 99.9% accuracy, 85.71% f1-score, 93% precision, and 98% AUC Curves and outperformed existing ML and DL algorithms commonly employed in credit card fraud detection.

Nguyen et al. (2020) explored the utilization of deep learning methods in the context of credit card fraud detection, contrasting their efficacy with traditional machine learning algorithms. Their paper meticulously delineated the challenges inherent in this domain, including data scarcity, pronounced class imbalance, dynamic fraud strategies, and the prevalence of false alarms. Through empirical analysis conducted on three distinct financial datasets, the authors demonstrated the superior performance of deep learning approaches over conventional models. Their findings underscored the potential of these novel techniques to significantly enhance the efficacy of real-world credit card fraud detection systems.

The research conducted under the title "Developing a Credit Card Fraud Detection Model using Machine Learning Approaches" aimed to construct a robust credit card fraud detection model employing machine learning methodologies to discern transactions as either fraudulent or legitimate. Khan et al. (2022) employed Logistic Regression, Artificial Neural Networks, and Support Vector Machines as classifiers in their investigation. Evaluation of these classifiers involved the utilization of various performance metrics such as accuracy, precision, recall, and ROC curve analysis, revealing that Support Vector Machine exhibited superior performance compared to other models. This study addressed the challenge of imbalanced data in fraud detection by implementing resampling techniques during model assessment. The findings of this research offer a comprehensive examination of diverse machine learning approaches tailored towards the effective detection of credit card fraud.

Madhurya et al. (2022) delved into the exploration of machine learning techniques for detecting credit card fraud in their paper titled "Exploratory analysis of credit card fraud detection using machine learning techniques." They underscored the increasing necessity for such systems owing to the surge in online financial transactions and associated fraudulent activities. The study conducted a comparative analysis of various algorithms including Support Vector Machine, Gradient Boost, and Random Forest to ascertain their effectiveness in fraud detection. Furthermore, the research addressed the challenge of class imbalance in datasets and proposed solutions to enhance algorithm performance. While Logistic Regression exhibited high accuracy, the study revealed that KNN demonstrated superior classification capability for credit card fraud detection due to its adaptive learning capacity. Additionally, the paper outlined potential avenues for future research aimed at augmenting prediction accuracy.

Suryanarayana et al. (2018) conducted an extensive investigation into credit card fraud detection utilizing a range of machine learning algorithms including Naïve Bayes, Decision Tree Algorithm, K-Nearest Neighbors Algorithm, Support Vector Machines, Logistic Regression, and Artificial Neural Networks. Their study employed real-life financial transaction data from an e-commerce organization, consisting of 100,000 records. To enhance classifier performance and reduce training and operating time, the dataset underwent rigorous preprocessing. This preprocessing involved thorough exploration of the dataset's feature space and addressing the inherent imbalance within the data. Evaluation of model performance was conducted using metrics such as Accuracy, Sensitivity, and Specificity. The findings of the study revealed that Logistic Regression-based approaches demonstrated superior performance, achieving the highest accuracy among the tested algorithms.

Sharma et al. (2021) conducted a comparative analysis of machine learning models aimed at detecting credit card fraud. The study juxtaposed the efficacy of logistic regression, SVM, random forest, and neural networks in this domain. To tackle the inherent imbalance in the dataset of credit card transactions, the researchers employed the Synthetic Minority Over-sampling Technique (SMOTE). Their findings revealed that the Artificial Neural Network (ANN) model surpassed the others, achieving an impressive F1 score of 0.91.

In their study, Sulaiman et al. (2022) scrutinized machine learning (ML) methodologies deployed in detecting credit card fraud, shedding light on the exponential surge in credit card usage and the corresponding escalation in fraud incidents. The paper elucidated the pivotal role played by various ML algorithms in scrutinizing customer data to pinpoint fraudulent activities, underscoring ML's significance within the financial sector. Through a comprehensive review of literature on ML techniques for credit card fraud detection (CCFD), the authors navigated issues pertaining to data confidentiality, advocating for a hybrid solution integrating neural networks. They astutely identified challenges inherent in CCFD, including transactional diversity and data imbalance, thereby advocating for an innovative solution that not only ensures user privacy but also effectively detects fraud. This comprehensive analysis underscores the imperative of ML in combatting credit card fraud and underscores the necessity for pioneering solutions that address privacy concerns.

Olowookere, T. A., and Adewale, O. S. (2020) addressed in their research paper "A Framework for Detecting Credit Card Fraud with Cost-Sensitive Meta-Learning

Ensemble Approach" the persistent threat of credit card fraud to global financial institutions. They proposed a novel framework that integrated meta-learning ensemble techniques and cost-sensitive learning to enhance fraud detection. The framework involved training three base classifiers (Decision Tree, Multi-Layer Perceptron, and K-Nearest Neighbors) on historical credit card transaction data. Additionally, a cost-sensitive logistic regression meta-classifier was trained using predictions from the base classifiers and an instance-based cost matrix. The results indicated that the proposed ensemble classifier consistently outperformed individual base classifiers across varying fraud rates in the dataset. This approach effectively detected fraudulent transactions regardless of the proportion of fraud cases in the data. The study underscored the significance of amalgamating ensemble methods and cost-sensitive learning to bolster credit card fraud detection efforts.

In the research article "A Soft Voting Ensemble Learning Approach for Credit Card Fraud Detection", Mim, M. A., Majadi, N., & Mazumder, P. (2024) addressed the pressing issue of credit card fraud, intensified by the surge in daily fraudulent transactions. The authors highlighted the challenge posed by class imbalance, where fraudulent transactions were disproportionately fewer than legitimate ones, hindering accurate fraud detection. To tackle this, they proposed a soft voting ensemble learning approach specifically tailored for credit card fraud detection. Through rigorous evaluation against various sampling techniques including oversampling, undersampling, and hybrid sampling, the efficacy of the soft-voting approach was demonstrated. It surpassed the performance of individual classifiers, achieving notable metrics such as a false negative rate (FNR) of 0.0306, precision of 0.9870, recall of 0.9694, f1-score of 0.8764, and AUROC of 0.9936. The study underscored the importance of effective fraud detection in upholding the integrity of payment systems. By leveraging ensemble techniques, this research provided valuable insights for enhancing credit card fraud detection amidst the challenge of class imbalance.

In their study, Afriyie et al. (2023) compared three machine learning models—Logistic Regression, random forest, and decision trees—for classifying and predicting fraudulent credit card transactions. The results showed that random forest achieved the highest accuracy (96%) and an area under the curve (AUC) value of 98.9% in predicting and detecting fraud. It was observed that credit card holders above 60 years old were most affected by fraudulent transactions, which typically occurred between 22:00 GMT and 4:00 GMT. This research provided valuable insights for improving credit card fraud detection systems.

Credit card fraud detection was increasingly vital with the growing prevalence of cashless transactions (Lilhore & Patil, 2018). The paper emphasized the significance of ensuring genuine credit card usage while preventing fraudulent activities. The study explored the role of data mining and machine learning techniques in this domain, particularly in analyzing transaction history datasets to detect anomalies in spending behaviors. A hybrid approach employing AdaBoost and majority voting demonstrated promising accuracy rates in fraud detection. Additionally, the utilization of KNN algorithm and outlier detection methods was found effective in minimizing false alarms and enhancing fraud detection rates. Decision trees were employed for transaction classification based on diverse features, while deep learning models like Convolutional Neural Networks (CNNs) also showed potential. The research highlighted several challenges in credit card fraud detection, including stakeholders' reluctance to share information due to confidentiality concerns, the imperative for computational efficiency in real-time detection, and the need for adaptive approaches to counter evolving fraudster tactics. In conclusion, safeguarding financial transactions against credit card fraud necessitated the amalgamation of innovative techniques, as outlined in this survey.

The research paper by Islam et al. (2023) titled "An Ensemble Learning Approach for Anomaly Detection in Credit Card Data with Imbalanced and Overlapped Classes" revealed the challenges of detecting anomalies, such as credit card fraud, in transaction data due to overlapping class samples and imbalanced class distribution. Existing learning algorithms were noted to potentially exhibit bias towards majority class samples. The Credit Card Anomaly Detection (CCAD) model, as proposed, leveraged base learners and meta-learning ensemble techniques. It combined four outlier detection algorithms as base learners and XGBoost as the meta learner. To address data imbalance and overfitting, the model employed stratified sampling and k-fold cross-validation. CCAD demonstrated superior performance over existing approaches in detecting anomalies from minority class instances. Experimental results on two datasets, namely Credit Card Fraud and Credit Card Default Payment, illustrated the effectiveness of the proposed model in improving anomaly detection rates in credit card transactions. The study emphasized the significance of ensemble techniques in handling imbalanced data and enhancing overall performance.

In the study conducted by Osegi and Jumbo (2021), a comparative analysis of credit card fraud detection methods was performed, focusing on Simulated Annealing trained Artificial Neural Networks (SA-ANN), Hierarchical Temporal Memory (HTM), and Long Short-Term Memory (LSTM) based ANNs. It was found that HTM, inspired by the neocortex and capable of continual learning, showed competitive performance with SA-ANN, achieving an average classification performance ratio (ACPR) of approximately 1:1. Additionally, HTM outperformed LSTM-ANN by a factor of 2:1 in benchmark datasets. These findings suggested that HTM, with its real-time processing capabilities and biological inspiration, held promise for addressing the challenges of credit card fraud detection faced by financial institutions.

In the study conducted by Fang, Zhang, and Huang (2019), the authors addressed the growing concern of credit card fraud in the context of the expanding e-commerce landscape. They highlighted the inherent challenges in detecting fraudulent transactions, primarily stemming from the imbalance between normal and fraudulent data sets. The research focused on leveraging machine learning techniques for credit card fraud detection, particularly employing the Light Gradient Boosting Machine (LightGBM) algorithm. The features considered in their analysis encompassed various aspects such as customer demographics, transaction details, and categorical variables like consumption and job categories. To handle categorical features, they employed one-hot encoding, while addressing data imbalance through the Smote algorithm. Their findings indicated that LightGBM achieved a remarkable recall rate of 99% on real-world datasets. Furthermore, comparisons with other machine learning algorithms like Random Forest

(RF) and Gradient Boosting Machine (GBM) demonstrated the superior performance of LightGBM. The study also emphasized the importance of exploring fraud rings through relationship maps and advocated for further investigation using diverse real-world datasets to enhance the model's generalizability. In conclusion, the proposed LightGBM model showcased efficient credit card fraud detection capabilities with high recall rates, warranting future research focus on refining detection methodologies and identifying fraud rings.

In their study, Warghade, Desai, and Patil (2020) addressed the significant problem of credit card fraud in online transactions, noting its increased risk due to the expanding use of credit cards. They emphasized the necessity for efficient fraud detection algorithms to minimize losses in the finance industry. The researchers highlighted the highly imbalanced nature of credit card fraud datasets, where fraudulent transactions are significantly fewer than legitimate ones, impacting the learning and prediction phases of machine learning algorithms. To address this imbalance, oversampling and undersampling methods were employed, with the Synthetic Minority Oversampling Technique (SMOTE) utilized to generate synthetic minority class instances. Data transformation techniques were also implemented to reduce variance, alongside the identification of outliers based on local density deviation and the random isolation of observations to detect anomalies. Classification and regression models, including Linear models, were utilized, with the Local Outlier Factor (LOF) achieving an accuracy of 99.66%, Isolation Forest 99.74%, and Support Vector Machine (SVM) 45.84%. Evaluation metrics such as Precision, Recall, and F1 score were employed to assess the performance of each algorithm. This article explored techniques for credit card fraud detection, emphasizing the challenges posed by imbalanced datasets and the critical role of accurate fraud detection.

In the research paper “Credit Card Fraud Detection Using a New Hybrid Machine Learning Architecture”, Malik, Khaw, Belaton, Wong, and Chew proposed a novel hybrid machine learning architecture aimed at enhancing credit card fraud detection. The research entailed an investigation into seven hybrid machine learning models utilizing a real-world dataset. The evaluation metrics included recall (sensitivity), measuring the proportion of actual fraud cases correctly predicted, and precision, assessing the accuracy of predicted fraud cases. Additionally, the F1-measure, which balances precision and recall, and the misclassification rate, indicating the percentage of misclassified observations by the model, were considered. In the context of fraud detection, Type-I error (false positive) is particularly costly as it necessitates further investigation, while Type-II error (false negative) represents missed fraudulent activities, which are typically more severe than false allegations. The document extensively discussed the performance of hybrid machine learning models, highlighting Adaboost + LGBM as the standout performer in terms of AUROC measure, and deemed it the best hybrid model for the dataset under study (Malik et al., 2022).

In the research paper titled "Credit Card Fraud Detection using Pipelining and Ensemble Learning" by Bagga, S., Goyal, A., Goyal, N., & Gupta, A. (2020), the authors explored the challenges and techniques involved in detecting credit card fraud, an issue with significant implications for the financial industry. The study emphasized the dynamic nature of fraudulent behavior patterns and the inherent imbalance within datasets, complicating the detection process. Data mining emerged as a pivotal tool in addressing credit card fraud detection challenges, with various techniques such as Logistic Regression, K-Nearest Neighbors, Random Forest, Naive Bayes, Multilayer Perceptron, AdaBoost, Quadrant Discriminative Analysis, Pipelining, and Ensemble Learning being employed. Fraud detection encountered hurdles such as imbalanced datasets and the ever-evolving nature of fraudulent and legitimate transaction behaviors. To evaluate model performance, metrics including Precision, Recall, F1 Score, Matthews Correlation Coefficient (MCC), and Balanced Classification Rate (BCR) were employed. The results indicated that the proposed Ensemble Learning and Pipelining methods surpassed other classifiers in effectively identifying credit card fraud instances. The study aimed to enhance fraud detection accuracy, particularly within highly imbalanced datasets, showcasing promising outcomes for real-world application.

In the article titled "Credit Card Fraud Detection Using Unsupervised Machine Learning Algorithms" by Bodepudi (2021), the author addressed the escalating risk of credit card fraud amidst the surge in e-commerce and online transactions. This heightened risk poses significant challenges for banks tasked with detecting fraudulent activities. The article highlighted the utilization of anomaly detection techniques, particularly outlier detection, as a means to identify transactions or events deviating significantly from normal behavioral patterns. Among the unsupervised machine learning algorithms employed for this purpose were Isolation Forest, Local Outlier Factor (LOF), and One-Class SVM. In the study, Isolation Forest emerged as the top-performing algorithm, achieving an impressive accuracy rate of 99.74%. Following closely behind, LOF demonstrated an accuracy of 99.65%, while One-Class SVM yielded a comparatively lower accuracy score of 70.09%. These findings underscore the efficacy of unsupervised machine learning algorithms in bolstering credit card fraud detection efforts, offering banks and financial institutions valuable tools to combat fraudulent activities in an increasingly digitized financial landscape.

In their paper titled "Credit Card Fraud Detection Using Supervised Learning Approach," More, R. S., Awati, C. J., Shirgave, S. K., Deshmukh, R. J., & Patil, S. S. (2020) addressed the pressing issue of credit card fraud within the financial sector. They noted the limitations of traditional rule-based methods, citing their time-consuming and costly nature. The researchers advocated for the application of machine learning techniques, with a particular focus on Random Forest, as an effective means of fraud detection. Their system utilized Random Forest to classify credit card transaction alerts as either fraudulent or legitimate. Additionally, they implemented a learning-to-rank approach to prioritize alerts based on their perceived severity. Performance evaluation of the proposed system encompassed metrics such as precision, F1 score, recall, and accuracy. The results revealed an impressive accuracy rate of 97.93% on a dataset comprising 100,000 transactions. Despite encountering class imbalance issues, the model demonstrated robust performance. Further analysis showcased Random Forest's superiority over Decision Tree and Naive Bayes classifiers. This comparison underscores the efficacy of Random Forest in discerning fraudulent activities within credit card transactions. In conclusion, the study proposed a streamlined and efficient approach to credit card fraud detection through the application of machine learning techniques, offering a promising avenue for combating financial fraud.

In their paper titled "Credit Card Fraud Detection in Payment Using Machine Learning Classifiers," Mijwil, M. M., and Salem, I. E. (2020) addressed the classification challenge of fraud detection in payment transactions. They proposed the utilization of machine learning classifiers to differentiate between regular and fraudulent transactions. The authors implemented three distinct classifiers: Naïve Bayes, Decision Trees, and Bagging Ensemble Learner. These classifiers underwent evaluation using a dataset comprising over 297,000 credit card transactions sourced from Kaggle. Performance assessment metrics included Precision, Recall, and the area under the Precision-Recall Curve (PRC). The PRC area for class 0 (non-fraudulent transactions) ranged between 99.9% and 100%, indicating exceptional performance in detecting legitimate transactions. Decision Trees emerged as the top-performing classifier, achieving an accuracy rate of 94.12% specifically in predicting fraudulent transactions. In summary, the findings underscored the efficacy of the Decision Tree classifier for credit card fraud detection within the context of the study. The research contributes valuable insights toward enhancing security within payment systems, thus offering significant implications for the financial sector.

The study titled "Predictive Modelling for Credit Card Fraud Detection Using Data Analytics" (Patil et al., 2018) highlighted the global challenge posed by credit card and online net banking fraud. The researchers aimed to improve fraud detection accuracy in the face of increasing transaction volumes. They introduced a framework designed to process large datasets efficiently, integrating it with Hadoop for fraud prediction. Data extraction from Hadoop involved transforming it into raw data files, which were then used to develop analytical models. Three models—Logistic Regression, Decision Tree, and Random Forest—were utilized in the study. Evaluation of these models utilized confusion matrices, demonstrating that Random Forest outperformed Logistic Regression and Decision Tree in terms of Accuracy, Precision, and Recall. The research focused on detecting fraud in credit card transactions through the application of big data analytics and machine learning algorithms. The proposed framework aimed to achieve real-time fraud prediction while minimizing risk and ensuring high levels of customer satisfaction.

The study by Gupta et al. (2023) highlighted credit card fraud as a significant challenge for businesses due to the increasing volume of fraudulent transactions. Detecting such activities was crucial to protect clients from unauthorized charges. The study focused on modeling credit card fraud detection by employing multiple classifiers and data balancing techniques. The dataset suffered from an imbalance, which negatively affected model performance. XGBoost showed promising results, achieving a precision score of 0.91 and an accuracy score of 0.99 on the imbalanced data. Further enhancement was observed with the implementation of Random Oversampling, which raised precision and accuracy scores to 0.99 when used alongside XGBoost. The study recognized the importance of employing data sampling techniques, like Random Oversampling, to improve model performance in identifying fraudulent activities. It emphasized the significance of data balancing in achieving optimal performance for credit card fraud detection.

CHAPTER III:

METHODOLOGY - ANALYSIS AND DESIGN

3.1 Data Pre-processing

The dataset utilized in this paper highlights a significant imbalance, a factor that deserves careful consideration. Among the extensive collection of transactions totaling 284,807, merely 492 instances are classified as fraudulent. This substantial contrast in numbers underscores the complexity posed by the distribution of the dataset. Such an imbalance can pose challenges for analysis and interpretation, requiring specialized techniques to address effectively. The distribution of transactions is depicted below:

Genuine Transactions	284, 315	99.83%
Fraudulent Transactions	492	0.17%
Total Transactions	284, 807	

Table 3.1
Dataset Distribution

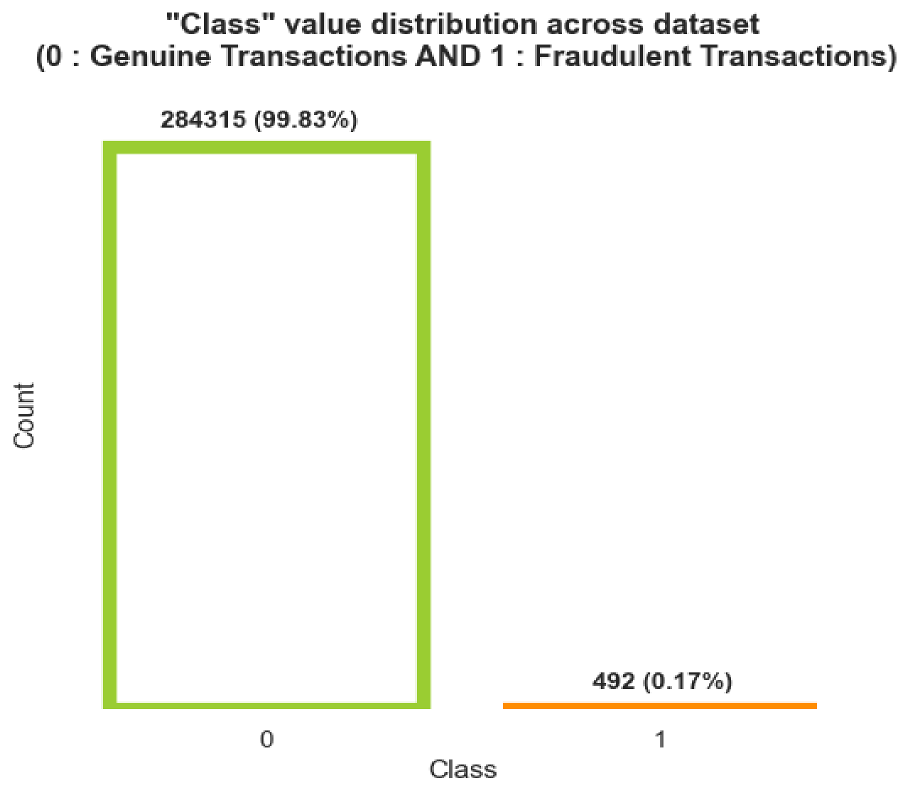

Figure 3.1
"Class" value distribution across dataset

In the dataset used for this analysis, many features have been anonymized to protect confidentiality. These anonymized features are labeled from V1 to V28 and are all numerical and scaled. However, three columns, namely "Time," "Amount," and "Class," remain unaltered.

The "Time" column indicates the time elapsed in seconds between each transaction and the first transaction recorded in the dataset. The "Amount" column denotes the transaction amount, while the "Class" column serves as the target variable. In this context, a value of 0 in the "Class" column signifies a genuine transaction, whereas a value of 1 indicates a fraudulent transaction.

Analysis of the data confirms that there are no NULL or MISSING values in the columns.

We analyze the distribution of the "Time" and "Amount" features concerning the target feature "Class" to assess their potential impact on the target variable. By examining these distributions, we aim to understand whether variations in time and transaction amounts correlate with the classification of transactions as genuine or fraudulent. The graphical representations depicting these distributions are provided below for visual reference and analysis.

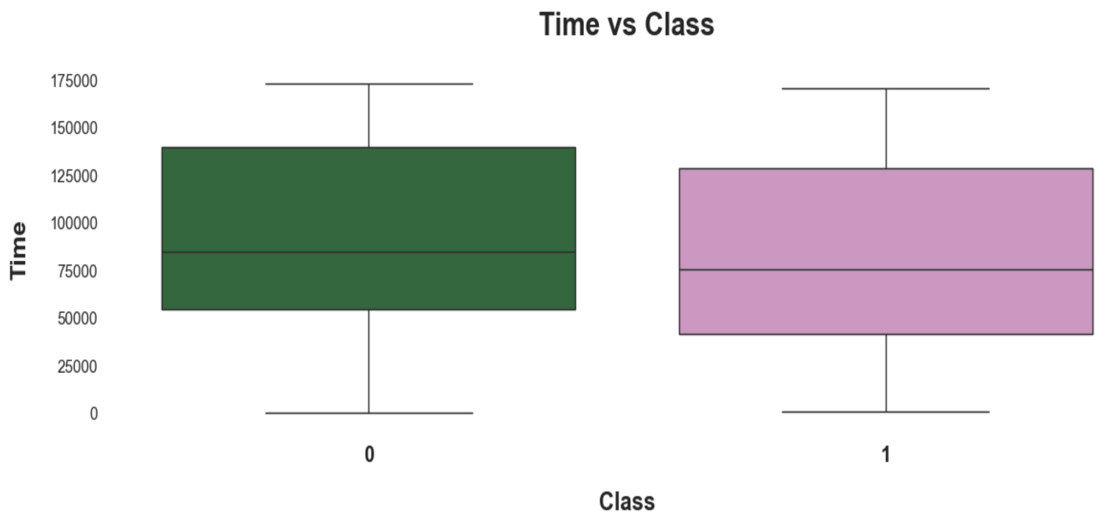

Figure 3.2
“Time” distribution against “Class”

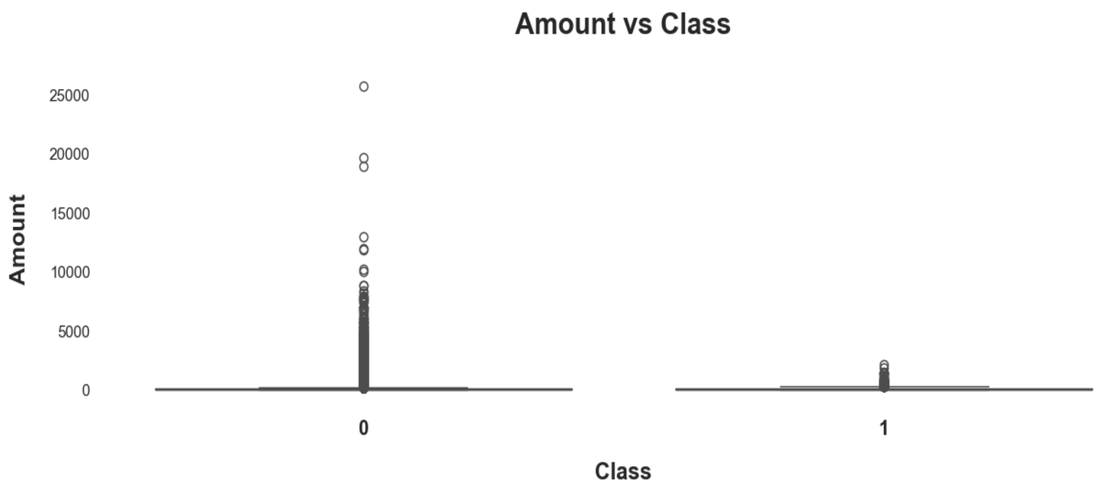

Figure 3.3
“Amount” distribution against “Class”

Above provided visual representations indicate that “Time” feature of the dataset may not have significant influence on the target feature "Class." This observation arises from the fact that the distribution of "Time" remains largely consistent across both genuine and fraudulent transactions. However, a notable disparity is observed in the distribution of the "Amount" attribute between genuine and fraudulent transactions. This discrepancy suggests that the transaction amount may indeed play a role in distinguishing between genuine and fraudulent transactions.

Both “Amount” and “Time” columns are not normalized as is visible from the numeric analysis of this column.

Amount	
Count	284,807
Mean	88.349619
Standard Deviation	250.12011
Minimum	0
25% Percentile	5.6
50% Percentile	22
75% Percentile	77.165
Maximum	25,691.16

Table 3.2
"Amount" column before scaling

Time	
Count	284,807
Mean	94,813.859575
Standard Deviation	47,488.145955
Minimum	0
25% Percentile	54,201.5
50% Percentile	84,692.0
75% Percentile	139,320.5
Maximum	172,792

Table 3.3
"Time" column before scaling

Feature scaling plays a crucial role in enhancing the performance of Machine Learning algorithms by enabling them to effectively identify patterns within the dataset and make accurate predictions. As such, we employ the min-max method to scale the "Amount" and "Time" columns.

After normalization, a detailed numerical analysis of the "Amount" and "Time" column is conducted to gain insights into its distribution and characteristics. This analysis aids in understanding the range, distribution, and statistical properties of transaction amounts, which are essential for further analysis and modeling.

Amount	
Count	284,807
Mean	0.003439
Standard Deviation	0.009736
Minimum	0.000000
25% Percentile	0.000218
50% Percentile	0.000856
75% Percentile	0.003004
Maximum	1.000000

Table 3.4
"Amount" column after scaling

Time	
Count	284,807
Mean	0.548717
Standard Deviation	0.274828
Minimum	0.000000
25% Percentile	0.313681
50% Percentile	0.490138
75% Percentile	0.806290
Maximum	1.000000

Table 3.5
"Time" column after scaling

The visual representations provided below illustrate the distribution of the "Amount" and “Time” columns both before and after the scaling process. These figures offer insights into how the values of the "Amount" and “Time” attributes are distributed across the dataset before and after applying the scaling technique. This analysis aids in understanding the impact of scaling on the distribution of transaction amounts and its implications for subsequent data analysis and modeling.

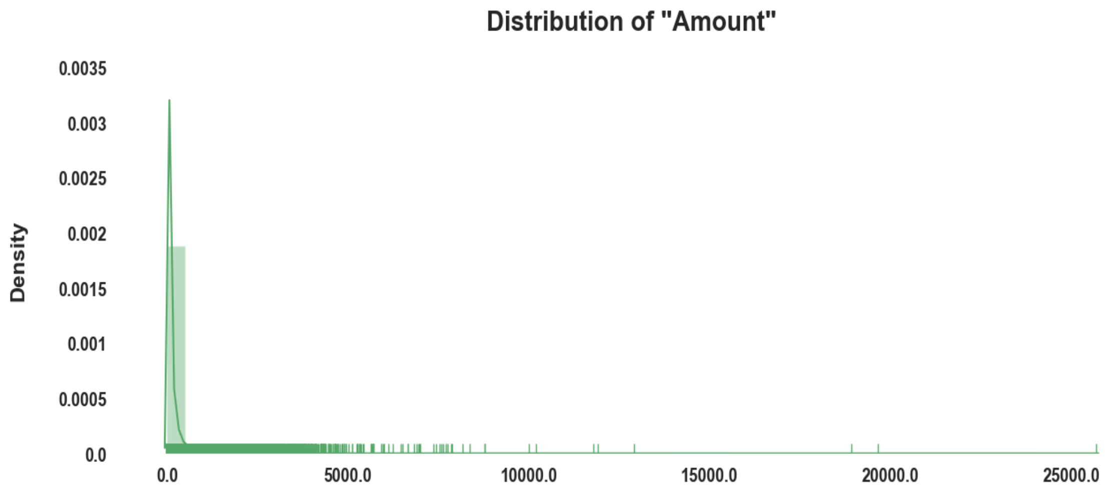

Figure 3.4
Distribution of “Amount” before Scaling

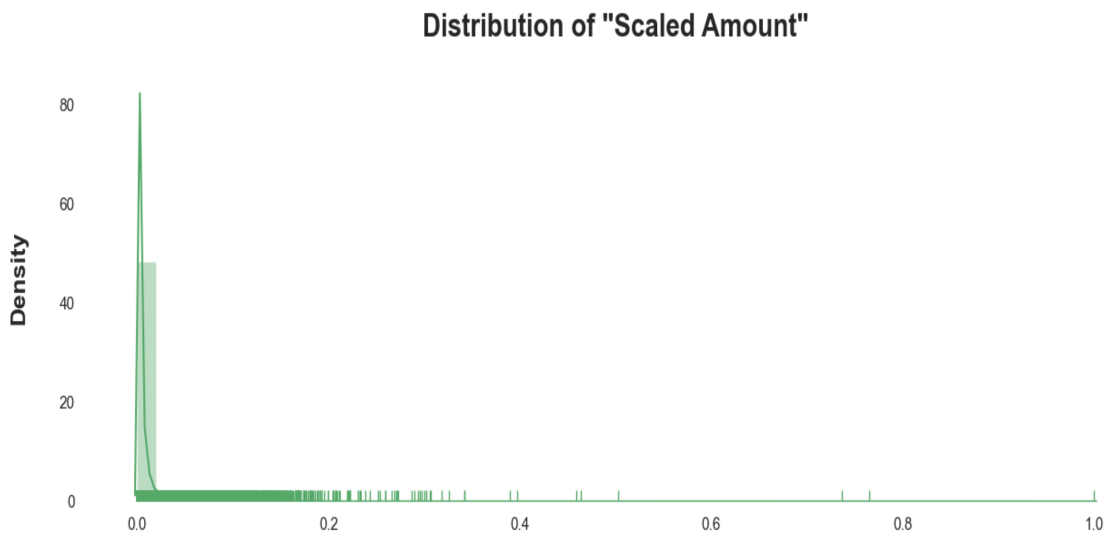

Figure 3.5
Distribution of "Amount" after Scaling

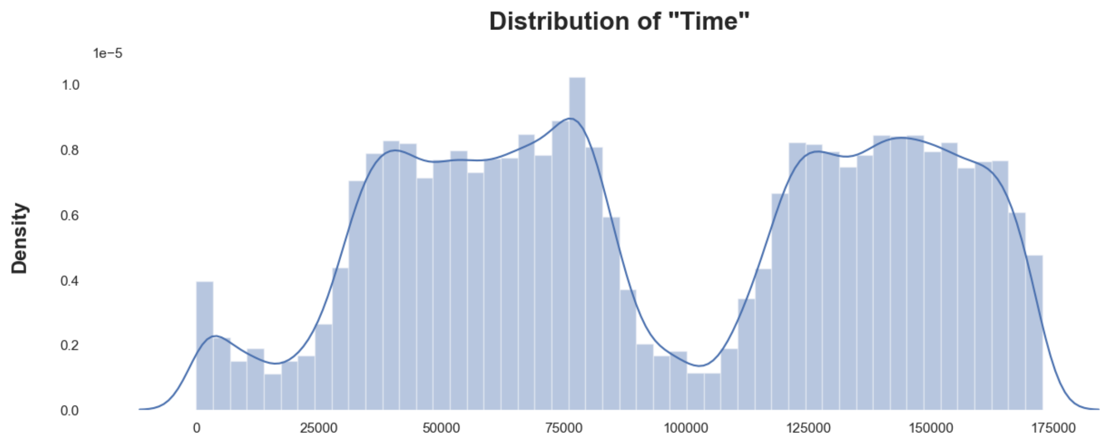

Figure 3.6
Distribution of "Time" before Scaling

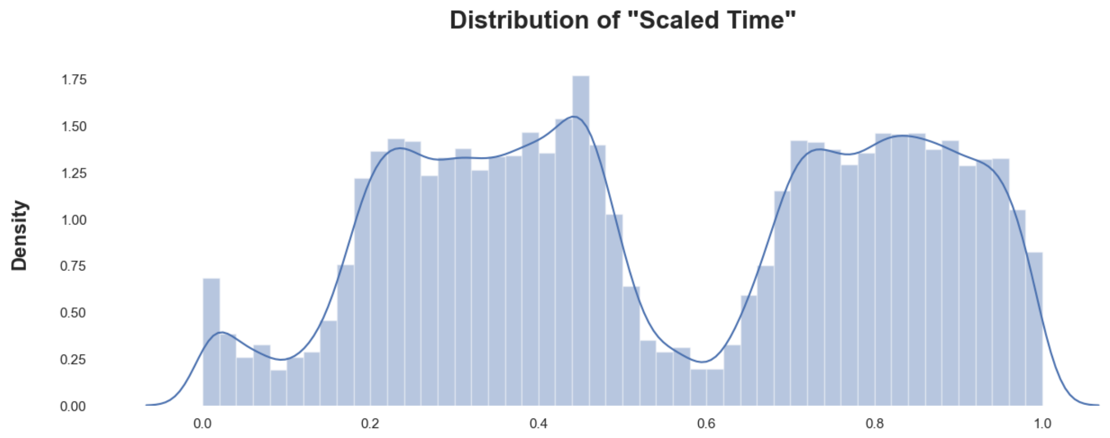

Figure 3.7
Distribution of "Time" after Scaling

3.2 Outliers Identification, Treatment and Removal

With all features now scaled, the subsequent step involves identifying and addressing any outliers present in the data..

Outliers, those elusive data points that stray far from the norm within a dataset, wield significant influence on the integrity and accuracy of analyses. These exceptional data points exhibit characteristics that diverge markedly from the majority of observations, thereby disrupting the expected distribution of the data. Their presence can skew fundamental statistical parameters like the mean and standard deviation, rendering them less representative of the dataset as a whole.

In the realm of machine learning, where models collect patterns from training data to make predictions, outliers pose a formidable challenge. When models are trained on datasets riddled with outliers, they risk learning from distorted representations of the data.

As a consequence, the predictive accuracy and performance of these models may suffer, as they struggle to discern genuine patterns amidst the noise introduced by outliers.

To mitigate these adverse effects, it is imperative to proactively identify and address outliers prior to feeding the data into machine learning algorithms. By doing so, researchers can ensure that the models are trained on clean, representative data, thereby enhancing their ability to generalize and make accurate predictions on unseen data. Various techniques, ranging from statistical methods to advanced algorithms, can be employed to detect and manage outliers effectively, safeguarding the integrity and efficacy of machine learning analyses.

The box plot, a staple visualization technique in data analysis, offers a comprehensive view of the distribution and dispersion of data points within a dataset. It provides a clear depiction of the median, quartiles, and any potential outliers present across various features.

In the context of the Credit Card fraud dataset, employing box plots allows for a detailed examination of each feature's distribution in relation to the "Class" variable, which denotes whether a transaction is genuine or fraudulent. By plotting box plots for all features against the "Class" variable, researchers can discern any significant deviations or outliers that may exist within the dataset.

These box plots provide valuable insights into the spread of data points within each feature and how they correlate with the occurrence of fraudulent transactions. Visual inspection of the box plots can help identify any anomalies or irregularities that warrant further investigation, ultimately enhancing the robustness and reliability of the analysis conducted on the dataset.

