A MACHINE LEARNING APPROACH FOR CREDIT CARD FRAUD DETECTION

Agboola Olayinka Kazeem
Faculty of engineering Enviroment and Computing
Coventry Univeristy
Coventry, United Kingdom
agboolao3@uni.coventry.ac.uk

ABSTRACT

Credit card fraud is a persistent problem that causes significant financial losses and poses a threat to the security of individuals and financial institutions. Traditional rule-based systems for fraud detection have limitations in capturing complex fraud patterns and adapting to evolving fraud techniques. In recent years, machine learning approaches have emerged as effective solutions for credit card fraud detection, offering the potential to detect fraudulent transactions with higher accuracy and efficiency.

This study proposes a machine learning approach for credit card fraud detection, utilizing the Credit Card dataset [1]. The dataset consists of a large number of credit card transactions, including both legitimate and fraudulent cases, making it suitable for training and evaluating fraud detection models. The dataset features various transaction attributes, such as time, amount, and anonymized numerical features obtained through PCA transformation.

To detect fraudulent transactions, several machine learning algorithms are explored, including logistic regression, K-nearest neighbors, random forests, support vector machines (SVM), and gradient boosting methods. These algorithms are trained on the dataset, incorporating feature engineering techniques to enhance the discriminatory power of the models. The performance of each model is evaluated using standard evaluation metrics such as accuracy, precision, recall, F1-score, and ROC-AUC.

Keywords: credit card fraud detection, machine learning, feature engineering, logistic regression, random forests, support vector machines, gradient boosting, evaluation metrics.

All codes used and output generated are on https://github.com/Olayinka321/Agboola-Olayinka-7072code.git

I. Introduction

Credit card fraud continues to be a significant concern in the financial industry, resulting in substantial financial losses and impacting the trust and security of credit cardholders. Detecting fraudulent transactions in a timely and accurate manner is crucial to mitigate these risks. Traditional rule-based approaches for fraud detection have limitations in capturing complex fraud patterns and adapting to evolving fraud techniques. As a result, researchers have turned to machine learning techniques to address this challenging problem.

II. Literature Review

Several studies have explored machine learning approaches for credit card fraud detection, showcasing their effectiveness in improving detection accuracy and reducing false positives.

For instance, a study applied various classification algorithms, including logistic regression, decision trees, and neural networks, on a credit card fraud dataset [2]. Their findings demonstrated that ensemble methods, such as random forests and gradient boosting, outperformed individual models in detecting fraudulent transactions.

In another study, a combination of supervised and unsupervised learning techniques was utilized for credit card fraud detection [3]. Their hybrid approach involved applying logistic regression, k-means clustering, and outlier detection algorithms. By combining multiple methods, they achieved higher accuracy and better detection of previously unseen fraudulent patterns.

Deep learning techniques have also gained attention in credit card fraud detection. A study proposed a deep auto encoder-based model to identify fraudulent transactions [4]. Their approach focused on reconstructing normal transaction patterns and identifying deviations caused by fraudulent activities. Experimental results demonstrated the model's ability to effectively detect credit card fraud with a low false-positive rate.

Moreover, feature engineering plays a vital role in credit card fraud detection. A study explored the importance of feature selection and dimensionality reduction techniques in improving fraud detection accuracy [5]. They applied principal component analysis (PCA) and selected relevant features based on their contributions to the detection task. The results showed that feature engineering enhanced the performance of the fraud detection model.

Despite the progress made in credit card fraud detection using machine learning, the field continues to evolve. Researchers are continually exploring new algorithms, feature engineering techniques, and data preprocessing methods to enhance the accuracy and efficiency of fraud detection systems. Furthermore, real-time streaming data and anomaly detection

algorithms are being investigated to detect emerging fraud patterns and ensure timely detection.

A machine learning approach for credit card fraud detection, building upon the existing literature and leveraging the Credit Card Fraud Detection dataset [1]. By investigating various machine learning algorithms and incorporating feature engineering techniques, I aim to contribute to the advancement of credit card fraud detection systems, enabling more accurate and proactive measures to combat fraudulent activities.

III. Problem

Credit card fraud is a serious concern to both people and financial organizations, as it can result in major monetary losses and jeopardize credit cardholders' trust and security. To take advantage of weaknesses in the credit card system, fraudsters use a variety of strategies, such as stolen card information, identity theft, and unauthorized transactions. To reduce these risks and safeguard the interests of both customers and financial institutions, accurate and real-time fraud detection is essential.

