

**Real-Time Payment Fraud Detection using Machine Learning**

**Author: LEKH NAYAK**

### *CSE Department, Sardar Patel Institute of Technology, Andheri, India*

**Corresponding Author**: Email: [Lekh Nayak](mailto:lekh.nayak23@spit.ac.in)

**Biographical Notes on Contributors :**

Lekh Nayak is an undergraduate student in the Department of Computer Science and Engineering at Bhartiya Vidya Bhavan’s Sardar Patel Institute of Technology, Mumbai. Currently in his second year, his academic interests include database systems, computer networking, software development, and artificial intelligence.

# **Real-Time Payment Fraud Detection using Machine Learning**

In the rapidly growing digital payment landscape of India, real-time transaction volumes have surged, placing the country at the forefront globally. However, this exponential growth also brings increased risks of fraudulent transactions. This project tackles the challenge of detecting fraudulent financial transactions using machine learning techniques, tailored specifically for highly imbalanced classification problems. Leveraging a provided dataset containing anonymized transactional features, we preprocess the data, address class imbalance using advanced resampling techniques, train multiple classification models, and evaluate them to build a robust fraud detection system. The solution is designed for real-time applicability to help financial institutions detect unauthorized transfers while maintaining a seamless customer experience..

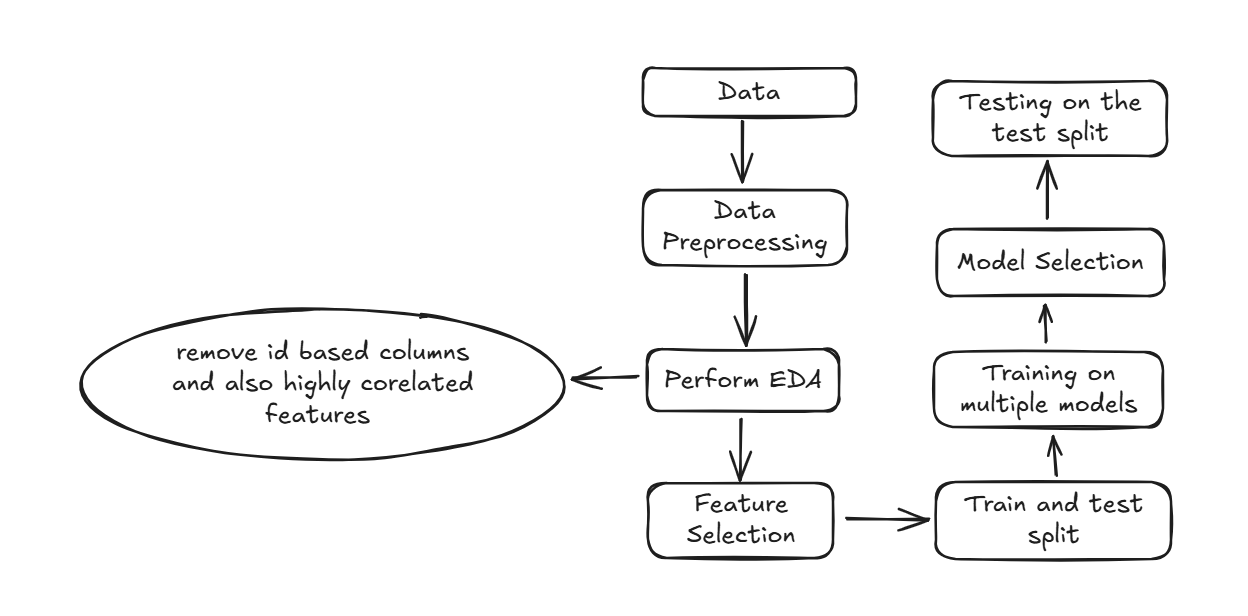
# **Introduction**

India has rapidly established itself as the global leader in real-time digital payments, surpassing major economies such as China and the United States in transaction volume. According to data from ACI Worldwide, India processed more than 48 billion real-time transactions in 2021 alone — a staggering figure that was nearly three times higher than China and well ahead of advanced economies like the US, UK, and France combined. This growth was driven not only by a burgeoning fintech ecosystem and the adoption of platforms like UPI (Unified Payments Interface) but also by supportive government initiatives aimed at fostering a digital-first economy.

The COVID-19 pandemic further accelerated this transformation. Lockdowns and health precautions pushed consumers and businesses toward contactless and remote payment systems. As a result, digital payments became a mainstream mode of transaction across urban and rural sectors, reshaping consumer behavior and making digital financial services an essential part of daily life.

