**CREDIT CARD FRAUD DETECTION**

**USING MACHINE LEARNING**

Project submitted to the

SRM University – AP, Andhra Pradesh

for the partial fulfillment of the requirements to award the degree of

**Bachelor of Technology**

In

**Computer Science and Engineering**

**School of Engineering and Sciences**

Submitted by

**Bugginni Roshini -AP22110010141**

**Likhitha Sri Gandham -AP2211001447**

**Parimala Shivani Mandava -AP22110010431**

**A picture containing text

Description automatically generated**

Under the Guidance of

**Dr.Murali Krishna Enduri**

**SRM University–AP**

**Neerukonda, Mangalagiri, Guntur**

**Andhra Pradesh – 522 240**

**[November, 2024]**

# Certificate

Date: 15-Jan-25

This is to certify that the work present in this Project entitled “**Credit Card Fraud Detection using Machine Learning**” has been carried out by **Bugginni Roshini** under my supervision. The work is genuine, original, and suitable for submission to the SRM University – AP for the award of Bachelor of Technology in **Computer Science and Engineering**.

**Supervisor**

(Signature)

Prof. / Dr. Murali Krishna Enduri

Designation: Asst. Professor

Affiliation: SRM University-AP

**Co-supervisor**

(Signature)

Prof. / Dr. [Name]

Designation,

Affiliation.

# 

# Acknowledgments

We extend our heartfelt gratitude to everyone who has contributed to the successful completion of this project. This achievement is a testament to their unwavering dedication, hard work, patience, and guidance. We owe a special thanks to our mentor Dr.Murali Krishna Enduri, whose invaluable advice and support have been instrumental from the project's inception to its conclusion. His generosity in sharing his knowledge and insights has been a guiding light throughout our journey. We are also immensely grateful to our friends for their support, as well as anyone else who has contributed to this project. We recognize and appreciate the contributions of countless others—directly or indirectly—who played a role in shaping the outcome of this project. We are conscious of the fact that a project of this magnitude could not have been successful without the help of many individuals. We are extremely grateful to all those who have aided us in any way, whether it be directly or indirectly. We thank you all for your support. In closing, we would like to express our sincere appreciation to all those who have been a part of this project. Your contributions, no matter how big or small, have been invaluable. We are profoundly grateful for the support and encouragement we have received and feel humbled by the generosity and amount of support we have been given, and we thank all those involved for their invaluable contributions.

Thank you all for being an integral part of this accomplishment.

**Table of Contents**

Certificate…………………………………………………………………………………….3

Acknowledgements…………………………………………………………………………5

Abstract………………………………………………………………………………………8

Abbreviations……………………………………………………………………………….10

List of Tables ……………………………………………………………………………….12

List of Figures………………………………………………………………………………14

List of Equations…………………………………………………………………………...16

1. Introduction……………………………………………………………………………...19

1. 1.1. Project Goals……………………………………………………………………19
2. 1.2. Advantages of Machine Learning Models………………………………….20
3. 1.3 Data Source……………………………………………………………………...20

2. Related Work…………………………………………………………………………….23

3. Methodology……………………………………………………………………………..25

1. 3.1 Techniques and Packages……………………………………………………...25
2. 3.2 End-to-End Methodology……………………………………………………..28
3. 3.3 Challenges and Areas for Improvement in Credit Card Fraud Detection Using Machine Learning…………………………………………………………. 31

4. Result……………………………………………………………………………………...34

5. Conclusion and Future Work…………………………………………………………..40

6. References………………………………………………………………………………...42

**Abstract**

Credit card fraud poses a significant financial and security challenge worldwide, resulting in billions of dollars in losses annually. This project aims to address this critical issue by developing a machine learning-based model to detect fraudulent transactions effectively. The model leverages historical credit card transaction data to identify patterns indicative of fraud. By employing a diverse set of machine learning algorithms, including K-Nearest Neighbors (KNN), XGB Classifier, Logistic Regression, Random Forest, and Decision Trees, the project evaluates the performance of each approach in accurately distinguishing between fraudulent and legitimate transactions.

The primary objective is to enable early detection of fraudulent activities, ensuring that customers’ money is safeguarded and unauthorized charges are prevented. This will positively impact customers by ensuring their funds are recovered and their accounts remain secure. The machine learning models will be trained and tested on a credit card transaction dataset, and their effectiveness will be evaluated based on metrics such as precision, recall, accuracy, and confusion matrix. This project not only demonstrates the applicability of machine learning in fraud detection but also highlights the importance of advanced algorithms in mitigating financial losses and enhancing the security of digital transactions.

Index Terms: Credit Card Fraud Detection, Fraud Detection, Fraudulent Transactions, Machine Learning, XGB Classifier, K-Nearest Neighbors, Random Forest, Logistic Regression, Decision Tree.