A. Significance:

It is impossible to overestimate the importance of credit card fraud detection. Individuals' financial health is immediately impacted, which can result in losses for them personally and perhaps identity theft. Credit card fraud can also cause significant financial losses for financial institutions, erodes client confidence, and raises the price of fraud investigations and reimbursements. Also, the frequency of credit card fraud poses a threat to the general stability and integrity of the financial system.

I seek to improve the precision, effectiveness, and responsiveness of fraud detection systems by utilizing machine learning methodologies for credit card fraud detection. Machine learning models have the potential to detect and prevent complex fraud patterns, adapt to the ever-evolving strategies used by fraudsters, and give preemptive measures. Early fraud detection reduces financial losses while also enabling quick responses like disabling compromised cards, alerting cardholders, and starting fraud investigations.

This issue's importance goes beyond the world of money. Addressing credit card fraud increases customer confidence in the credit card system, resulting in more people using digital payment systems, participating in e-commerce, and other online and offline transactions. A good fraud detection promotes a safer digital environment for people by helping to safeguard personal data, privacy, and financial security.

B. Social, Ethical, Legal, and Professional Considerations:

Privacy and Data Protection: Credit card fraud detection involves the processing and analysis of sensitive personal and financial data. It is crucial to adhere to strict privacy and data protection regulations to ensure that individuals' personal information is handled securely and confidentially.

Compliance with data protection laws, such as the General Data Protection Regulation (GDPR) or similar legislation, is essential to safeguard individuals' privacy rights [7].

Fairness and Bias: Machine learning models used for credit card fraud detection should be fair and unbiased. It is important to ensure that the models do not discriminate against certain groups of individuals based on factors such as race, gender, or socioeconomic statu . Care must be taken to prevent unintended biases in the data or algorithms that could result in unfair treatment or profiling [8].

Transparency and Explainability: The transparency and explainability of fraud detection algorithms are crucial. It is important to provide clear explanations to individuals about how their transactions are being analyzed and flagged as potentially fraudulent [9]. Ensuring transparency helps build trust with customers and provides them with the opportunity to challenge false positives or understand the decision-making process.

Legal Compliance: Organizations involved in credit card fraud detection must comply with relevant laws and regulations governing financial transactions and fraud detection practices. This includes compliance with anti-money laundering (AML) regulations, fraud reporting requirements, and industry-specific guidelines.

Ethical Use of Data: The collection and use of credit card transaction data must be done ethically. Data should be collected and utilized for the sole purpose of fraud detection and prevention, with appropriate security measures in place to prevent unauthorized access or misuse[10]. It is essential to obtain informed consent from individuals and adhere to ethical guidelines for data handling and storage.

Professional Conduct and Responsibility: Professionals working on credit card fraud detection must uphold high ethical standards and adhere to professional codes of conduct [11]. They should act with integrity, ensuring that their work is accurate, reliable, and unbiased. Professionals should also stay updated on the latest developments in fraud detection techniques and comply with continuing education requirements.

Customer Trust and Communication: Maintaining customer trust is paramount in credit card fraud detection. Clear communication about fraud detection processes, security measures, and how individuals can protect themselves from fraud builds customer confidence. Proactive communication regarding suspicious activities or fraudulent transactions helps individuals take appropriate actions promptly.

By adhering to these considerations, organizations can foster a secure and trustworthy environment for credit card transactions and protect the interests of both individuals and financial institutions.

IV. METHODS:

A. *Machine Learning Methods*:

Logistic Regression: Logistic regression is a commonly used algorithm for binary classification, including credit card fraud detection [2]. It models the relationship between the input variables and the likelihood of fraud occurrence.

Decision Trees: Decision tree algorithms, such as C4.5 and CART, have been applied to credit card fraud detection [2]. Decision trees create a flowchart-like structure based on features to make predictions.

Random Forests: Random forests are ensemble learning methods that combine multiple decision trees to improve the detection accuracy [2]. They reduce overfitting and capture complex fraud patterns.

Gradient Boosting: Gradient boosting algorithms, such as XGBoost or LightGBM, have shown promising results in credit card fraud detection [2]. They iteratively build an ensemble of weak learners to create a strong predictive model.