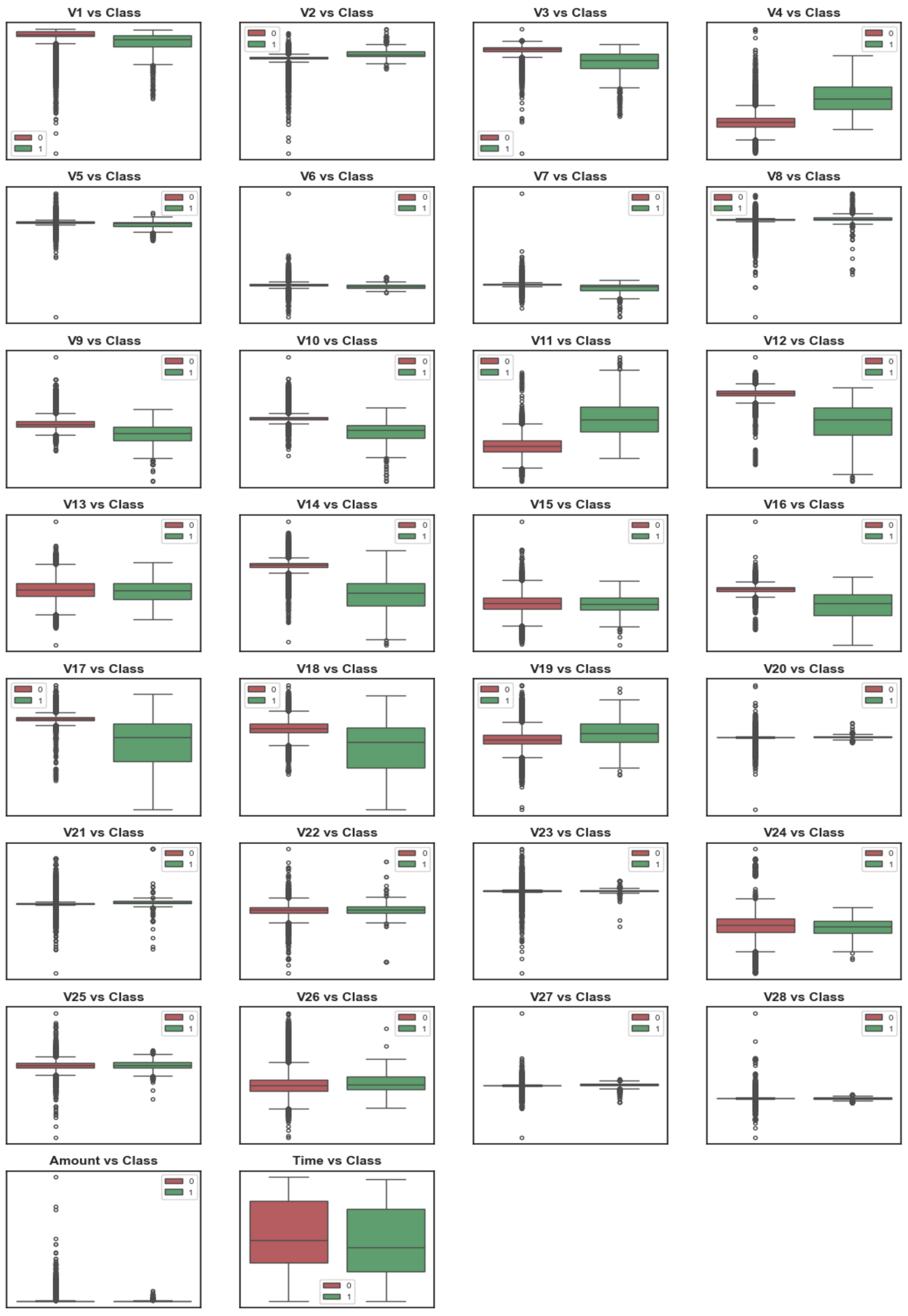

Figure 3.8
Boxplots – before treatment of outliers

The figure above distinctly illustrates that several features exhibit outliers when compared against the "Class" values of 0 and 1. Therefore, it becomes imperative to address these outliers in order to enhance the dataset's suitability for a wide range of Machine Learning algorithms.

In this research, Inter Quartile Range (IQR) method is used for the detection and treatment of outliers. In this method, data is sorted in ascending order and then divided into 4 equal parts.

First part:	0 to 25th percentile of data
Second part:	25th to 50th percentile of data
Third part:	50th to 75th percentile of data
Fourth part:	75th to 100th percentile of data

The data at 25th, 50th and 75th percentiles are denoted by Q1, Q2 and Q3 respectively.

Inter Quartile Range (IQR) is calculated as the difference between Q3 (75th percentile) and Q1 (25th percentile) of the dataset.

The IQR can be used to identify outliers by defining limits on the data points that are a factor "k" of the IQR below the 25th percentile (Q1) or above the 75th percentile (Q3). This method sets up the normal data range with lower limit as (Q1 – k * IQR) and upper limit as (Q3 + k * IQR). Any data point that is outside this data range is considered as outliers. The common value for the factor "k" is the value 1.5. A factor “k” of 3 or more can be used to identify values that are extreme outliers.

In the Credit Card fraud dataset, fraudulent records are very less in the numbers. So, a high value of “k” needs to be applied. This way only very extreme outliers in the fraudulent records will be treated. For genuine records, a lower value of “k” is being used.

For this research, “k” values of 1.5 and 3 are used for genuine and fraudulent records respectively.

Outlier identification and treatment is a multi-step process. At high level, below are the main steps:

1) Identification of outliers for each feature against the column “Class” using IQR method

2) Capture the feature name, “Class” value and the outlier percentage for each pair of feature and “Class” values where outliers are present in the dataset

3) Treat the outliers captured in the above Step 2 using the Quantile based flooring and capping

4) Repeat Step 1 & 2 to identify and capture the extreme outliers present in the dataset after the treatment carried out in Step 3

5) Remove the extreme outliers from the dataset identified in Step 4

Step 1: Using the IQR method as described above, outliers are identified in the dataset against the feature “Class”. First outliers % in each feature of the dataset calculated using the IQR method for Class=0 (Genuine records) and Class=1 (Fraudulent records). For the features where the outliers % is higher than the pre-defined Outliers Threshold then those features are considered for outliers treatment in Step 2. For this research, Outliers Threshold for Genuine records (Class=0) and Fraudulent records (Class=1) are 1% and 10% respectively. So, if any feature having more than 1% data identified as outliers by IQR method for Class=0 then that feature is identified for the outliers treatment. Similarly, if any feature having more than 10% data identified as outliers by IQR method for Class=1 then that feature would be treated for outliers.

Step 2: Result of above Step 1 captured below.

Outliers % in features evaluated for Genuine records (Class=0) (Outliers Threshold: 1%)

Sl. No.	Feature Name	Outliers %
1	V27	13.72
2	Amount	11.21
3	V28	10.62
4	V20	9.72
5	V8	8.43
6	V6	8.03
7	V23	6.48
8	V12	5.29
9	V21	5.02

10	V14	4.85
11	V2	4.69
12	V5	4.26
13	V4	3.84
14	V19	3.56
15	V10	3.21
16	V7	3.05
17	V9	2.85
18	V16	2.78
19	V18	2.58
20	V17	2.48
21	V1	2.43
22	V26	1.97
23	V25	1.87
24	V24	1.68
25	V13	1.18
26	V3	1.09
27	V15	1.01

Table 3.6
Outliers % in features for Genuine records

Outliers % in features evaluated for Fraudulent records (Class=1) (Outliers Threshold: 10%)

Sl. No.	**Feature Name**	**Outliers %**
1	V8	11.38

Table 3.7
Outliers % in features for Fraudulent records

Boxplots for the identified features against Class values of 0 or 1 given below.

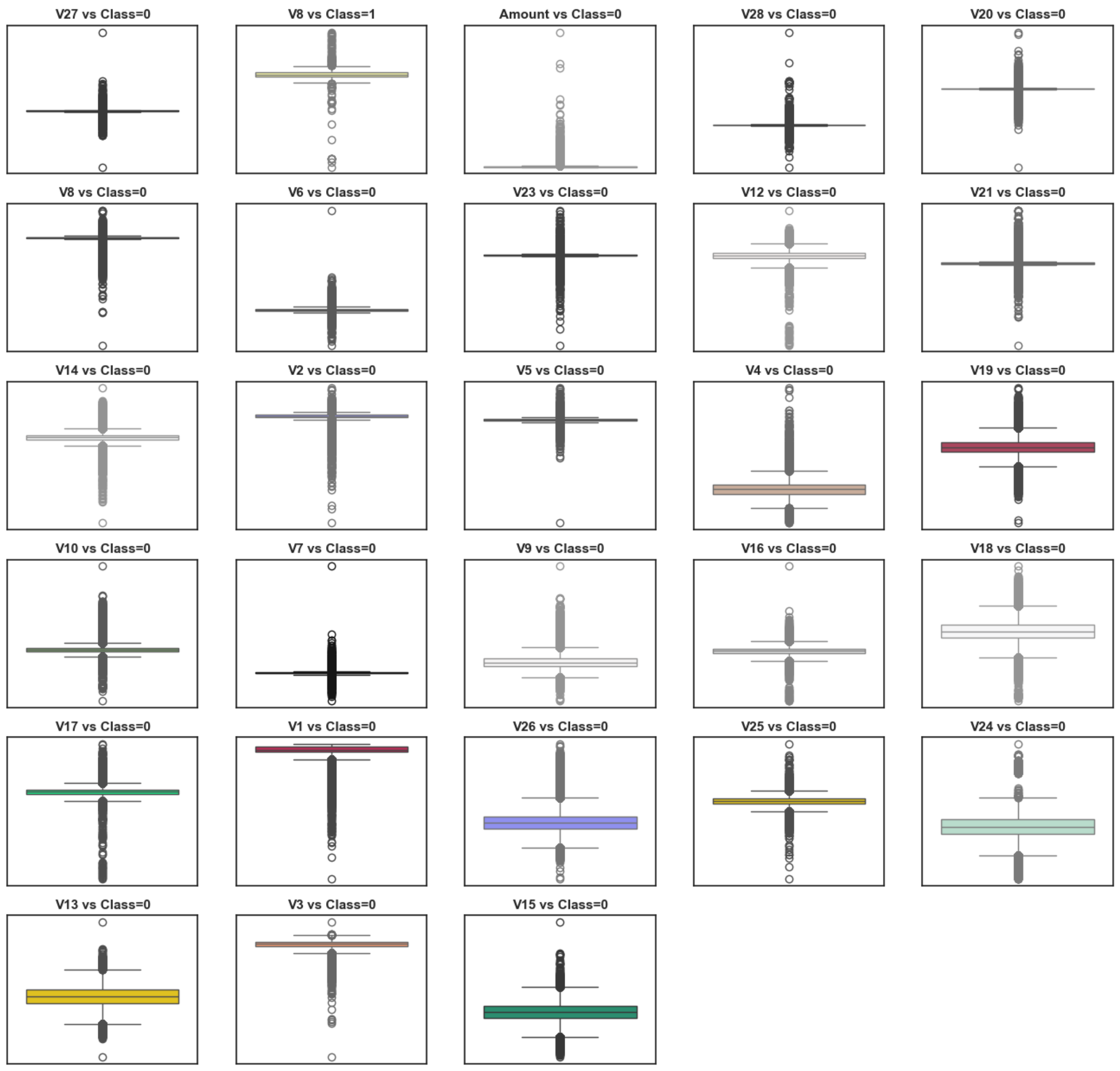

Figure 3.9
Boxplots for features which need outliers treatment

Step 3: To treat the outliers identified in the above Step 2, we would use Percentile based flooring and capping method. In this method, outliers are capped at a certain value above or equal to the 90th percentile value or floored at a factor below or equal to the 10th percentile value. These limits are called Upper Percentile and Lower Percentile respectively. Any data points that are lesser than the Lower Percentile are replaced with the Lower Percentile value and any data points that are higher than the Upper Percentile are replaced with the Upper Percentile value. This method applied on each pair of features and "Class" from the identified list in Step 2.

For this research, below mentioned limits are considered.

Genuine records (Class=0):

Upper Percentile = 90%   Lower Percentile = 10%

Fraudulent records (Class=1):

Upper Percentile = 95%   Lower Percentile = 5%

Step 4: after the outlier treatment in Step 3, above Steps 1 & 2 repeated to identify and capture the extreme outliers present in the dataset. List of extreme outliers given below:

Sl. No.	Feature Name	Class Value	Outliers %
1	Amount	0	11.21

Table 3.8
Extreme Outliers % in features after outliers treatment

These extreme outliers remain in the dataset even after performing the Percentile based Capping and Flooring in Step 3.

Step 5: Extreme outliers are removed from the dataset using the Inter Quartile Range (IQR) method as described above. In total, 31,861 genuine records are removed from the dataset as part of the treatment of the extreme outliers. Dataset size reduced from 284,807 to 252,946. Genuine and Fraudulent records count changed from 284,315 and 492 to 252,454 and 492 respectively.

After implementing outlier treatment techniques, such as removing or adjusting extreme values, we can examine the boxplots of all the features in the Credit Card dataset in relation to the "Class" variable. These boxplots provide a visual representation of the distribution of each feature for both fraudulent and genuine transactions.

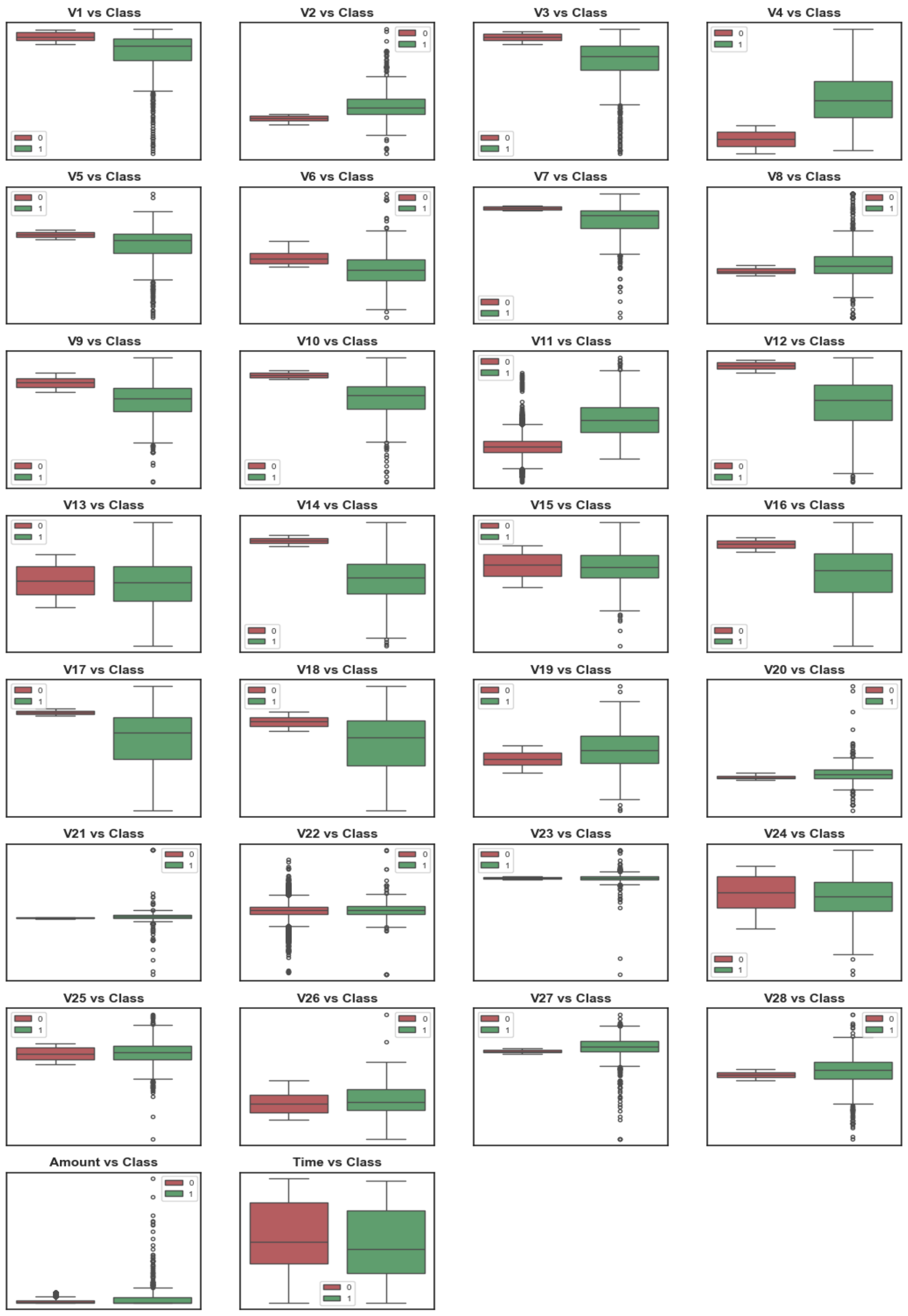

Figure 3.10
Boxplots – after treatment of outliers

After the treatment of outliers, the Boxplots demonstrate a considerable enhancement in the dataset's outlier situation. This enhancement is especially prominent in the depiction of genuine transactions (Class=0), where the identification and handling of outliers were more thorough in comparison to fraudulent transactions (Class=1). The reason behind this less stringent approach towards fraudulent transactions lies in their lower frequency within the dataset, as opposed to genuine transactions. Consequently, there's a greater likelihood of sacrificing valuable insights from fraudulent transactions if outlier treatment is excessively aggressive.

3.3 Data Correlation

The figure above illustrates the correlation among the different features in the Credit Card fraud dataset. Correlation analysis helps in understanding the relationships between variables, which is crucial for building accurate predictive models. A correlation value close to 1 indicates a strong positive correlation, while a value close to -1 indicates a strong negative correlation. Features with a correlation value close to 0 indicate little to no correlation.

Examining the correlation matrix allows us to identify potential multicollinearity issues, where two or more features are highly correlated with each other. Multicollinearity can lead to unstable and unreliable model estimates. Therefore, it's essential to address multicollinearity by either removing one of the correlated features or using dimensionality reduction techniques such as principal component analysis (PCA).

Additionally, understanding the correlation structure of the dataset helps in feature selection, as highly correlated features may provide redundant information to the model. By selecting features that are less correlated with each other but highly correlated with the target variable (fraudulent transactions in this case), we can improve the model's predictive performance.

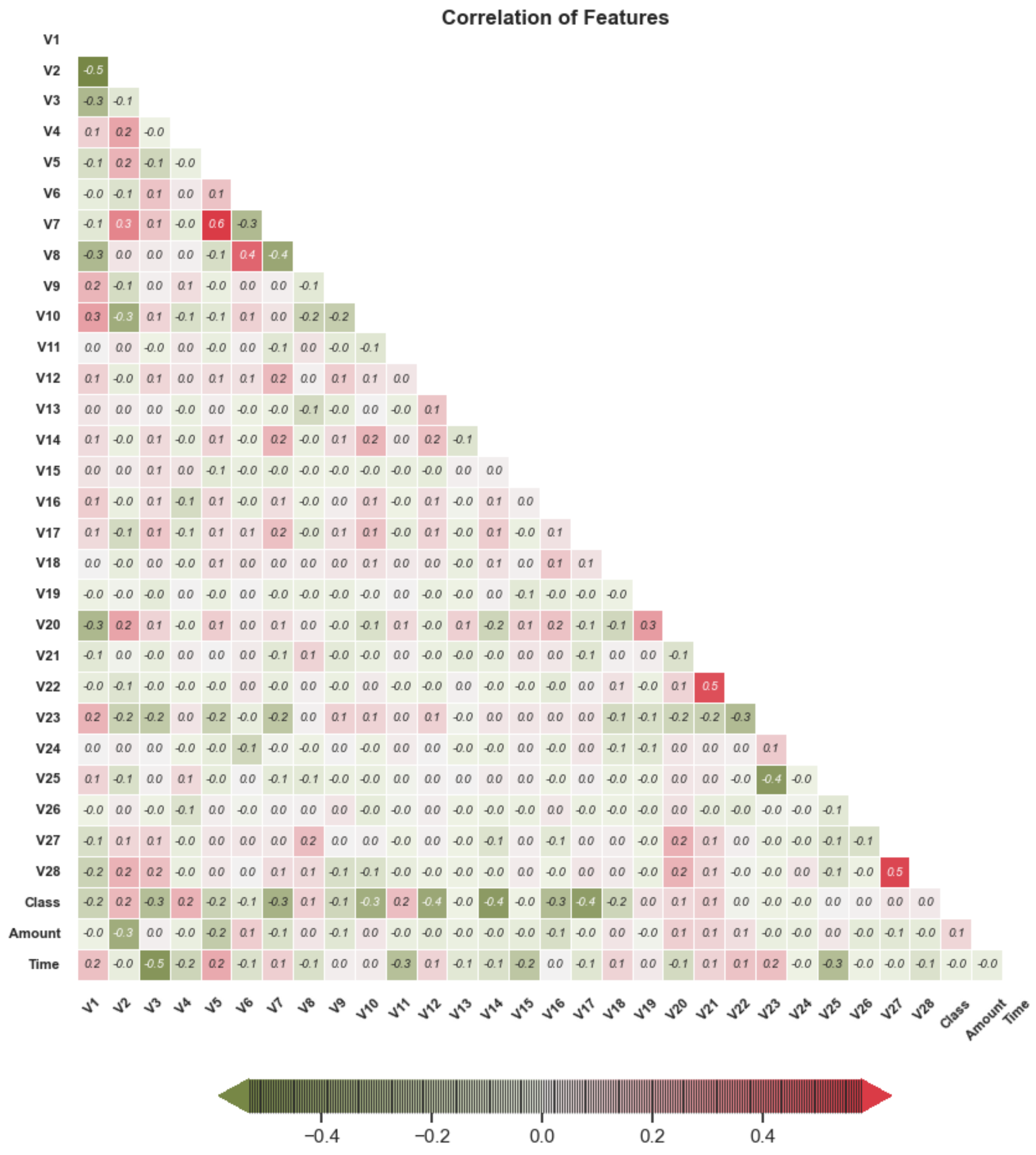

Figure 3.11
Correlation in the dataset

Below plot displays the correlation of all features in the Credit Card fraud dataset against the column "Class".

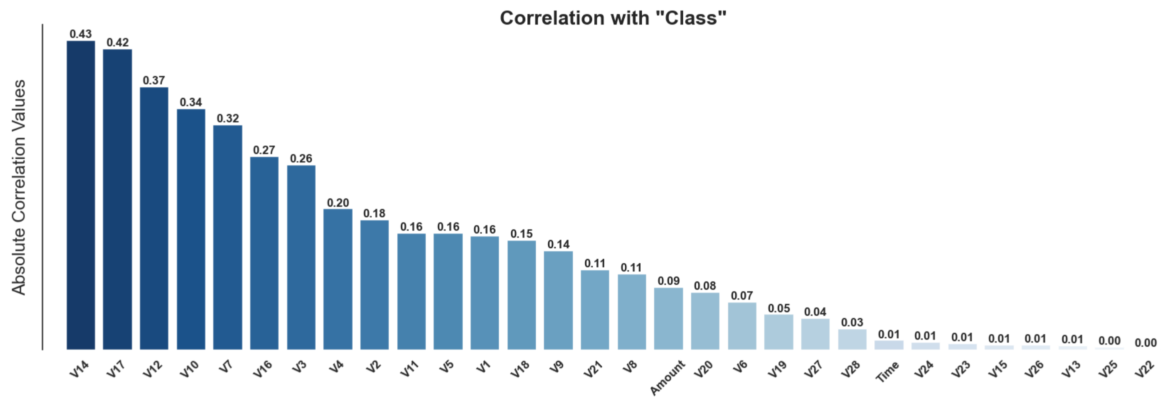

Figure 3.12
Correlation of features with "Class" variable

After analyzing the plots, it's evident that some features exhibit correlations with each other. However, none of these feature pairs have a correlation coefficient exceeding 0.6, which is the threshold considered for identifying strong correlations. As a result, there's no notable multicollinearity concern in this dataset that would necessitate further attention or treatment.

3.4 Data Imbalance

The imbalance in the Credit Card fraud dataset, with only 492 out of 252,946 records being fraudulent, raises concerns about the effectiveness of Machine Learning algorithms. When such a significant disparity exists between the number of fraudulent and genuine transactions, it can skew the algorithm's ability to accurately detect fraudulent activity. With the vast majority of transactions being legitimate, the model may struggle to identify the relatively rare instances of fraud. This imbalance underscores the importance of addressing class imbalance issues in the dataset preprocessing stage to ensure the robustness and reliability of the Machine Learning model.

Genuine Transactions	252,454	99.81 %
Fraudulent Transactions	492	0.19 %
Total Transactions	252,946	

Table 3.9
Data imbalance in the dataset

The following figure illustrates the imbalance in the dataset concerning the target column "Class":

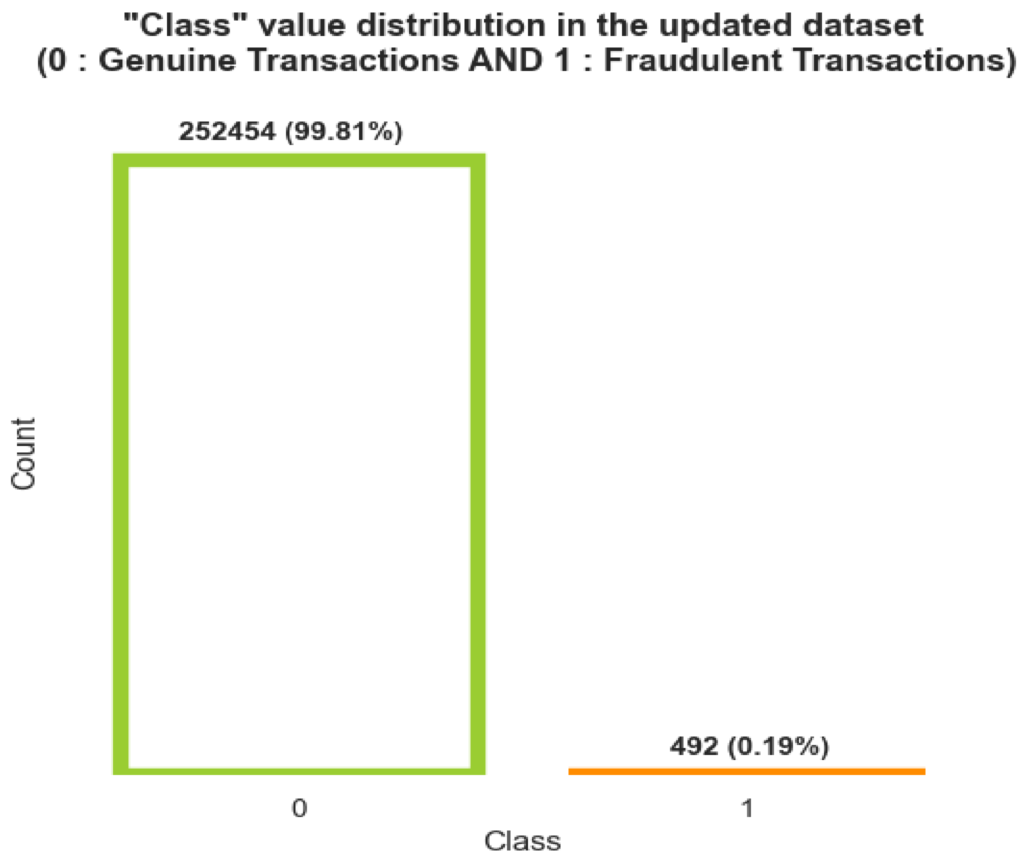

Figure 3.13
Data imbalance in the dataset

In an imbalanced dataset, the distribution of classes is heavily skewed, with one class significantly outnumbering the other. This creates a challenge for Machine Learning algorithms because they are trained to minimize errors during predictions. When faced with imbalanced data, classifiers tend to prioritize accuracy on the majority class at the expense of the minority class.

In our specific case, where only 0.19% of transactions are fraudulent while the rest are genuine, classifiers may become overly biased towards predicting transactions as genuine. This bias arises because predicting all transactions as genuine would result in a high overall accuracy due to the overwhelming dominance of the majority class. For instance, if the classifier simply labels every transaction as genuine, it would be correct about 99.81% of the time, just by guessing the majority class.

However, this approach completely overlooks the minority class of fraudulent transactions, resulting in poor detection of actual fraud. As a result, the classifier's performance in identifying fraudulent transactions, which is often the primary concern, may suffer significantly.

Achieving a balanced dataset isn't just a preference; it's strongly advised before utilizing any Machine Learning algorithm.

To tackle the problem of data imbalance, we predominantly use following techniques:

1) Under sampling

This technique, known as undersampling, aims to create a more balanced distribution of classes by reducing the number of instances belonging to the majority class. By selectively removing instances from the majority class, the dataset's class distribution becomes more equitable, with a closer alignment between the number of instances in the majority and minority classes. Undersampling is a strategy commonly employed to address imbalanced datasets, particularly when the majority class vastly outnumbers the minority class.

2) Over sampling

In oversampling, the goal is to bolster the representation of the minority class in the training dataset. This can be accomplished by either replicating existing instances of the minority class or by generating new synthetic examples that resemble the minority class instances. By increasing the number of minority class examples, the dataset becomes more balanced, mitigating the effects of class imbalance and improving the performance of machine learning models.

These techniques help to bring appropriate balance in the training dataset before the dataset is fed into Machine Learning algorithms.

Combining over-sampling and under-sampling techniques creates a more balanced approach to tackling class imbalance in the dataset. While under-sampling reduces the majority class instances to level the playing field, it risks losing crucial information inherent in those samples. On the other hand, over-sampling duplicates minority class instances to ensure their representation, potentially leading to an overemphasis on these instances and model overfitting. However, by employing both methods simultaneously, we mitigate these drawbacks and achieve a more nuanced solution. This hybrid approach preserves essential information from the majority class while adequately representing the minority class, thereby enhancing the model's robustness and accuracy in addressing class imbalance issues.

In this study, we implemented the SMOTEENN technique from the imbalanced-learn library. SMOTEENN combines two approaches: SMOTE, an over-sampling method, and ENN, an under-sampling technique.

SMOTE, or Synthetic Minority Oversampling Technique, is a popular method employed to tackle class imbalance in datasets. SMOTE’s functioning detailed below:

1. Selection of Instances: SMOTE begins by randomly selecting an instance from the minority class, which is the class with fewer samples.

2. Nearest Neighbors: Once an instance is chosen, SMOTE identifies its k nearest neighbors from the same class. The value of k is a parameter set by the user.

3. Synthetic Sample Creation: From the selected neighbors, SMOTE randomly picks one neighbor and creates a synthetic sample. This synthetic sample is generated by calculating the difference between the selected neighbor and the chosen instance. The difference is then multiplied by a random value between 0 and 1, and added to the chosen instance's feature values. This process effectively generates a new sample that lies

somewhere along the line connecting the chosen instance and its selected neighbor in the feature space.

By repeating this process for multiple instances in the minority class, SMOTE effectively increases the number of samples in the minority class, thereby helping to balance the dataset.

The Edited Nearest Neighbors Rule (**ENN**) is a technique utilized to identify and eliminate ambiguous and noisy examples within a dataset. Here's how ENN operates:

1. Nearest Neighbors Calculation: ENN begins by computing the three nearest neighbors for each record in the dataset. These neighbors are determined based on their similarity in feature space.

2. Misclassification Detection: Once the nearest neighbors are identified, ENN checks if the selected record is misclassified by its three nearest neighbors. If the selected record is incorrectly classified by its neighbors, indicating ambiguity or noise in the dataset, it is flagged for removal.

3. Under-sampling Procedure: ENN is primarily applied to the examples of the majority class. Any majority class examples that are misclassified as belonging to the minority class by their nearest neighbors are removed from the dataset. This helps in balancing the class distribution by eliminating potentially misleading majority class examples.

SMOTEENN, which stands for Synthetic Minority Oversampling Technique combined with Edited Nearest Neighbours, is a powerful approach employed in this research to address the issue of imbalanced datasets. By integrating the capabilities of both SMOTE and ENN, SMOTEENN effectively generates synthetic examples of the minority class while also eliminating noisy and ambiguous instances from the majority class. This comprehensive technique ensures that the training dataset maintains a balanced representation of both classes, thereby enhancing the performance and reliability of machine learning models.

3.5 Data Preparation

Partitioning the dataset into training and test datasets is a crucial step in machine learning model development. It involves splitting the available data into two subsets: one for training the model and the other for evaluating its performance.

The training dataset is used to train the machine learning model, allowing it to learn patterns and relationships within the data. This involves feeding the model with input data and corresponding target labels, enabling it to adjust its parameters to minimize prediction errors.

On the other hand, the test dataset is kept separate from the training data and is used to assess the model's performance. It serves as a proxy for real-world data that the model has not seen before. By evaluating the model's predictions on this unseen data, we can estimate how well it generalizes to new, unseen instances.

Splitting the dataset into training and test sets helps prevent overfitting, where the model learns to memorize the training data rather than generalize to new data. It also allows us to obtain an unbiased estimate of the model's performance on unseen data, which is crucial for assessing its real-world applicability.

First the test dataset is carved out of the main dataset using the below logic.

In preparing the test dataset, we adopted a specific approach. We extracted half of the fraudulent transactions while tripling the number of genuine transactions. Consequently, the test dataset comprised a total of 246 fraudulent transactions and 738 genuine transactions. This strategy ensured a balanced representation of both types of transactions for effective testing of our machine learning models. Distribution of the test dataset is given below:

Genuine Transactions	738	75 %
Fraudulent Transactions	246	25 %
Total Transactions	984	

Table 3.10
Distribution of "Class" in the Test dataset

The distribution of the test dataset, comprising 246 fraudulent transactions and 738 genuine transactions, is visualized in the figure below.

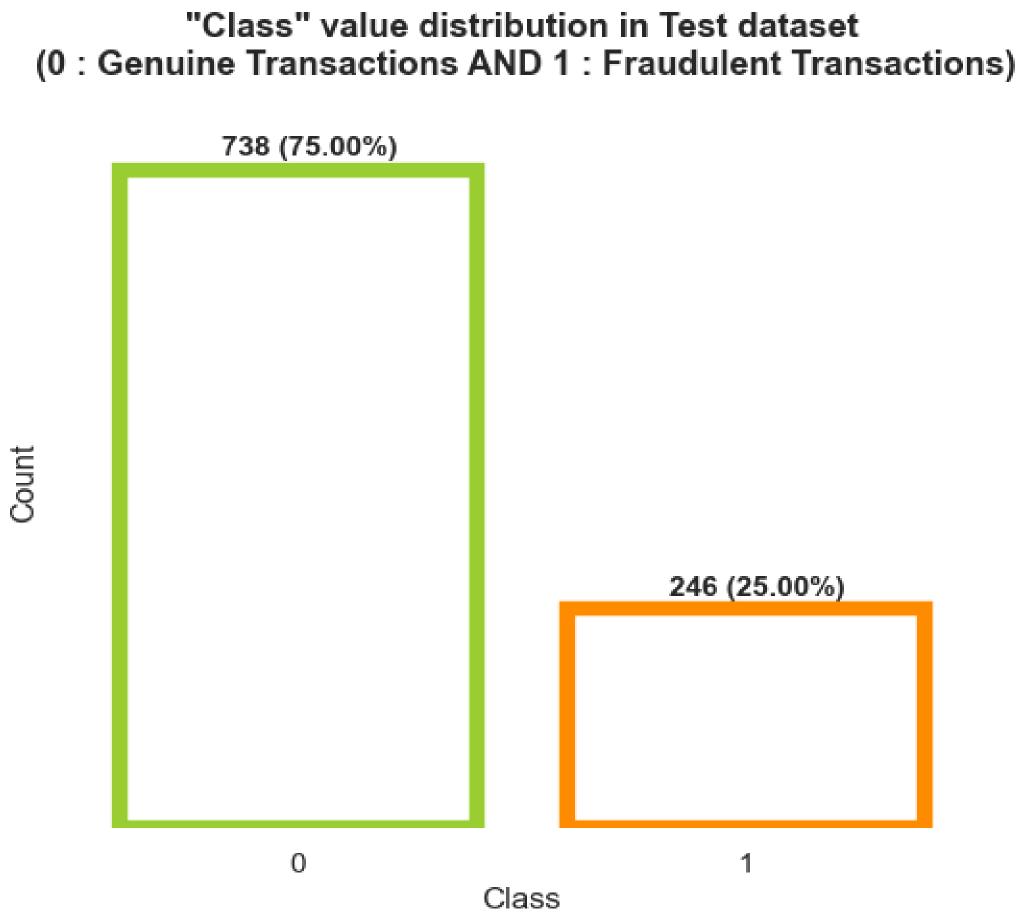

Figure 3.14
Distribution of "Class" in the Test dataset

For the training dataset, we've picked 251,716 genuine transactions and included the remaining 50% of fraudulent transactions, making sure to cover a wide range of deceptive behaviors. We carefully selected genuine transactions from the entire dataset, excluding those already set aside for testing. So, our training set has 251,716 genuine transactions and 246 carefully chosen fraudulent transactions.

The distribution of the training dataset before balancing is illustrated below:

Genuine Transactions	251,716	99.90 %
Fraudulent Transactions	246	0.10 %
Total Transactions	251,962	

Table 3.11
Distribution of "Class" in the Train dataset before balancing

The distribution is illustrated in the figure below.

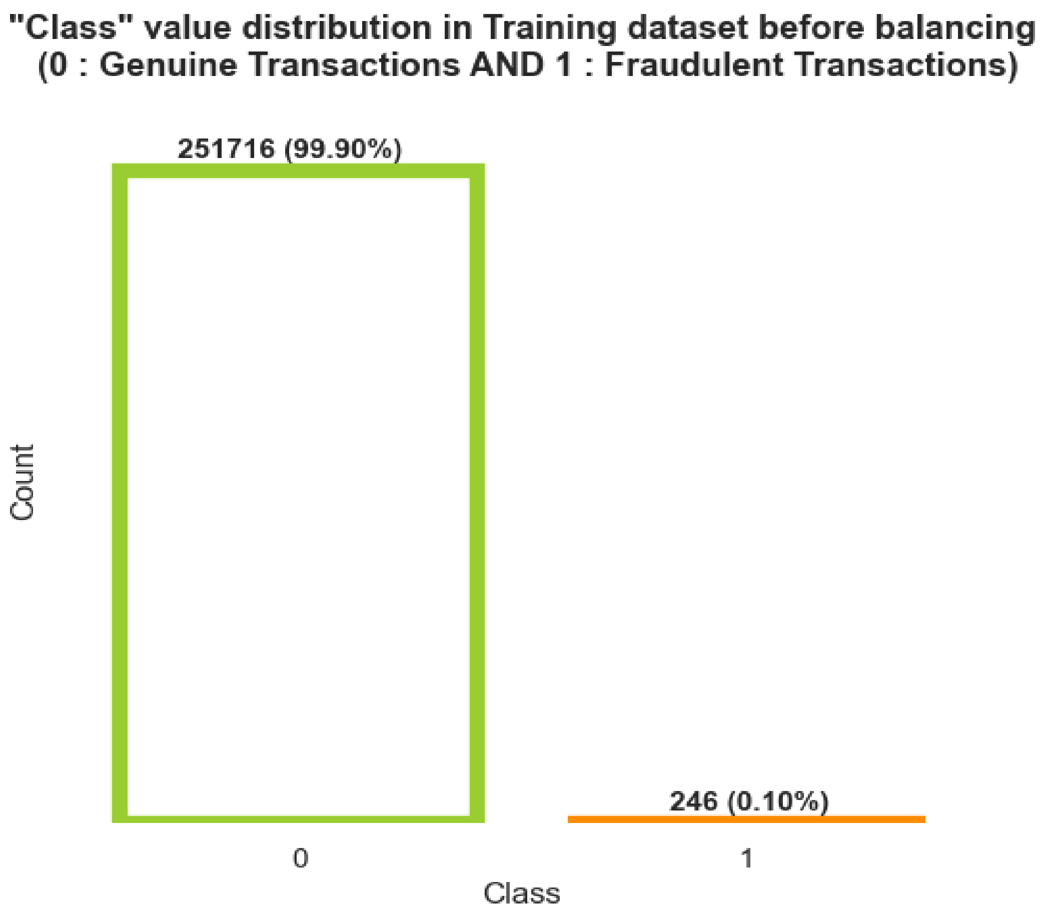

Figure 3.15
Distribution of "Class" in the Train dataset before balancing

After partitioning the training dataset, it's subjected to SMOTEENN processing, a technique that tackles class imbalance. First, SMOTE is used to oversample the minority class, generating synthetic instances. Then, ENN is applied to remove noisy examples from the majority class. This results in a balanced dataset, with 251,693 genuine transactions and 83,905 fraudulent transactions. The distribution of the training dataset after balancing is illustrated below:

Genuine Transactions	251,691	75 %
Fraudulent Transactions	83,905	25 %
Total Transactions	335,596	

Table 3.12
Distribution of "Class" in the Train dataset after balancing

The distribution is illustrated in the figure below.