# Abbreviations

# 1) SMOTE - Synthetic Minority Oversampling Technique

# 2) EDA - Exploratory Data Analysis

# 3) KNN - K-Nearest Neighbors

# 4) XG Boost - Extreme Gradient Boosting

# 5) RF - Random Forest

# 6) DT - Decision Tree

# 7) TP - True Positive

# 8) TN - True Negative

# 9) FP - False Positive

# 10) FN - False Negative

# 11) FPR - False Positive Rate

# 12) TPR - True Positive Rate

# List of Tables

Table 1. Performance metrics of the machine-learning techniques….…………35

# List of Figures

Figure 1: Heatmap showing correlation matrix of the dataset…………………………21

Figure 2: Data Flowchart of the project….………………………………………………..28

Figure 3: Class distribution( fraud vs non-fraud)…..………………………….……......29

Figure 4: Confusion Matrix for Logistic Regression…………………………………….36

Figure 5: Confusion Matrix for KNN……………………………………….…………….36

Figure 6: Confusion Matrix for Decision Tree……..…………………………………….37

Figure 7: Confusion Matrix for XG Boost…………………………………………………37

Figure 8: Confusion Matrix for Random Forest………………………………………….38

# List of Equations

Equation 1. Equation for Logistic Regression ………………………................................25

Equation 2. Equation for KNN (K-Nearest Neighbor)……..…...…...………………….26

Equation 3. Equation for Decision Tree………………………………...............................26

Equation 4. Equation for XG Boost…….………..………….……........…………………..27

Equation 5. Equation for Random Forest……………….………………………………..27

Equation 6. Equation for Accuracy………………………………………………………..34

Equation 7: Equation for Precision………………………………………………………..35

Equation 8: Equation for Recall……………………………………………………………35

Equation 9: Equation for F1-Score…………………………………………………………35

Equation 10: Equation for Specificity Score………………………………………………34

# 

# 1. Introduction

In today’s digital age, credit cards have become an essential part of everyday life, providing convenience and accessibility for millions of users worldwide. However, this growing reliance on credit cards has been accompanied by a significant increase in fraudulent activities, posing a serious threat to financial security and customer trust. According to statistics from 2021, the global number of credit card users reached 2.8 billion in 2019, with 70% of users owning at least one card. Reports of credit card fraud in the United States alone surged by 44.7% from 2019 to 2020. This alarming rise includes two primary forms of fraud: identity thieves opening new accounts under victims’ names, which saw a 48% increase, and fraudulent use of existing accounts through stolen information, which grew by 9% during the same period (Daly, 2021).

These statistics highlight the urgency for effective solutions to combat credit card fraud. The rising prevalence of such activities has motivated this project to address the issue analytically using advanced machine learning techniques. This project aims to detect fraudulent transactions with high accuracy and efficiency by leveraging historical credit card transaction data. The XG Boost algorithm has been found to provide a good estimate of the generalization error and to be resistant to overfitting.

**1.1. Project Goals**

The primary goal of this project is to identify fraudulent credit card transactions, ensuring that customers are not wrongly charged for products or services they did not purchase. The project will employ multiple machine learning algorithms, including K-Nearest Neighbors (KNN), Random Forest, XG Boost, Decision Tree, and Logistic Regression, to analyze transaction data and detect fraudulent patterns. A comparative evaluation of the models' performance will determine the most effective method for fraud detection. The outcomes will be presented with visualizations and statistical insights, providing a comprehensive understanding of the results.

These statistics emphasize the critical need for efficient strategies to tackle credit card fraud. The increasing frequency of such incidents has driven the development of this project, which seeks to analyze and address the issue using advanced machine learning methods. By utilizing historical credit card transaction data, this study focuses on identifying fraudulent activities with precision and effectiveness.

Additionally, this project explores previous literature and alternative techniques for fraud detection, contributing to a deeper understanding of the domain and improving the reliability of the proposed solutions. By addressing this critical issue, the project aims to enhance financial security, safeguard customer accounts, and restore trust in digital payment systems.

**1.2. Advantages of Machine Learning Models**Top of Form

Bottom of Form

* XG Boost selects the best feature rather than the most important feature among a random subset of data resulting in a better model.

Machine learning (ML) models surpass traditional fraud detection techniques in performance and efficiency, offering the ability to identify thousands of patterns within large datasets. ML provides valuable insights into customer activity by analyzing user behavior, such as app usage, payment habits, and transaction methods. Some key advantages of using ML for fraud detection include:

* Faster Detection

In real-time, machine learning algorithms can rapidly detect deviations from normal transaction patterns and user behaviors. By identifying anomalies, such as unexpected increases in transaction amounts or sudden location changes, ML minimizes the risk of fraudulent activities and ensures safer transactions.