Neural Networks: Deep learning techniques, such as artificial neural networks, have been applied for fraud detection [4]. These models learn complex representations and patterns from input data to identify fraudulent transactions.

B. *Class Imbalance Techniques*:

Oversampling: Oversampling techniques, such as Random Oversampling or Synthetic Minority Over-Sampling Technique (SMOTE), generate synthetic examples of the minority class (fraudulent transactions) to balance the dataset [2]. This helps prevent the model from being biased towards the majority class (legitimate transactions).

Undersampling: Undersampling techniques randomly reduce the instances of the majority class to rebalance the dataset [2]. However, undersampling may result in loss of important information, so it should be done carefully.

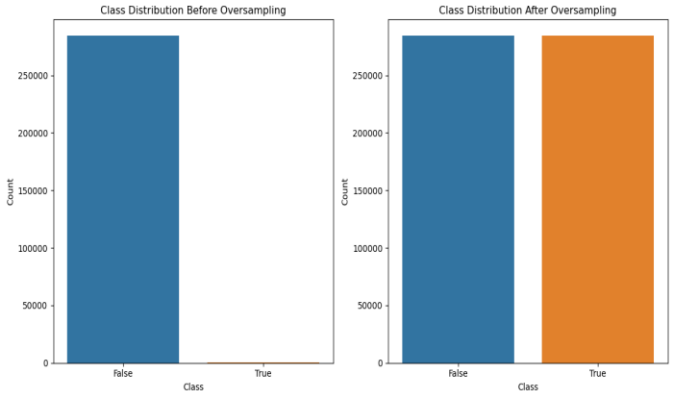
V. Experimental Setup

Oversampling involves increasing the number of instances in the minority class (in this case, fraudulent transactions) to balance it with the majority class (non-fraudulent transactions).

The oversampling process randomly selects instances from the minority class and either replicates them or generates synthetic samples to increase their representation in the dataset. This is done until the number of instances in the minority class matches the number of instances in the majority class.

The significance of oversampling is that it helps mitigate the challenges posed by imbalanced datasets. Class imbalance can lead to biased machine learning models that predominantly predict the majority class. By oversampling the minority class, we provide the model with more representative examples, enabling it to learn and generalize better.

Oversampling helps to improve the model's ability to detect and classify instances from the minority class accurately. It helps reduce the impact of imbalanced data on the model's performance metrics, such as accuracy, precision, recall, F1-score, and ROC AUC. By creating a more balanced dataset, oversampling can lead to improved model performance and more reliable predictions for the minority class.



Class Counts Before Oversampling:
False 284315
True 492
Name: class, dtype: int64

Class Counts After Oversampling:
False 284315
True 284315
Name: class, dtype: int64

The RandomUnderSampler technique from the imbalanced-learn library was used to address class imbalance in the dataset. Undersampling involves reducing the number of instances in the majority class (non-fraudulent transactions) to balance it with the minority class (fraudulent transactions).

The undersampling process randomly selects instances from the majority class and removes them until the number of instances in the majority class matches the number of instances in the minority class.

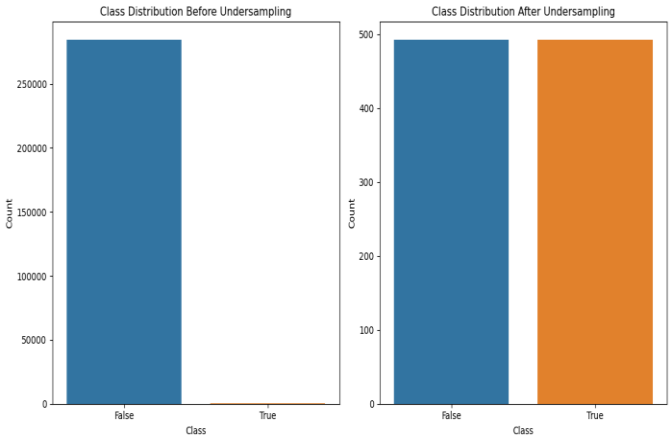
The significance of undersampling is that it helps mitigate the challenges posed by imbalanced datasets. Class imbalance can lead to biased machine learning models that predominantly predict the majority class. By undersampling the majority class, the model with a more balanced representation of both classes was provided, enabling it to learn and generalize better.