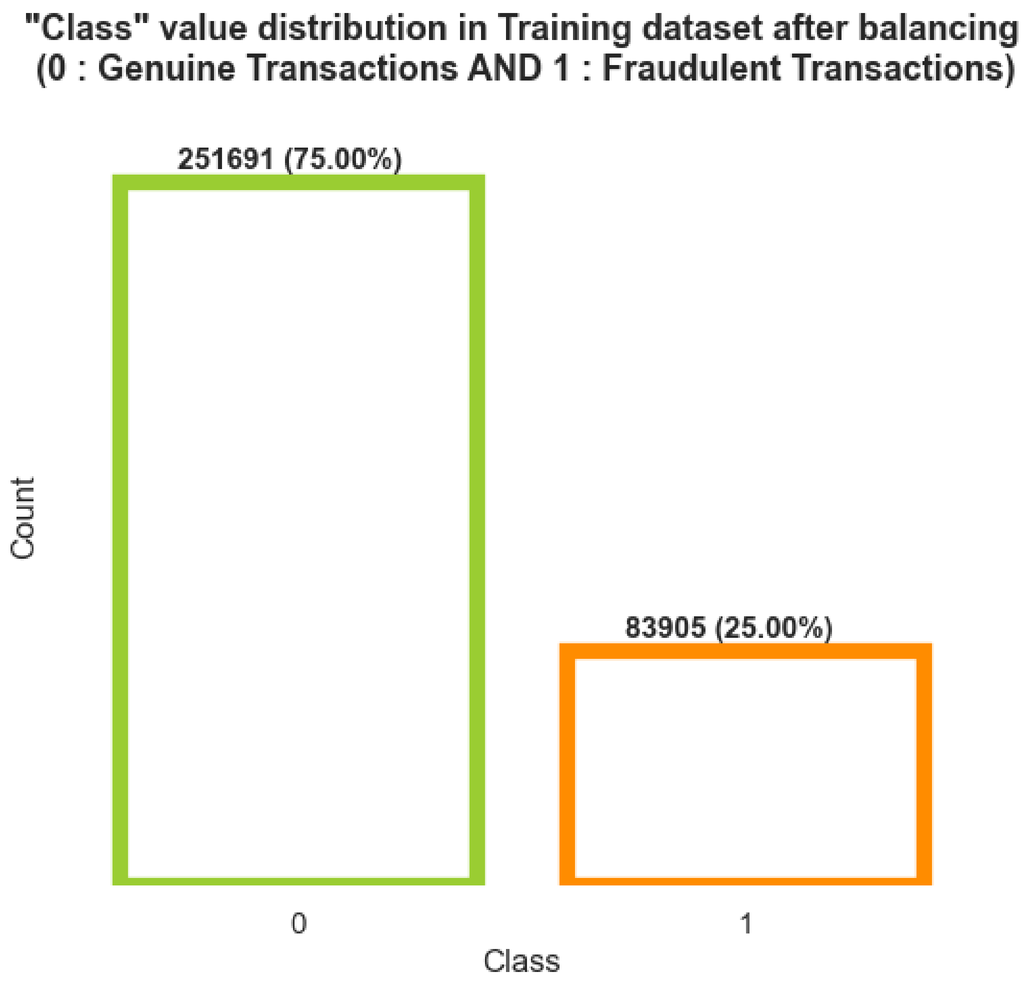

Figure 3.16
Distribution of "Class" in the Train dataset after balancing

After removing the "Class" column from both our training and test datasets, we proceed to structure our data into arrays. This process involves organizing our data into four distinct arrays: X_train, X_test, y_train, and y_test. These arrays will serve as the input and output data for our Machine Learning algorithms and classifiers, enabling us to effectively train and evaluate our models.

CHAPTER IV:
RESULTS AND IMPLEMENTATION

In this study, the following Machine Learning algorithms/classifiers are used to train and predict credit card fraud using the prepared dataset.

4.1 Machine Learning Algorithms/Classifiers

4.1.1 Logistic Regression

Logistic Regression is a versatile Machine Learning tool commonly used for tackling binary classification tasks. It works by taking various independent variables as input and then estimating the likelihood that a given data point belongs to a specific class. This algorithm scrutinizes the connections between these independent variables and the binary outcome of interest. In logistic regression, this outcome is typically binary, representing two distinct classes such as genuine or fraudulent transactions, denoted as 0 and 1 respectively.

The algorithm doesn't just provide a simple classification; instead, it offers probabilistic predictions, assigning each data point a probability score between 0 and 1. These scores reflect the likelihood that the data point falls into one class or the other. To finalize the classification, a predefined threshold is applied to these probabilities. If the predicted probability exceeds the threshold, the data point is labeled as belonging to one class; otherwise, it's assigned to the other class. This flexible approach allows for nuanced decision-making, making logistic regression a valuable tool in various domains, including fraud detection in financial transactions.

4.1.2 Gaussian Naïve Bayes

Naive Bayes is a classification algorithm known for its simplicity, speed, and effectiveness. It's dubbed "naïve" because it operates under the assumption that each feature in the dataset contributes independently and equally to the final classification. Despite this oversimplified approach, Naive Bayes often performs well in practice. It's based on the Bayes' theorem, a fundamental concept in probability theory that deals with conditional probabilities.

4.1.3 KNeighbors Classifier

K Nearest Neighbors (KNN) is a simple yet powerful classification algorithm. For a given data point, KNN helps find its closest neighbors. The algorithm calculates distances (usually Euclidean distance) between the new point and all other points in the dataset. KNN selects the K nearest points based on these distances. These neighbors form a neighborhood for the data points under consideration. Once the neighborhood identified, the data point gets a label based on the majority vote among its K neighbors. So, if most of the neighbors out of K neighbors are “genuine” records then the data point gets labeled as “genuine”.

4.1.4 Linear Support Vector Classifier

Linear Support Vector Classifier (LSVC) is a straightforward yet effective tool for classifying data, especially when the classes can be separated by a straight line or plane. Its flexibility and robustness make it a popular choice in machine learning applications.

The goal of the Linear SVC is to categorize data points into different classes. It achieves this by finding the best straight line or plane that separates these classes. This separation is based on the features (attributes) of the data points.

The SVC aims to maximize the margin between the decision boundary (the separating line/plane) and the nearest data points from each class. A larger margin helps in robust generalization, meaning the model performs well on unseen data.

The support vectors are the data points that lie closest to the decision boundary. These points are crucial because they significantly influence the position of the decision boundary. The model focuses on these support vectors during training.

Regularization Parameter is an important hyperparameter of LSVC which affects the trade-off between margin size and classification accuracy. A smaller value of this hyperparameter allows for a wider margin which might lead to more misclassifications. On the other hand, a higher value allows better accuracy of classifications but might result in overfitting.

Linear SVC works well when classes are linearly separable (meaning they can be separated by a straight line or plane). Despite its simplicity, Linear SVC is a powerful algorithm for binary classification. Its ability to handle large datasets and high-dimensional feature spaces makes it widely applicable across various domains.

4.1.5 Decision Tree Classifier

A Decision Tree is a supervised machine learning algorithm primarily used for classification tasks. It operates by recursively splitting the dataset into subsets based on features that effectively separate different classes.

Key components of a Decision Tree:

Decision Nodes: These points in the tree split the data into branches based on feature values. Each decision node represents a question about a feature.

Leaf Nodes: These are the endpoints of the tree where the final class label decision is made. Each leaf node corresponds to a specific class.

The algorithm selects the best feature to split the data at each decision node. It evaluates criteria like Gini impurity or information gain to determine the most informative feature.

The tree is constructed recursively, repeatedly splitting the data until stopping criteria are met (e.g., maximum depth, minimum samples, or purity improvement).

To classify a new data point, it traverses the tree from the root node down to a leaf node, following the decision paths based on feature values.

Decision trees are interpretable and suitable for tasks where human understanding matters. They handle both numerical and categorical data and are robust to outliers. However, overfitting can occur if the tree grows too deep or when the dataset is noisy. Techniques like pruning help mitigate overfitting.

Decision Trees strike a balance between interpretability and performance, making them valuable tools in various domains.

4.1.6 Keras Classifier

Keras is a popular and easy-to-use deep learning library that works on TensorFlow, CNTK, or Theano. It's designed for fast experimentation and focuses on being user-friendly, modular, and customizable. For classification tasks, we use the KerasClassifier wrapper, which simplifies building and training models. It requires a function to create the model and accepts settings like epochs and batch size. Typically, a basic classification model includes input, hidden, and output layers, with "ADAM" as the optimizer and "binary_crossentropy" as the loss function. Overall, Keras makes deep learning accessible even for beginners, enabling quick model creation and training.

4.1.7 Random Forest Classifier

A Random Forest Classifier is a popular machine learning algorithm used for classification tasks. It belongs to the ensemble learning methods, which combine the predictions of multiple individual models to improve the overall performance.

A Random Forest Classifier is constructed from a group of decision trees. Each decision tree is trained on a random subset of the training data and makes decisions based on features to classify instances into different classes.

To create multiple subsets of the original dataset, Random Forest uses a technique called bootstrapping. This involves randomly selecting samples with replacement. Some instances may appear multiple times in a subset, while others may not appear at all.

At each node of each decision tree, a random subset of features is considered for splitting. This randomness ensures that the trees in the forest are diverse and not highly correlated.

When predicting a new instance, each decision tree independently predicts the class. The class with the most votes among all the trees is chosen as the final prediction.

Random Forest algorithm is robust to overfitting because they combine predictions from multiple models, reducing the variance of the overall model. They can handle large datasets with high dimensionality. They provide estimates of feature importance, which can be useful for understanding the data.

4.1.8 Isolation Forest

The Isolation Forest is an anomaly detection algorithm with the primary purpose of identifying outliers within large datasets by isolating them within a tree-based framework.

The algorithm starts by randomly selecting a feature from the dataset. Next, it chooses a split value within the range of that feature. This process creates a partition that separates data points into two groups. The data undergoes recursive partitioning based on the selected feature and split value. At each step, the algorithm continues to split the data into smaller subsets. This recursive process repeats until each data point is isolated or until a predefined maximum tree depth is reached.

For each data point, the algorithm computes an anomaly score. The anomaly score reflects the number of partitions (splits) needed to isolate that data point. Generally, anomalies (outliers) require fewer partitions because they significantly deviate from the majority of data points. Finally, the algorithm applies a threshold to the computed anomaly scores. Data points with scores above the threshold are classified as anomalies, while those below it are considered normal.

Isolation Forest efficiently identifies anomalies, making it suitable for large datasets. It is robust against outliers. Unlike some other methods, it doesn’t rely on specific data distributions.

The Isolation Forest algorithm provides an effective way to detect anomalies without assuming any specific data distribution.

4.1.9 Adaboost Classifier

Adaptive Boosting (AdaBoost) is an ensemble method that constructs a robust classifier by combining multiple weak classifiers. This begins by training an initial model (often a simple decision tree) on the training data. Subsequent models focus on correcting errors made by the previous ones. These models are added iteratively until either the training set is perfectly predicted or a maximum number of models is reached.

AdaBoost assigns weights to instances in the training data. Instances that are difficult to classify receive higher weights, emphasizing their importance. Well-classified instances receive lower weights.

Weak learners in AdaBoost are typically decision trees with just one split, known as “decision stumps”. These simple models contribute to the ensemble by focusing on specific features.

AdaBoost sequentially adds weak models, each trained using the weighted training data. The process continues until a predetermined number of weak learners is reached or no further improvement can be made.

AdaBoost adapts dynamically, emphasizing challenging instances and gradually improving classification accuracy.

4.1.10 LightGBM Classifier

LightGBM (Light Gradient Boosting Machine), made by Microsoft, is a powerful tool for machine learning. It's great for ranking and classification tasks. LightGBM creates decision tree algorithms for ranking and classification. It's based on gradient boosting, which is a strong way to combine many weak models into a powerful one.

LightGBM puts together weak learners, like decision trees, to make a strong classifier. It's really fast, especially when using a GPU.

Multiple parameters can be selected to tune the performance of the model like boosting type, maximum number of leaves in a tree, learning rate etc.

LightGBM is used in finance, healthcare, marketing, and recommendation systems. Its flexibility and accuracy make it a popular choice for predicting outcomes.

The key difference between LightGBM and other decision tree-based algorithms lies in their tree growth strategy. While traditional methods usually expand trees horizontally (level-wise), LightGBM takes a vertical approach (leaf-wise). In practical terms, this means that LightGBM constructs trees by adding more leaves rather than growing them horizontally across levels. Consequently, leaf-wise growth often results in lower loss compared to level-wise growth. However, it’s essential to keep in mind that LightGBM might tend to overfit when dealing with small datasets.

4.1.11 CatBoost Classifier

CatBoost, developed by Yandex, is an open-source boosting library used for classification, regression, and ranking tasks. It simplifies preprocessing by automatically handling categorical features and utilizes Symmetric Weighted Quantile Sketch (SWQS) to handle missing data and prevent overfitting.

With its default settings, CatBoost achieves outstanding performance, reducing the need for extensive parameter tuning. It even includes built-in cross-validation for selecting optimal parameters. This model employs robust techniques to prevent overfitting and generalize well to new data. Additionally, it simplifies modeling by internally scaling all columns.

CatBoost also offers a fast GPU-accelerated version, making it suitable for training on large datasets. It's a powerful and user-friendly tool for various predictive tasks, especially well-suited for multiclass classification.

4.1.12 XGBoost Classifier

XGBoost, short for eXtreme Gradient Boosting, is a powerful and flexible gradient boosting library designed for efficiency and versatility. XGBoost combines predictions from multiple models, like decision trees, to create a strong classifier. It improves accuracy by iteratively correcting errors made by previous models.

XGBoost efficiently constructs trees for solving various data science problems. It can handle massive datasets and works on major distributed environments like Hadoop. This is widely used for regression and classification tasks due to its reliability and competitive performance.

XGBoost is a go-to choice for accurate predictive modeling, especially when dealing with complex datasets and ensemble techniques.

4.1.13 MLP Classifier

The MLP Classifier, also known as the Multi-layer Perceptron classifier, is a neural network-based algorithm that resides within the Scikit-Learn library. This classifier operates using the multilayer perceptron (MLP) architecture. Data flows unidirectionally through its layers, from input to output. Each layer consists of neurons that process the input data.

The primary focus of the MLP Classifier is to learn non-linear relationships between input features and output labels. During training, it adjusts weights associated with connections between neurons to minimize the loss function.

The MLP Classifier comprises multiple hidden layers, each fully connected to the next. Neurons within these layers apply nonlinear activation functions to transform input data. The MLP Classifier is primarily used for classification tasks, where it assigns input data to predefined classes.

The versatility of the MLP Classifier lies in its ability to handle complex data relationships. As an integral part of Scikit-Learn, it's widely accessible and commonly used in various machine learning applications.

The MLP Classifier is a valuable tool for creating neural network-based classifiers, especially when dealing with intricate data patterns. Its ease of use and competitive performance make it a popular choice among data scientists and practitioners.

4.1.14 Bagging Classifier

A Bagging Classifier is a machine learning technique used for classification tasks. It works by creating multiple subsets of the original dataset through a process called bootstrapping, where data is randomly sampled with replacement. Each subset is used to train a separate base classifier, such as a decision tree.

The key idea behind Bagging is to reduce variance and improve accuracy by combining the predictions of multiple classifiers. This is achieved through a voting mechanism, where the most common prediction among all classifiers is chosen as the final prediction.

Bagging classifiers offer several benefits. First, they help prevent overfitting by training on different subsets of data. Second, they improve accuracy by aggregating predictions from multiple classifiers. Third, they are robust to noisy data and outliers since they average out their effects across multiple models.

4.1.15 Dynamic Ensemble

Dynamic Ensemble Selection (DES) is a method in ensemble learning used for classification predictions. Unlike traditional ensemble methods, which combine predictions from multiple models, DES dynamically chooses the best-suited model for each new example based on its unique characteristics.

1. Model Training: Initially, multiple machine learning models are trained using the training dataset. These models form a pool of potential choices for making predictions.

2. Example-Specific Selection: When a prediction is needed for a new example, DES identifies instances in the training data that are most similar to the new example. It accomplishes this by using a k-nearest neighbor model, which finds the closest examples in the dataset.

3. Evaluation: Once the similar instances are identified, all models in the pool are evaluated based on their performance on this subset of similar examples.

4. Best Model Selection: Finally, the model that performs the best on this subset of similar examples is selected to make the prediction for the new example.

Because of the architecture, DES often outperforms any single model in the pool by selecting the most suitable model for each example. This adaptability helps improve classification accuracy. Also, unlike traditional ensemble methods that rely on averaging predictions from multiple models, DES provides an alternative approach by selecting the best-performing model for each example. This can lead to more accurate predictions, especially in cases where models in the ensemble have different strengths and weaknesses.

4.1.16 Stacking

Stacking, also known as Stacked Generalization, is an ensemble technique in machine learning. Like other ensemble methods, it blends multiple machine learning algorithms to achieve superior performance compared to any individual algorithm used alone.

Stacking divides participating machine learning algorithms into two layers. The first layer, known as the base model layer, comprises two or more diverse machine learning models. These base models are trained independently on the same dataset.

The second layer, called the meta model, acts as the final model in the ensemble, producing the ultimate prediction. The predictions made by the base models serve as input for the meta model, which learns to combine them for a more accurate overall outcome.

Stacking typically opts for varied and sophisticated models as base models. These could include decision trees, neural networks, support vector machines, or other robust algorithms.

Simple models are favored for the meta model. Linear models like logistic regression or linear regression are commonly employed. The meta model learns to effectively weigh the predictions from the base models.

In essence, stacking harnesses the strengths of different models by integrating them in a hierarchical structure, resulting in enhanced predictive performance.

4.2 Performance Evaluation Criteria

The Confusion Matrix serves as a crucial metric for evaluating the accuracy and efficacy of machine learning models, especially in classification tasks involving multiple distinct classes. Below are the key points related to this tool:

- Structure: Represented as a two-dimensional table, the Confusion Matrix consists of "Actual" and "Predicted" dimensions, each containing sets of class labels.
- Rows and Columns: Rows correspond to actual classifications, while columns denote predicted classifications.
- Performance Metrics: From the Confusion Matrix, essential performance metrics such as accuracy, precision, recall, and F1-score can be derived.

In essence, the Confusion Matrix offers a comprehensive insight into the performance of a classification model across various classes.

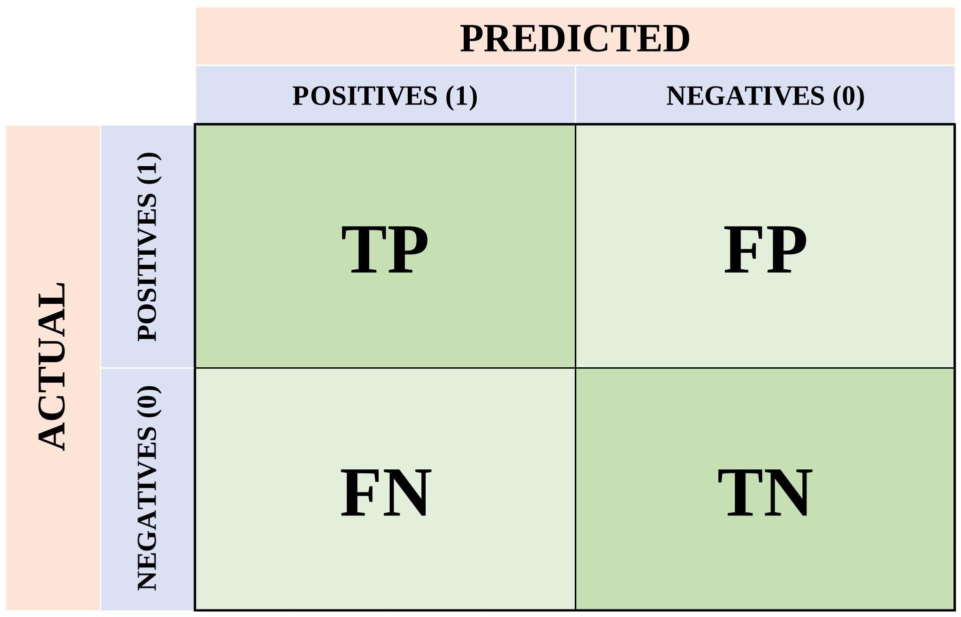

Figure 4.1
Confusion Matrix

Output of the above-mentioned Machine Learning Algorithms are evaluated using the four basic parameters viz.

TP – True Positive – *fraudulent* transactions classified as fraudulent

TN – True Negative – *genuine* transactions classified as genuine

FP – False Positive – *genuine* transactions classified as fraudulent

FN – False Negative - *fraudulent* transactions classified as genuine

Major performance metrics for Machine Learning Algorithms are as below:

Accuracy is a fundamental metric in evaluating the performance of a machine learning model. It provides insight into how well the model predicts outcomes correctly across all classes.

To calculate accuracy, we look at the number of correct predictions made by the model and compare it to the total number of predictions. Specifically, in the numerator, we consider the sum of True Positives (TP) and True Negatives (TN), which are instances where the model correctly predicts the positive and negative classes, respectively. In the denominator, we include the total number of predictions, which is the sum of True Positives, True Negatives, False Positives (FP), and False Negatives (FN). False Positives occur when the model incorrectly predicts the positive class, while False Negatives occur when the model incorrectly predicts the negative class.

By dividing the number of correct predictions by the total number of predictions, accuracy provides a simple and intuitive measure of the model's overall correctness. It is expressed as a percentage, with higher values indicating better performance. However, accuracy may not be the most suitable metric in imbalanced datasets, where one class dominates the others, as it may give a misleading impression of the model's performance. Therefore, it is essential to consider other metrics, such as precision, recall, and F1-score, in conjunction with accuracy to obtain a comprehensive evaluation of the model's effectiveness.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Precision is a crucial metric that assesses the precision or exactness of a machine learning model's positive predictions. It provides valuable insight into the model's ability to correctly identify positive instances, especially in scenarios where false positives can be costly.

To calculate precision, we focus on the number of correct positive predictions made by the model and compare it to the total number of positive predictions. Specifically, in the numerator, we consider True Positives (TP), which are instances where the model correctly identifies positive cases. In the denominator, we include the total number of positive predictions, which is the sum of True Positives and False Positives (FP). False Positives occur when the model incorrectly predicts a positive case when it is actually negative.

By dividing the number of correct positive predictions by the total number of positive predictions, precision quantifies the model's precision in identifying positive cases. It is expressed as a percentage, with higher values indicating a more precise model. Precision is particularly useful in scenarios where the cost of false positives is significant, as it helps ensure that positive predictions are accurate and reliable.

$$Precision = \frac{TP}{TP + FP}$$

Recall, also known as Sensitivity, is a critical metric in evaluating the effectiveness of a machine learning model, especially in scenarios where the cost of false negatives is high. Recall measures the model's ability to correctly identify all positive instances out of all actual positive cases.

To calculate recall, we focus on the number of correct positive predictions made by the model and compare it to the total number of positive instances in the dataset. Specifically, in the numerator, we consider True Positives (TP), which are instances where the model correctly identifies positive cases. In the denominator, we include the total number of positive instances, which is the sum of True Positives and False Negatives (FN). False Negatives occur when the model incorrectly predicts a negative case when it is actually positive.

By dividing the number of correct positive predictions by the total number of positive instances, recall quantifies the model's ability to capture all positive cases. It is expressed as a percentage, with higher values indicating a higher proportion of positive instances correctly identified by the model. Recall is particularly valuable in applications where missing positive instances can have significant consequences, as it ensures that the model identifies as many positive cases as possible.

$$Recall = \frac{TP}{TP + FN}$$

The **F1 Score** is a comprehensive metric that integrates both Recall and Precision into a single measure of a machine learning model's performance. Unlike Recall and Precision, which focus on specific aspects of prediction accuracy, F1 considers both false positives and false negatives, making it particularly effective in scenarios with imbalanced classes. The F1 Score is calculated as the harmonic mean of Precision and Recall.

To calculate the F1 Score, we use the formula:

$$F1Score = 2 * \frac{Recall * Precision}{Recall + Precision}$$

In the numerator, we compute twice the product of Recall and Precision, while in the denominator, we sum Recall and Precision. By taking the harmonic mean, the F1 Score strikes a balance between Precision and Recall, giving equal weight to both metrics.

The F1 Score is especially useful in situations where accuracy alone may be misleading due to class imbalances. For instance, in medical diagnostics, where the prevalence of a disease may be low, the F1 Score provides a more accurate assessment of a model's performance by considering both false positives and false negatives. In summary, the F1 Score offers a comprehensive evaluation of a model's predictive capability, making it a valuable metric for assessing performance in classification tasks, particularly in scenarios with imbalanced classes.

RECALL gives us information about a classifier's performance with respect to **FALSE NEGATIVES** (how many fraudulent transactions model missed to classify correctly), while PRECISION gives us information about its performance with respect to **FALSE POSITIVES** (how many fraudulent transactions model predicted correctly). F1

SCORE is a balance between RECALL and PRECISION. This combines both these metrics into one metric as weighted average of the two.

AUROC, or "Area Under the Receiver Operating Characteristics," is a widely used evaluation metric for classification tasks. It comprises two key components: AUC, short for "Area Under the Curve," and ROC, which stands for "Receiver Operating Characteristics."

The ROC curve is a probability curve, while AUC represents the degree of separability between different classes. A higher AUC indicates a better ability of the model to distinguish between target classes, with positive instances correctly identified as positive and negative instances correctly identified as negative.

In essence, AUROC quantifies how effectively a model can discern between different classes, making it a crucial metric for evaluating classification models. Higher AUC values signify superior predictive performance in classifying positive and negative instances.

The ROC curve is graphed with the True Positive Rate (TPR) plotted against the False Positive Rate (FPR), with TPR represented on the y-axis and FPR on the x-axis.

The True Positive Rate (TPR), also known as sensitivity, is a critical metric in classification tasks. It measures the proportion of positive instances that are correctly identified by the model out of all actual positive instances.

Mathematically, TPR is calculated as the ratio of True Positives (TP) to the sum of True Positives and False Negatives (FN), represented as:

$$TPR = \frac{TP}{FN + TP}$$

In practical terms, TPR tells us how well the model captures positive instances among all the positive examples present in the dataset. A higher TPR indicates that the model is effective at correctly identifying positive cases, making it an essential measure of the model's sensitivity to positive instances.

The False Positive Rate (FPR) is a crucial metric in classification analysis, providing insights into the model's propensity to incorrectly classify negative instances as positive. It quantifies the proportion of negative data points that are erroneously classified as positive among all actual negative instances.

Mathematically, FPR is calculated as the ratio of False Positives (FP) to the sum of False Positives and True Negatives (TN), expressed as:

$$FPR = \frac{FP}{FP + TN}$$

In practical terms, FPR helps assess the model's specificity by indicating the rate of false alarms or false positives generated by the model. A lower FPR signifies that the model is better at distinguishing between negative instances and positive ones. Therefore, minimizing the FPR is essential in scenarios where correctly identifying negative instances is critical, such as in medical diagnostics or fraud detection.

The True Positive Rate (TPR) and False Positive Rate (FPR) are both metrics that fall within the range of [0, 1]. These values are computed at various threshold levels, typically ranging from 0.00 to 1.00, with increments such as 0.02 or 0.04. By computing TPR and FPR at different threshold values, we can plot them on a graph known as the Receiver Operating Characteristic (ROC) curve.

The ROC curve depicts the relationship between TPR (sensitivity) and FPR (1 - specificity) across different threshold levels. Specifically, TPR is plotted on the y-axis, while FPR is plotted on the x-axis. Each point on the ROC curve represents the TPR and FPR values obtained at a specific threshold level.

The Area Under the Curve (AUC) of the ROC curve quantifies the overall performance of the classification model. It represents the area under the ROC curve, which ranges from 0 to 1. A higher AUC value indicates better discrimination ability of the model, with values closer to 1 indicating superior performance.

In summary, the ROC curve is a graphical representation of the trade-off between TPR and FPR at various threshold levels. The AUC provides a single metric to evaluate the overall performance of the classification model based on the ROC curve, with higher AUC values indicating better performance.

A model with an AUC close to 1 signifies excellent performance, indicating a high degree of separability between the positive and negative classes. In other words, the model demonstrates a strong ability to distinguish between the two classes effectively.

For example, if a model achieves an AUC value of 0.7, it implies that there is a 70% chance that the model can correctly discriminate between instances belonging to the positive class and those belonging to the negative class. This indicates a reasonably good ability to differentiate between the two classes, although there is still room for improvement.

The AUC value serves as a valuable metric for evaluating the discriminative power of a classification model. Higher AUC values suggest better model performance, with values approaching 1 indicating a higher likelihood of accurate classification.

In the context of Credit Card Fraud detection, the emphasis is placed on detecting fraudulent transactions (Positives) rather than accurately identifying genuine transactions (Negatives). The primary goal is to develop a model that excels in identifying instances of fraud, as the consequences of missing fraudulent activity can be severe for both the cardholder and the financial institution.

In this scenario, the performance of the model in detecting fraudulent transactions takes precedence over its ability to accurately classify genuine transactions. While it's essential to minimize false positives (misclassifying genuine transactions as fraudulent), the priority lies in maximizing true positives (correctly identifying fraudulent transactions).

A model is considered more suitable for fraud detection if it demonstrates high sensitivity in detecting fraudulent activity while maintaining an acceptable level of false positives. Therefore, the focus is on optimizing the model's ability to detect fraudulent transactions, even if it results in a slightly higher rate of false positives in genuine transactions.

In Credit Card Fraud detection, the primary objective is to develop a model that effectively identifies fraudulent transactions while ensuring that genuine transactions are not excessively misclassified as fraudulent. This approach prioritizes the detection of fraud to mitigate financial losses and protect cardholders from fraudulent activities.

The overall "Accuracy" of a model reflects its ability to correctly identify both True Positives and True Negatives among all instances. It provides a broad assessment of the model's performance across all classes.

On the other hand, "Precision" places greater emphasis on minimizing False Negatives, aiming to ensure that the positive predictions made by the model are indeed accurate. Conversely, "Recall" focuses on reducing False Positives, striving to capture as many positive instances as possible from the actual positives.

When both "Precision" and "Recall" are equally important, the "F1" score becomes crucial. This metric represents the harmonic mean of Precision and Recall, providing a balanced evaluation of the model's performance. By considering both Precision and Recall simultaneously, the F1 score effectively captures the trade-off between these two metrics, ensuring a comprehensive assessment of the model's effectiveness.

In the realm of Credit Card Fraud detection, the primary objective is to accurately identify fraudulent transactions while minimizing false negatives. Detecting a fraudulent transaction is of paramount importance, as failure to do so can lead to significant financial losses and reputational damage for the financial institution.

When a transaction is erroneously classified as fraudulent, the financial institute may need to address the concerns of the affected customer. However, there are typically no direct financial repercussions associated with such misclassifications. On the other hand, if the model fails to detect a fraudulent transaction, allowing it to be processed as a genuine transaction, the consequences can be far-reaching.

Misclassifying fraudulent transactions as genuine can result in substantial financial losses for the institution, followed by a loss of trust and confidence from customers. This erosion of trust can have profound implications for the organization's business operations, as customer confidence in the institution's ability to safeguard their financial assets would be severely undermined.

Given these risks, financial institutions prioritize the accurate detection of fraudulent transactions, even if it means that a small number of genuine transactions may be incorrectly flagged as fraudulent. The overarching goal is to maintain the integrity of the financial system and preserve customer trust by minimizing the occurrence of undetected fraudulent activity.

In evaluating the performance of models for Credit Card Fraud detection, the significance of various metrics differs. Among these metrics, "Recall" holds the highest importance, followed closely by "Precision." While "Accuracy" and "AUC" are also considered, they are accorded lesser importance compared to "Recall" and "Precision."

"Recall" is of paramount importance in Credit Card Fraud detection as it measures the model's ability to correctly identify fraudulent transactions among all actual fraudulent instances. Maximizing Recall ensures that the model captures as many instances of fraud as possible, thereby minimizing the risk of undetected fraudulent activity.

Following "Recall," "Precision" holds significant importance. Precision focuses on minimizing False Positives, ensuring that the transactions flagged as fraudulent are indeed fraudulent. High Precision indicates that the model accurately identifies fraudulent transactions, reducing the occurrence of false alarms and unnecessary investigations.

While "Accuracy" is a fundamental metric that assesses overall model performance, it may not provide a complete picture in the context of imbalanced datasets, such as those encountered in Credit Card Fraud detection. Similarly, "AUC" evaluates the model's ability to discriminate between classes, but its significance may be overshadowed by the critical need to maximize Recall and Precision in fraud detection scenarios.

In summary, for Credit Card Fraud detection, prioritizing Recall and Precision over Accuracy and AUC ensures that the model effectively identifies fraudulent transactions while minimizing false alarms. This strategic emphasis reflects the critical importance of accurately detecting fraud to mitigate financial losses and maintain customer trust.

Based on the insights outlined above, this research will utilize a composite metric called "Model Score" to evaluate the performance of the models. "Model Score" will be derived from a weighted combination of "Accuracy," "Precision," "Recall," and "AUC" scores, reflecting their respective importance in the context of the research.

To calculate the "Model Score," priority will be given to "Recall" followed by "Precision," acknowledging their critical roles in Credit Card Fraud detection. "Accuracy" and "AUC" will be accorded lesser weightage compared to "Recall" and "Precision" due to their secondary significance in fraud detection scenarios.

The "Model Score" will be computed using the following weightage scheme: 0.35 for "Recall," 0.25 for "Precision," 0.2 for "Accuracy," and 0.2 for "AUC." This distribution of weightage reflects the research's emphasis on maximizing the model's ability to accurately detect fraudulent transactions while balancing overall performance across multiple metrics.

Notably, the "F1" score will not be included in the evaluation criteria for model ranking. This decision is based on the consideration that both "Precision" and "Recall" are already accounted for in the calculation of the "Model Score," rendering the "F1" score redundant in this context.

The proposed approach ensures a comprehensive assessment of model performance by integrating multiple key metrics into the "Model Score," with a tailored weightage scheme that aligns with the research objectives and priorities in Credit Card Fraud detection.

Formula used to calculate “Model Score” is given below:

MODEL SCORE = 0.35 * RECALL + 0.25 * PRECISION + 0.2 * ACCURACY + 0.2 * AUC

Therefore, the "Model Score" gets calculated using the below weightage:

RECALL: 35%

PRECISION: 25%

ACCURACY: 20%

AUC: 20%

4.3 Model Evaluation

4.3.1 Logistic Regression

4.3.1.1 Confusion Matrix

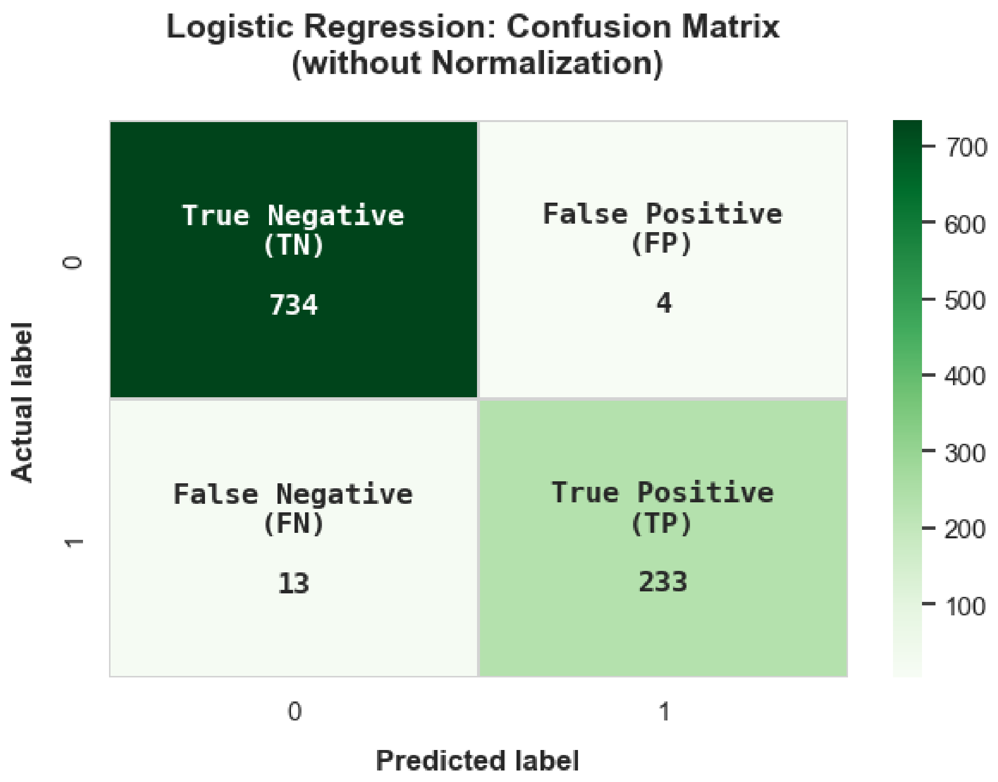

Figure 4.2
Logistic Regression: Confusion Matrix (without Normalization)

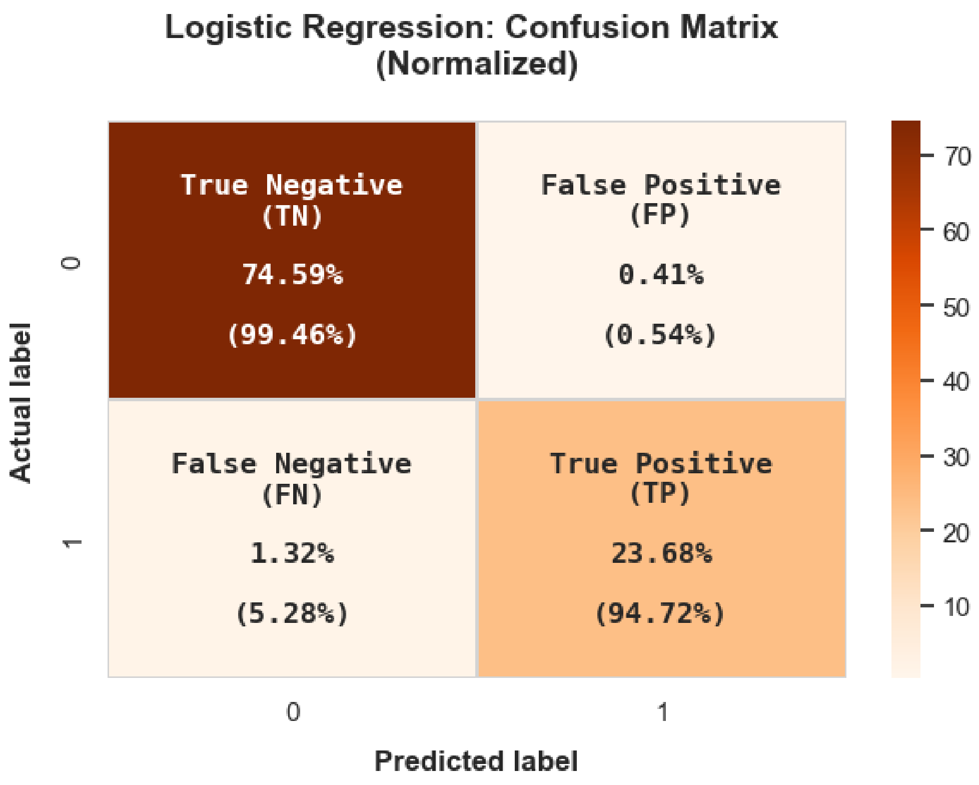

Figure 4.3
Logistic Regression: Confusion Matrix (Normalized)

In Logistic Regression, 99.46% of genuine transactions (representing 74.59% of all transactions) and 94.72% of fraudulent transactions (comprising 23.68% of all transactions) are accurately detected. However, it fails to identify 5.28% of fraudulent transactions (accounting for 1.32% of all transactions) and 0.54% of genuine transactions (equating to 0.41% of all transactions).