* Greater Accuracy

Traditional fraud detection methods often produce errors at payment gateways, which can lead to blocking legitimate customers. ML models, when trained with sufficient data, achieve higher accuracy and precision, reducing false positives and minimizing the need for time-consuming manual reviews.

* Enhanced Efficiency with Larger Datasets

Once machine learning models recognize various transactional patterns and behaviors, they can handle large datasets effectively. These algorithms can process vast amounts of data in seconds, delivering real-time insights and enabling better decision-making for fraud prevention.

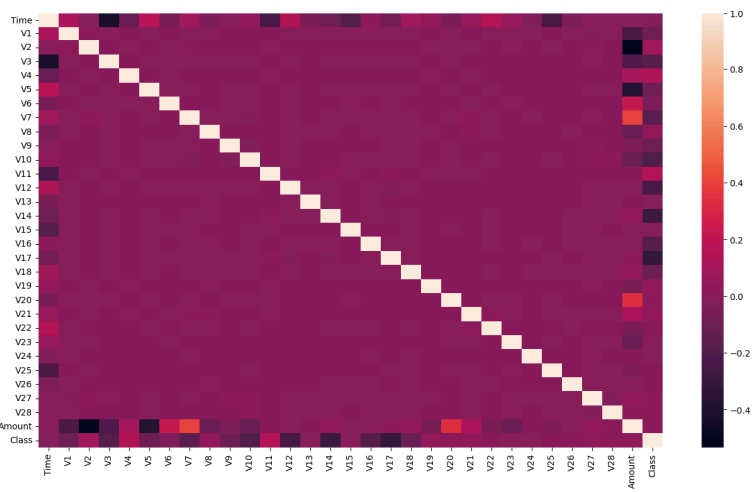
**1.3 Data Source**

The dataset used in this project was sourced from Kaggle, an open-access platform. It contains information on transactions conducted over two days in 2013 by European credit card users. The dataset includes 31 attributes and 284,808 rows. Out of these, 28 attributes are numerical variables that have been anonymized using PCA transformation to protect customer privacy. The remaining three attributes are: “Time,” which represents the time elapsed in seconds between each transaction and the first transaction; “Amount,” which indicates the monetary value of the transaction; and “Class,” a binary variable where "1" denotes a fraudulent transaction and "0" represents a legitimate transaction.

Dataset : <https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud/data>

Fig. shows the correlation matrix of the dataset.

The correlation matrix graphically gives us an idea of how features correlate with each other and can help us predict the most relevant features for the prediction.



# 2. Related Work

Detecting fraud in credit card transactions often involves analyzing historical data to identify irregularities or unusual behaviors. Traditional methods have typically relied on rule-based systems that operate using predefined criteria, such as flagging transactions that exceed set limits or originate from suspicious locations. However, these systems face limitations in adapting to new and sophisticated fraud strategies, often resulting in high false positive rates, which can inconvenience legitimate users.

In recent years, machine learning has gained significant attention in fraud detection due to its ability to process large datasets and uncover complex patterns. Supervised learning algorithms, including logistic regression, decision trees, and support vector machines (SVM), are commonly employed. These methods use labeled data to train models capable of classifying transactions as either fraudulent or legitimate. For situations where labeled data is unavailable, unsupervised learning techniques like clustering and anomaly detection are used to identify deviations from normal transaction patterns and flag them as potential fraud.

Extensive research has been conducted to improve fraud detection systems using machine learning and artificial intelligence. Dal Pozzolo et al. (2015) demonstrated the effectiveness of ensemble methods, such as random forests and boosting algorithms, in addressing the issue of imbalanced fraud datasets. They emphasized the importance of techniques like resampling and cost-sensitive learning to tackle the challenge of skewed class distributions commonly found in fraud detection data.

Similarly, Bahnsen et al. (2016) explored cost-sensitive logistic regression to reduce false positives and improve the trade-off between precision and recall. Their work highlighted the importance of balancing model sensitivity and the costs associated with misclassification to ensure practical applicability.

Deep learning has also been pivotal in advancing fraud detection. Fiore et al. (2019) utilized deep autoencoders to identify anomalies in transaction data, achieving superior accuracy compared to traditional approaches. Additionally, recurrent neural networks (RNNs) have been applied to sequential transaction data, capturing temporal dependencies that provide insights into fraudulent behavior.