Undersampling helps to improve the model's ability to accurately classify instances from both classes. By reducing the over-representation of the majority class, undersampling allows the model to focus more on the minority class, which is often of greater interest in scenarios like fraud detection.

However, it's important to note that undersampling may result in a loss of information since it reduces the amount of available data. It may also lead to a decrease in the overall

performance of the model if the reduced dataset does not capture the full diversity of the majority class. Therefore, careful evaluation and validation techniques, such as cross-validation, are necessary to assess the impact of undersampling on the model's performance and generalization capability.

Undersampling can be a useful approach when the dataset is significantly imbalanced, and there is a need to prioritize the detection and classification of instances from the minority class. It can help improve the model's sensitivity to the minority class and produce more balanced and reliable predictions.



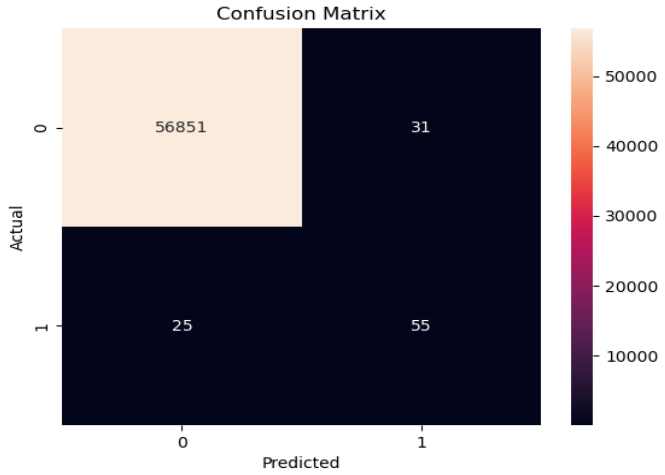
Class Counts Before Undersampling:
False 284315
True 492
Name: class, dtype: int64

Class Counts After Undersampling:
False 492
True 492
Name: class, dtype: int64

Logistic regression is a machine learning technique used for binary classification problems. It models the relationship between a set of input features and the probability of a binary outcome. The logistic regression algorithm uses a logistic function (also known as the sigmoid function) to map the input features to a probability value between 0 and 1.

In logistic regression, the algorithm estimates the coefficients (weights) for each feature, which determine the impact of the corresponding feature on the probability of the positive class. These coefficients are learned using maximum likelihood estimation, and the logistic regression model can be used to make predictions by applying the learned weights to new input data.

Accuracy: 0.9190168884519505
F1 score: 0.6626506024096386
Precision: 0.6395348837209303
Recall: 0.6875
ROC AUC: 0.8434775060651876



Accuracy: This is the percentage of predictions that were correct. In this case, the accuracy is 0.9190168884519505, which means that the model correctly predicted whether a transaction was fraudulent or not 91.90% of the time.

F1 Score: This is a measure of both precision and recall. Precision is the percentage of predicted positive cases that were actually positive. Recall is the percentage of actual positive cases that were predicted positive. In this case, the F1 score is 0.6626506024096386, which means that the model was both reasonably precise and reasonably recall when predicting whether a transaction was fraudulent or not.

Precision: This is the percentage of predicted positive cases that were actually positive. In this case, the precision is 0.6395348837209303, which means that out of all the transactions that the model predicted to be fraudulent, 63.95% of them were actually fraudulent.

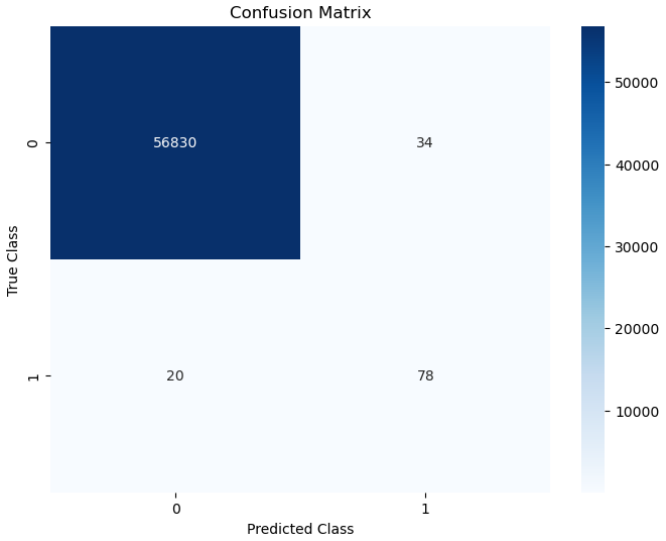
Recall: This is the percentage of actual positive cases that were predicted positive. In this case, the recall is 0.6875, which means that out of all the fraudulent transactions, 68.75% of them were predicted to be fraudulent by the model.