4.3.1.2 Performance Metrics

Model Name	Accuracy	Precision	Recall	AUC	Model Score

Logistic Regression	98.27%	98.31%	94.72%	98.64%	97.11%

Table 4.1
Logistic Regression – Performance Metrics

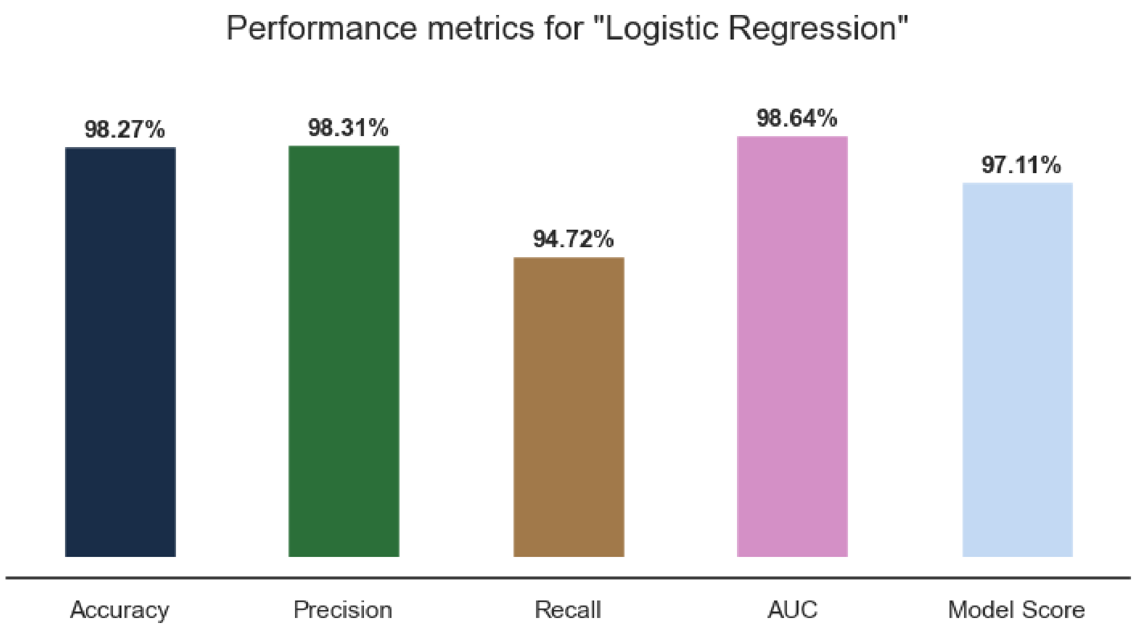

Figure 4.4
Logistic Regression – Performance Metrics

4.3.2 Gaussian Naïve Bayes

4.3.2.1 Confusion Matrix

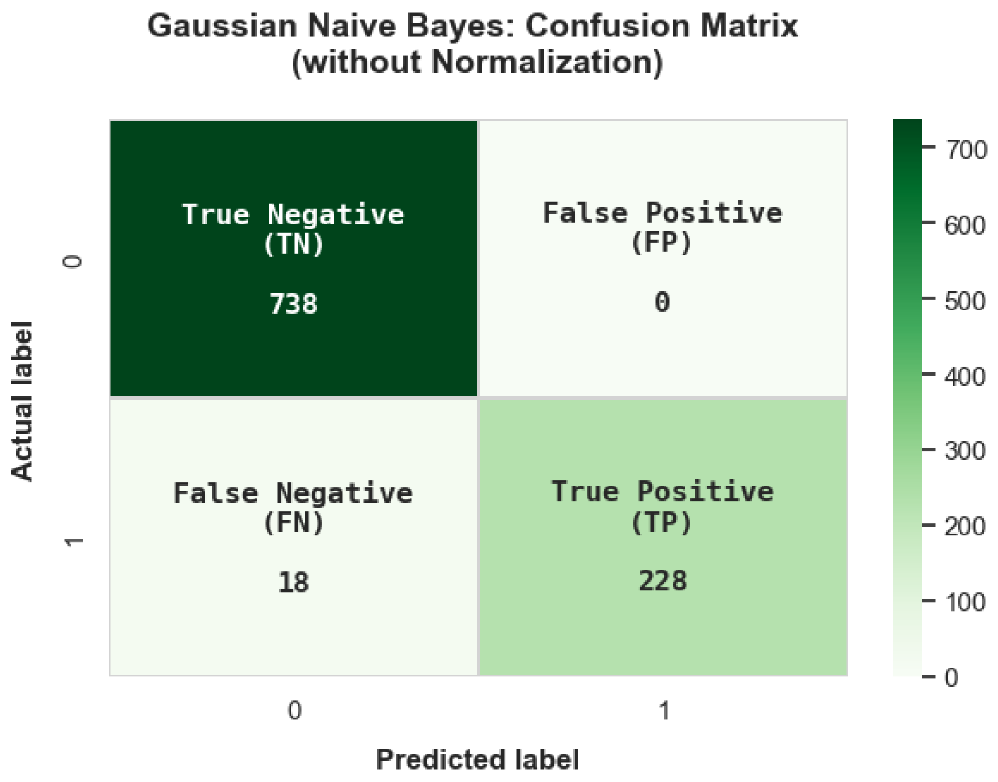

Figure 4.5
Gaussian Naïve Bayes: Confusion Matrix (without Normalization)

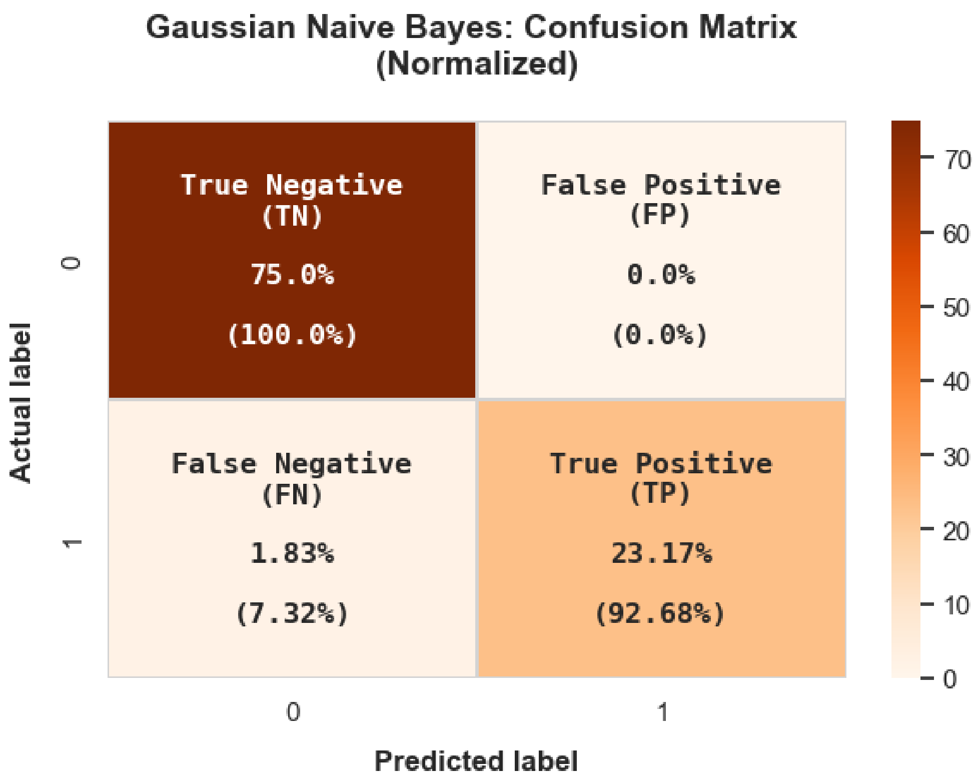

Figure 4.6
Gaussian Naïve Bayes: Confusion Matrix (Normalized)

Gaussian Naïve Bayes exhibited slightly lower performance compared to Logistic Regression. It successfully identified all genuine transactions, accounting for 75% of all transactions, while accurately detecting 92.68% of fraudulent transactions, which constituted 23.17% of all transactions. However, it failed to detect 7.32% of fraudulent transactions, representing 1.83% of the total transactions. This failure to identify fraudulent transactions is concerning in the context of Credit Card Fraud detection, highlighting the importance of improving the model's ability to detect such instances.

4.3.2.2 Performance Metrics

Model Name	Accuracy	Precision	Recall	AUC	Model Score
Gaussian Naïve Bayes	98.17%	100.00%	92.68%	98.67%	96.81%

Table 4.2
Gaussian Naïve Bayes – Performance Metrics

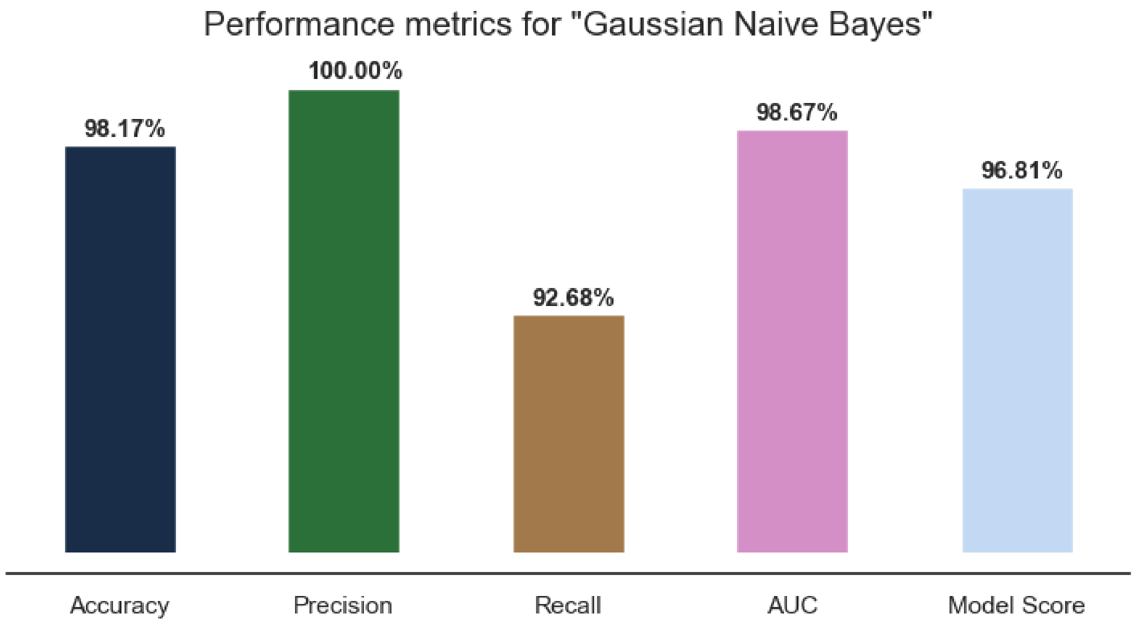

Figure 4.7
Gaussian Naïve Bayes – Performance Metrics

4.3.3 Linear Support Vector Classifier

4.3.3.1 Confusion Matrix

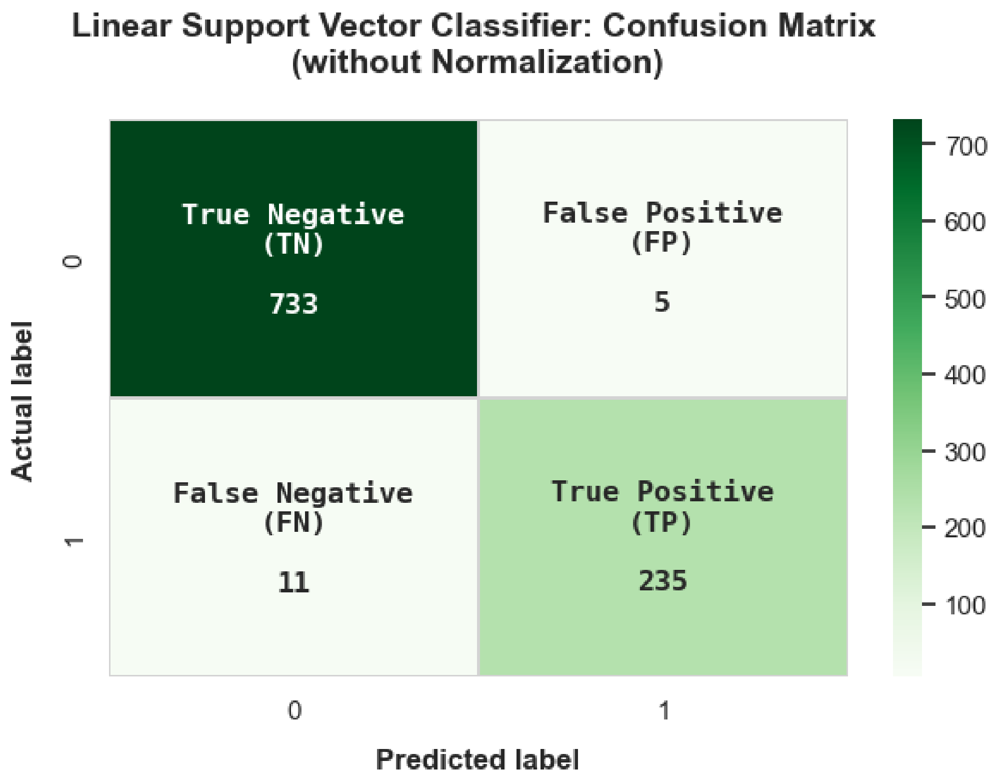

Figure 4.8
Linear Support Vector Classifier: Confusion Matrix (without Normalization)

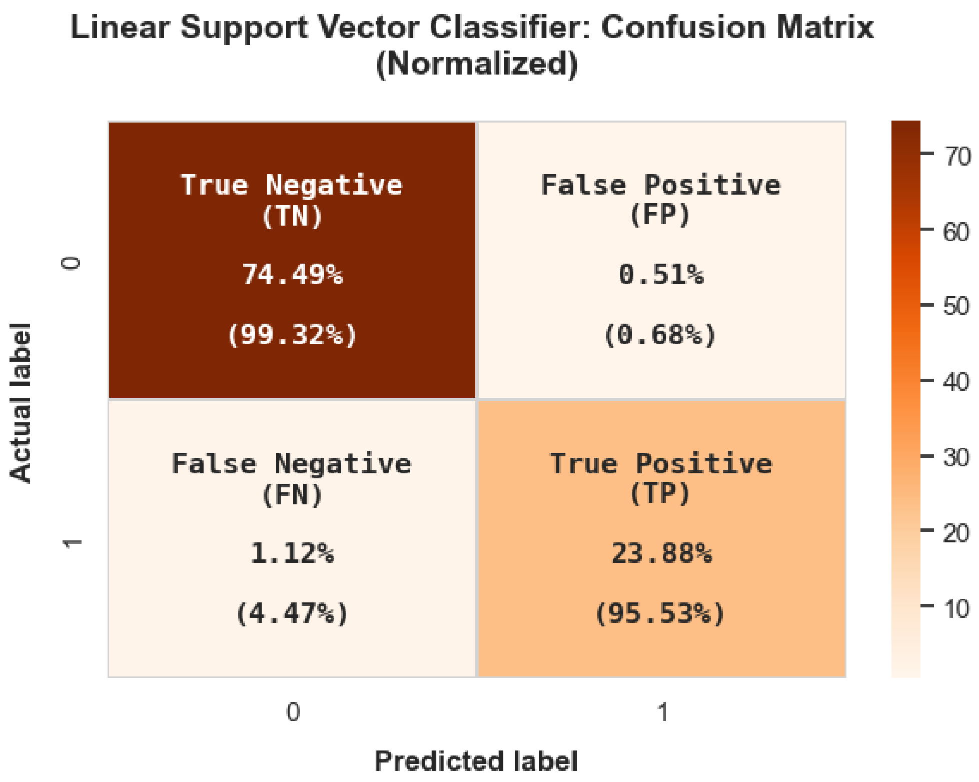

Figure 4.9
Linear Support Vector Classifier: Confusion Matrix (Normalized)

The Linear Support Vector Classifier successfully identified 99.32% of genuine transactions, representing 74.49% of all transactions, and accurately detected 95.53% of fraudulent transactions, constituting 23.88% of the total transactions. However, it missed detecting 4.47% of fraudulent transactions, amounting to 1.12% of all transactions, and 0.68% of genuine transactions, which accounted for 0.51% of the total transactions.

4.3.3.2 Performance Metrics

Model Name	Accuracy	Precision	Recall	AUC	Model Score
Linear Support Vector Classifier	98.37%	97.92%	95.53%	97.43%	97.07%

Table 4.3
Linear Support Vector Classifier – Performance Metrics

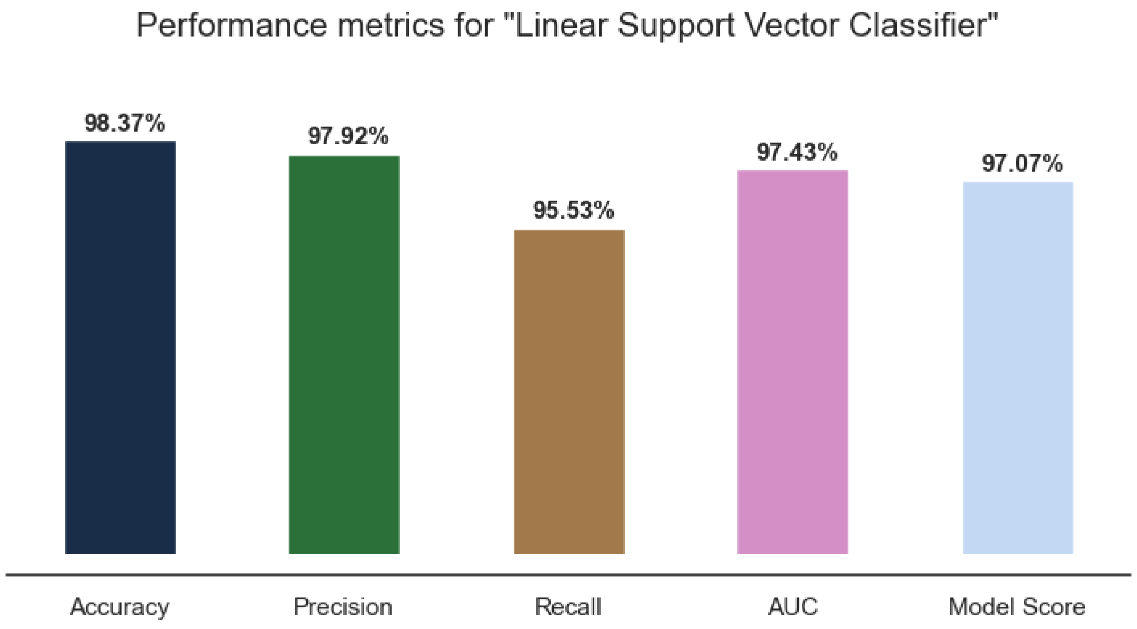

Figure 4.10
Linear Support Vector Classifier – Performance Metrics

4.3.4 KNeighbors Classifier

4.3.4.1 Confusion Matrix

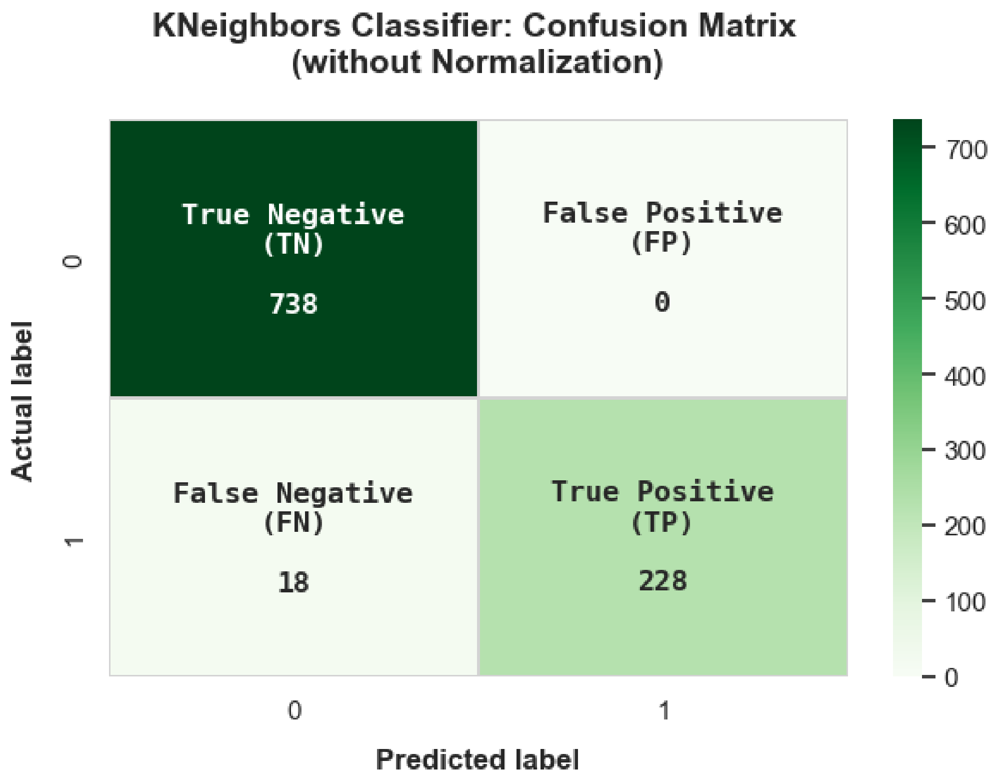

Figure 4.11
KNeighbors Classifier: Confusion Matrix (without Normalization)

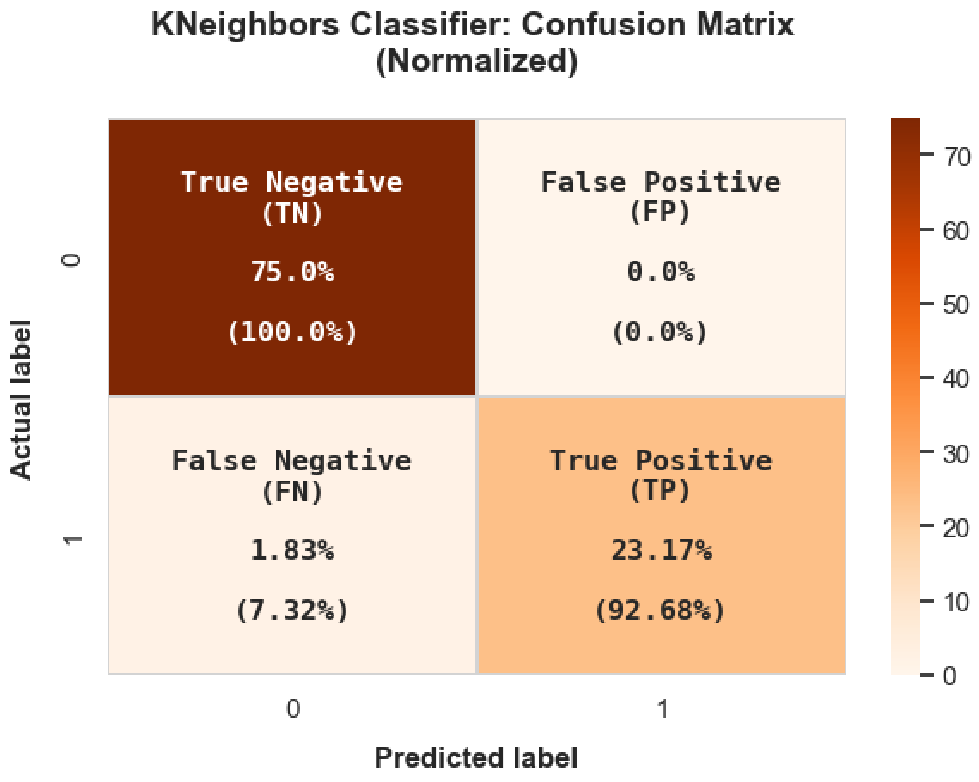

Figure 4.12
KNeighbors Classifier: Confusion Matrix (Normalized)

The KNeighbors Classifier achieved a perfect detection rate for genuine transactions, capturing 75% of the total transactions. However, its performance in identifying fraudulent transactions was slightly lower, with an accuracy of 92.68%, representing 23.17% of the total transactions. Notably, it failed to detect 7.32% of fraudulent transactions, amounting to 1.83% of the total transactions.

4.3.4.2 Performance Metrics

Model Name	Accuracy	Precision	Recall	AUC	Model Score
KNeighbors Classifier	98.17%	100.00%	92.68%	96.94%	96.46%

Table 4.4
KNeighbors Classifier – Performance Metrics

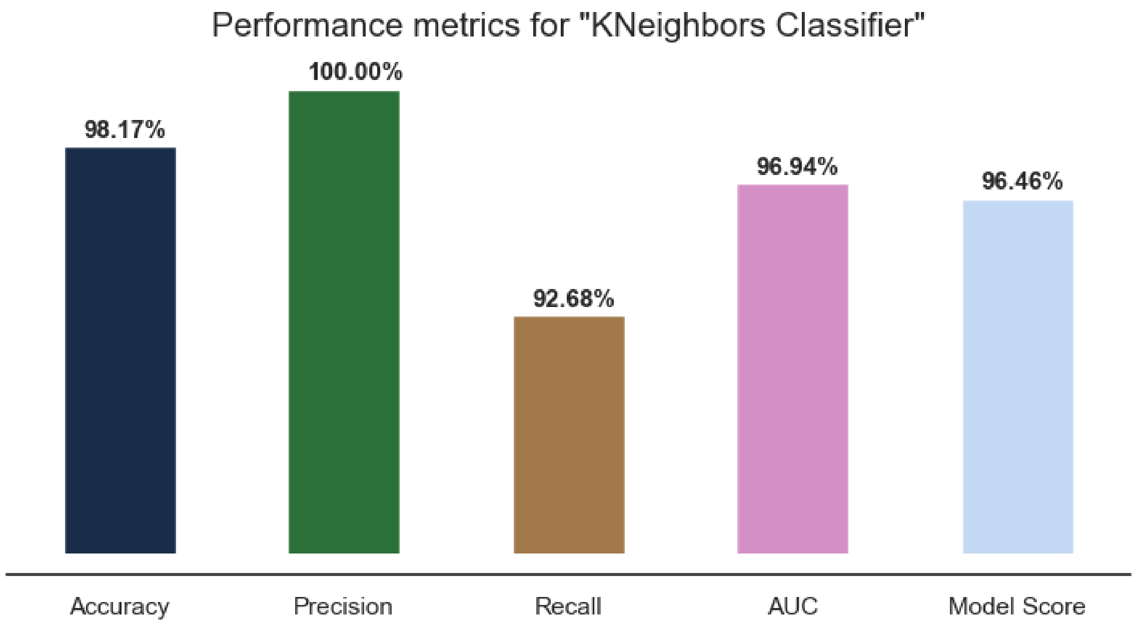

Figure 4.13
KNeighbors Classifier – Performance Metrics

4.3.5 Random Forest Classifier

4.3.5.1 Confusion Matrix

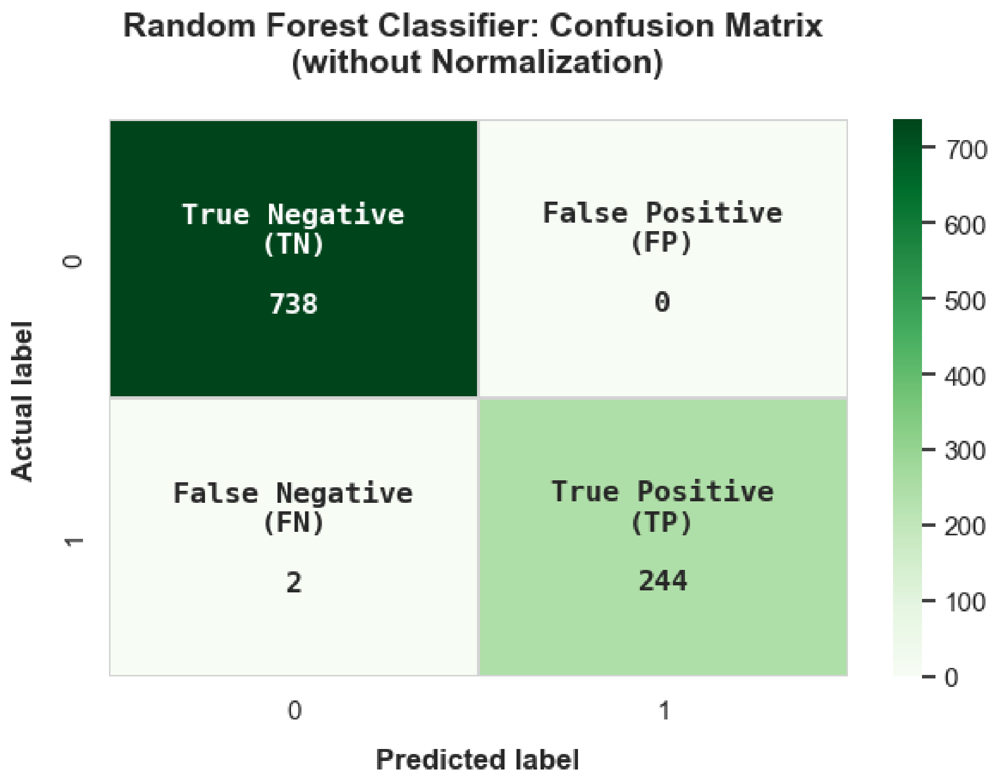

Figure 4.14
Random Forest Classifier: Confusion Matrix (without Normalization)

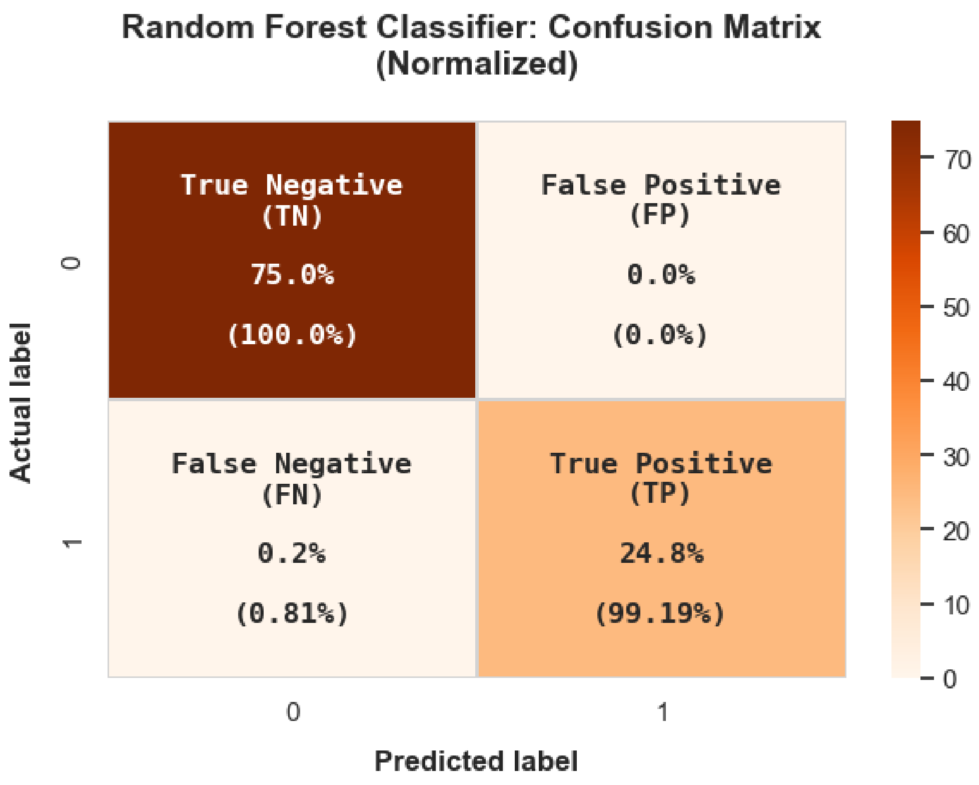

Figure 4.15
Random Forest Classifier: Confusion Matrix (Normalized)

The Random Forest Classifier exhibited exceptional performance in detecting genuine transactions, achieving a perfect detection rate of 100%, which accounted for 75% of all transactions. Moreover, it demonstrated remarkable accuracy in identifying fraudulent transactions, with a detection rate of 99.19%, representing 24.8% of the total transactions. Impressively, this model missed only 0.81% of fraudulent transactions, equivalent to just 0.2% of the total transactions.