# 

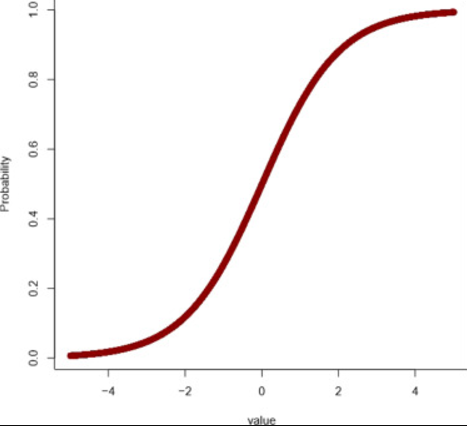
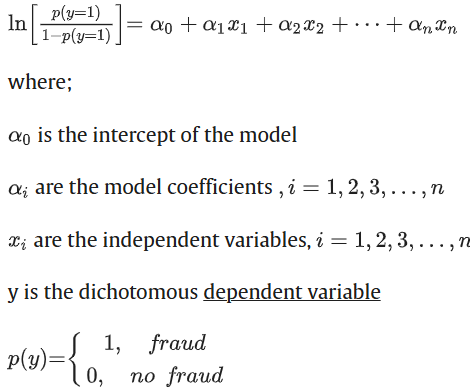
# 3. Methodology

## 3.1 Techniques and Packages

In this report, we have used various machine learning techniques like Logistic Regression, KNN, Decision Tree, Random Forest, and XG Boost.

1. **Logistic Regression:** Logistic Regression serves as a straightforward yet effective method for identifying fraudulent transactions. By analyzing input features and calculating their weighted influence, it estimates the likelihood of a transaction being fraudulent.

* Determining whether a transaction is fraudulent based on factors such as the amount, timestamp, and geographical location.
* Advantage: Simple to implement and interpret and best suited for datasets with a linear relationship between features and outcomes.

Eq. Equation of Logistic regression

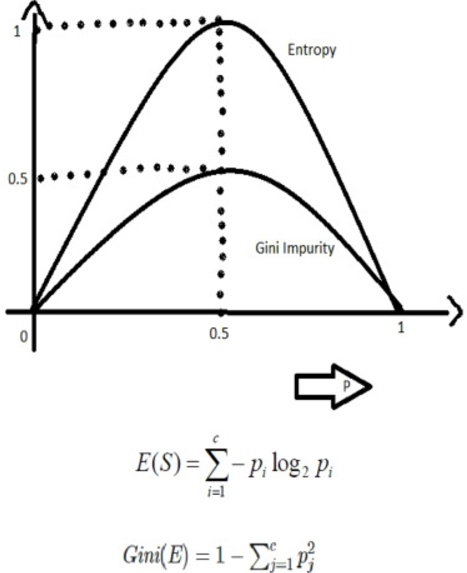
1. **KNN:** KNN identifies the status of a transaction by examining its similarity to nearby data points. Transactions are classified based on the majority class among their closest neighbors in the dataset.

* Spotting anomalies within smaller datasets or focused groups of transactions, such as those sharing similar timestamps or geographic locations.
* Advantages: Easy to understand and implement and also useful for detecting localized anomalies and capturing unique patterns
* Challenges: Computationally demanding for large datasets and requires careful selection of the "k" value and distance metrics for optimal results.

Eq. Equation of KNN (Euclidean Distance Formula)

1. **Decision Tree:** Decision Trees use a tree-like structure to divide the data into subsets based on feature conditions, generating interpretable rules for classifying transactions.

* Creating decision rules to classify transactions while providing stakeholders with a transparent reasoning process.
* Advantages: Produces clear, easy-to-follow decision paths and also handles both numerical and categorical variables effectively
* Challenges: May overfit the training data, especially when the dataset is highly complex.



Eq. Equation of Decision Tree

1. **Random Forest:** Random Forest builds multiple decision trees using different data and feature subsets and combines their outputs to improve classification accuracy while reducing overfitting.

* Classifying fraudulent transactions in datasets with multiple features, such as user demographics, transaction type, and merchant details.
* Advantages: Works well with large and high-dimensional datasets and also Mitigates overfitting through the ensemble approach.

Eq. Equation for Random Forest

1. **XG Boost:** XG Boost, short for "Extreme Gradient Boosting," is a highly popular and widely used machine learning algorithm known for its exceptional performance in handling large datasets and achieving state-of-the-art results in various machine learning tasks, including classification and regression. A notable strength of XG Boost is its ability to manage missing values efficiently, enabling it to work effectively with real-world data without requiring extensive preprocessing. Additionally, XG Boost is designed with built-in support for parallel processing, allowing it to train models on large datasets quickly and efficiently.

* XG Boost is an advanced ensemble method that uses gradient boosting to train a series of decision trees iteratively, focusing on minimizing classification errors. This makes it highly suitable for detecting fraud in datasets with significant imbalances.
* Detecting rare fraudulent activities in datasets where genuine transactions vastly outnumber fraudulent ones, leveraging XG Boost to identify hard-to-detect patterns.
* Advantages: Highly accurate and efficient due to its advanced optimization and regularization techniques and also effectively addresses class imbalance through weighted loss functions.