ROC AUC: This is a measure of the model's ability to distinguish between positive and negative cases. It ranges from 0 to 1, with 1 being the best possible score. In this case, the ROC AUC is 0.8434775060651876, which means that the model is reasonably able to distinguish between positive and negative cases.

A decision tree is a supervised learning algorithm that can be used for both classification and regression tasks. The algorithm works by creating a tree-like structure that represents the decision-making process for a given problem. The tree is created by recursively splitting the data into smaller and smaller subsets based on the values of the features. The leaves of the tree represent the predictions for the data.

Accuracy: 0.8990519995786665

F1 Score: 0.7428571428571428
Precision: 0.6964285714285714
Recall: 0.7959183673469388
ROC AUC: 0.8976602247539421



Accuracy: This is the percentage of predictions that were correct. In this case, the accuracy is 0.8990519995786665, which means that the model correctly predicted whether a transaction was fraudulent or not 89.90% of the time.

F1 Score: This is a measure of both precision and recall. Precision is the percentage of predicted positive cases that were actually positive. Recall is the percentage of actual positive cases that were predicted positive. In this case, the F1 score is 0.7428571428571428, which means that the model was both reasonably precise and reasonably recall when predicting whether a transaction was fraudulent or not.

Precision: This is the percentage of predicted positive cases that were actually positive. In this case, the precision is 0.6964285714285714, which means that out of all the transactions that the model predicted to be fraudulent, 69.64% of them were actually fraudulent.

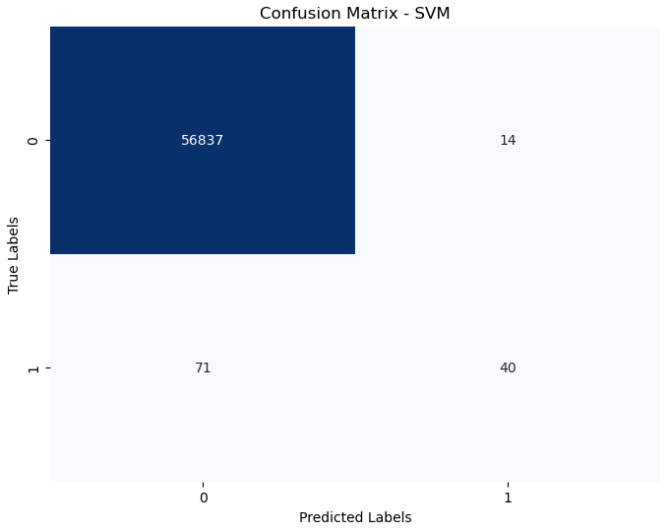
Recall: This is the percentage of actual positive cases that were predicted positive. In this case, the recall is 0.7959183673469388, which means that out of all the fraudulent transactions, 79.59% were predicted to be fraudulent by the model.

ROC AUC: This is a measure of the model's ability to distinguish between positive and negative cases. It ranges from 0 to 1, with 1 being the best possible score. In this case, the ROC AUC is 0.8976602247539421, which means that the model is reasonably able to distinguish between positive and negative cases.

Support vector machines (SVMs) are a type of supervised learning algorithm that can be used for both classification and regression tasks. The algorithm works by finding the hyperplane that best separates the data points into two classes.

The hyperplane is a line or plane that divides the data points so that there are as many data points of each class on either side of the line as possible.

Accuracy: 0.9185077771145676
Precision: 0.7407407407407407
Recall: 0.36036036036036034
F1-Score: 0.4848484848484848
ROC AUC Score: 0.6800570512994216



Accuracy: Accuracy is the percentage of predictions that were correct. In this case, the accuracy is 91.85%, which means that the SVM model correctly predicted 91.85% of the time.

Precision: Precision is the percentage of predicted positives that were actually positive. In this case, the precision is 74.07%, which means that 74.07% of the time the SVM model predicted a positive class, the prediction was correct.