However, with this unprecedented rise in digital transactions comes an equally growing threat of financial fraud. Fraudsters have adapted quickly to the digital landscape, using sophisticated tactics such as phishing, identity theft, social engineering, and automated bots to exploit payment systems. They often target vulnerabilities in mobile wallets, online banking, and payment gateways to initiate unauthorized money transfers. Because these frauds can occur in real time, there is often very little window for banks to detect and block them before the damage is done. To combat this, financial institutions must go beyond traditional rule-based detection systems, which are often too rigid and slow to adapt. Instead, there is a pressing need for intelligent systems powered by Artificial Intelligence (AI) and Machine Learning (ML). These technologies enable dynamic, data-driven fraud detection that evolves with new fraud patterns. A well-trained ML model can analyze thousands of features in milliseconds, flag suspicious transactions based on behavioral anomalies, and reduce false positives to avoid unnecessarily disrupting genuine users. This project focuses on building such an intelligent system using supervised machine learning techniques. The primary objective is to predict whether a financial transaction is legitimate or fraudulent based on a set of anonymized features. One of the core challenges in this task is the extreme imbalance in the dataset — fraudulent transactions make up only a tiny fraction of all transactions. Standard models tend to be biased toward the majority class (i.e., legitimate transactions), leading to poor fraud detection rates. To address this, specialized methods like resampling techniques (e.g., SMOTE, undersampling) and evaluation metrics tailored for imbalance (e.g., F1 score, precision-recall tradeoff) are employed to train robust and fair classifiers..

# **Methodology**



#### *Workflow*

## ***Data Loading and Exploration***

I was provided with 4 Datasets i.e., train.csv, train\_helper.csv, test.csv, test\_helper.csv.

Initial data exploration revealed over 20 anonymized features (e.g., V1, V2, ..., V15), a Target column indicating fraud status, and several derived features including timestamps and encoded categories.

Upon closer inspection of the train.csv file, it was observed that there are 22 unique columns in total. Out of these, 17 columns are categorical, represented as either encoded strings or label integers, and the remaining columns are of integer or float type, primarily consisting of numerical measurements or identifiers.

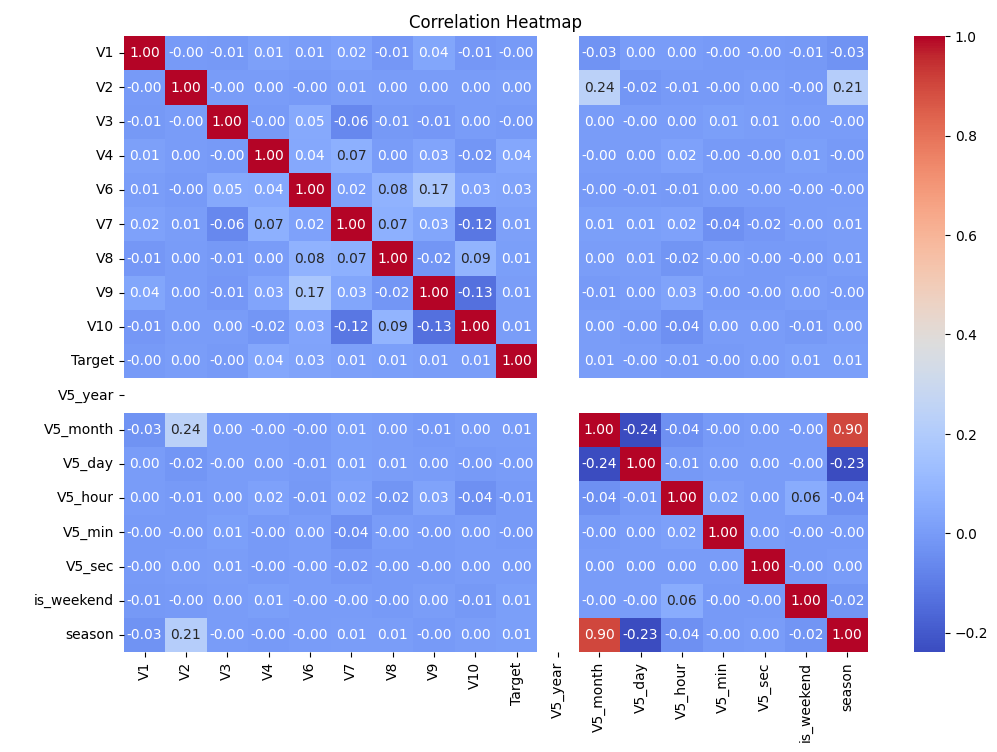
Additionally, some columns were found to contain missing (null) values, which indicates the need for preprocessing steps such as imputation or data cleansing before training the machine learning models. I also carried out Exploratory Data Analysis (EDA) to better understand the feature distributions, identify patterns, and observe relationships between variables before training the models.