4.3.5.2 Performance Metrics

Model Name	Accuracy	Precision	Recall	AUC	Model Score
Random Forest Classifier	99.80%	100.00%	99.19%	99.75%	99.63%

Table 4.5
Random Forest Classifier – Performance Metrics

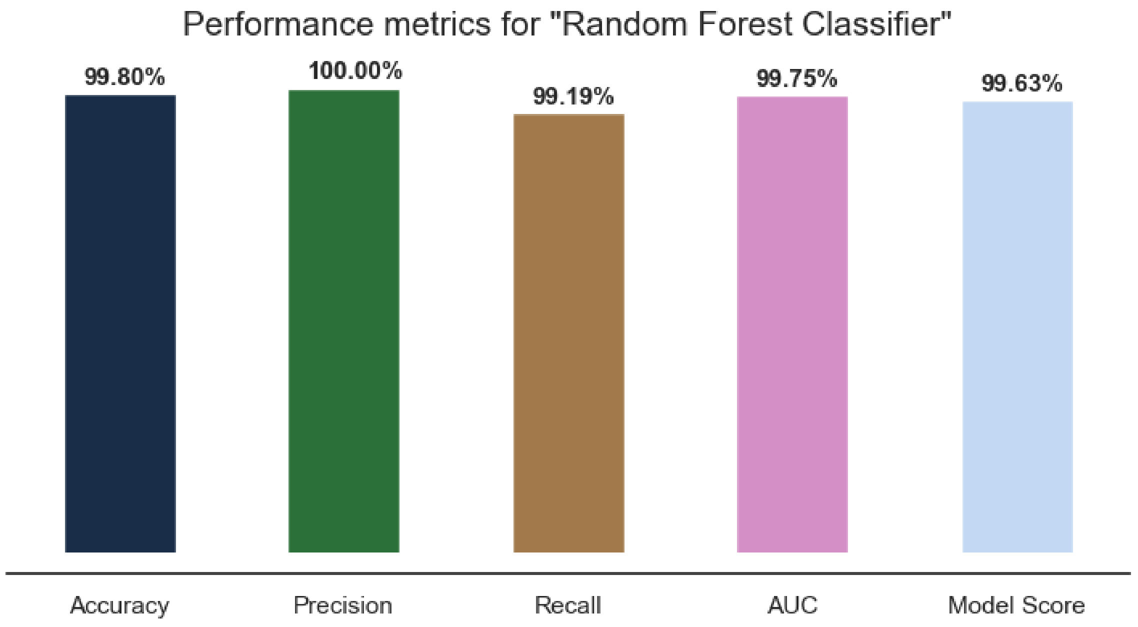

Figure 4.16
Random Forest Classifier – Performance Metrics

4.3.6 Isolation Forest

4.3.6.1 Confusion Matrix

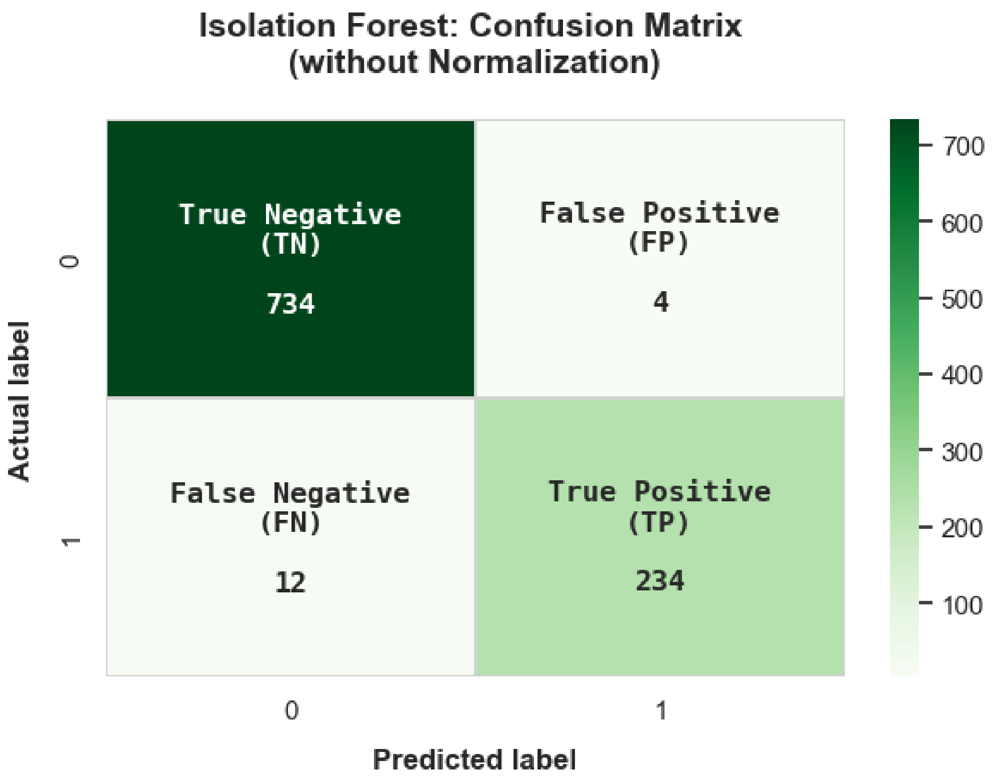

Figure 4.17
Isolation Forest: Confusion Matrix (without Normalization)

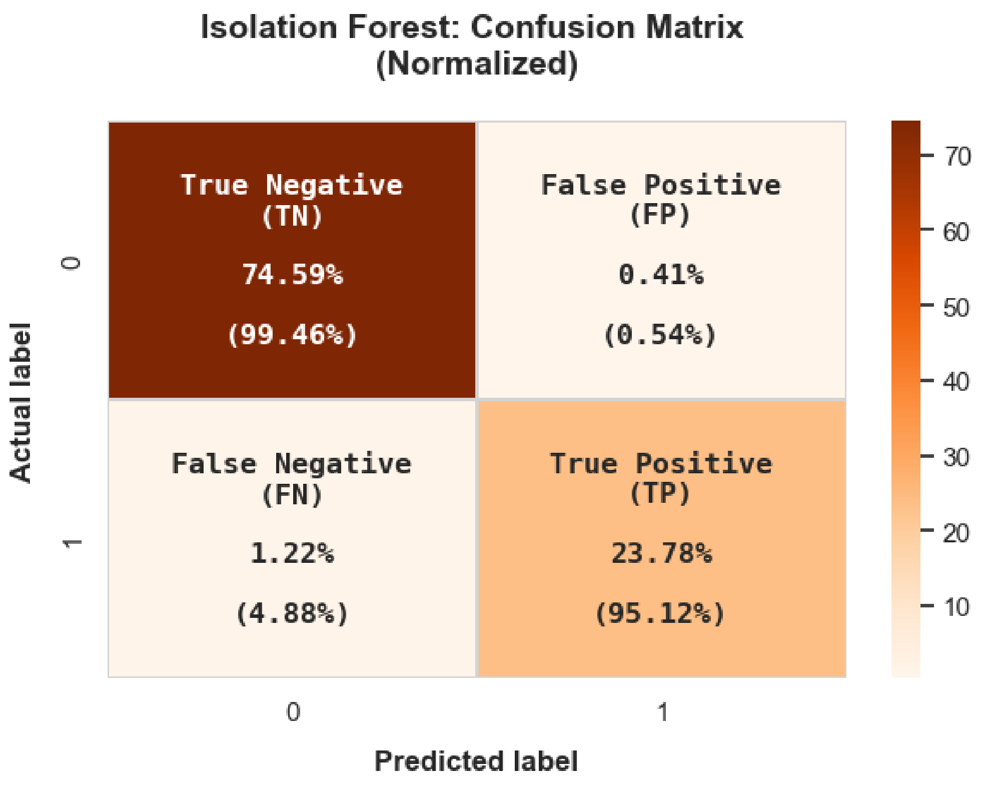

Figure 4.18
Isolation Forest: Confusion Matrix (Normalized)

Isolation Forest was able to detect 99.46% of genuine records and 95.12% of fraudulent records correctly which denotes 74.59% and 23.78% of the total records respectively. However, this model failed to detect 4.88% of the fraudulent records and 0.54% of the genuine records correctly.

4.3.6.2 Performance Metrics

Model Name	Accuracy	Precision	Recall	AUC	Model Score
Isolation Forest	98.37%	98.32%	95.12%	97.29%	97.01%

Table 4.6
Isolation Forest – Performance Metrics

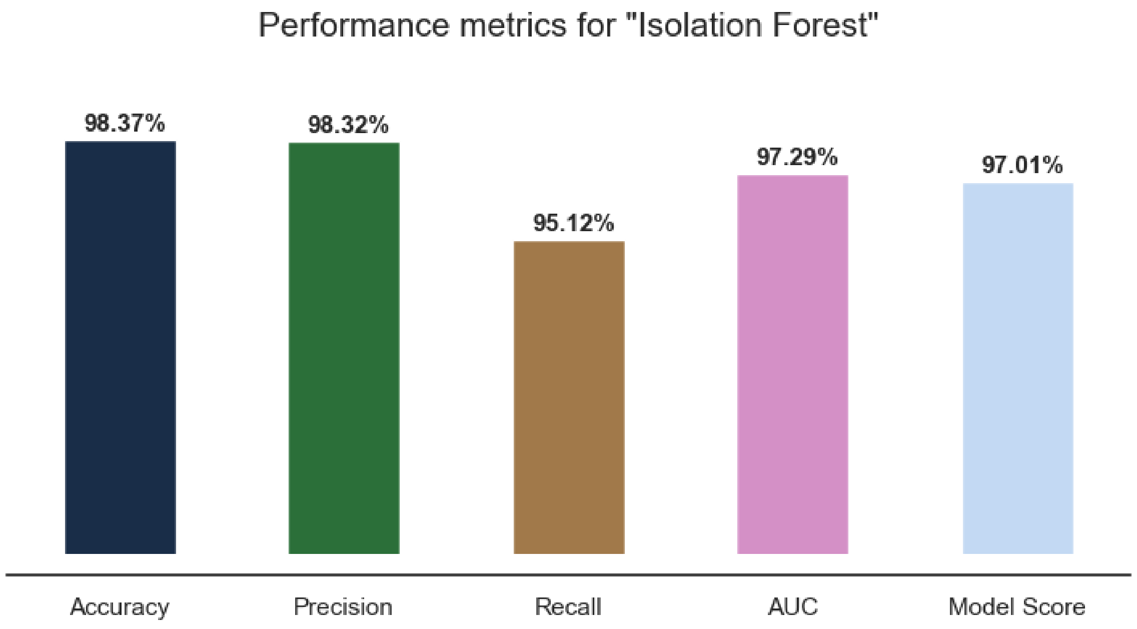

Figure 4.19
Isolation Forest – Performance Metrics

4.3.7 Bagging Classifier

4.3.7.1 Confusion Matrix

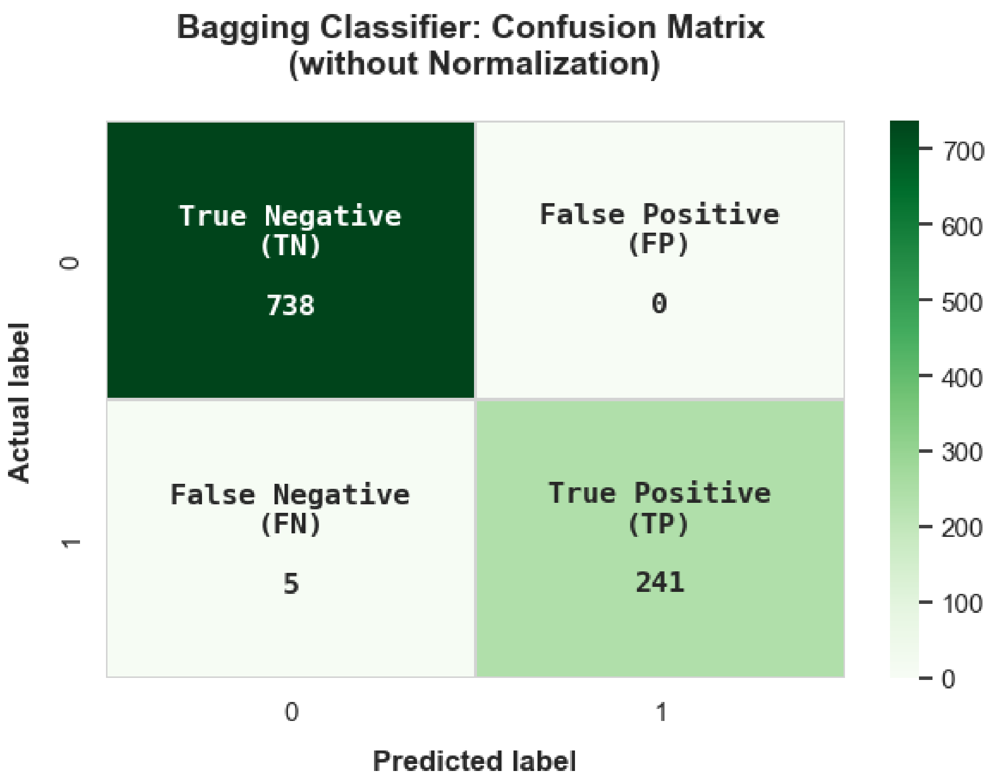

Figure 4.20
Bagging Classifier: Confusion Matrix (without Normalization)

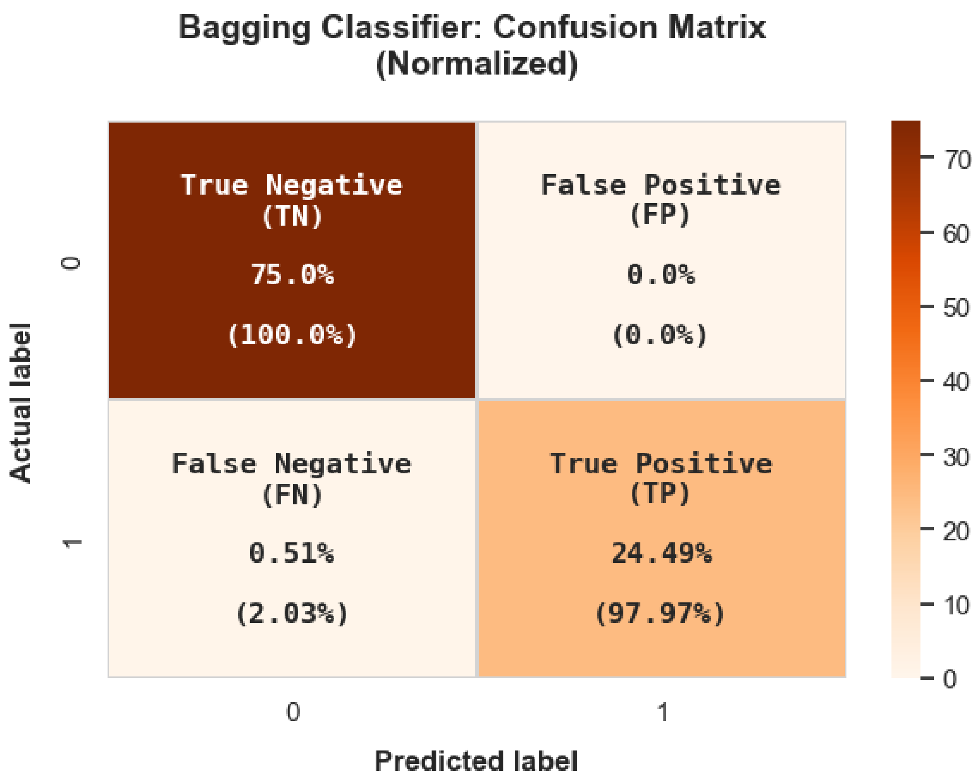

Figure 4.21
Bagging Classifier: Confusion Matrix (Normalized)

The Bagging Classifier excelled in identifying genuine transactions, achieving a flawless detection rate of 100%, which comprised 75% of all transactions. Additionally, it demonstrated strong performance in detecting fraudulent transactions, with an accuracy of 97.97%, representing 24.49% of the total transactions. However, the model missed detecting 2.03% of fraudulent transactions, accounting for 0.51% of the total transactions.

4.3.7.2 Performance Metrics

Model Name	Accuracy	Precision	Recall	AUC	Model Score
Bagging Classifier	99.49%	100.00%	97.97%	99.95%	99.18%

Table 4.7
Bagging Classifier – Performance Metrics

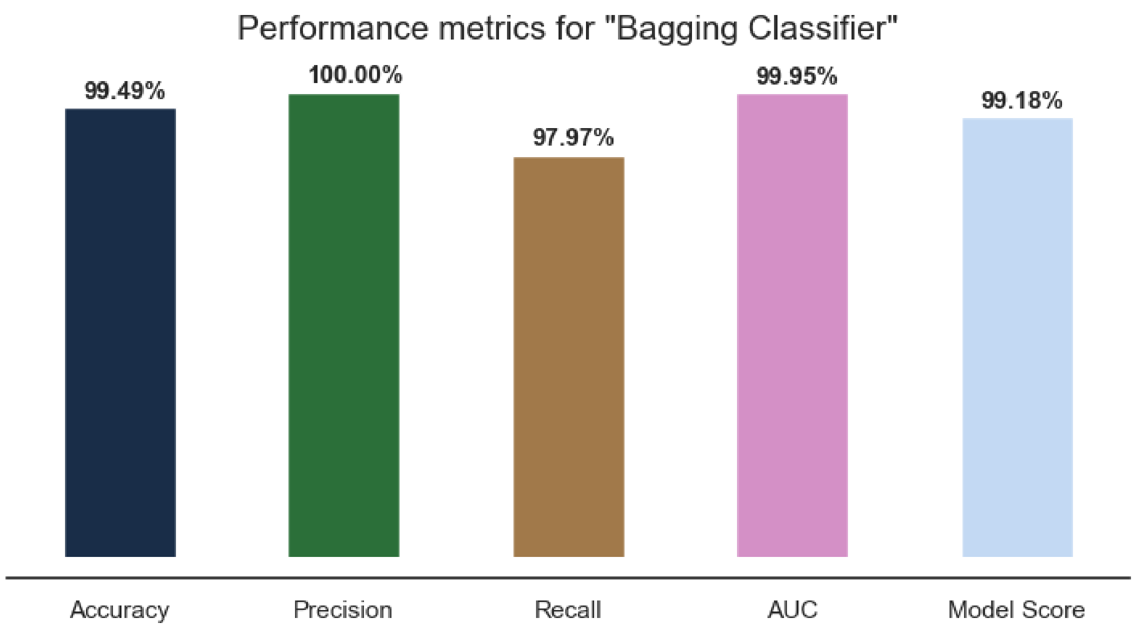

Figure 4.22
Bagging Classifier – Performance Metrics

4.3.8 Decision Tree Classifier

4.3.8.1 Confusion Matrix

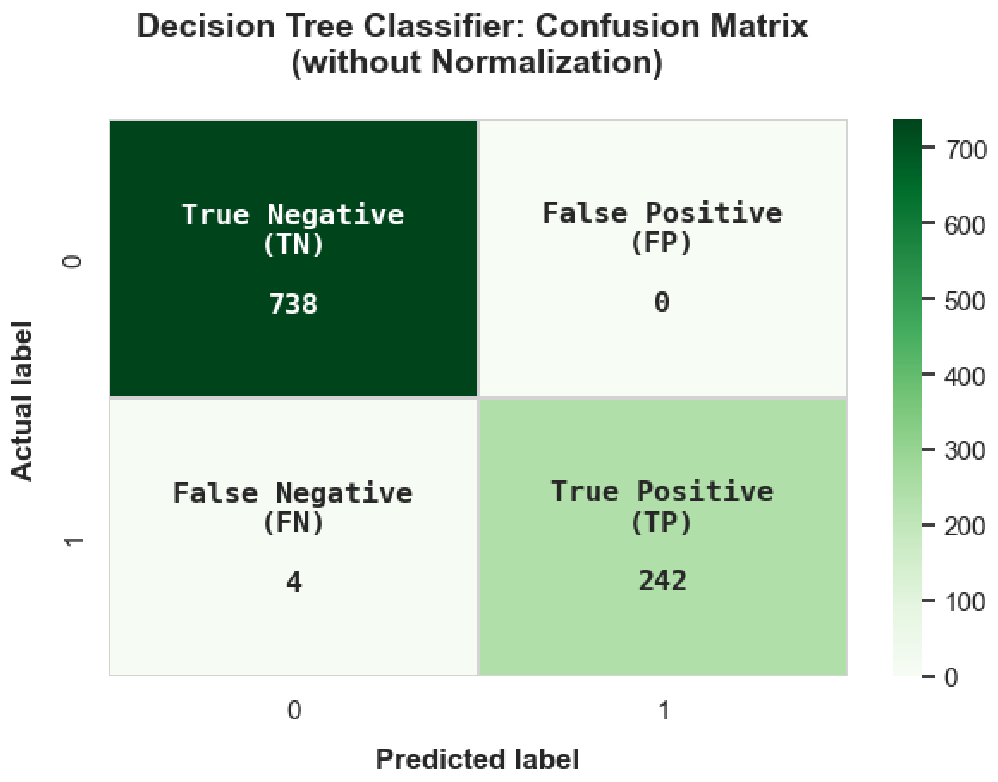

Figure 4.23
Decision Tree Classifier: Confusion Matrix (without Normalization)

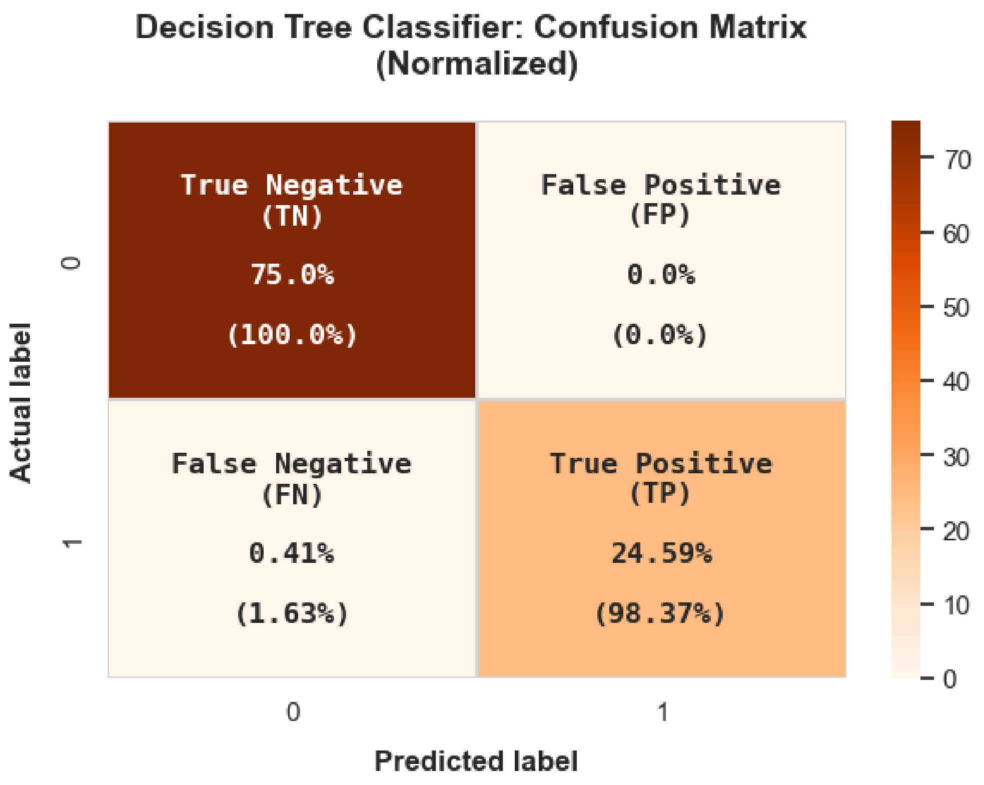

Figure 4.24
Decision Tree Classifier: Confusion Matrix (Normalized)

The Decision Tree Classifier effectively identified all genuine transactions, achieving a perfect detection rate of 100%, which constituted 75% of the total transactions. Moreover, it demonstrated strong performance in detecting fraudulent transactions, with an accuracy of 98.37%, representing 24.59% of the total transactions. However, the model misclassified 1.63% of fraudulent transactions, accounting for 0.41% of the total transactions.

4.3.8.2 Performance Metrics

Model Name	Accuracy	Precision	Recall	AUC	Model Score
Decision Tree Classifier	99.59%	100.00%	98.37%	99.19%	99.19%

Table 4.8
Decision Tree Classifier – Performance Metrics

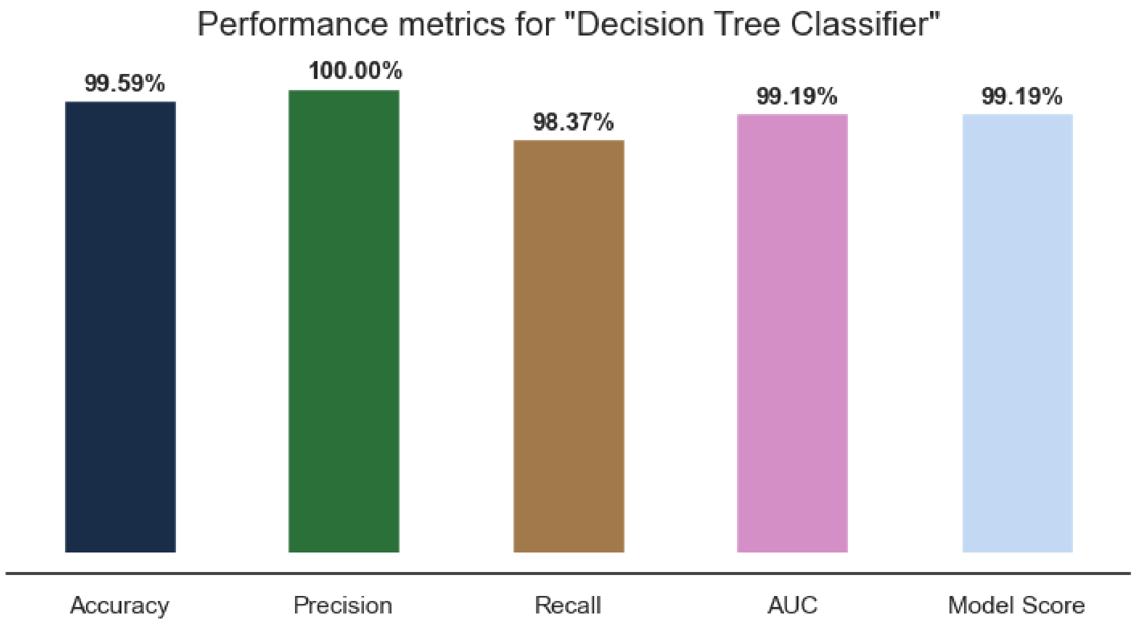

Figure 4.25
Decision Tree Classifier – Performance Metrics

4.3.9 Keras Classifier

4.3.9.1 Confusion Matrix

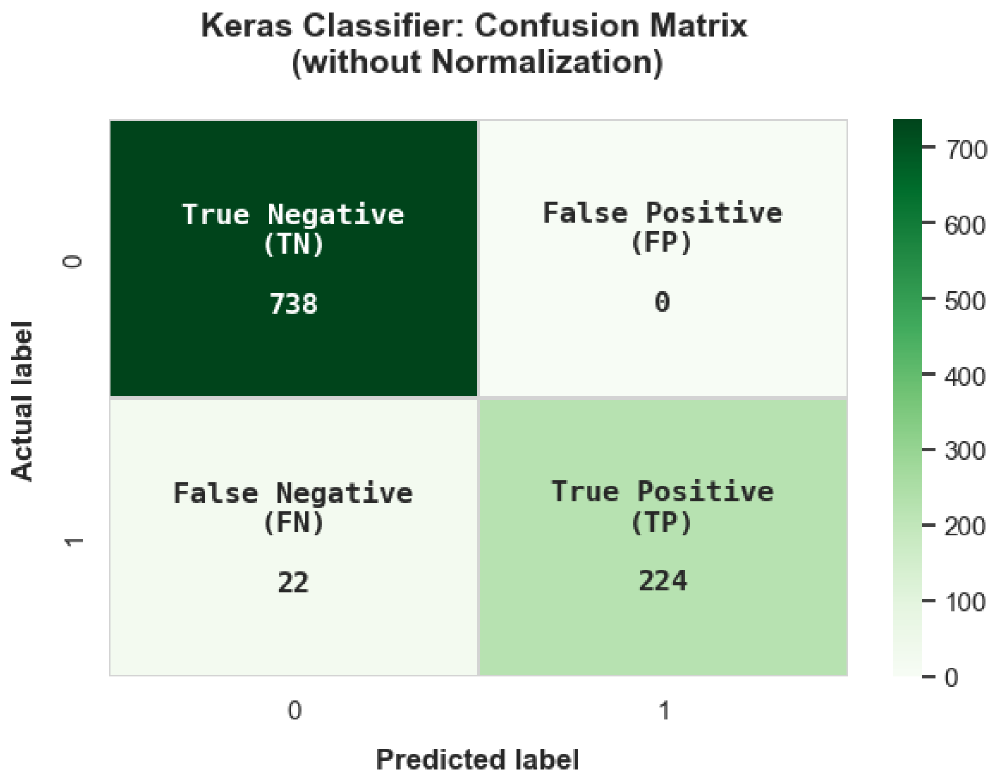

Figure 4.26
Keras Classifier: Confusion Matrix (without Normalization)

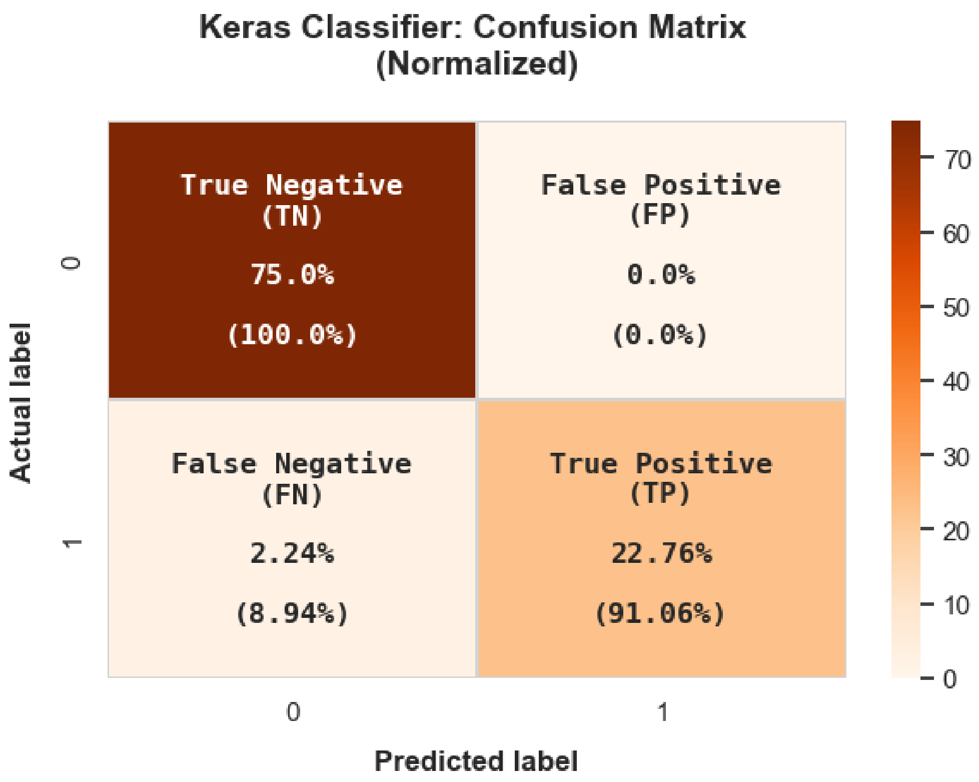

Figure 4.27
Keras Classifier: Confusion Matrix (Normalized)

The Keras Classifier accurately classified all genuine records, achieving a perfect classification rate of 100%, which constituted 75% of the total transactions. Additionally, it demonstrated strong performance in classifying fraudulent transactions, with an accuracy of 91.06%, representing 22.76% of the total transactions. However, the model incorrectly classified 8.94% of fraudulent transactions, accounting for 2.24% of the total transactions.

4.3.9.2 Performance Metrics

Model Name	Accuracy	Precision	Recall	AUC	Model Score
Keras Classifier	97.76%	100.00%	91.06%	97.87%	96.00%

Table 4.9
Keras Classifier – Performance Metrics

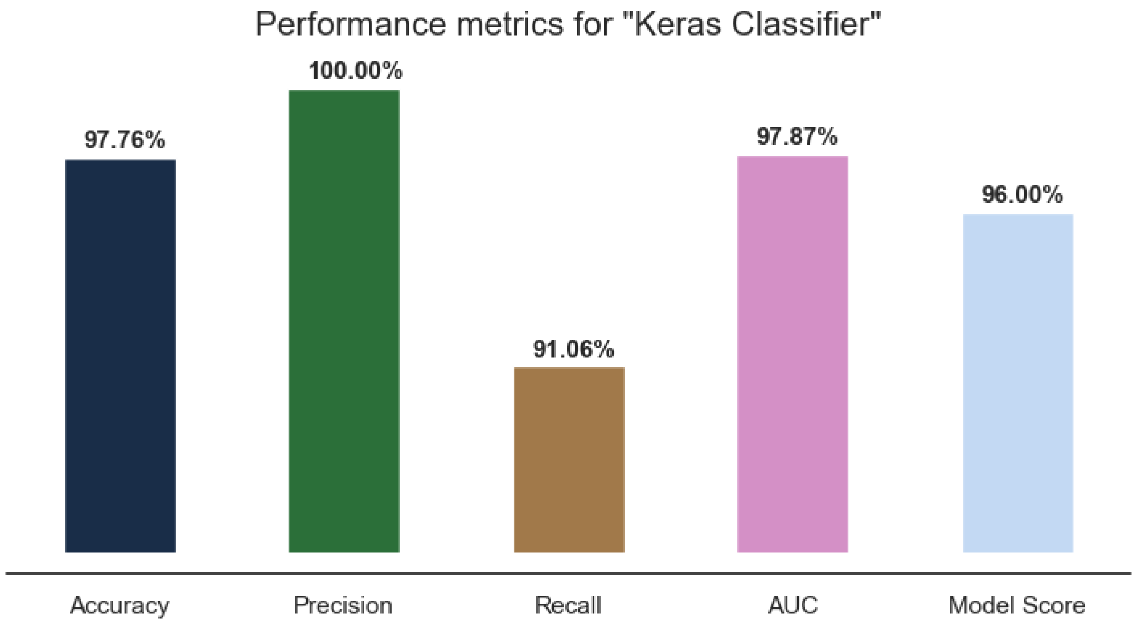

Figure 4.28
Keras Classifier – Performance Metrics

4.3.10 Adaboost Classifier

4.3.10.1 Confusion Matrix

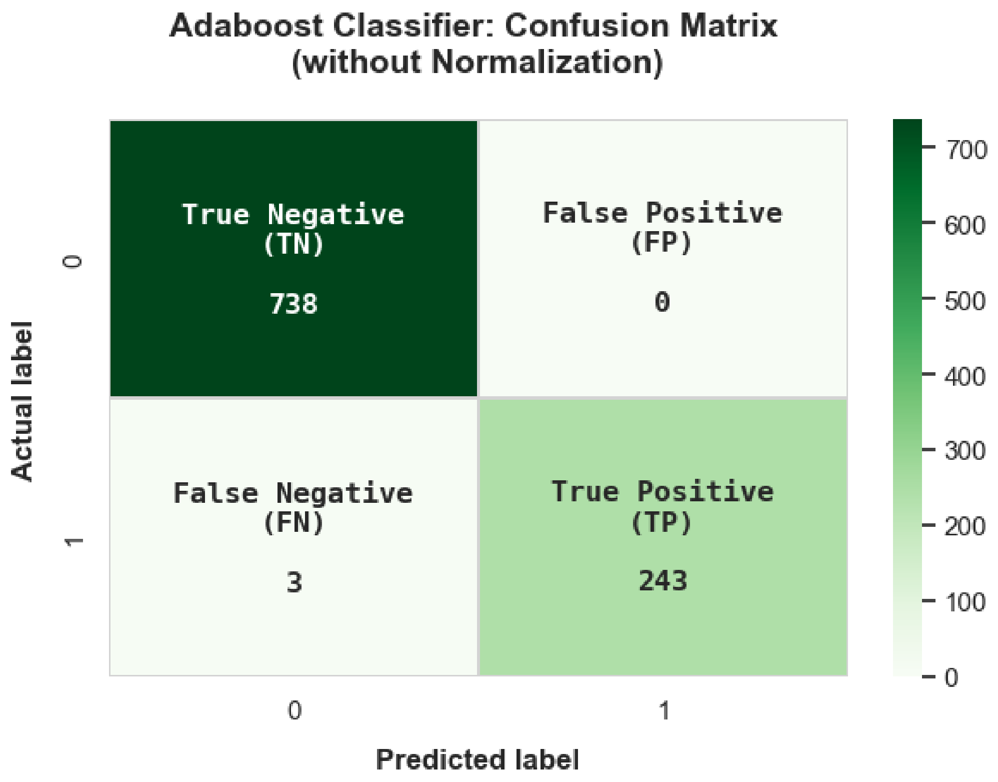

Figure 4.29
Adaboost Classifier: Confusion Matrix (without Normalization)

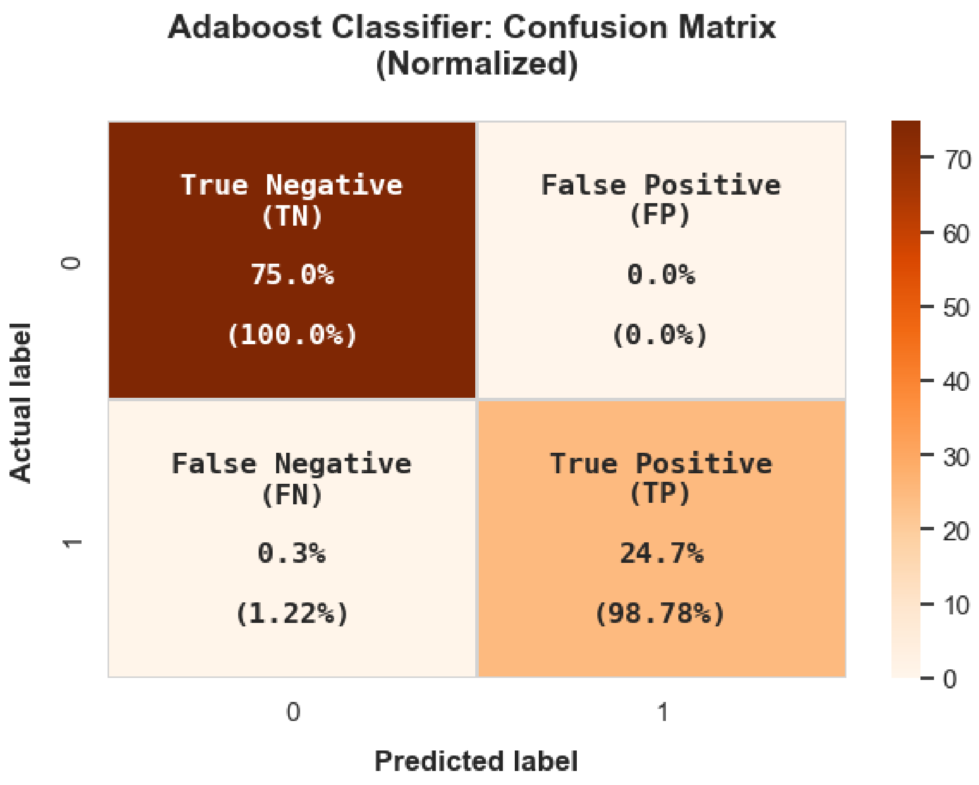

Figure 4.30
Adaboost Classifier: Confusion Matrix (Normalized)

The Adaboost Classifier performed remarkably well in identifying both genuine and fraudulent transactions. It correctly flagged all genuine records, which constituted 75% of the total transactions, achieving a flawless detection rate. Furthermore, it exhibited an impressive accuracy of 98.78% in identifying fraudulent transactions, encompassing nearly a quarter of the total transactions. Despite its high accuracy, the model encountered a minor setback as it misclassified 1.22% of fraudulent records, equivalent to only 0.3% of the total transactions.