**Obj(θ)=**

Eq. Equation for XG Boost (objective function)

In this report, we have used various packages such as:

* Numpy: Provides support for numerical operations and working with arrays and also used for efficient mathematical computations and handling large datasets.
* Pandas: Used for data manipulation and analysis, especially with tabular data (e.g., .csv files).
* matplotlib.pyplot: For creating static, interactive, and dynamic visualizations.
* Seaborn: Built on top of Matplotlib, used for making statistical data visualizations more intuitive and attractive.
* scipy.stats (specifically norm): Provides statistical functions; here, the norm might be used to work with normal distributions.
* sklearn.preprocessing:
* MinMaxScaler: Scales data to a range (default [0, 1]).
* StandardScaler: Standardizes data (mean=0, variance=1).
* RobustScaler: Scales data using statistics less sensitive to outliers.
* sklearn.model\_selection:
* train\_test\_split: Splits data into training and test sets for model evaluation.
* GridSearchCV: Optimizes hyperparameters by exhaustively searching over specified parameter grids.
* imblearn.over\_sampling.SMOTE:

Synthetic Minority Oversampling Technique (SMOTE) is used to balance datasets by generating synthetic samples for the minority class.

* sklearn.linear\_model.LogisticRegression:

Logistic regression algorithm for binary or multiclass classification tasks.

* sklearn.neighbors.KNeighborsClassifier:

k-Nearest Neighbors (KNN) algorithm for classification tasks.

* sklearn.tree.DecisionTreeClassifier:

Decision Tree algorithm for classification and regression tasks.

* sklearn.ensemble.RandomForestClassifier:

Random Forest ensemble method for classification and regression tasks.

* xgboost.XGBClassifier:

Gradient boosting model for classification tasks.

## 3.2 End-to-End Methodology

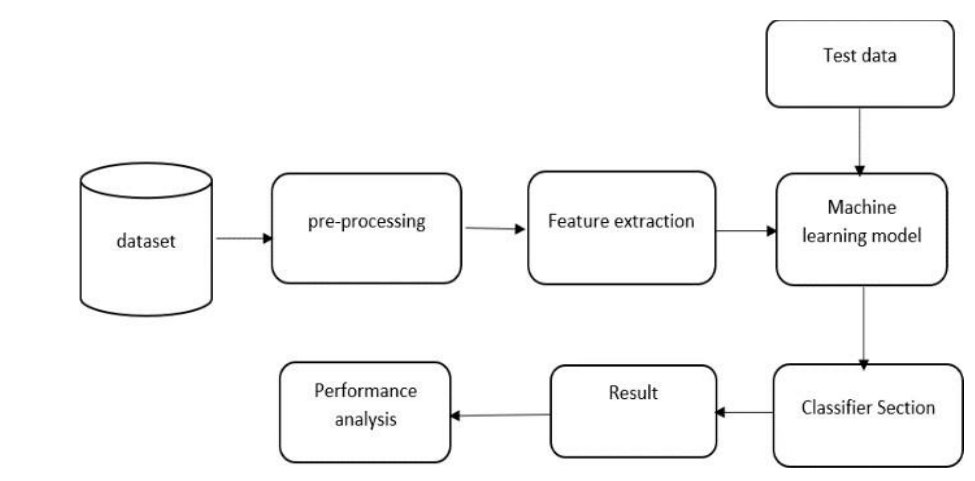


Fig. Data Flowchart in the project

1. **Data Collection**

* Objective: Gather a dataset of credit card transactions, including information such as transaction amount, time, location, and fraud label. Real-world datasets often include highly imbalanced data (e.g., more legitimate transactions than fraudulent ones).
* Dataset: <https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud/data>

1. **Data Cleaning**

* Remove duplicate transactions and handle missing entries (e.g., impute with the mean or use domain-specific logic).
* Ensure all numerical features are consistent in scale (e.g., currency values in a uniform format).

1. **Exploratory Data Analysis**

* Visualize distributions of features like transaction amounts or times.
* Investigate correlations between features and fraud labels.
* Detect anomalies and outliers that may indicate fraud trends.



Fig. Class Distribution(Fraud vs Non-Fraud)

1. **Advance Data Cleaning and Preprocessing**

* Visualize distributions of features like transaction amounts or times.
* Investigate correlations between features and fraud labels.
* Detect anomalies and outliers that may indicate fraud trends.

1. **Data Transformation**

* Scale features using techniques like MinMaxScaler or StandardScaler to ensure consistent ranges.
* Convert categorical variables (e.g., merchant type) into numerical formats using one-hot encoding.

1. **Feature Engineering**

* Generate new features such as transaction frequency per card or average transaction amount per day.
* Use domain expertise to add relevant features (e.g., time between transactions, location-based risk scores).

1. **Handling Imbalanced Data**

* Apply oversampling techniques like SMOTE to generate synthetic fraud samples.
* Use under-sampling to reduce the size of the majority class.
* Employ class weighting in model training to penalize misclassification of fraud cases more heavily.