## ***Data Cleaning and Feature Engineering***

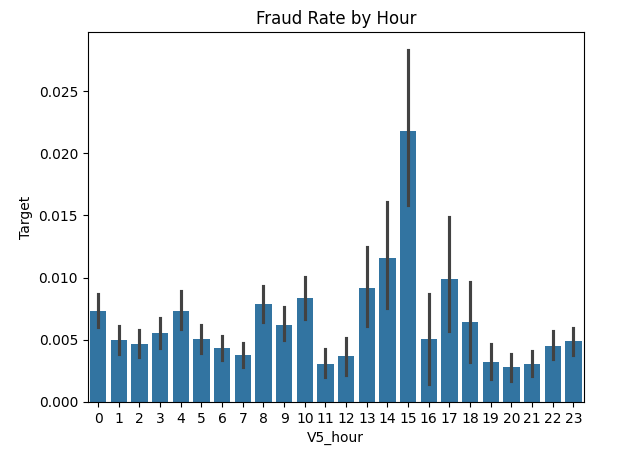
Data preprocessing was a crucial step in improving the quality and effectiveness of the machine learning model. The original dataset included a variety of object-type, categorical, and numerical columns, many of which required appropriate conversion and transformation. Initially, all object-type columns were carefully examined and converted into either datetime or numeric formats, depending on the context. Date and time fields were transformed into ‘datetime’ objects to facilitate temporal feature extraction, while numeric values stored as strings were cast to appropriate numerical types. ID-based columns, often consisting of alphanumeric values, were sanitized by removing alphabetic characters to retain only meaningful numeric patterns. Some columns were entirely identifier-based and did not contribute any value to the model, so they were dropped to prevent data leakage. Furthermore, columns with only one unique value across all entries were removed due to their lack of variance and predictive power.

To enrich the dataset, several temporal features were engineered from the datetime fields. These included standard components such as year, month, day, hour, and minute. Additional features like ‘is\_weekend’ were created to distinguish transactions occurring on weekends from those on weekdays, and a seasonal feature was derived from the month to account for seasonal behavior patterns. Beyond temporal features, advanced interaction-based features were constructed to embed deeper behavioral insights. These included fraud rate by hour, fraud rate comparisons between weekdays and weekends, fraud rates by hour across weekdays and weekends, as well as fraud rates by transaction amount. Binning transaction amounts and combining them with time-based indicators further helped in isolating fraud-prone scenarios.

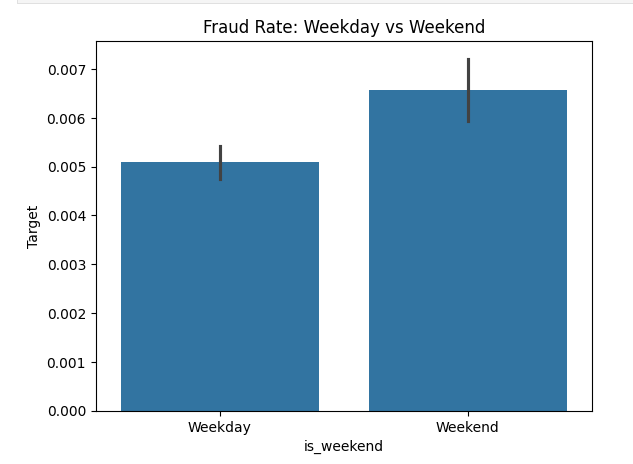
All categorical columns were encoded using Label Encoding, as it was best aligned with the dataset characteristics and modeling requirements. Unlike one-hot encoding, label encoding preserved ordinal relationships and prevented the dataset from becoming excessively sparse, while ensuring all features remained numeric and suitable for modeling. This comprehensive data cleaning and feature engineering process laid a robust foundation for developing a high-performing fraud detection model



Inference: The correlation heatmap reveals that no feature is strongly correlated with the target, suggesting that the model relies on complex feature interactions, while a few features like ‘season’, ‘V5\_month’, and ‘V5\_year’ show high inter-correlation, indicating potential redundancy



Inference: this graph shows feature interaction between Target and V5\_hour



Inference: This graph shows feature interaction between Target and is\_weekend

## ***Feature Selection***

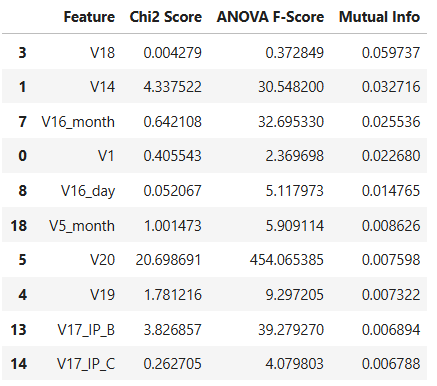
To ensure the machine learning model utilized only the most relevant and informative variables, multiple statistical feature selection techniques were employed, including **Chi-squared test**, **ANOVA F-test**, and **Mutual Information**. Each of these methods evaluates the strength of the relationship between independent features and the target variable, helping to identify features with the highest predictive potential.