4.3.10.2 Performance Metrics

Model Name	Accuracy	Precision	Recall	AUC	Model Score
Adaboost Classifier	99.70%	100.00%	98.78%	99.39%	99.39%

Table 4.10
Adaboost Classifier – Performance Metrics

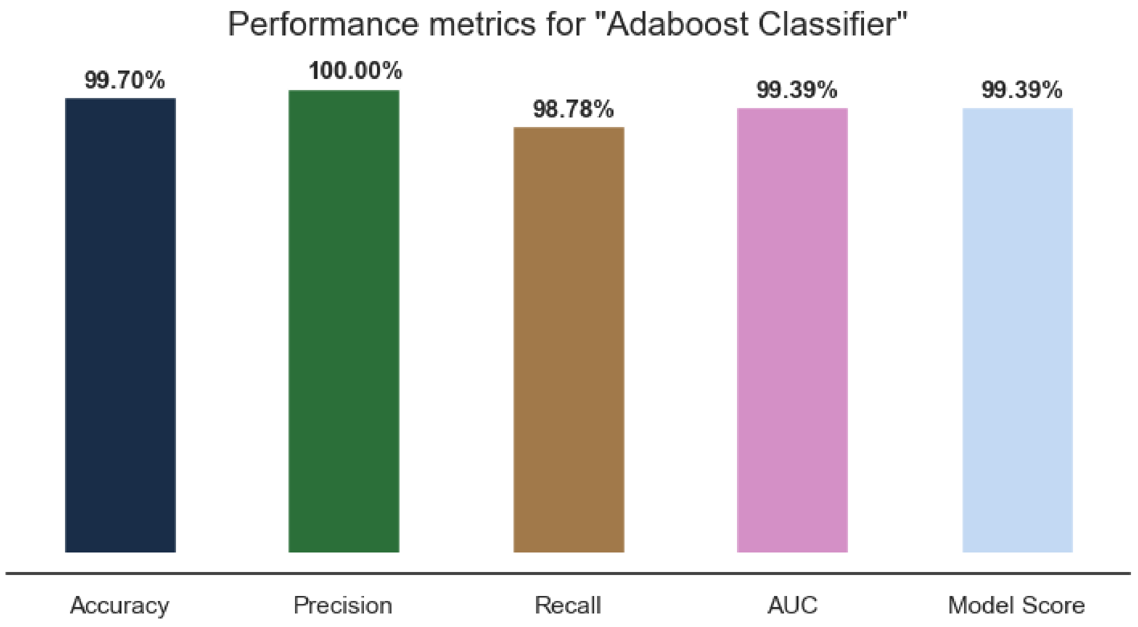

Figure 4.31
Adaboost Classifier – Performance Metrics

4.3.11 LightGBM Classifier

4.3.11.1 Confusion Matrix

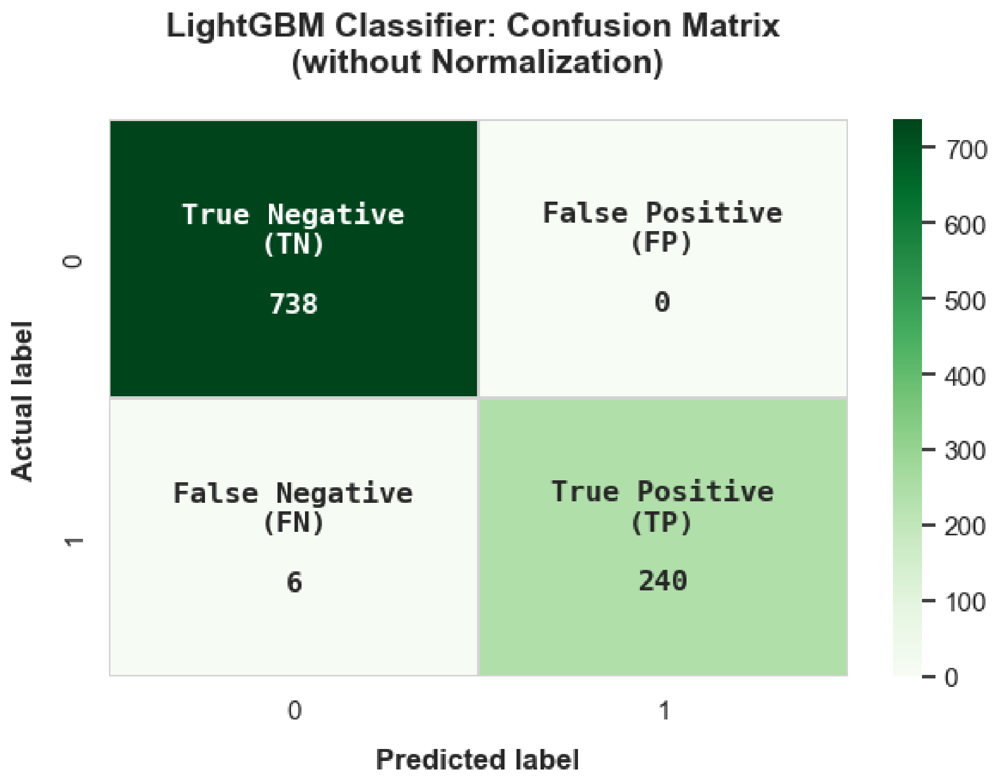

Figure 4.32
LightGBM Classifier: Confusion Matrix (without Normalization)

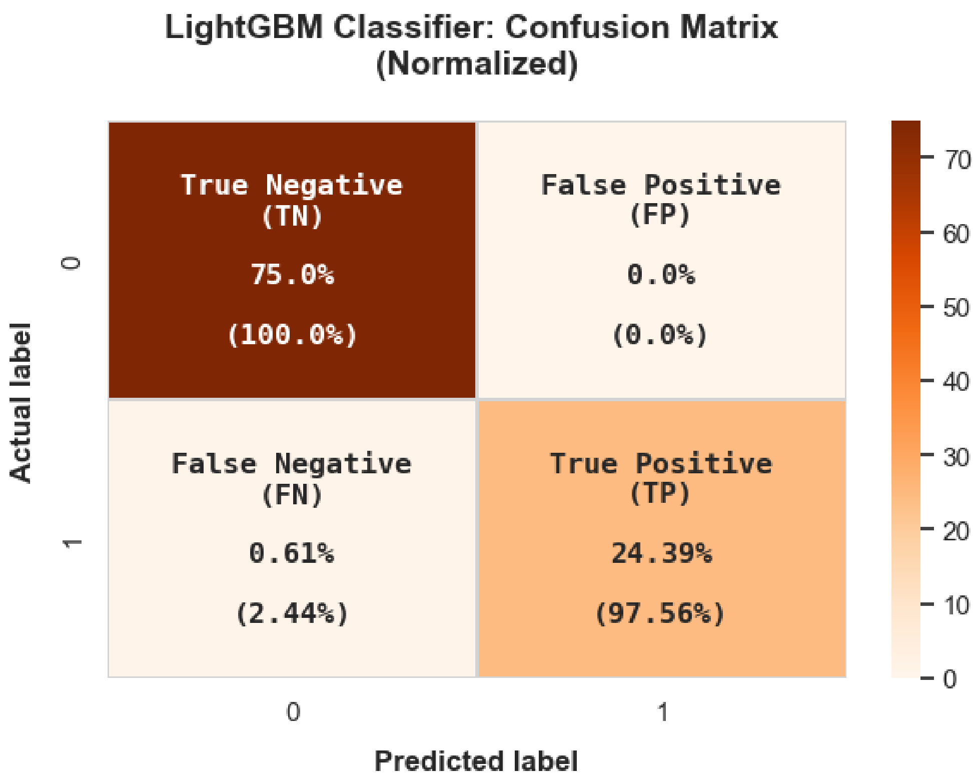

Figure 4.33
LightGBM Classifier: Confusion Matrix (Normalized)

The LightGBM Classifier demonstrated impressive accuracy in distinguishing between genuine and fraudulent transactions. It performed flawlessly in detecting all genuine records, which constituted 75% of the total dataset. Additionally, it showed strong capability by accurately identifying 97.56% of fraudulent transactions, accounting for nearly a quarter of the total records. Despite its overall high accuracy, the model encountered a slight challenge as it misclassified 2.44% of fraudulent records, equivalent to 0.61% of the total records.

4.3.11.2 Performance Metrics

Model Name	Accuracy	Precision	Recall	AUC	Model Score
LightGBM Classifier	99.39%	100.00%	97.56%	99.77%	98.98%

Table 4.11
LightGBM Classifier – Performance Metrics

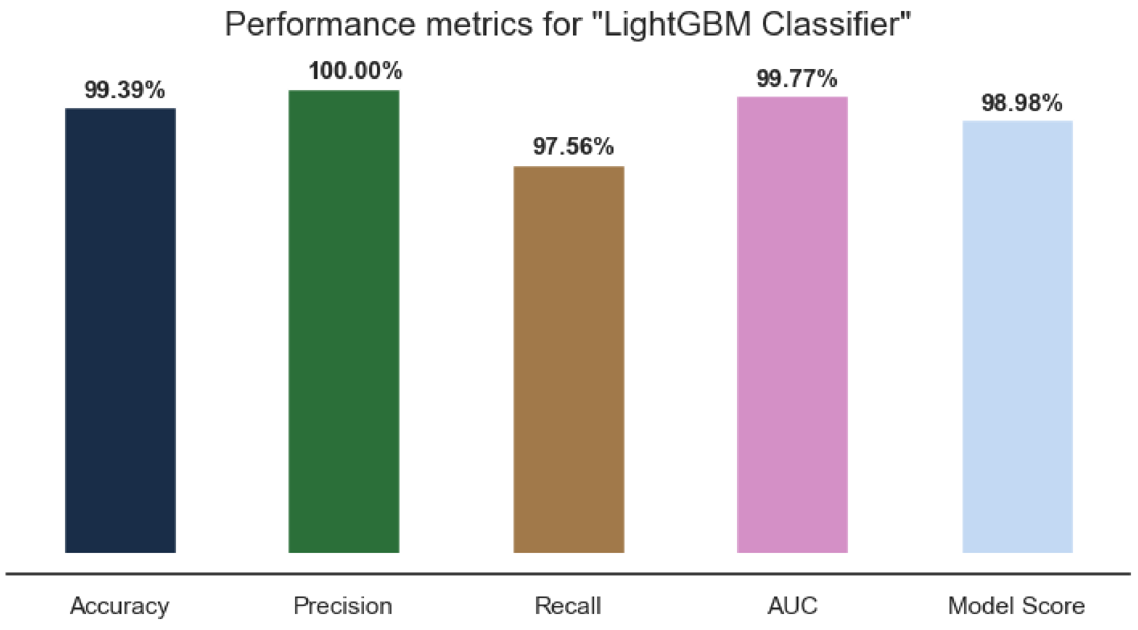

Figure 4.34
LightGBM Classifier – Performance Metrics

4.3.12 Catboost Classifier

4.3.12.1 Confusion Matrix

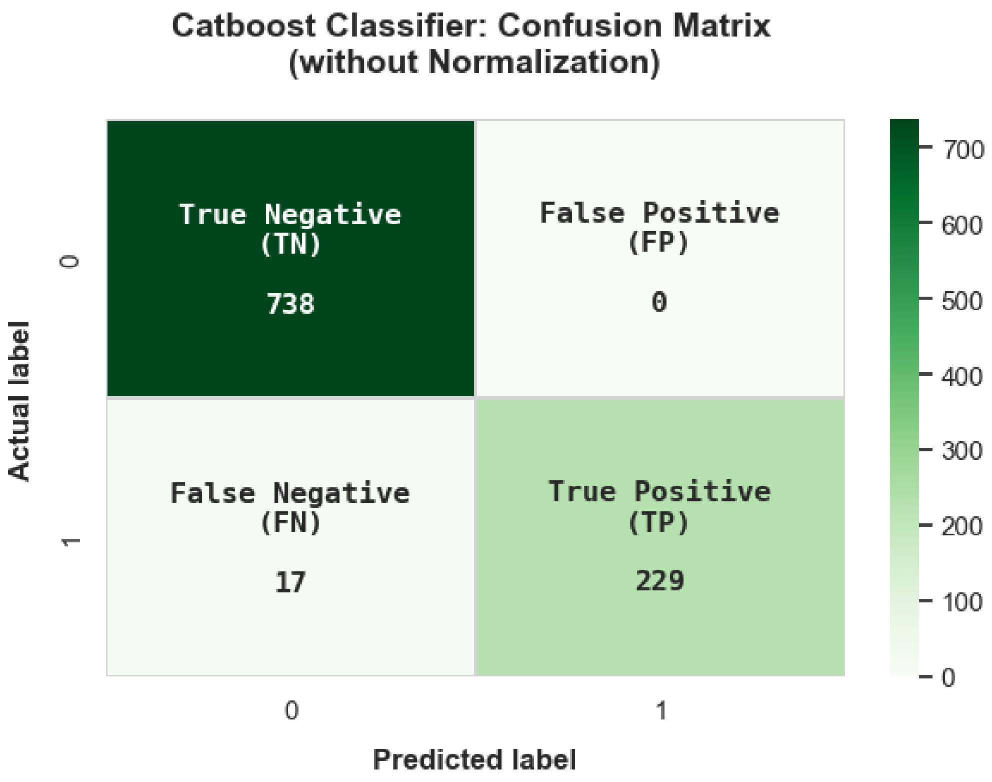

Figure 4.35
Catboost Classifier: Confusion Matrix (without Normalization)

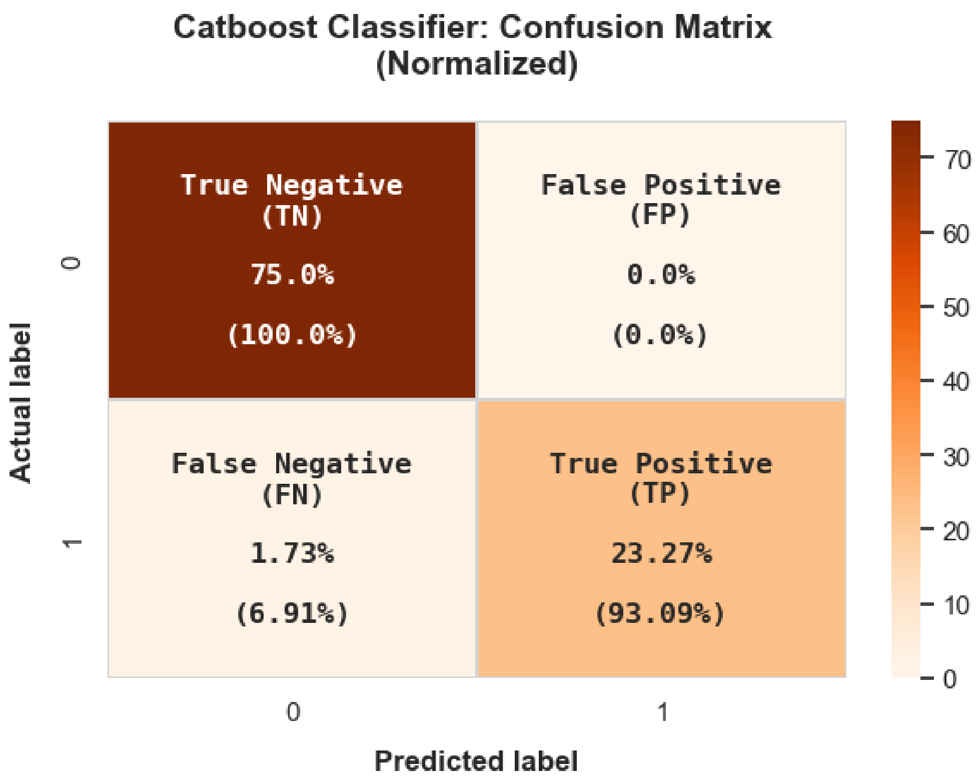

Figure 4.36
Catboost Classifier: Confusion Matrix (Normalized)

The Catboost Classifier demonstrated impressive precision in distinguishing between genuine and fraudulent transactions. It flawlessly identified all genuine records, which comprised 75% of the total transactions. Additionally, it displayed a commendable accuracy of 93.09% in identifying fraudulent transactions, representing nearly a quarter of the total transactions. However, the model encountered a notable challenge as it exhibited a higher misclassification rate for fraudulent records, reaching 6.91%, equivalent to 1.73% of the total transactions.

4.3.12.2 Performance Metrics

Model Name	Accuracy	Precision	Recall	AUC	Model Score
Catboost Classifier	98.27%	100.00%	93.09%	99.59%	97.15%

Table 4.12
Catboost Classifier – Performance Metrics

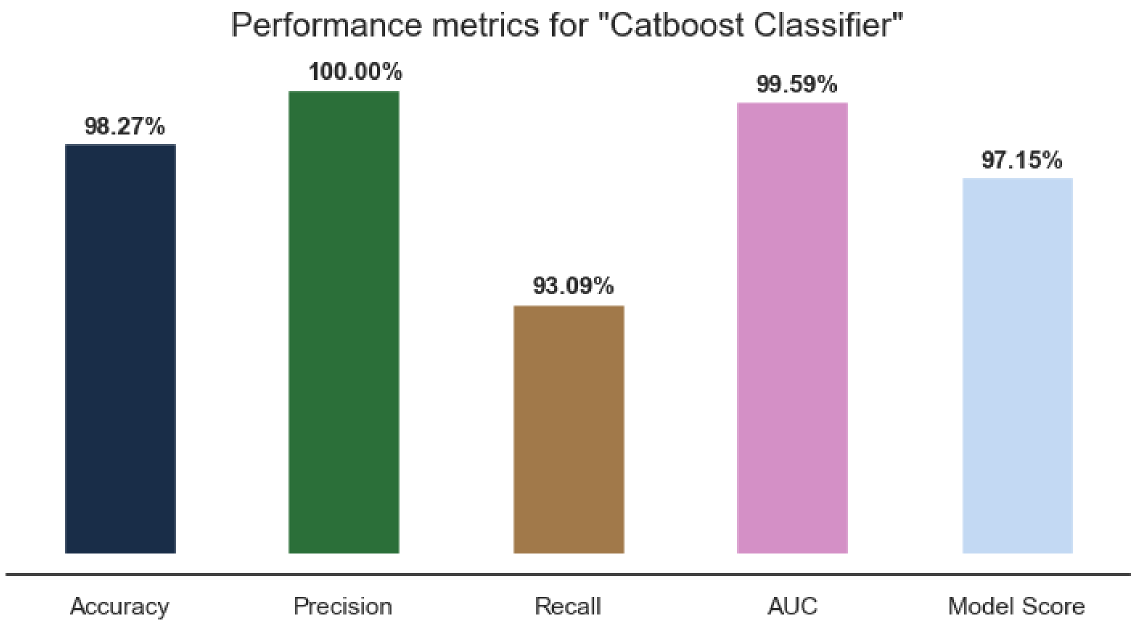

Figure 4.37
Catboost Classifier – Performance Metrics

4.3.13 XGboost Classifier

4.3.13.1 Confusion Matrix

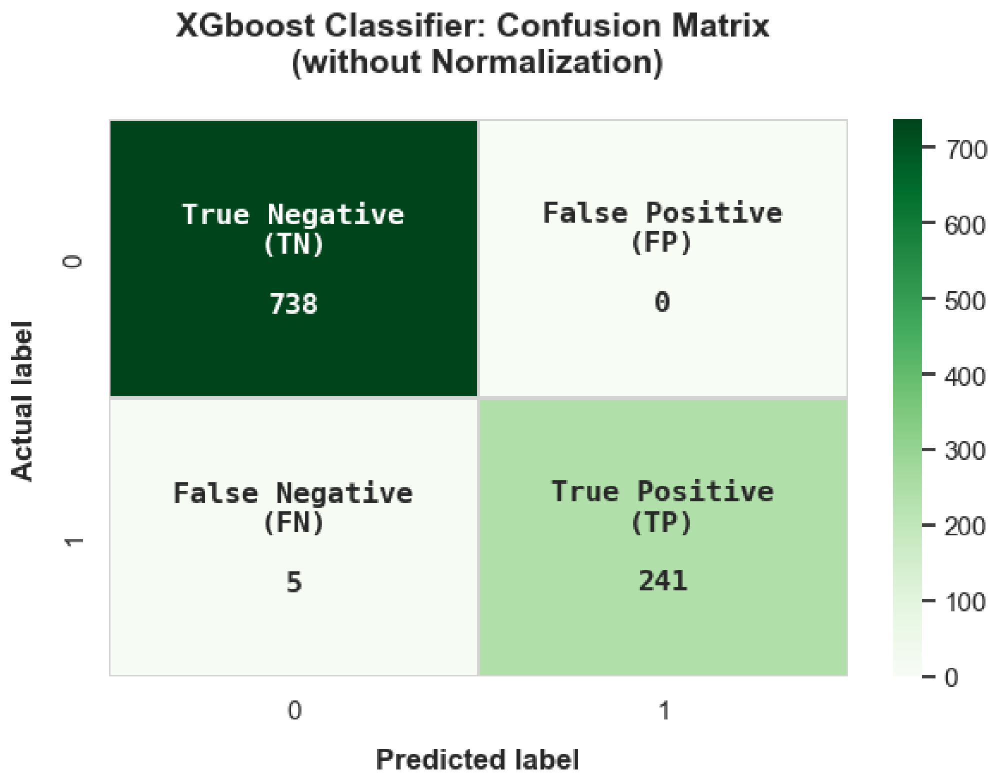

Figure 4.38
XGboost Classifier: Confusion Matrix (without Normalization)

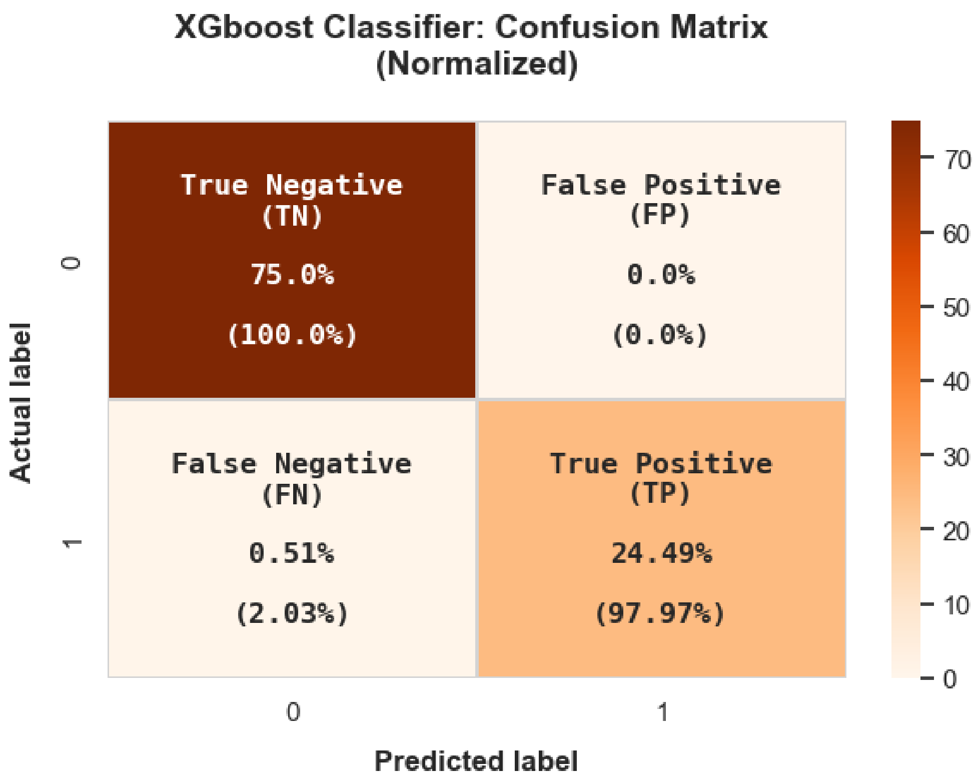

Figure 4.39
XGboost Classifier: Confusion Matrix (Normalized)

The XGBoost Classifier showcased impressive precision in distinguishing between genuine and fraudulent transactions. It flawlessly identified all genuine records, constituting 75% of the total transactions. Additionally, it demonstrated robust accuracy by successfully detecting 97.97% of fraudulent transactions, which comprised nearly a quarter of the total transactions. Despite its overall accuracy, the model encountered a slight challenge as it misclassified 2.03% of fraudulent transactions, accounting for 0.51% of the total transactions.

4.3.13.2 Performance Metrics

Model Name	Accuracy	Precision	Recall	AUC	Model Score
XGboost Classifier	99.49%	100.00%	97.97%	99.91%	99.17%

Table 4.13
XGboost Classifier – Performance Metrics

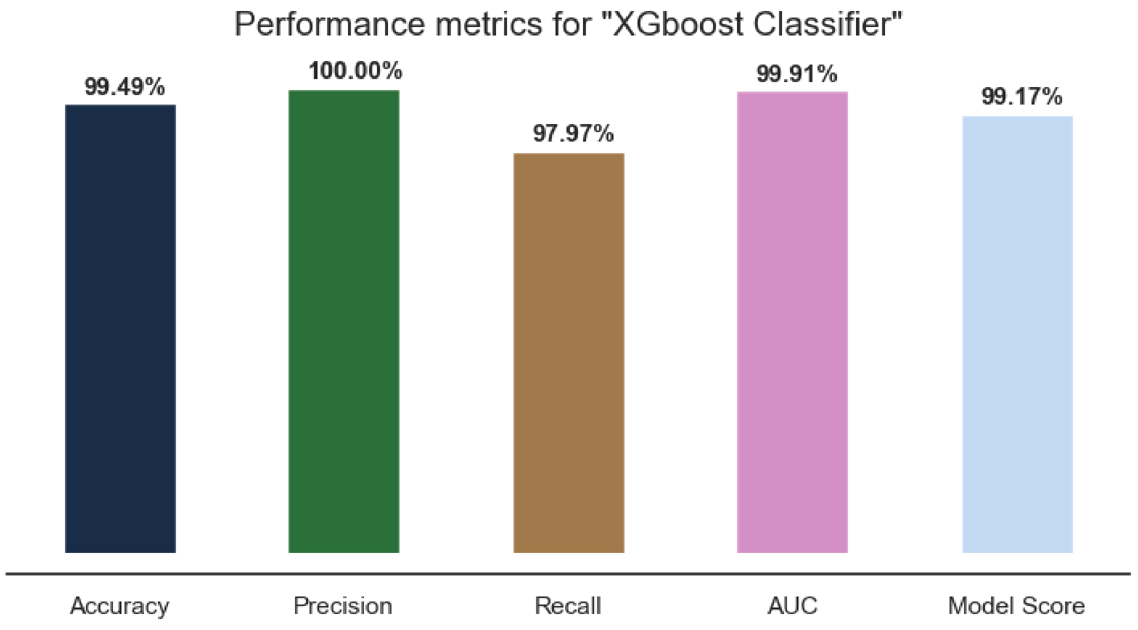

Figure 4.40
XGboost Classifier – Performance Metrics

4.3.14 MLP Classifier

4.3.14.1 Confusion Matrix

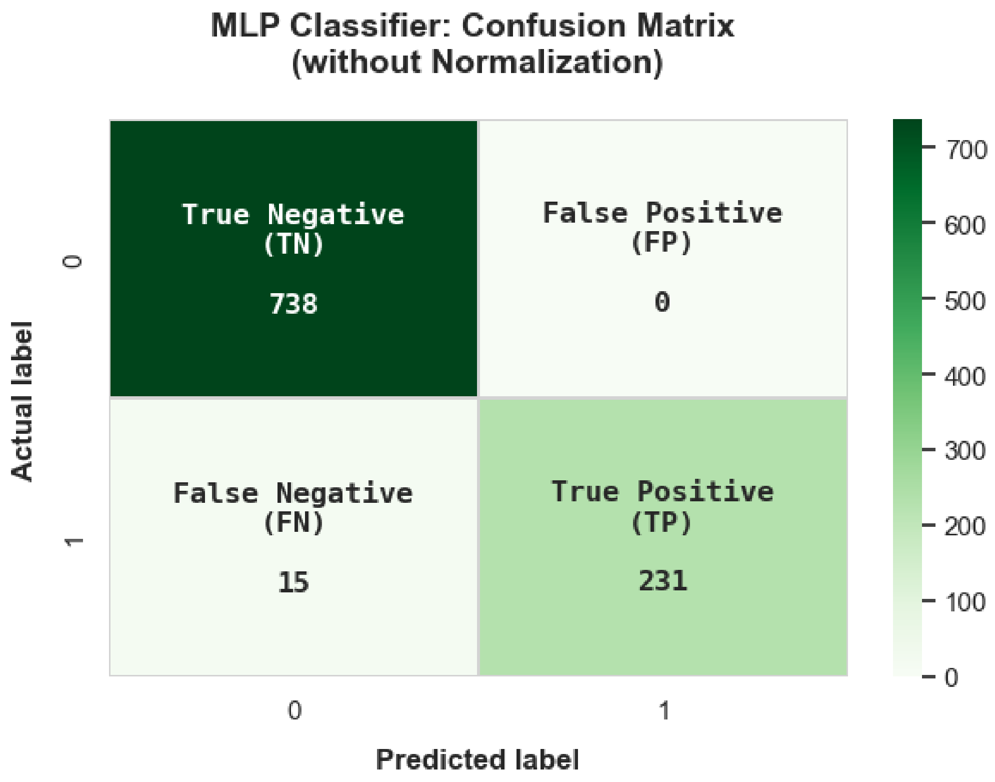

Figure 4.41
MLP Classifier: Confusion Matrix (without Normalization)

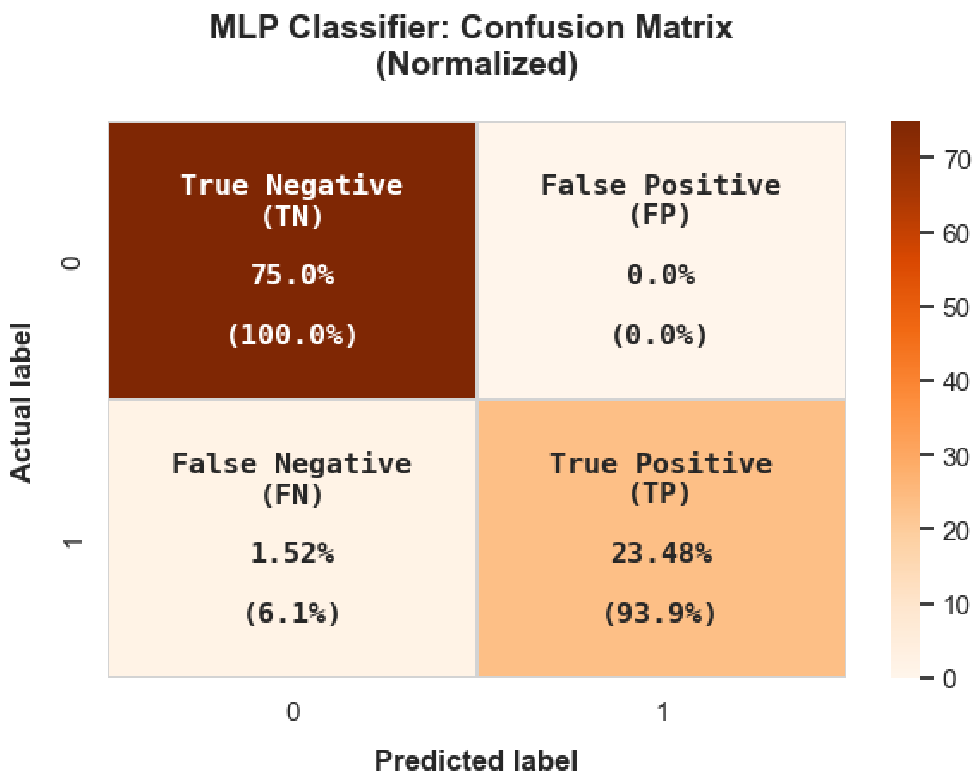

Figure 4.42
MLP Classifier: Confusion Matrix (Normalized)

The MLP Classifier demonstrated strong accuracy in distinguishing between genuine and fraudulent transactions. It successfully classified all genuine transactions, comprising 75% of the total records, with 100% accuracy. Moreover, it displayed a commendable performance by correctly identifying 93.9% of fraudulent transactions, which constituted 23.48% of the total records. However, the model encountered a notable challenge in classifying 6.1% of fraudulent records, accounting for 1.52% of the total records.

4.3.14.2 Performance Metrics

Model Name	Accuracy	Precision	Recall	AUC	Model Score
MLP Classifier	98.48%	100.00%	93.90%	98.73%	97.31%

Table 4.14
MLP Classifier – Performance Metrics

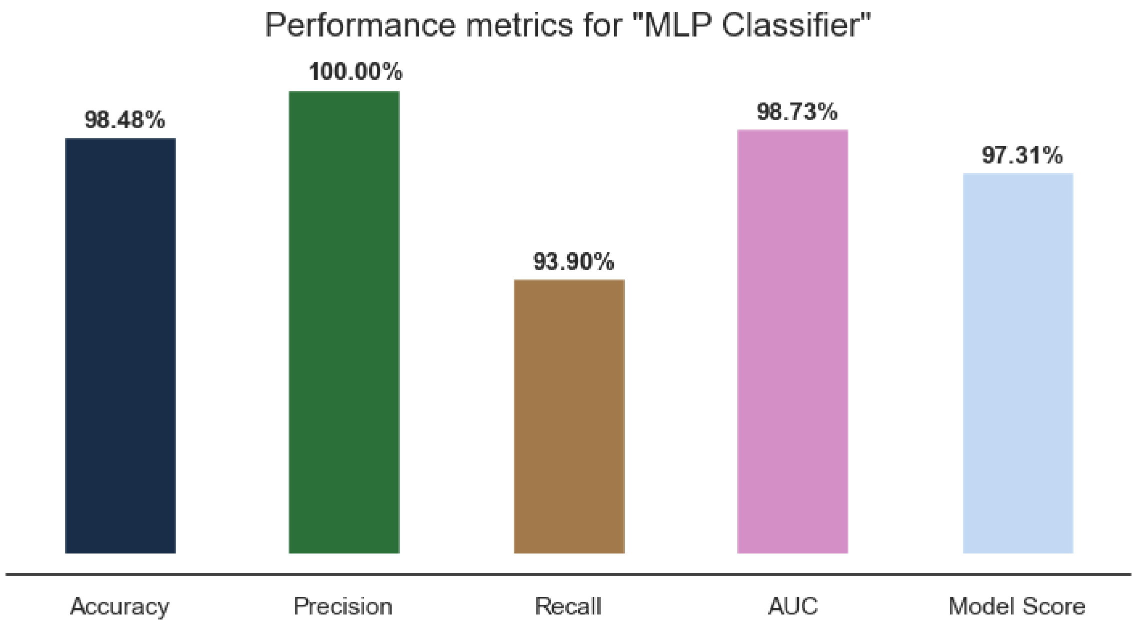

Figure 4.43
MLP Classifier – Performance Metrics

4.3.15 Dynamic Ensemble

4.3.15.1 Confusion Matrix

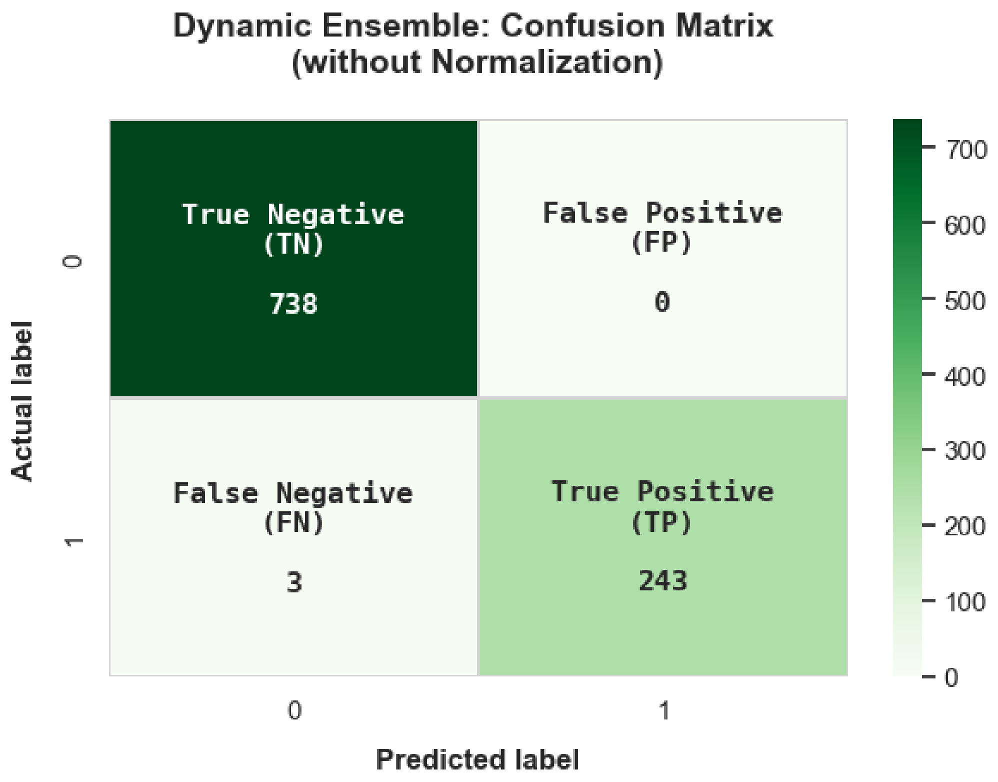

Figure 4.44
Dynamic Ensemble: Confusion Matrix (without Normalization)

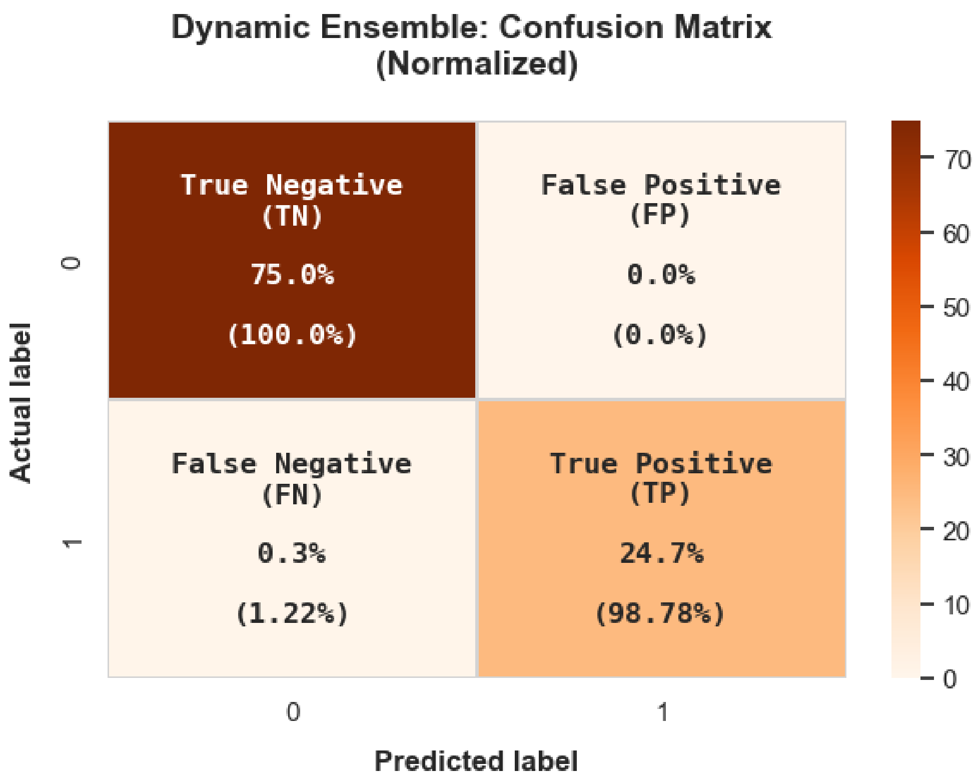

Figure 4.45
Dynamic Ensemble: Confusion Matrix (Normalized)

The Dynamic Ensemble model demonstrated exceptional accuracy in discerning between genuine and fraudulent transactions. It successfully identified all genuine transactions, which made up 75% of the total transactions, achieving a flawless detection rate. Additionally, it displayed robust performance by accurately detecting 98.78% of fraudulent transactions, representing nearly a quarter of the total transactions. Despite its overall accuracy, the model encountered a minor issue, misclassifying 1.22% of fraudulent transactions, which accounted for 0.3% of the total transactions.