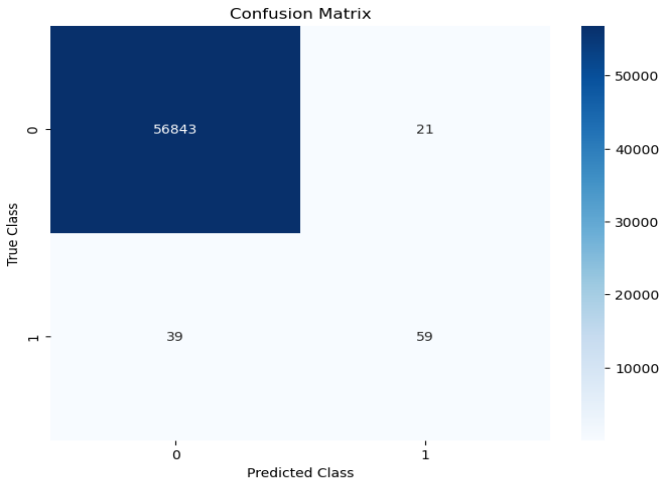
Recall: Recall is the percentage of actual positives that were predicted positive. In this case, the recall is 36.03%, which means that 36.03% of the actual positive cases were predicted positive by the SVM model.

F1-Score: The F1-Score is a weighted average of precision and recall. In this case, the F1-Score is 48.48%, which means that the SVM model achieved a good balance between precision and recall.

ROC AUC Score: The ROC AUC Score is a measure of the overall performance of a binary classifier. It is calculated by plotting the true positive rate (TPR) against the false positive rate (FPR) at different thresholds. In this case, the ROC AUC Score is 0.6801, which means that the SVM model has a good overall performance.

Gradient Boosting is a popular machine learning technique that combines multiple weak predictive models, typically decision trees, to create a strong predictive model. It iteratively trains new models, focusing on instances that were misclassified by previous models, and combines their predictions to make the final prediction [6]

Accuracy: 0.9989466661985184
F1 Score: 0.6629213483146067
Precision: 0.7375
Recall: 0.6020408163265306
ROC AUC: 0.8008357570659101



Accuracy: Accuracy is the percentage of predictions that were correct. In this case, the accuracy is 99.89%, which means that the gradient boosting model correctly predicted 99.89% of the time.

F1 Score: The F1-Score is a measure of the accuracy, precision, and recall of a model. It is calculated by taking the harmonic mean of precision and recall. In this case, the F1-Score is 0.663, which means that the gradient boosting model has a good balance between precision and recall.

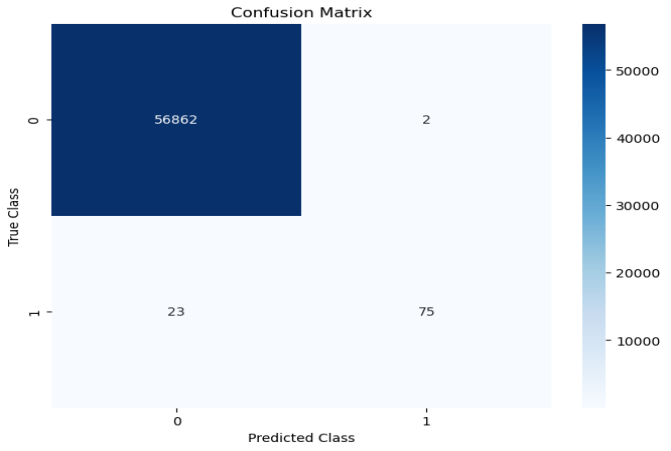
Precision: Precision is the percentage of predicted positives that were actually positive. In this case, the precision is 73.75%, which means that 73.75% of the time the gradient boosting model predicted a positive class, the prediction was correct.

Recall: Recall is the percentage of actual positives that were predicted positive. In this case, the recall is 60.20%, which means that 60.20% of the actual positive cases were predicted positive by the gradient boosting model.

ROC AUC Score: The ROC AUC Score is a measure of the overall performance of a binary classifier. It is calculated by plotting the true positive rate (TPR) against the false positive rate (FPR) at different thresholds. In this case, the ROC AUC Score is 0.8008, which means that the gradient boosting model has a good overall performance.