1. **Splitting Data**

* Use stratified sampling to maintain the class imbalance in all splits.
* Typical split: 70% training, 15% validation, 15% testing.

1. **Model Selection**

* Common models: Logistic Regression, Decision Trees, Random Forest, XG Boost, KNN**.**
* Start with simple models (e.g., Logistic Regression) and progress to complex ones as needed.

1. **Model Training**

* Optimize for metrics like recall (to capture fraud cases) and precision (to reduce false positives).
* Use techniques like cross-validation for robust training.

1. **Hyperparameter Tuning**

* Techniques: Grid Search, Random Search, or Bayesian Optimization.
* Examples: Adjusting tree depth for Random Forest or learning rates for XG Boost.

1. **Model Evaluation**

* **Accuracy:** Overall correctness (not always suitable for imbalanced data).
* **Precision:** Fraction of identified fraud cases that are correct**.**
* **Recall:** Fraction of actual fraud cases correctly identified**.**
* **F1-Score**: Balances precision and recall.
* **Confusion Matrix**: The confusion matrix provides actionable insights into model strengths and weaknesses, enabling targeted improvements in fraud detection systems.

1. **Model Improvement**

* Adjust hyperparameters.
* Incorporate new features or refine existing ones.
* Combine models using ensemble methods.

1. **Final Model and Deployment**

* Deploy as part of a real-time fraud detection system integrated with payment gateways.
* Use APIs or microservices for seamless integration.

1. **Monitor and Maintain the Model**

* Regularly monitor metrics like Accuracy, recall, and precision to detect model drift.
* Update the model with new transaction data to adapt to emerging fraud patterns.
* Set up alerts for significant drops in performance or spikes in false positives.
  1. **Challenges and Areas for Improvement in Credit Card Fraud Detection Using Machine Learning**

Credit card fraud detection poses several challenges that require careful consideration to enhance the performance of machine learning models:

1. **Imbalanced Datasets**  
   A significant challenge in fraud detection is the imbalance between fraudulent and non-fraudulent transactions, as legitimate transactions far outnumber fraudulent ones. This imbalance can lead to biased predictions favoring the majority class, resulting in reduced detection rates for fraudulent transactions.
2. **Evaluation Metrics**  
   Relying solely on accuracy as a performance metric is insufficient for imbalanced datasets. Metrics such as precision, recall, and the F1 score are crucial for evaluating the model’s performance in distinguishing between the minority (fraudulent) and majority (legitimate) classes.
3. **Model Selection and Tuning**  
   Effective fraud detection requires selecting the most appropriate model and optimizing its hyperparameters. Without detailed hyperparameter tuning, the model’s performance may not reach its full potential. Exploring additional models or ensemble methods could yield better results.
4. **Feature Engineering**  
   Feature selection and engineering play a vital role in improving model performance. Identifying and leveraging the most important features or creating new, informative features can enhance the predictive power of the model.
5. **Cross-Validation**  
   Cross validation ensures the robustness and generalizability of the model to unseen data. Skipping this step may result in overfitting or underfitting, leading to unreliable predictions.
6. **Dataset Balancing**  
   Addressing data imbalance is critical for effective fraud detection. Techniques such as the Synthetic Minority Oversampling Technique (SMOTE) can help balance the dataset, ensuring the model learns from both fraudulent and legitimate transactions effectively.
7. **Computational Efficiency**  
   Advanced models like XG Boost and clustering algorithms can be computationally intensive, especially with large datasets. Optimizing the code and ensuring efficient use of computational resources is crucial to reduce training and inference times.

By addressing these challenges, machine learning models for credit card fraud detection can be made more robust, accurate, and computationally efficient, ensuring better protection against fraud in real-world applications.

# 4. Result

We have evaluated the results of the model based on the various performance measures. Performance measures are essential metrics used to evaluate the effectiveness of a machine learning algorithm. Performance measures are used to compare the performance of different models on a given data set and to determine which model provides the best results. Performance measures can be used to identify areas in which the model can be improved and to determine the overall accuracy of the model.

The most common performance measures used for Deep Learning and Transfer Learning Techniques:

1. Accuracy
2. Precision
3. Recall
4. F1-Score
5. Specificity

**Accuracy :** The accuracy score gauges how accurate a categorization model is all in all. The ratio of accurate forecasts (true positives and true negatives) to all predictions is calculated. The formula is:

Equation 6

**Recall:** The sensitivity score, sometimes referred to as recall or true positive rate (TPR), gauges how well a model can identify positive cases. It determines the ratio of accurate positive forecasts to all actual positive cases. The formula is:

Equation 7

**Specificity Score:** The specificity score gauges how well a model can recognize negative cases. It determines the ratio of accurate negative predictions to all actual negative cases. The formula is:

Equation 10

**Precision:** The precision score measures the fraction of accurate positive predictions and overall positive predictions. Instead, then concentrating on the model's overall accuracy, it emphasizes the accuracy of positive predictions. The formula is:

Equation 8

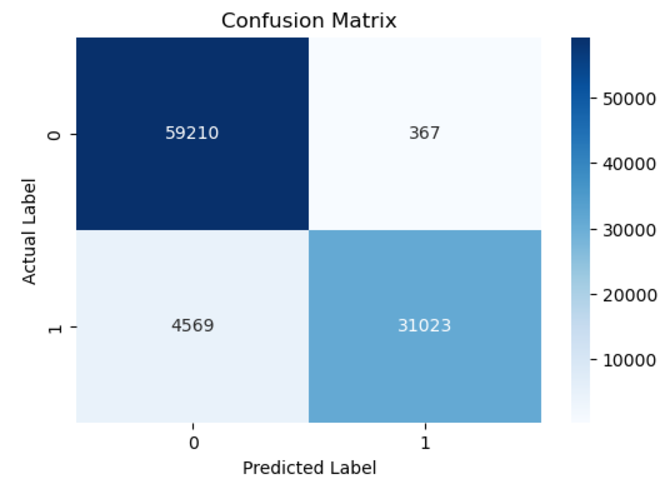
**F1-Score:** The F1 score is a performance metric that has a harmonic mean of precision and recall into a single value, providing a balanced measure, especially useful for imbalanced datasets like credit card fraud detection.

*F*1=2 × Precision + Recall   
 Precision × Recall​

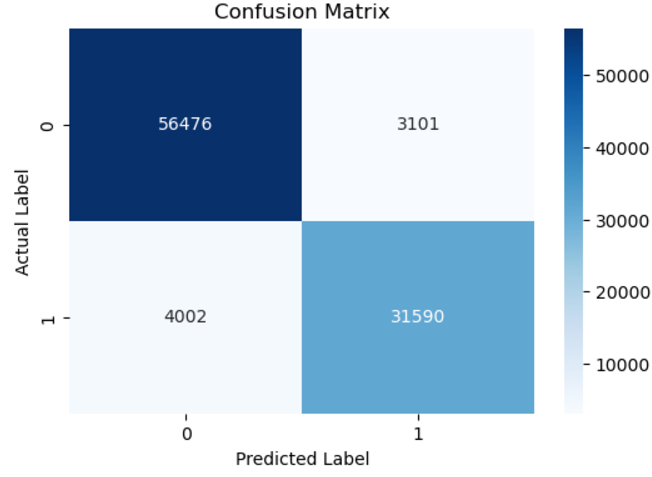
Equation 9

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Performance metrics** | **Accuracy** | **Precision** | **Recall** | **F1-Score** |
| **Logistic Regression** | **0.9481** | **0.9883** | **0.8716** | **0.9263** |
| **KNN** | **0.9254** | **0.9106** | **0.8876** | **0.8989** |
| **Decision Tree** | **0.9467** | **0.9339** | **0.9227** | **0.9283** |
| **XG Boost** | **0.9865** | **0.9873** | **0.9765** | **0.9819** |
| **Random Forest** | **0.9698** | **0.9696** | **0.97991** | **0.9797** |

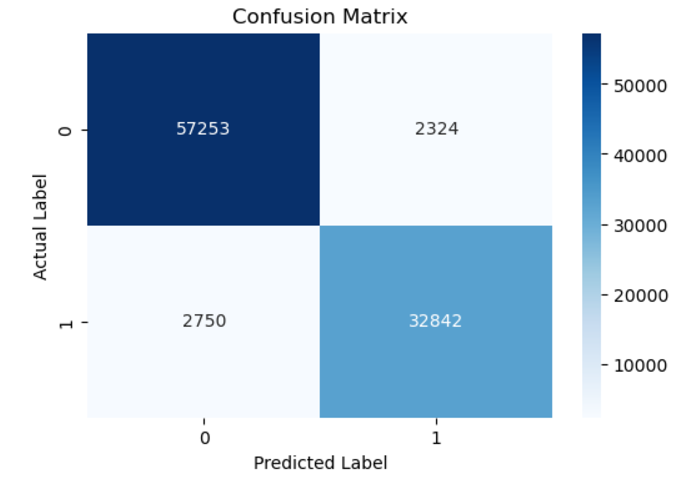
Table: Indicates the performance metrices of the techniques used