4.3.15.2 Performance Metrics

Model Name	Accuracy	Precision	Recall	AUC	Model Score
Dynamic Ensemble	99.70%	100.00%	98.78%	99.96%	99.50%

Table 4.15
Dynamic Ensemble – Performance Metrics

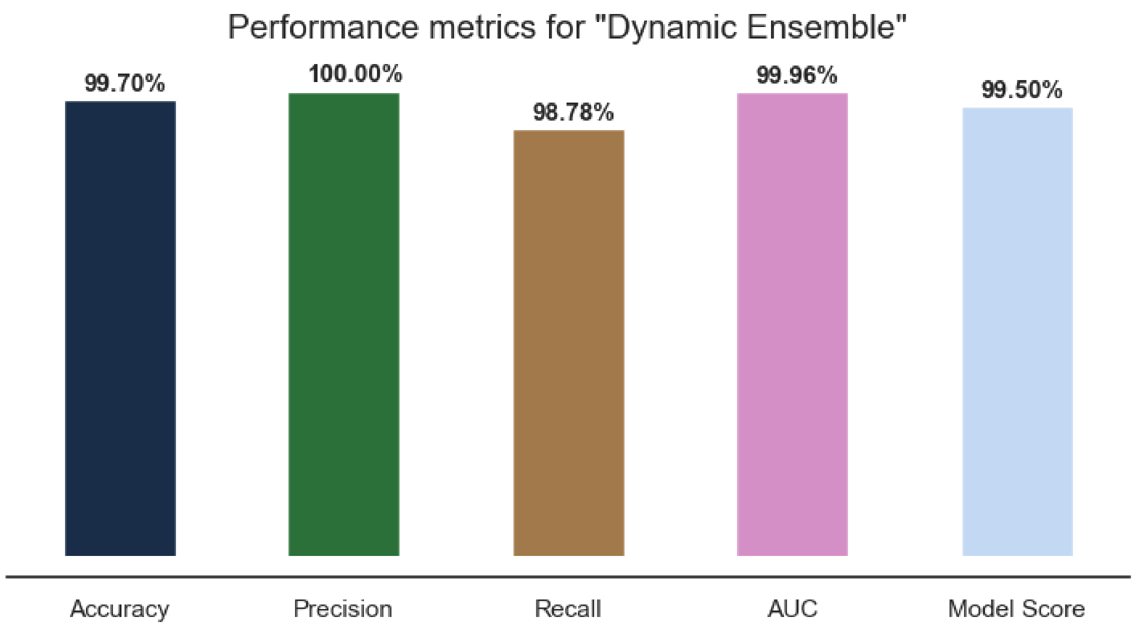

Figure 4.46
Dynamic Ensemble – Performance Metrics

4.3.16 Stacking

4.3.16.1 Confusion Matrix

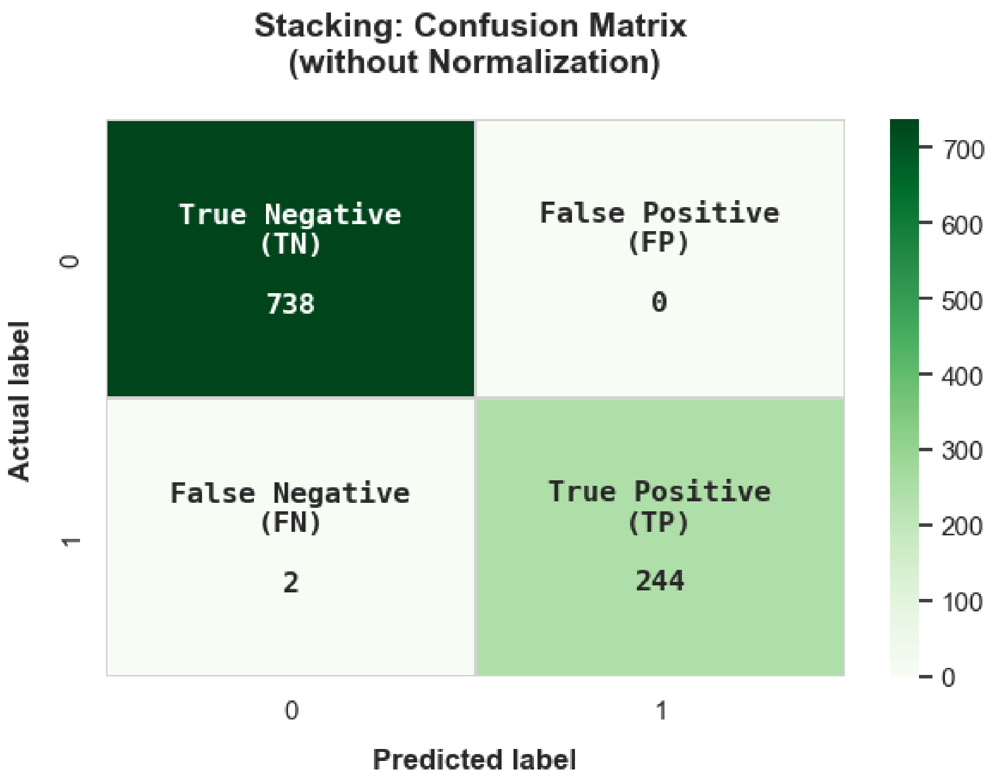

Figure 4.47
Stacking: Confusion Matrix (without Normalization)

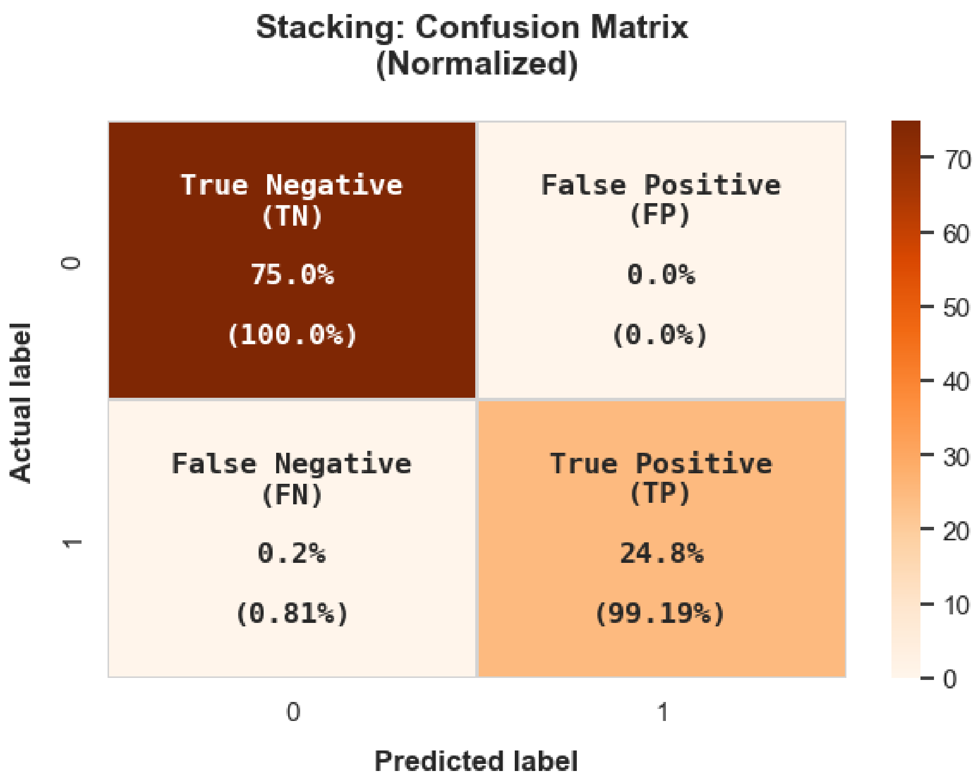

Figure 4.48
Stacking: Confusion Matrix (Normalized)

The Stacking model showcased exceptional accuracy in distinguishing between genuine and fraudulent transactions. It flawlessly detected all genuine transactions, which comprised 75% of the total transactions. Additionally, it demonstrated robust performance by accurately identifying 99.19% of fraudulent transactions, representing nearly a quarter of the total transactions. Despite its high accuracy, the model encountered a minor issue, as it misclassified only 0.81% of fraudulent transactions, accounting for 0.2% of the total transactions.

4.3.16.2 Performance Metrics

Model Name	Accuracy	Precision	Recall	AUC	Model Score
Stacking	99.80%	100.00%	99.19%	99.66%	99.61%

Table 4.16
Stacking – Performance Metrics

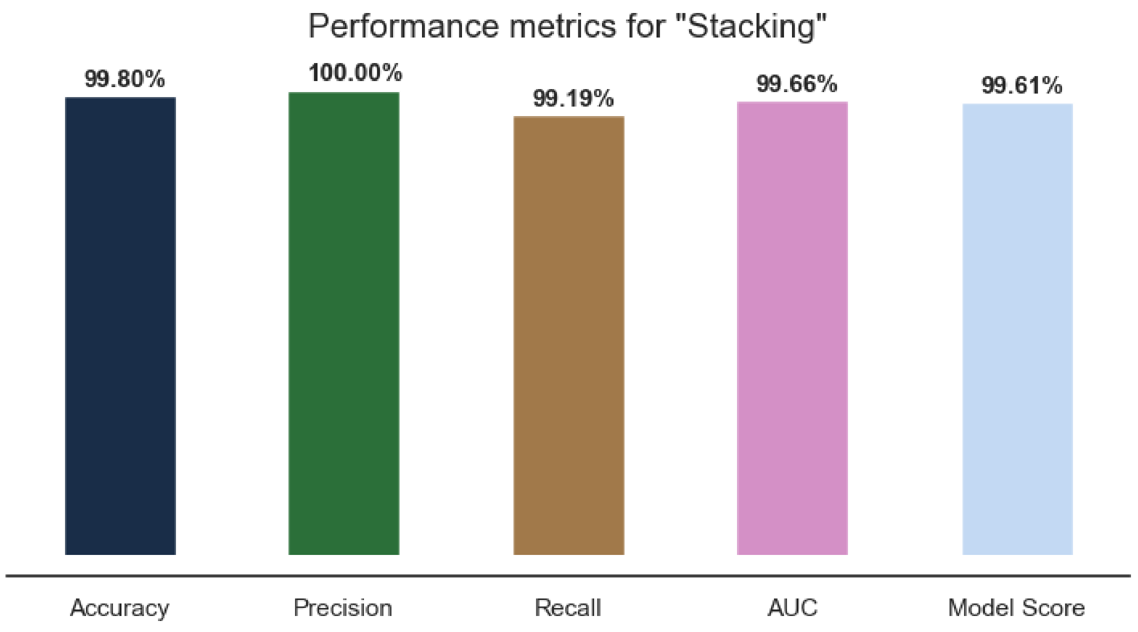

Figure 4.49
Stacking – Performance Metrics

4.4 Models Performance Evaluation

The following table illustrates the comparison of all models across various metrics including "Accuracy," "Precision," "Recall," "AUC," and "Model Score."

Model Name	Accuracy	Precision	Recall	AUC	Model Score
Logistic Regression	98.27%	98.31%	94.72%	98.64%	97.11%
Gaussian Naïve Bayes	98.17%	100.00%	92.68%	98.67%	96.81%
Linear Support Vector Classifier	98.37%	97.92%	95.53%	97.43%	97.07%
Kneighbors Classifier	98.17%	100.00%	92.68%	96.94%	96.46%
Random Forest Classifier	99.80%	100.00%	99.19%	99.75%	99.63%
Isolation Forest	98.37%	98.32%	95.12%	97.29%	97.01%
Bagging Classifier	99.49%	100.00%	97.97%	99.95%	99.18%
Decision Tree Classifier	99.59%	100.00%	98.37%	99.19%	99.19%
Keras Classifier	97.76%	100.00%	91.06%	97.87%	96.00%
Adaboost Classifier	99.70%	100.00%	98.78%	99.39%	99.39%
LightGBM Classifier	99.39%	100.00%	97.56%	99.77%	98.98%
Catboost Classifier	98.27%	100.00%	93.09%	99.59%	97.15%
XGboost Classifier	99.49%	100.00%	97.97%	99.91%	99.17%
MLP Classifier	98.48%	100.00%	93.90%	98.73%	97.31%
Dynamic Ensemble	99.70%	100.00%	98.78%	99.96%	99.50%
Stacking	99.80%	100.00%	99.19%	99.66%	99.61%

Table 4.17
Performance Metrics Comparison for all models in tabular format

In the above table, highest value for each performance metrics highlighted in "light green" and lowest value highlighted in "light orange".

The performance comparison details of the models are presented in the following figure.

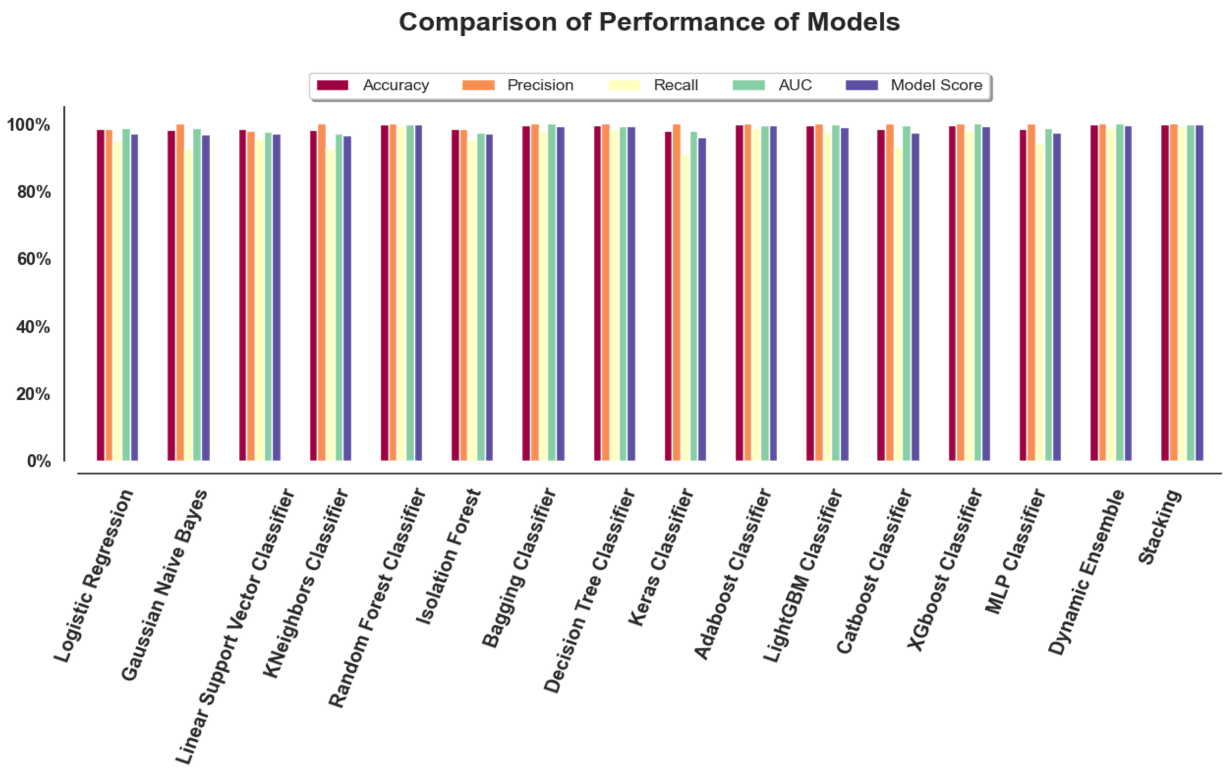

Figure 4.50
Performance Metrics Comparison for all models in graphical format

In the sections below, we'll examine how the models compare in terms of individual performance metrics.

4.4.1 ACCURACY Comparison

Model Name	Accuracy
Logistic Regression	98.27%
Gaussian Naïve Bayes	98.17%
Linear Support Vector Classifier	98.37%
Kneighbors Classifier	98.17%
Random Forest Classifier	99.80%
Isolation Forest	98.37%
Bagging Classifier	99.49%
Decision Tree Classifier	99.59%
Keras Classifier	97.76%
Adaboost Classifier	99.70%
LightGBM Classifier	99.39%
Catboost Classifier	98.27%
XGboost Classifier	99.49%
MLP Classifier	98.48%
Dynamic Ensemble	99.70%
Stacking	99.80%

Table 4.18
Comparison of "Accuracy" of Models

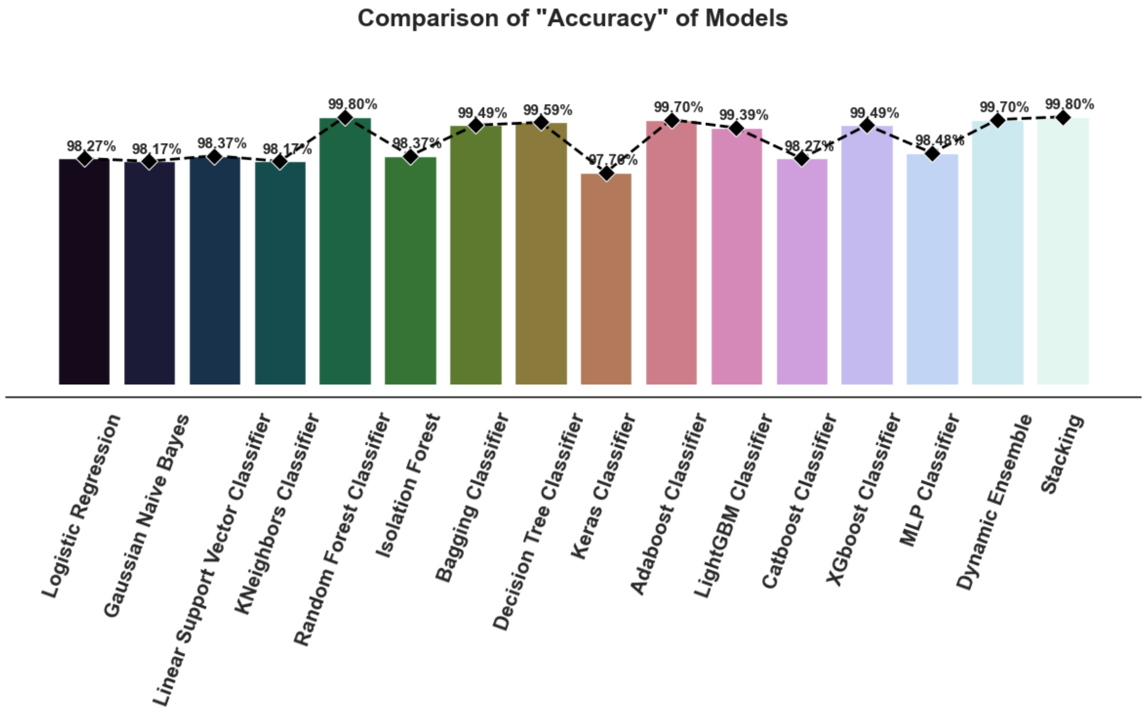

Figure 4.51
Comparison of "Accuracy" of Models

Accuracy is a metric that gauges the overall correctness of a model's predictions by comparing them to all predictions generated by the model. It doesn't differentiate between the effectiveness of the model in predicting genuine transactions versus fraudulent ones.

The accuracy of the models falls within a range of 97.76% to 99.80%. Among all evaluated models, Random Forest Classifier and Stacking achieved the highest accuracy scores, while Logistic Regression performed the least accurately. Adaboost and Dynamic Ensemble with accuracy of 99.70% followed closely.

4.4.2 PRECISION Comparison

Model Name	Precision
Logistic Regression	98.31%
Gaussian Naïve Bayes	100.00%
Linear Support Vector Classifier	97.92%
Kneighbors Classifier	100.00%
Random Forest Classifier	100.00%
Isolation Forest	98.32%
Bagging Classifier	100.00%
Decision Tree Classifier	100.00%
Keras Classifier	100.00%
Adaboost Classifier	100.00%
LightGBM Classifier	100.00%
Catboost Classifier	100.00%
XGboost Classifier	100.00%
MLP Classifier	100.00%
Dynamic Ensemble	100.00%
Stacking	100.00%

Table 4.19
Comparison of "Precision" of Models

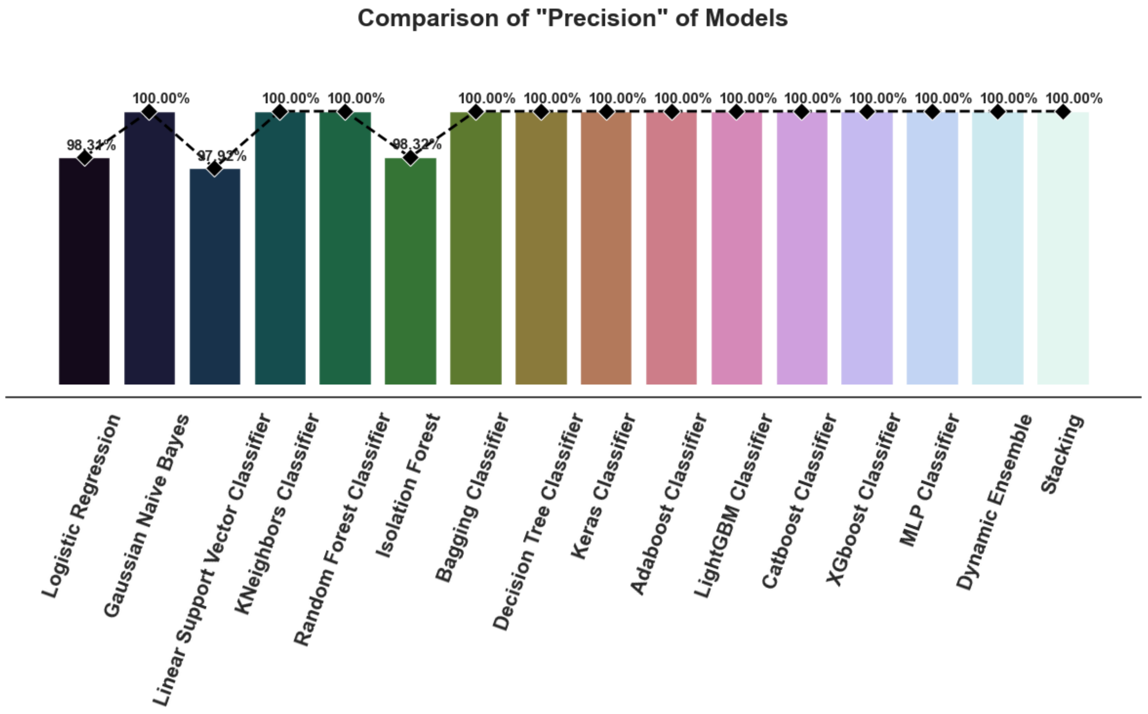

Figure 4.52
Comparison of "Precision" of Models

Precision assesses the precision of the model by focusing on the accuracy of positive predictions. It measures what percentage of all positive predictions are correct. This metric is particularly important in evaluating the effectiveness of the model in minimizing false positives. In the context of Credit Card Fraud detection research, precision measures how accurately the model identifies fraudulent transactions.

The precision scores of the models vary between 100.00% and 97.92%. Notably, the Gaussian Naïve Bayes, KNeighbors Classifier, Random Forest Classifier, Bagging Classifier, Decision Tree Classifier, Keras Classifier, Adaboost Classifier, LightGBM Classifier, Catboost Classifier, XGBoost Classifier, MLP Classifier, Dynamic Ensemble, and Stacking all achieved a perfect precision score of 100.00%. However, Linear Support

Vector Classifier exhibited the lowest precision score at 97.92%. It's worth mentioning that precision is the only evaluation metric where models achieved a perfect score of 100%.

4.4.3 RECALL Comparison

Model Name	**Recall**
Logistic Regression	94.72%
Gaussian Naïve Bayes	92.68%
Linear Support Vector Classifier	95.53%
Kneighbors Classifier	92.68%
Random Forest Classifier	99.19%
Isolation Forest	95.12%
Bagging Classifier	97.97%
Decision Tree Classifier	98.37%
Keras Classifier	91.06%
Adaboost Classifier	98.78%
LightGBM Classifier	97.56%
Catboost Classifier	93.09%
XGboost Classifier	97.97%
MLP Classifier	93.90%
Dynamic Ensemble	98.78%
Stacking	99.19%

Table 4.20
Comparison of "Recall" of Models

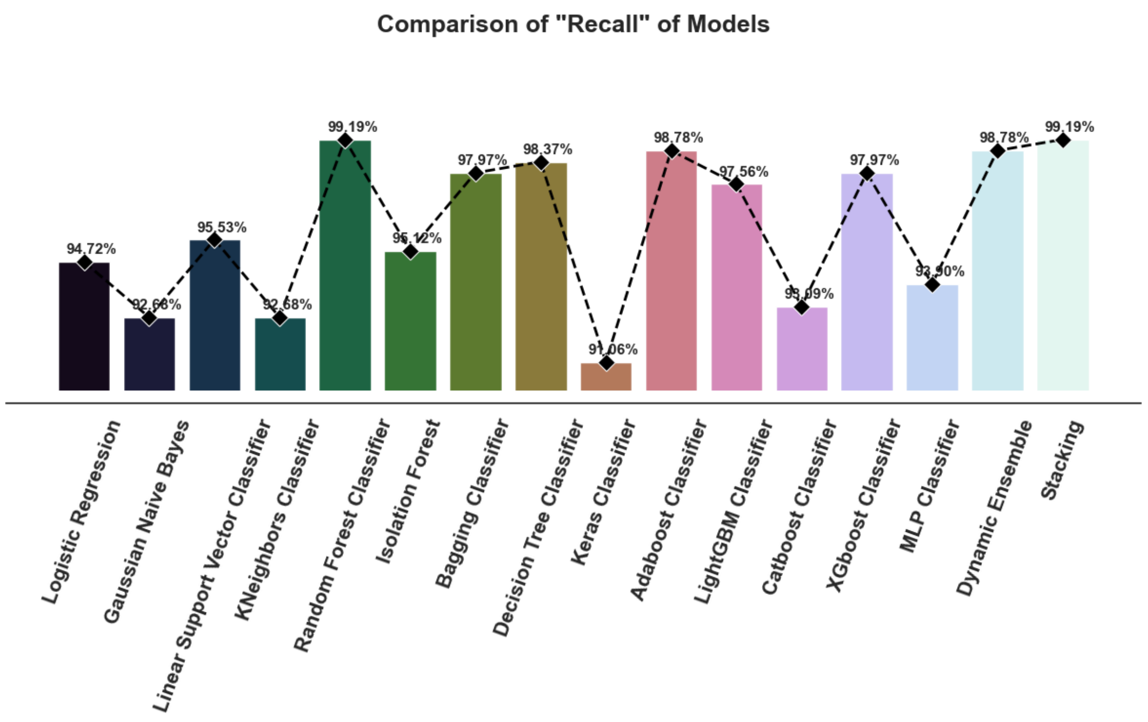

Figure 4.53
Comparison of "Precision" of Models

Recall evaluates the sensitivity of the model by assessing its effectiveness in predicting fraudulent transactions. It measures the proportion of all positive cases that are correctly predicted. This metric is particularly valuable in understanding the model's ability to minimize false negatives. In the context of Credit Card Fraud detection research, recall quantifies how many fraudulent transactions the model failed to classify accurately.

Recall, the most crucial metric for this research, measures how effectively models identify fraudulent transactions. Scores range from 91.06% to 99.19%. The Random

Forest Classifier and Stacking top the list with a recall score of 99.19%. In contrast, the Keras Classifier scored the lowest at 91.06%. The Gaussian Naïve Bayes and KNeighbors Classifier closely trailed behind at 92.68%. Notably, recall is the only metric where some models fell below the 96% mark.

4.4.4 AUC Comparison

AUC (Area under the ROC curve) is a crucial metric linked to the ROC (Receiver Operating Characteristic) curve, a standard tool in evaluating the predictive performance of Machine Learning models. The ROC curve visually illustrates how a model's classification threshold affects its performance. It comprises two key aspects: True-Positive Rate (TPR) and False-Positive Rate (FPR). TPR signifies the probability of correctly identifying fraudulent transactions, while FPR indicates the likelihood of misclassifying genuine transactions as fraudulent.

The ROC curve plots the TPR against the FPR across various classification thresholds. An ideal classifier's ROC curve aligns perfectly along the upper-left corner of the chart, reflecting consistently high TPR (equal to 1) regardless of the FPR value. This scenario signifies optimal performance, where the model effectively identifies fraudulent transactions while minimizing misclassifications of genuine ones.

Below are the ROC curves for the Machine Learning models assessed in this study:

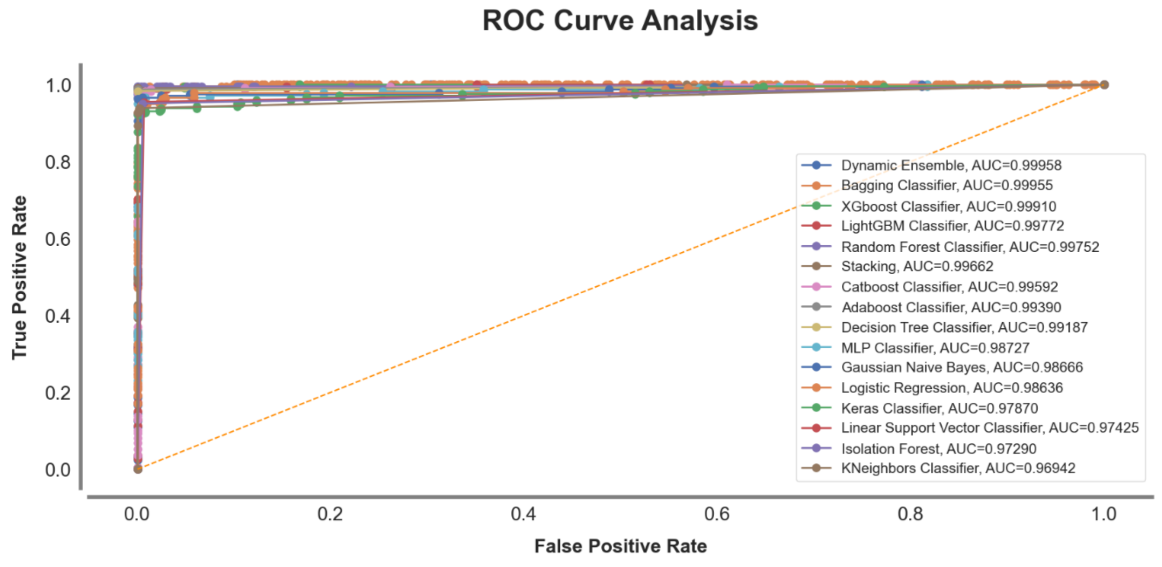

Figure 4.54
Receiver Operating Characteristic (ROC) Curve

Given that the ROC curves of the evaluated models predominantly cluster towards the upper left corner of the plot, the image below presents an alternative version of the chart, focusing on a zoomed-in view of the upper left corner.

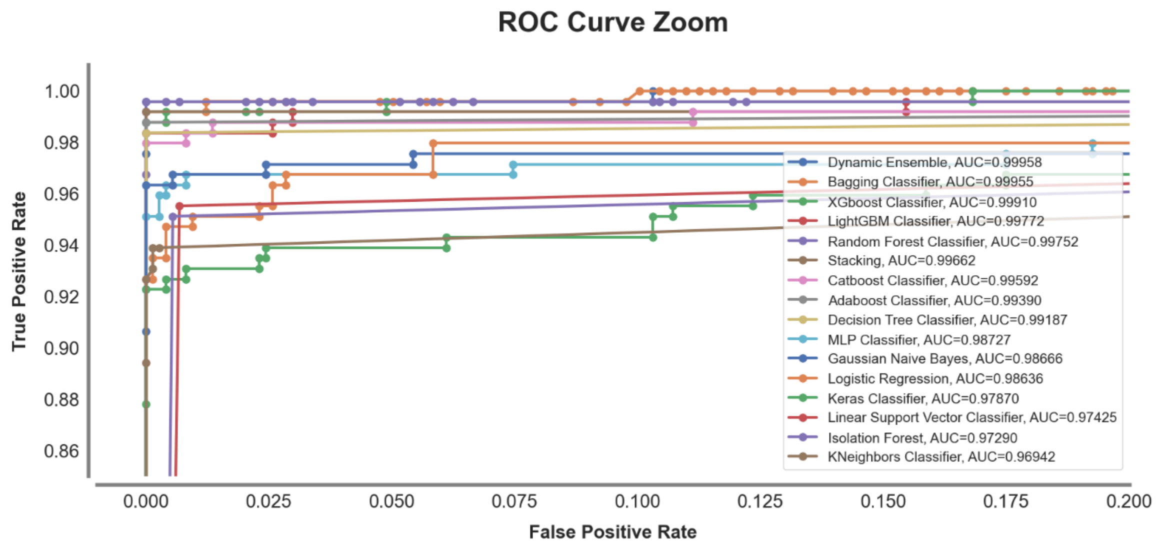

Figure 4.55
Receiver Operating Characteristic (ROC) Curve – Upper Left Corner Zoomed

AUC, short for Area Under the ROC Curve, serves as a comprehensive measure of a model's performance. It ranges between 0 and 1, with higher values indicating better performance. When the AUC is higher, it signifies that the model achieves higher True-Positive Rates and lower False-Positive Rates across various classification thresholds.

AUC of the evaluated models are given below:

Model Name	**AUC**
Logistic Regression	98.64%
Gaussian Naïve Bayes	98.67%
Linear Support Vector Classifier	97.43%
Kneighbors Classifier	96.94%
Random Forest Classifier	99.75%
Isolation Forest	97.29%
Bagging Classifier	99.95%
Decision Tree Classifier	99.19%
Keras Classifier	97.87%
Adaboost Classifier	99.39%
LightGBM Classifier	99.77%
Catboost Classifier	99.59%
XGboost Classifier	99.91%
MLP Classifier	98.73%
Dynamic Ensemble	99.96%

Stacking	99.66%

Table 4.21
Comparison of "AUC" of Models

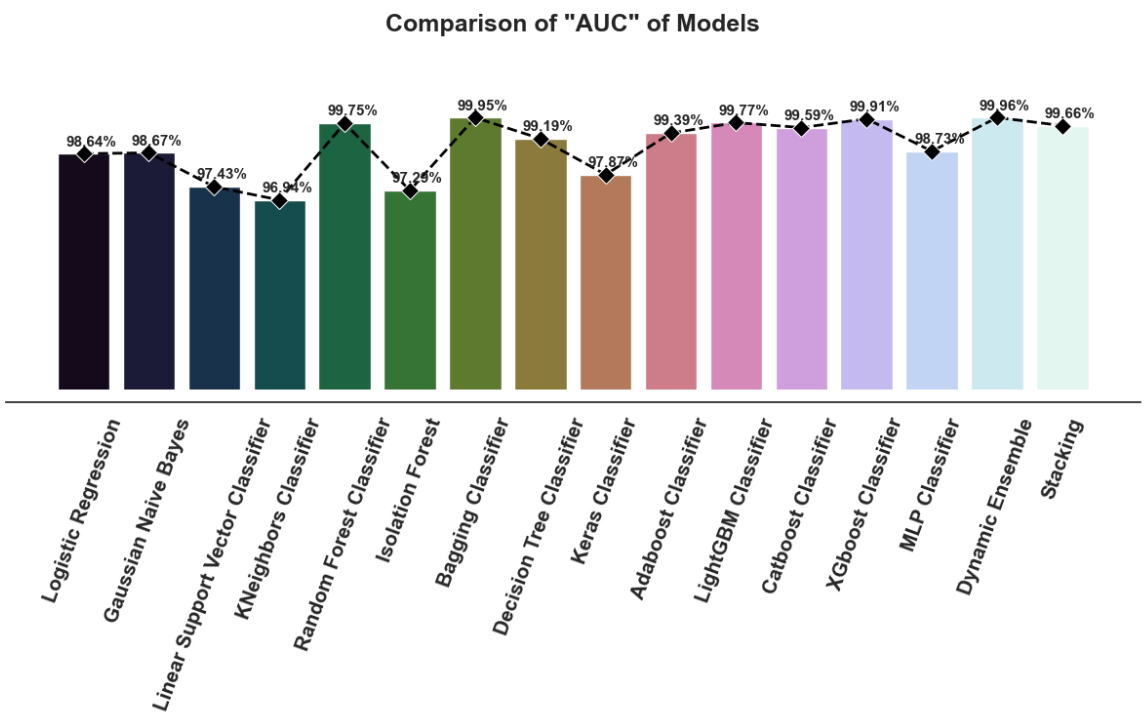

Figure 4.56
Comparison of "AUC" of Models

The AUC values for all evaluated models ranged from 96.94% to 99.96%. Dynamic Ensemble (99.96%) achieved the highest scores, while KNeighbors Classifier (96.94%) scored the lowest. Additionally, Random Forest Classifier (99.75%), Bagging Classifier (99.95%), Decision Tree Classifier (99.19%), Adaboost Classifier (99.39%), LightGBM Classifier (99.77%), CatBoost Classifier (99.59%), XGBoost Classifier (99.91%), and Stacking (99.66%) achieved AUC scores above 99%.