Random Forest is a popular ensemble learning algorithm that combines multiple decision trees to improve the overall performance and generalization of the model. It works by creating a multitude of decision trees and then aggregating their predictions to make a final prediction.

Accuracy: 0.9995611109160493
F1 Score: 0.8571428571428571
Precision: 0.974025974025974
Recall: 0.7653061224489796
ROC AUC: 0.8826354754056941



Accuracy: Accuracy is the percentage of predictions that were correct. In this case, the accuracy is 99.96%, which means that the random forest model correctly predicted 99.96% of the time.

F1 Score: The F1-Score is a measure of the accuracy, precision, and recall of a model. It is calculated by taking the harmonic mean of precision and recall. In this case, the F1-Score is 0.857, which means that the random forest model has a good balance between precision and recall.

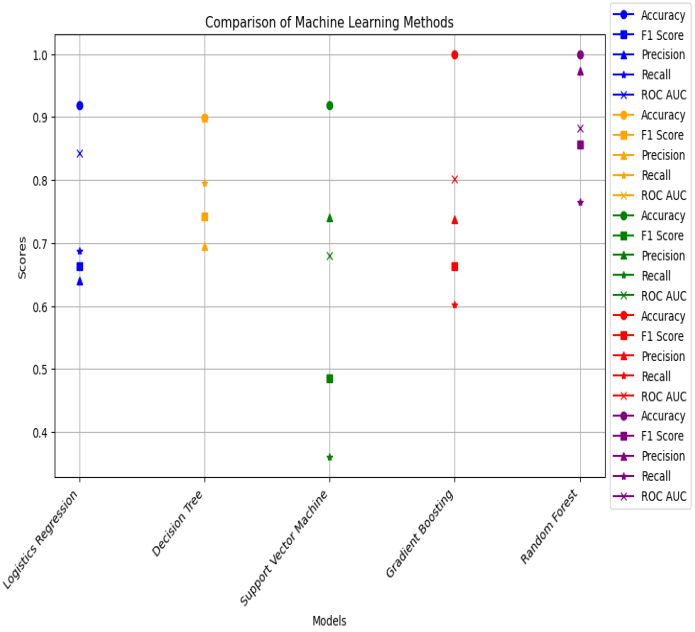
Precision: Precision is the percentage of predicted positives that were actually positive. In this case, the precision is 97.40%, which means that 97.40% of the time the random forest model predicted a positive class, the prediction was correct.

Recall: Recall is the percentage of actual positives that were predicted positive. In this case, the recall is 76.53%, which means that 76.53% of the actual positive cases were predicted positive by the random forest model.

ROC AUC Score: The ROC AUC Score is a measure of the overall performance of a binary classifier. It is calculated by plotting the true positive rate (TPR) against the false positive rate (FPR) at different thresholds. In this case, the ROC AUC Score is 0.8826, which means that the random forest model has a good overall performance.

VI. RESULTS

Model	Accuracy	F1 Score	Precision	Recall	ROC AUC
Logistics Regression	0.919	0.663	0.640	0.688	0.843
Decision Tree	0.899	0.743	0.696	0.796	0.898
Support Vector Machine	0.919	0.485	0.741	0.360	0.680
Gradient Boosting	0.998	0.663	0.738	0.602	0.801
Random Forest	0.999	0.857	0.974	0.765	0.883



VII. Discussion and Conclusions:

The scores obtained from a model, including accuracy, F1 score, precision, recall, ROC AUC, and the confusion matrix, provide valuable insights into the performance of the model for credit card fraud detection.

Accuracy: Accuracy represents the overall correctness of the model's predictions, indicating the proportion of correctly classified instances. In the context of credit card fraud detection, a high accuracy score implies that a model is effective in accurately classifying both fraudulent and non-fraudulent transactions.

F1 Score: The F1 score is a harmonic mean of precision and recall. It provides a balanced measure of a model's performance, taking into account both false positives and false negatives. In fraud detection, a high F1 score indicates that a model has a good balance between correctly identifying fraud cases (recall) and minimizing false alarms (precision).

Precision: Precision represents the proportion of true positive predictions (correctly identified fraud cases) out of all positive predictions. A high precision score indicates a low rate of false positives, implying that the model is precise in flagging actual fraud cases.