****

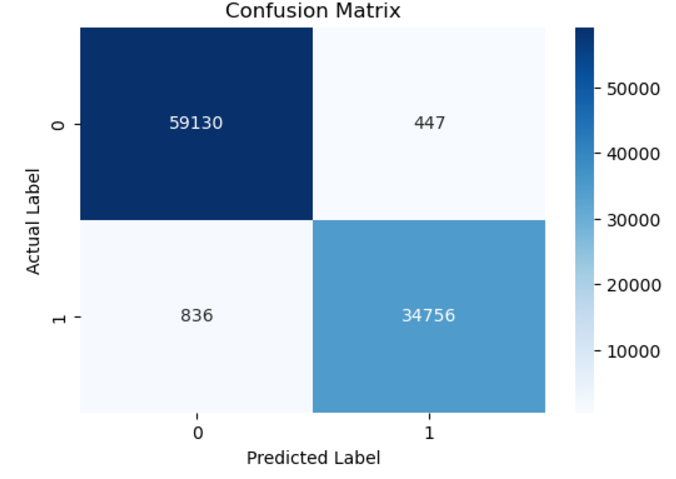
**Fig 1. Confusion Matrix for Logistic Regression**

****

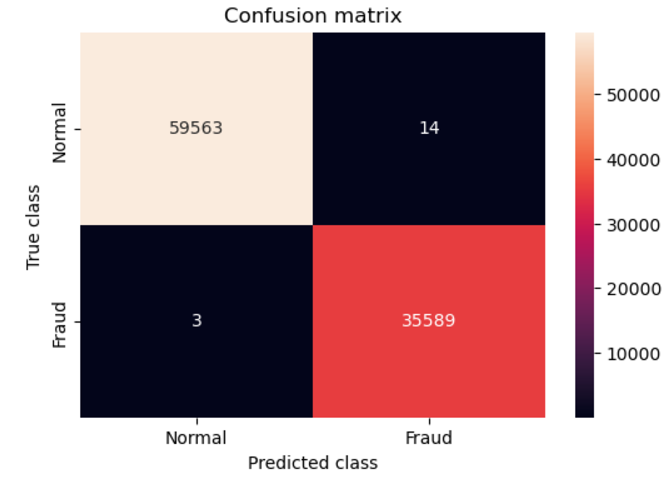
**Fig2. Confusion Matrix for KNN**

****

**Fig3. Confusion Matrix for Decision Tree**

****

**Fig4. Confusion Matrix for XG Boost**

****

**Fig5. Confusion Matrix for Random Forest**

**After evaluating various classifiers on a balanced dataset created using the SMOTE technique, it was observed that the XG Boost Classifier outperformed all other models. It achieved an impressive** **accuracy of 0.9865, a Precision of 0.9873, a recall of 0.9765, and an F1-score of 0.9819. These results indicate that XG Boost is highly effective at handling balanced datasets and delivers superior performance compared to other classifiers.**

# 

# 5. Conclusion and Future Work

The findings indicate that the XG Boost classifier demonstrated the best performance in terms of accuracy and other evaluation metrics, including Precision, recall, and F1-score, for detecting fraudulent credit card transactions.

* After balancing the dataset, XG Boost maintained an accuracy of 0.9865, a Precision of 0.9873, a recall of 0.9765, and an F1-score of 0.9819, for scenarios where minimizing false negatives (incorrectly classifying fraud as legitimate) is a priority, models like XG Boost and Random Forest are ideal choices due to their high precision and sensitivity.

Based on the balance of accuracy and other performance metrics, XG Boost is recommended as the most effective model for detecting fraudulent credit card transactions. However, the choice of the best model should consider specific requirements, computational constraints, and the unique objectives of the fraud detection system. However, models like **XG Boost and Random Forest** are ideal choices due to their high precision and sensitivity Additional fine-tuning, extensive testing, and validation are advised before deploying the model in a real-world production environment.

The model can be enhanced in several ways, such as applying it to diverse datasets of varying sizes and structures or experimenting with different data splitting ratios and algorithms. For instance, integrating telecom data to track the cardholder's location could significantly improve fraud detection. By analyzing the cardholder's real-time location, discrepancies can be flagged more effectively. For example, if a cardholder is in Dubai but a transaction is recorded in Abu Dhabi simultaneously, it would be identified as potential fraud. This approach can provide a more accurate and context-aware fraud detection mechanism.

# 6. References:

1. <https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud/data>

2.<https://www.kaggle.com/code/mennakhaled2/credit-card-fraud-detection-end-to-end-ml-model#1-Data-Concepts%F0%9F%93%9D>

3.<https://www.infosysbpm.com/blogs/bpm-analytics/machine-learning-for-credit-card-fraud-detection.html>

4.<https://www.sciencedirect.com/science/article/pii/S2772662223000036>

5.<https://www.geeksforgeeks.org/ml-credit-card-fraud-detection>

6. [https://towardsdatascience.com/the-random-forestalgorithm-d457d499ffcd](%20https://towardsdatascience.com/the-random-forestalgorithm-d457d499ffcd)

7.<https://ijcsmc.com/docs/papers/April2021/V10I4202112.pdf>

8.<https://xgboost.readthedocs.io/en/stable/>