4.4.5 MODEL SCORE Comparison

Model Name	Model Score
Logistic Regression	97.11%
Gaussian Naïve Bayes	96.81%
Linear Support Vector Classifier	97.07%
Kneighbors Classifier	96.46%
Random Forest Classifier	99.63%
Isolation Forest	97.01%
Bagging Classifier	99.18%
Decision Tree Classifier	99.19%
Keras Classifier	96.00%
Adaboost Classifier	99.39%
LightGBM Classifier	98.98%
Catboost Classifier	97.15%
XGboost Classifier	99.17%
MLP Classifier	97.31%
Dynamic Ensemble	99.50%
Stacking	99.61%

Table 4.22
Comparison of "Model Score" of Models

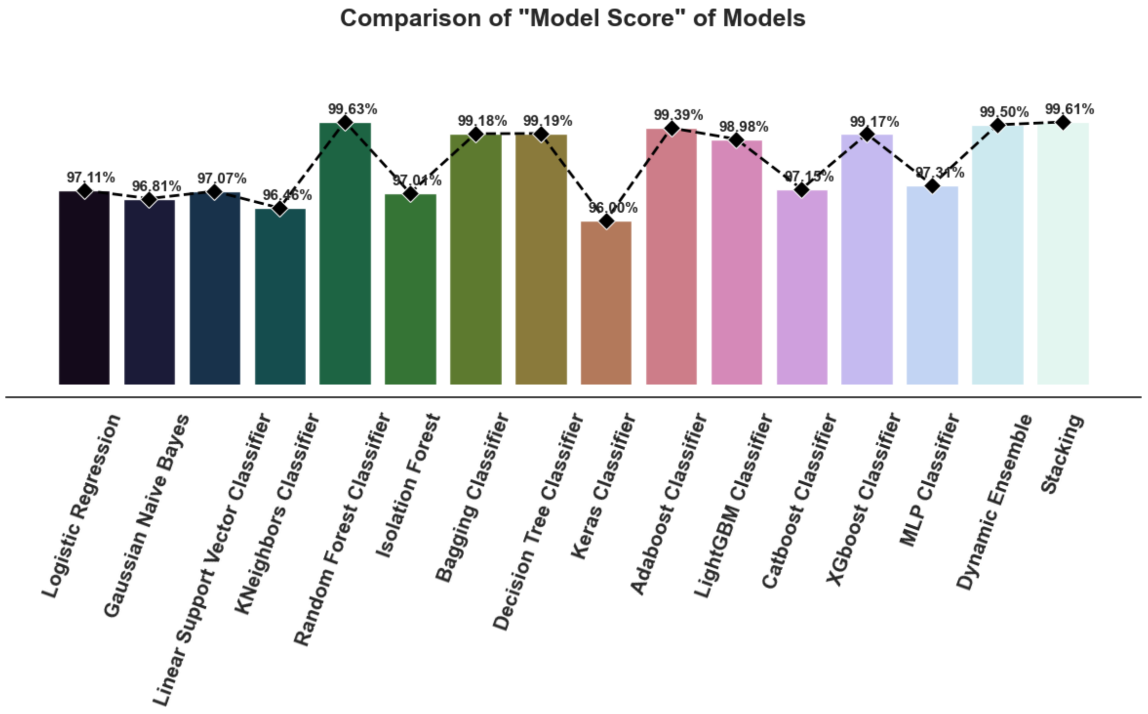

Figure 4.57
Comparison of "Model Score" of Models

Model Score is a comprehensive evaluation metric that combines the weighted averages of the "Accuracy," "Precision," "Recall," and "AUC" scores of the models. It provides a holistic assessment of the model's overall performance. The formula used to calculate the Model Score is as follows:

MODEL SCORE = 0.35 * RECALL + 0.25 * PRECISION + 0.2 * ACCURACY + 0.2 * AUC

So, "Model Score" gives 35%, 25%, 20% and 20% weightage to "Recall", "Precision", "Accuracy" and "AUC" respectively.

CHAPTER V:
DISCUSSION

5.1 Analysis of Models' Performance

In this study, 16 Machine Learning models were analyzed and compared based on several performance metrics such as "Accuracy", "Precision", "Recall" and "AUC". Subsequently, a derived evaluation criterion was developed to rank the performance of all models for the Credit Card Fraud detection task using the provided dataset. This metric, termed "Model Score," was calculated using the formula provided below.

MODEL SCORE = 0.35 * RECALL + 0.25 * PRECISION + 0.2 * ACCURACY + 0.2 * AUC

Before running the models, thorough preparation of the data was carried out. This included tasks like splitting the data into training and testing sets and addressing any imbalances in the data, as discussed in the paper. To optimize the performance of each model and minimize bias, we conducted exhaustive hyperparameter tuning using Grid Search. Eventually, each model was trained using the best combination of parameters found through Grid Search. These meticulous steps in data preparation and parameter tuning ensured that all models performed to their fullest potential.

The performance evaluation revealed that the majority of the evaluated models achieved scores above 95% across all performance metrics. However, a few models fell short of the 96% mark in the "Recall" metric.

Random Forest Classifier emerged as a standout performer, closely trailed by Stacking. Both models exhibited remarkable performance across key metrics such as “Accuracy”, “Precision”, and “Recall” with Random Forest Classifier holding a slight edge over Stacking in terms of “AUC”. However, when it comes to “AUC”, Dynamic Ensemble took the lead, securing the highest score among all evaluated models.

In terms of “Accuracy” and “Recall” Keras Classifier achieved the lowest scores among all models. Linear Support Vector Classifier obtained the lowest score for “Precision” while KNeighbors Classifier performed the least in terms of “AUC”.

Interestingly, thirteen out of the sixteen models evaluated in this study attained a perfect score of 100% for “Precision”.

Among the sixteen models evaluated, eight achieved impressive scores of over 99% for “Accuracy” and nine models scored above 99% for “AUC”. However, only two models—the Random Forest Classifier and Stacking—managed to surpass the 99% threshold for “Recall”.

All evaluated sixteen models scored above 96% for “Model Score”, seven scored above 99%.

From the test outcomes outlined in this paper, it's evident that Random Forest Classifier excelled as the top performer among all sixteen models, achieving the highest “Model Score” of 99.63%. Stacking closely followed with a score of 99.61%, while Dynamic Ensemble secured the third position with a score of 99.50%. Conversely, the least effective model was the Keras Classifier, with a “Model Score” of 96.00%.

In general, Ensemble models or Boosted models outperformed individual Machine Learning models. These ensemble approaches, which combine the predictions of multiple models, showed enhanced predictive capabilities. Moreover, all evaluated models achieved higher scores for “Precision” compared to “Recall”. This indicates that these models were more successful in correctly identifying positive cases while minimizing false positives.

5.2 Limitations of the Study

Here are the limitations encountered throughout this study:

Limited access to real data: Credit card transaction data is laden with sensitive information regarding individuals' financial behaviors. However, accessing this data directly for research purposes is restricted due to stringent privacy regulations and ethical considerations. Financial institutions prioritize safeguarding customer privacy and data security above all else. Consequently, there's a significant scarcity of publicly available datasets concerning credit card fraud, compelling researchers to rely on anonymized or synthetic datasets.

Anonymization entails the removal of personal identifiers such as names and account numbers from consumer data, while retaining transaction patterns intact. Alternatively, synthetic data involves the creation of simulated datasets that mimic real-world transactions. This approach allows researchers to train Machine Learning models without jeopardizing consumer privacy.

Despite these efforts, both anonymized and synthetic datasets may fall short in capturing the intricate nuances of actual fraud. Consequently, the effectiveness of trained models in real-world applications could be compromised.

Feature selection: In the realm of Credit Card fraud studies, feature selection entails the meticulous process of sifting through the available transaction dataset to identify the most relevant features. It's crucial to recognize that not all features contribute equally to fraud detection; some may introduce noise or prove irrelevant. Effective feature selection is paramount as it can significantly enhance model performance and mitigate the risk of overfitting.

However, Credit Card fraud datasets tend to be complex and high-dimensional, making the task of feature selection challenging. The presence of irrelevant or redundant features can obscure patterns and hinder model accuracy. Therefore, it's essential to pinpoint the most meaningful features to ensure robust fraud detection.

Complicating matters further, privacy concerns dictate that most publicly available Credit Card fraud datasets are either anonymized or synthetic. This means that all real features are either concealed or modified to preserve consumer privacy. As a result, this constraint imposes limitations on proper feature selection during model training, ultimately impacting both the effectiveness and interpretability of the models.

Evolving fraud profiles: Credit card transaction profiles are not static; they undergo constant change over time. What might have been considered a legitimate transaction in the past could now be flagged as fraudulent. As time progresses, fraudsters adapt their strategies, often imitating legitimate behaviors to evade detection. In this rapidly evolving landscape, static models fall short in keeping up with the dynamic nature of fraud.

To address this challenge, adaptive models are essential. These models have the capability to continuously learn and adjust based on new data inputs. They are crucial in ensuring that our fraud detection systems remain effective over time. By continuously ingesting real-time data, adaptive models can adapt their weights, recalibrate thresholds, and remain vigilant in identifying anomalies even as the underlying patterns change.

As technology advances, so do the tactics employed by fraudsters. Therefore, it's imperative that our fraud detection models evolve in tandem. Only by embracing adaptive approaches can we effectively combat the ever-evolving landscape of fraud.

Benchmarking challenges: Acquiring credit card fraud datasets poses a considerable challenge, as they are often closely guarded by financial institutions. The availability of publicly accessible datasets containing authentic credit card transactions is extremely limited, making it arduous for researchers to obtain the necessary data for testing and refining their models. Despite ongoing efforts, access to such datasets remains restricted, hindering progress in fraud detection research.

Adding to the complexity, even when researchers within financial organizations publish their findings, they tend to do so selectively, disclosing only partial insights into their methodologies. Consequently, the comprehensive details essential for meaningful research often remain elusive to the broader research community. This lack of transparency stems from concerns surrounding privacy and regulatory constraints, compelling researchers to navigate a delicate balance between sharing insights and protecting sensitive information.

The absence of a transparent benchmarking framework further compounds the issue. Without access to shared datasets and collaborative platforms, researchers face significant challenges in evaluating the effectiveness of their fraud detection models against real-world data. As a result, the advancement and refinement of fraud detection algorithms are impeded by the inability to establish standardized benchmarks and facilitate meaningful comparisons across different methodologies and approaches.

Infrastructure challenges: Machine learning algorithms, particularly deep learning models, rely heavily on computational resources for both training and inference stages. Having access to ample computational power is pivotal in the development and deployment of robust fraud detection systems. Furthermore, one of the primary factors that contributes to the effectiveness of machine learning models is the fine-tuning of hyperparameters. However, fine-tuning involves exploring a vast range of values for multiple hyperparameters simultaneously, a task that demands high-end computational capabilities to execute efficiently.

5.3 Enhancements pursued in this study

Credit card fraud is a major threat to financial organizations. Risk landscape is very vast, ranging from financial to reputational. Considering the huge impact of this menace on the society, many researchers have tried exploiting Machine Learning algorithms to build an effective model to detect fraudulent transactions from a set of valid Credit Card transactions. Researchers tried various types of Machine Learning algorithms with different data preparations steps. Multiple performance metrics were used to evaluate the effectiveness of models in detecting credit card fraud.

However, during study of many of the past research papers, few areas were identified which can be optimized further to increase the effectiveness of Machine Learning models in analyzing the transactions dataset and detecting fraudulent transactions from the dataset. This study attempted some of the potential improvements areas to enhance the quality of the outcome. These steps are detailed below:

1) Analysis and comparison of Machine Learning models from various categories explored in this study. This enabled a detailed study of the strengths and weaknesses of different categories of models. Previous researchers explored subset of the available types of models.

Classical ML algorithms - Logistic Regression, Gaussian Naïve Bayes, KNeighbors Classifier, Linear Support Vector Classifier, Decision Tree Classifier and Isolation Forest

Ensemble model - Random Forest Classifier, Bagging Classifier, Dynamic Ensemble and Stacking

Neural Network model - Keras Classifier and MLP Classifier

Boosting models – Adaboost Classifier, LightGBM Classifier, Catboost Classifier and XGboost Classifier

2) Multi-steps detailed outlier treatment. In general, outliers get treated by removing the concerned records to make data manageable. Majority of the papers studied where outlier treatment carried out, followed this method. However, removal of records has the potential to remove some meaningful data from the dataset which might reduce quality of the outcome. This challenge is more prominent in case of fraudulent records where volume is much less compared to genuine records. So, any reduction of data from fraudulent records should be well thought through and should be the last resort in outlier treatment. In this study, outlier identification and treatment carried out in multiple phases using Inter Quartile Range (IQR) method. Different IQR factors were considered for genuine and fraudulent records. This enabled less aggressive outlier identification approach for fraudulent records. With the help of multi-step identification approach and different IQR factors for genuine and fraudulent records, there was no data

deletion from the fraudulent records. This helped in preserving the vital information in the dataset, particularly in case of scarce fraudulent records.

3) Optimum class imbalance handling. Class imbalance can heavily impact the efficiency of Machine Learning models. Normally any transaction fraud related dataset will be acutely imbalance as the number of fraudulent records would be much less compared to genuine records. Generally, class imbalance get treated either by under sampling of majority class or over sampling of minority class. Both these approaches aim to bring balance between genuine records and fraudulent records in the dataset before we feed the data to Machine Learning models. While over sampling of minority class can introduce noise in the dataset, under sampling of majority class can lead to vital information loss from the dataset. Most of the research papers studied as part of the preparation of this research, opted for majority class under sampling to bring balance between genuine records and fraudulent records in the dataset. This approach can fit well in the case where we don’t have problem related to data availability. However, in case of Credit Card transactions data, each record is vital and losing any data can seriously affect the quality of the model training. So, in this study, a balance of over sampling of minority class (fraudulent records) and under sampling of majority class (genuine records) was maintained to take benefit from the best of both worlds.

4) Exhaustive hyperparameters tuning. For almost all the Machine Learning models (except very few like Gaussian Naïve Bayes), hyperparameters tuning is

mandatory to align the model's effectiveness with the problem statement. Each of the hyperparameters play a different role for different types of data. There cannot be a standard model for all business requirements. However, hyperparameters tuning is not always given the importance which it deserves which in turn doesn't allow the model to perform optimally for a given business requirement. Majority of the research carried out related to Credit Card fraud detection are either silent on hyperparameters tuning or this subject is dealt like a footnote. From the research papers, it was not visible that due importance was given to this step in the execution phase. In this study, most of the execution time was spent on hyperparameters tuning for all the evaluated Machine Learning models except for Gaussian Naïve Bayes. Each of the available hyperparameters were tuned individually and then the combinations of the hyperparameters were tuned to get the best performance metrics for all the models. All these tunings helped the evaluated Machine Learning models to achieve higher Recall value which was the main performance metrics targeted in this study.

5) Model evaluation basis a derivative metrics. In general, Machine Learning models get evaluated for their effectiveness using the standard performance metrics like Accuracy, Precision, Recall or AUC. However, not all these metrics will have same significance to assess models for all the business problems. Depending on the need of the task, importance of these metrics will change. In case of Credit Card fraud detection scenario, meaning of these metrics are given below:

Accuracy: depicts the capability of the model to detect genuine transactions as genuine (True Negative) and fraudulent transactions as fraudulent (True Positive)

Precision: denotes how many fraudulent transactions model predicted correctly

Recall: indicates how many fraudulent transactions model missed to classify correctly

Area Under the Curve (AUC): tells how much a particular model is capable of distinguishing between target classes

For a fraud detection model, successfully detecting a fraudulent transaction is much more important than wrongly classifying a genuine transaction as fraudulent. Performance related to detection of genuine transactions is of lesser priority in case of fraud detection.

In previous studies, many of the researchers considered Accuracy as the measure to evaluate the effectiveness of a Machine Learning model. Some researchers considered all standard metrics like Accuracy, Precision, Recall and AUC. However, considering detection of fraudulent transactions is most important in a fraud detection system, this study considered Recall as the most important metric for performance evaluation of Machine Learning models. Precision was considered as the next important metric. Accuracy and AUC were given equal importance after Recall and Precision.

So, instead of going with any of the standard available performance metrics to measure the effectiveness of the evaluated Machine Learning models, this study considered a weighted combination of “Accuracy”, “Precision”, “Recall” and “AUC” for models’ performance measurement. Weightage of 0.35, 0.25, 0.2 and 0.2 were given to “Recall”, “Precision”, “Accuracy” and “AUC” scores respectively and a derived metric called “Model Score” was calculated. This method of measuring the effectiveness of Machine Learning models for Credit Card fraud detection is more aligned with the actual requirements to find out a model which is good in detecting both genuine and fraudulent transactions with higher focus on fraudulent cases.

6) Visualization of results. Programming languages like Python provide default graphs and charts to visualize the results of analysis or model output. But the visualization of results always takes less priority than the actual analysis and execution. However, visualization of results should be given due importance to make the visuals more appealing and understandable to the end users. Visualization bridges the gap between raw data and actionable insights. Visuals make complex information digestible, engaging, and understandable for end users. In this study, leveraging Python’s visualization capabilities, each graph, chart, and data table was meticulously plotted to help decision making.

CHAPTER VI:
CONCLUSION

6.1 Summary

This paper delved deeply into evaluating the effectiveness of various Machine Learning Models in detecting Credit Card Fraud using a publicly available dataset. The dataset used in the paper is available through Kaggle. Almost all the features of the downloaded dataset were anonymized because of security reasons. Also, the dataset was highly imbalanced. There were 99.83% genuine records and only 0.17% fraudulent records in the available dataset. Imbalanced situation was handled using a combination of Over-sampling and Under-sampling techniques. For this, SMOTEENN function was used on the training dataset. Test dataset was consisting of half of the fraudulent transactions and thrice number of genuine transactions from the original dataset. Outliers were removed from the highly correlated variables and final train and test dataset were fed into sixteen Machine Learning Models. All these models were tuned for hyper-parameters. The final performance of these models was checked using “Accuracy”, “Precision”, “Recall” and “AUC” scores.

Based on the tests conducted in this paper, Random Forest Classifier was found to be the best performing model and Keras Classifer performed worst. For Random Forest Classifier, the scores of “Accuracy”, “Precision” and “Recall” were highest among all the models evaluated in this paper. Whereas Keras Classifier scored least for “Accuracy”, “Precision” and “Recall”.

This paper considered a weighted sum of “Accuracy”, “Precision”, “Recall” and “AUC” scores to rank the effectiveness of the sixteen models in detecting Credit Card fraud. Weightage given were 35%, 25%, 20% and 20% to “Recall”, “Precision”, “Accuracy” and “AUC” respectively. Considering this criteria, the top three performing models in the descending order are Random Forest Classifier, Stacking and Dynamic Ensemble.

Availability of limited dataset related to Credit Card fraud in the public domain and anonymity of the features in the available dataset constrained the research activities like feature selection, correlation between features etc. A more detailed dataset would surely increase the effectiveness of the Machine Learning models.

6.2 Recommendations for future research

Drawing from the insights gained in this study, several recommendations are provided below to guide future research endeavors.

1) In this study, Ensemble and Boosted Machine Learning models stood out for their strong performance across different metrics. However, the Neural Network-based models, specifically the Keras Classifier and MLP Classifier, didn't perform as well, with their "Model Score" values below 96%. To enhance future research, a more focused effort is recommended to fine-tune Neural Network-based classifiers for improved performance. This could involve exploring different architectures, optimizing hyperparameters, and

refining training techniques to maximize their effectiveness in fraud detection tasks.

2) For the next phase of research, it's recommended to utilize a more robust GPU-enabled server with greater RAM capacity for executing Machine Learning models. This upgrade will allow for the exploration of wider ranges of hyperparameters, enhancing the fine-tuning process. Furthermore, consider experimenting with resource-intensive models like Support Vector Machine (SVM) using oversampled datasets. This approach could offer valuable insights into SVM's effectiveness in fraud detection tasks and help optimize its performance.
3) The absence of high-quality real-time Credit Card fraud datasets publicly available presents a significant hurdle in developing versatile Machine Learning models. To overcome this challenge, future researchers could explore the possibility of obtaining more comprehensive and diverse datasets from financial institutions. By collaborating with these institutions and leveraging their data resources, researchers can access richer datasets that better reflect the complexities of real-world fraudulent activities in the banking sector. This approach will enable the development of more effective and adaptable Machine Learning models for detecting Credit Card fraud.
4) Researchers should delve into privacy-preserving techniques that play a crucial role in safeguarding the confidentiality of consumers' sensitive information. By implementing such techniques, Machine Learning models can

operate effectively on real-world credit card transaction data without jeopardizing the privacy of credit card users. This exploration could involve the development and adoption of advanced encryption methods, secure multiparty computation, or differential privacy frameworks. By prioritizing privacy preservation, researchers can ensure that their fraud detection models uphold ethical standards while effectively combating Credit Card fraud..

REFERENCES

1) Afriyie, J. K., Tawiah, K., Pels, W. A., Addai-Henne, S., Dwamena, H. A., Owiredu, E. O., Ayeh, S. A., and Eshun, J. (2023). A supervised machine learning algorithm for detecting and predicting fraud in credit card transactions. *Decision Analytics Journal*. Advance online publication. https://doi.org/10.1016/j.dajour.2023.100163

2) Alam, Md. N., Podder, P., Bharati, S., and Mondal, M. R. H. (2021). Effective Machine Learning Approaches for Credit Card Fraud Detection, *Institute of Information and Communication Technology*, Bangladesh University of Engineering and Technology, Dhaka 1205, Bangladesh. https://doi.org/10.1007/978-3-030-73603-3_14

3) Alarfaj, F. K., Malik, I., Khan, H. U., Almusallam, N., Ramzan, M., and Ahmed, M. (2022) Credit Card Fraud Detection Using State-of-the-Art Machine Learning and Deep Learning Algorithms, *IEEE Access*, 10, pp. 39700-39715. doi: 10.1109/ACCESS.2022.3166891

4) Asrar, Shuja. "Debit, Credit Card Frauds on Rise, ATM Scams down: NCRB." *Mint*, 31 Aug. 2022, www.livemint.com/industry/banking/debit-credit-card-frauds-on-rise-atm-scams-down-ncrb-11661885877307.html.

5) Azhan, M., and Meraj, S. 2020, Credit Card Fraud Detection using Machine Learning and Deep Learning Techniques, in *Third International Conference on Intelligent Sustainable Systems [ICISS 2020]*

6) Babu, M. G., Kshirsagarhi, P., Mamatha, B., & Chippalkatti, P. (2020). A Machine Learning Approach for Credit Card Fraud Detection. *The Mattingley Publishing Co., Inc., 82*, 5237-5244.

7) Bagga, S., Goyal, A., Goyal, N., & Gupta, A. (2020). Credit Card Fraud Detection using Pipeling and Ensemble Learning. *International Conference on Smart Sustainable Intelligent Computing and Applications* ICITETM2020 (pp. 104–112). Procedia Computer Science, 173.

8) Bahnsen, A. C., Aouada, D., Stojanovic, A., and Ottersten, B. (2015) Detecting Credit Card Fraud Using Periodic Features, *2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA)*, pp. 208–213. https://doi.org/10.1109/ICMLA.2015.28

9) Best, Raynor de. “Card Fraud Value Worldwide.” *Statista*, 1 Feb. 2024, www.statista.com/statistics/1394119/global-card-fraud-losses/#:~:text=Card%20fraud%20losses%20across%20the%20world%20increased%20by,U.S.%20dollar%20coming%20from%20the%20United%20States%20alone.

10) Bhanusri, Andhavarapu, et al. Credit Card Fraud Detection Using Machine Learning Algorithms. *Quest Journals*, 2020, www.questjournals.org/jrhss/papers/vol8-issue2/B08020411.pdf.

11) Bodepudi, H. (2021). Credit Card Fraud Detection Using Unsupervised Machine Learning Algorithms. *International Journal of Computer Trends and Technology*, 69(8), 1-3. doi:10.14445/22312803/IJCTT-V69I8P101

12) C., Dr. V. P. (2020). Analysis of Performance on Classification Algorithms for Credit Card Fraud Detection. *Journal of Advanced Research in Dynamical and*

Control Systems, 12(SP3), 1403–1409. https://doi.org/10.5373/JARDCS/V12SP3/20201391

13) *Consumer Sentinel Network Data Book 2021*, www.ftc.gov/system/files/ftc_gov/pdf/CSN Annual Data Book 2021 Final PDF.pdf. Accessed 20 Oct. 2023.

14) "Credit Card Fraud Soars to 10-Year High." *Experian Plc*, www.experianplc.com/newsroom/press-releases/2023/credit-card-fraud-soars-to-10-year-high. Accessed 20 Oct. 2023.

15) Degenhard, J. "India: Number of Credit Cards in Use 2014-2029." *Statista*, 30 Jan. 2024, www.statista.com/forecasts/1150220/credit-cards-in-use-forecast-in-india.

16) Dhankhad, S., Mohammed, E., and Far, B. (2018). Supervised Machine Learning Algorithms for Credit Card Fraudulent Transaction Detection: A Comparative Study. *2018 IEEE International Conference on Information Reuse and Integration (IRI), 122–125.* https://doi.org/10.1109/IRI.2018.00025

17) Dighe, D., Patil, S., and Kokate, S. (2018). Detection of Credit Card Fraud Transactions Using Machine Learning Algorithms and Neural Networks: A Comparative Study. *2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA),* 1–6. https://doi.org/10.1109/ICCUBEA.2018.8697799

18) Dornadula, V. N. and S. Geetha. (2019). Credit Card Fraud Detection using Machine Learning Algorithms. Procedia Computer Science. *INTERNATIONAL CONFERENCE ON RECENT TRENDS IN ADVANCED COMPUTING 2019*

19) Fang, Y., Zhang, Y., & Huang, C. (2019) Credit Card Fraud Detection Based on Machine Learning. *Computers, Materials & Continua*, 61(1), pp. 185-195. doi:10.32604/cmc.2019.06144

20) Gao, J., Zhou, Z., Ai, J., Xia, B., and Coggeshall, S. (2019). Predicting Credit Card Transaction Fraud Using Machine Learning Algorithms. *Journal of Intelligent Learning Systems and Applications, 2019, 11, 33-63.* https://doi.org/10.4236/jilsa.2019.113003

21) Gupta, P., Varshney, A., Khan, M. R., Ahmed, R., Shuaib, M., & Alam, S. (2023). Unbalanced Credit Card Fraud Detection Data: A Machine Learning-Oriented Comparative Study of Balancing Techniques. *International Conference on Machine Learning and Data Engineering*. Procedia Computer Science, 218, 2575–2584 https://doi.org/10.1016/j.procs.2023.01.231

22) Hafeez, Asefa. "5 Most Popular Credit Card Frauds in India 2024." *Kuvera*, 15 Apr. 2024, kuvera.in/blog/top-5-credit-card-frauds-in-india-2024-kuvera/.

23) Ileberi, E., Sun, Y. and Wang, Z. (2022). A machine learning based credit card fraud detection using the GA algorithm for feature selection. *Journal of Big Data,* 9(24), doi: 10.1186/s40537-022-00573-8.

24) Islam, M.A., Uddin, M.A., Aryal, S., & Stea, G. (2023). An ensemble learning approach for anomaly detection in credit card data with imbalanced and overlapped classes. *Journal of Information Security and Applications*, 78, 103618. https://doi.org/10.1016/j.jisa.2023.103618

25) Kaggle dataset for Credit Card Fraud Detection. Available at: https://www.kaggle.com/mlg-ulb/creditcardfraud

26) Kazemi, Z., and Zarrabi, H. (2017). Using deep networks for fraud detection in the credit card transactions. *2017 IEEE 4th International Conference on Knowledge-Based Engineering and Innovation (KBEI), Tehran, Iran*

27) Khan, Shahnawaz, et al. Developing a Credit Card Fraud Detection Model Using Machine Learning Approaches. *International Journal of Advanced Computer Science and Applications (IJACSA)*, The Science and Information (SAI) Organization Limited, 2022, thesai.org/Publications/ViewPaper?Volume=13&Issue=3&Code=IJACSA&SerialNo=50.

28) Khare, N., and Sait, S. Y. (2018). Credit Card Fraud Detection Using Machine Learning Models and Collating Machine Learning Models. *2018 International Journal of Pure and Applied Mathematics, Volume 118, No. 20, 825-838.*

29) Lilhore, U.K., & Patil, V. (2018). A Survey on Different Data Mining & Machine Learning Methods for Credit Card Fraud Detection. *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*, 3(5), ISSN : 2456-3307.

30) Lucas, Yvan, and Johannes Jurgovsky. Credit Card Fraud Detection Using Machine Learning: A Survey. *arXiv.Org*, 13 Oct. 2020, arxiv.org/abs/2010.06479v1.

31) Madhurya, M J, et al. Exploratory Analysis of Credit Card Fraud Detection Using Machine Learning Techniques. *Global Transitions Proceedings*, Elsevier, 6 Apr. 2022, www.sciencedirect.com/science/article/pii/S2666285X22000425.

32) Malik, E.F., Khaw, K.W., Belaton, B., Wong, W.P., & Chew, X.Y. (2022) Credit Card Fraud Detection Using a New Hybrid Machine Learning Architecture. *Mathematics*, 10, 1480. doi:10.3390/math10091480

33) Mijwil, M. M., & Salem, I. E. (2020). Credit Card Fraud Detection in Payment Using Machine Learning Classifiers. *Asian Journal of Computer and Information Systems,* 8(4), December 2020. ISSN: 2321 – 5658.

34) Mim, M. A., Majadi, N., and Mazumder, P. (2024). A soft voting ensemble learning approach for credit card fraud detection. *Heliyon*. Advance online publication. https://doi.org/10.1016/j.heliyon.2024.e25466

35) Mittal, S., and Tyagi, S. (2019). Performance Evaluation of Machine Learning Algorithms for Credit Card Fraud Detection. *2019 9th International Conference on Cloud Computing, Data Science & Engineering (Confluence), 320–324.* https://doi.org/10.1109/CONFLUENCE.2019.8776925

36) More, R. S., Awati, C. J., Shirgave, S. K., Deshmukh, R. J., & Patil, S. S. (2020). Credit Card Fraud Detection Using Supervised Learning Approach. *International Journal of Scientific & Technology Research*, 9(10). ISSN 2277-8616.

37) Mullen, Caitlin. "Card Industry Faces $400B in Fraud Losses over next Decade, Nilson Says." *Payments Dive*, 14 Dec. 2021, www.paymentsdive.com/news/card-industry-faces-400b-in-fraud-losses-over-next-decade-nilson-says/611521/.

38) Nadim, A.H., Chowdhury, M.S., Sayem, I.M., and Mutsuddy, A. (2019). Analysis of Machine Learning Techniques for Credit Card Fraud Detection. *Conference Paper · December 2019* DOI: 10.1109/iCMLDE49015.2019.00019

39) Naidu, J. S. 2020, "₹615.39cr Lost to Debit, Credit Card Frauds." *Hindustan Times*, 11 February 2020, https://www.hindustantimes.com/mumbai-news/615-39cr-lost-to-debit-credit-card-frauds/story-E335UM0fj1dVKJYZcJ2zRN.html.

40) Nguyen, Thanh Thi, et al. Deep Learning Methods for Credit Card Fraud Detection. *arXiv.Org*, 7 Dec. 2020, arxiv.org/abs/2012.03754.

41) Olowookerea, Toluwase Ayobami, and Adewale , Olumide Sunday (2020). A framework for detecting credit card fraud with cost-sensitive meta-learning ensemble approach. *Scientific African*, ELSEVIER. https://doi.org/10.1016/j.sciaf.2020.e00464

42) Osegi, E.N. and Jumbo, E.F. (2021) Comparative analysis of credit card fraud detection in Simulated Annealing trained Artificial Neural Network and Hierarchical Temporal Memory', *Machine Learning with Applications*, 6, p. 100080. Available at: https://doi.org/10.1016/j.mlwa.2021.100080

43) Parmar, J., Patel, A. C., and Savsani, M., (2020). Credit Card Fraud Detection Framework – A Machine Learning Perspective. *International Journal of Scientific Research in Science and Technology* https://doi.org/10.32628/IJSRST207671

44) Patidar, R., and Sharma, L. (2011). Credit Card Fraud Detection Using Neural Network. *International Journal of Soft Computing and Engineering (IJSCE) ISSN: 2231-2307, Volume-1, Issue-NCAI2011, June 2011*

45) Patil, S., Nemade, V., & Soni, P. (2018). Predictive Modelling For Credit Card Fraud Detection Using Data Analytics. *International Conference on Computational Intelligence and Data Science* (ICCIDS 2018). Procedia Computer Science, 132, 385–395.

46) Puh, M. and Brkić, L. (2019). Detecting Credit Card Fraud Using Selected Machine Learning Algorithms. *42nd International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*

47) Pumsirirat, A., and Yan, L. (2018). Credit Card Fraud Detection using Deep Learning based on Auto-Encoder and Restricted Boltzmann Machine. *International Journal of Advanced Computer Science and Applications, 9(1).* https://doi.org/10.14569/IJACSA.2018.090103

48) Rajora, S., Li, D.-L., Jha, C., Bharill, N., Patel, O. P., Joshi, S., Puthal, D., and Prasad, M. (2018). A Comparative Study of Machine Learning Techniques for Credit Card Fraud Detection Based on Time Variance. *2018 IEEE Symposium Series on Computational Intelligence (SSCI), 1958–1963.* https://doi.org/10.1109/SSCI.2018.8628930

49) RB, Asha, and Suresh Kumar KR. Credit Card Fraud Detection Using Artificial Neural Network. *Global Transitions Proceedings*, Elsevier, 23 Jan. 2021, www.sciencedirect.com/science/article/pii/S2666285X21000066.

50) Roy, Dibyendu (2020), 'Comparative Study of Machine Learning Algorithms to Detect Credit Card Fraud', Master's dissertation, Liverpool John Moores University.

51) Maniraj, S P, Saini, Aditya, Ahmed, Shadab, Sarkar, Swarna Deep, and SRM Institute of Science and Technology, INDIA. (2019). Credit Card Fraud Detection using Machine Learning and Data Science. *International Journal of Engineering Research And, 08(09), IJERTV8IS090031*. https://doi.org/10.17577/IJERTV8IS090031

52) S, Varun Kumar K, et al. Credit Card Fraud Detection Using Machine Learning Algorithms. *International Journal of Engineering Research & Technology*, IJERT-International Journal of Engineering Research & Technology, 5 Aug. 2020, www.ijert.org/credit-card-fraud-detection-using-machine-learning-algorithms.

53) S. V. S. S., Lakshmi, and Kavila, S. D. (2018). Machine Learning For Credit Card Fraud Detection System. *International Journal of Applied Engineering Research ISSN 0973-4562 Volume 13, Number 24 (2018) pp. 16819-16824*

54) Saloni, and Rout, M. (2021). Analysis and Comparison of Credit Card Fraud Detection Using Machine Learning. *Advances in Electronics, Communication and Computing (Vol. 709, pp. 33–40). Springer Singapore.* https://doi.org/10.1007/978-981-15-8752-8_4

55) Security.org team (2023) *2023 Credit Card Fraud Report.* Available at: www.security.org/digital-safety/credit-card-fraud-report/. (Accessed: 20 October 2023).

56) Sharma, Pratyush, et al. Machine Learning Model for Credit Card Fraud Detection- A Comparative Analysis. *The International Arab Journal of Information Technology*, Nov. 2021, www.iajit.org/portal/images/year2021/no6/19792.pdf.

57) Shirgave, S. K., Awati, C. J., More, R., & Patil, S. S. (2019). A Review On Credit Card Fraud Detection Using Machine Learning. *International Journal of Scientific & Technology Research, 8*(10), ISSN 2277-8616.

58) Shukur, Hamzah Ali, and Sefer Kurnaz. Credit Card Fraud Detection Using Machine Learning Methodology. *International Journal of Computer Science and Mobile Computing*, Mar. 2019, ijcsmc.com/docs/papers/March2019/V8I3201941.pdf.

59) Singh, G., Gupta, R., Rastogi, A., Chandel, M. D. S., and Riyaz, A. (2012). A Machine Learning Approach for Detection of Fraud based on SVM. *International Journal of Scientific Engineering and Technology*, 5.

60) Sulaiman, Rejwan Bin, et al. Review of Machine Learning Approach on Credit Card Fraud Detection. *SpringerLink*, Springer Netherlands, 5 May 2022, link.springer.com/article/10.1007/s44230-022-00004-0.

61) Suryanarayana, S Venkata, et al. Machine Learning Approaches for Credit Card Fraud Detection. *International Journal of Engineering & Technology*, 2018, doi.org/10.14419/ijet.v7i2.9356.

62) Thennakoon, A., Bhagyani, C., Premadasa, S., and Mihiranga, S., Kuruwitaarachchi, N. (2019). Real-time Credit Card Fraud Detection Using Machine Learning. *https://www.researchgate.net/publication/334761474*

63) Tiwari, Pooja, et al. Credit Card Fraud Detection Using Machine Learning: A Study. *arXiv.Org*, 23 Aug. 2021, arxiv.org/abs/2108.10005.

64) Trivedi, N.K., Simaiya, S., Lilhore, U.K., and Sharma, S.K. (2020). An Efficient Credit Card Fraud Detection Model Based on Machine Learning Methods. *International Journal of Advanced Science and Technology 29.*

65) Tuyls, Karl, Maes, Sam, and Vanschoenwinkel, B. (2015). Machine Learning Techniques for Fraud Detection. *https://www.researchgate.net/publication/254198382*

66) Udeze, C.L., Eteng, I.E. and Ibor, A.E. (2022). Application of Machine Learning and Resampling Techniques to Credit Card Fraud Detection. *Journal of the Nigerian Society of Physical Sciences*, 4, p. 769. doi: 10.46481/jnsps.2022.769.

67) Varmedja, D., Karanovic, M., Sladojevic, S., Arsenovic, M., and Anderla, A. (2019). Credit Card Fraud Detection—Machine Learning methods. *2019 18th International Symposium INFOTEH-JAHORINA (INFOTEH), 1–5.* https://doi.org/10.1109/INFOTEH.2019.8717766

68) Wang, C., Wang, Y., Ye, Z., Yan, L., Cai, W., and Pan, S. (2018). Credit Card Fraud Detection Based on Whale Algorithm Optimized BP Neural Network. *2018 13th International Conference on Computer Science & Education (ICCSE), 1–4*. https://doi.org/10.1109/ICCSE.2018.8468855

69) Warghade, S., Desai, S., & Patil, V. (2020) Credit Card Fraud Detection from Imbalanced Dataset Using Machine Learning Algorithm. *International Journal of Computer Trends and Technology (IJCTT)*, Volume 68, Issue 3, March 2020. doi: 10.14445/22312803/IJCTT-V68I3P105

70) Yee, Ong Shu, et al. Credit Card Fraud Detection Using Machine Learning as Data Mining Technique. *ResearchGate*, Aug. 2018, www.researchgate.net/publication/326986162_Credit_Card_Fraud_Detection_Using_Machine_Learning_As_Data_Mining_Technique.

71) Yu, W.-F., and Wang, N. (2009). Research on Credit Card Fraud Detection Model Based on Distance Sum. *2009 International Joint Conference on Artificial Intelligence, 353–356.* https://doi.org/10.1109/JCAI.2009.146

72) Zareapoor, M., and Shamsolmoali, P. (2015). Application of Credit Card Fraud Detection: Based on Bagging Ensemble Classifier. *Procedia Computer Science, 48, 679–685.* https://doi.org/10.1016/j.procs.2015.04.201