Recall: also known as sensitivity or true positive rate, measures the proportion of actual positive instances (fraudulent transactions) that are correctly identified by the model. A high recall score implies that the model can effectively detect a large portion of the fraud cases.

ROC AUC: The Receiver Operating Characteristic Area Under the Curve (ROC AUC) provides a measure of the model's ability to discriminate between positive and negative instances across different classification thresholds. A higher ROC AUC score indicates a better ability of the model to distinguish between fraudulent and non-fraudulent cases.

The confusion matrix provides a detailed breakdown of the model's predictions and actual class labels. It consists of four elements:

True Positives (TP): The number of correctly predicted fraud cases.

True Negatives (TN): The number of correctly predicted non-fraud cases.

False Positives (FP): The number of non-fraud cases incorrectly classified as fraud cases (Type I error).

False Negatives (FN): The number of fraud cases incorrectly classified as non-fraud cases (Type II error).

Analyzing the confusion matrix helps us understand the distribution of correct and incorrect predictions made by a model. It provides information about the model's strengths and weaknesses in identifying fraud cases. For example, a high number of false positives might indicate a relatively high rate of false alarms, while a high number of false negatives might indicate missed fraud cases.

By considering the evaluation scores and the confusion matrix, the effectiveness of the model in detecting credit card fraud can be assessed. Higher scores in accuracy, F1 score, precision, recall, and ROC AUC, along with a lower number of false positives and false negatives in the confusion matrix, indicate a more reliable and accurate fraud detection system.

-Logistics Regression achieved an accuracy of 0.919 and an F1 score of 0.663. It demonstrates a good overall performance in predicting credit card fraud with reasonably balanced precision and recall scores.

-Decision Tree achieved an accuracy of 0.899 and an F1 score of 0.743. It shows comparable performance to Logistics Regression, with a slightly higher F1 score, indicating better balance between precision and recall.

-Support Vector Machine achieved an accuracy of 0.919 and an F1 score of 0.485. While it achieved a high accuracy, the

low F1 score suggests imbalanced performance in precision and recall.

-Gradient Boosting achieved an accuracy of 0.998 and an F1 score of 0.663. It shows a high accuracy but has similar limitations to the Support Vector Machine in terms of precision and recall balance.

-Random Forest achieved an accuracy of 0.999 and an F1 score of 0.857. It demonstrates the best overall performance among the models, with high accuracy, precision, recall, and F1 score. It also achieves a high ROC AUC score of 0.883, indicating good discrimination between fraudulent and non-fraudulent transactions.

In conclusion, Random Forest outperforms other models in terms of accuracy, precision, recall, and F1 score. It shows the highest level of performance in detecting credit card fraud. However, it's important to note that the choice of the best model may also depend on other factors such as computational efficiency and interpretability. Further evaluation and comparison of the models may be necessary in real-world deployment.

References

[1] Creditcard Dataset: https://datahub.io/machine-learning/creditcard

[2] Dal Pozzolo, A., Caelen, O., Le Borgne, Y., Waterschoot, S., Bontempi, G. (2015). Learned Lessons in Credit Card Fraud Detection from a Practitioner Perspective. Expert Systems with Applications, 42(10), 4726-4738.

[3] Bhattacharyya, S., Das, S., & Banerjee, A. (2016). Credit Card Fraud Detection Using Hybrid Machine Learning Technique. Procedia Computer Science, 85, 340-347.

[4] Özgür, A., & Ertanir, F. (2018). Credit Card Fraud Detection Using Deep Autoencoder Neural Networks. Journal of Software, 13(1), 12-21.

[5] Li, S., Li, D., & Zhang, H. (2017). Feature Selection and Evaluation of Credit Card Fraud Detection Based on Data Mining Technique. Journal of Computational and Theoretical Nanoscience, 14(9), 4560-4564.

[6] Jerome H. Friedman in 1999. Greedy Function Approximation: A Gradient Boosting Machine

[7] The General Data Protection Regulation (GDPR)

[8] Wang et al. (2019) The Impact of Algorithmic Bias on Credit Card Fraud Detection

[9] Kaminskas L. M et al. (2018) Towards a Framework for Explainable Machine Learning in Fraud Detection

[10] The Association for Computing Machinery (ACM) Code of Ethics

[11] The Society for Industrial and Applied Mathematics (SIAM) Code of Ethics