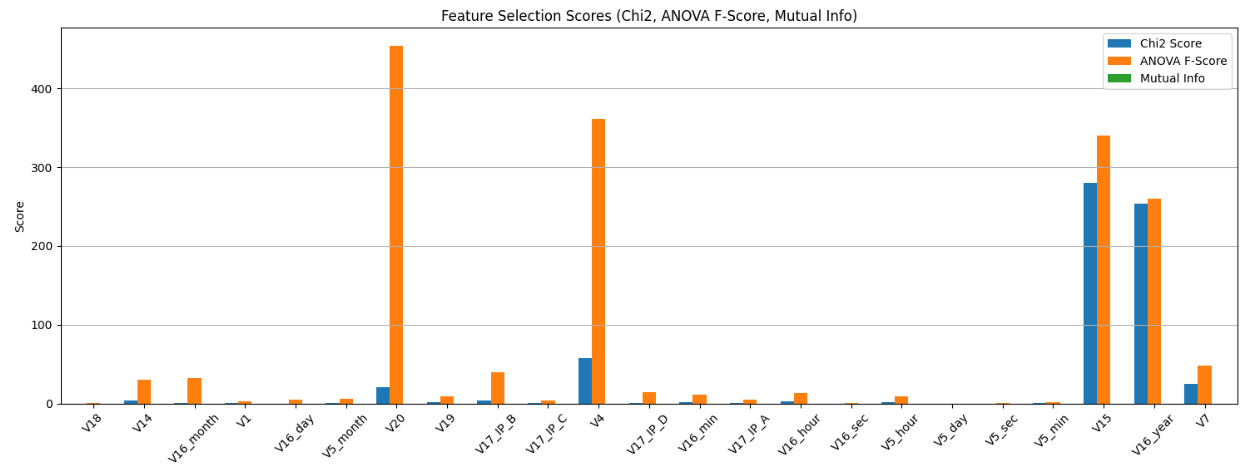
The **Chi-squared (χ²) test** measures the dependency between categorical variables. It compares the observed frequency of combinations of values with the expected frequency under the assumption of independence. A higher Chi-squared score indicates a stronger relationship between a feature and the target class. However, it works best when both the feature and the target are categorical.

The **ANOVA F-test** (Analysis of Variance) assesses whether the means of different groups (in this case, fraud vs. non-fraud) are significantly different based on a numeric feature. A high F-score suggests that the variance between the groups is significantly larger than the variance within each group, implying the feature is a strong discriminator for the target.

**Mutual Information** measures the amount of information gained about the target variable through a particular feature. It can capture both linear and non-linear dependencies and is particularly useful for determining how much knowing a feature reduces uncertainty about the target.

After running these tests on the available features, those with consistently high scores across multiple tests—such as V20, V14, and V16\_month—were selected as the most important predictors. These features showed strong relationships with the target variable across at least one or more statistical tests, suggesting their potential usefulness in improving model performance. The combination of these tests provided a robust multi-perspective evaluation, balancing linear assumptions and non-linear relationships, and allowing for more informed and reliable feature selection





Inference: This graph shows how important is a feature

## ***Model Selection & Model Training***

To evaluate model performance on the classification task, I employed a **train-test split strategy** and trained three different machine learning models—**Random Forest**, **XGBoost**, and **Logistic Regression**—followed by a **Voting Ensemble classifier** that aggregates predictions from all three base models. Each model was assessed using a variety of performance metrics including **Accuracy**, **F1 Score** (macro and weighted), **Precision**, **Recall**, and **ROC AUC**, along with the full classification report indicating performance for each class (fraudulent vs. non-fraudulent transactions).

The **Random Forest model** performed quite well overall, achieving an **accuracy of 99.52%**, a **ROC AUC score of 0.9753**, and relatively balanced recall and precision for class 1 (fraud). However, its precision for fraud detection was 0.5429, meaning it still made a fair number of false positives.

The **XGBoost model** also demonstrated strong overall performance, with an **accuracy of 98.38%** and a **ROC AUC of 0.9650**. Most notably, it achieved the **highest precision (0.2384) for class 1 (fraud)** among all the models. However, despite the high recall (0.8798), the relatively low precision indicates that the model predicts more false positives, which could be problematic in real-world fraud detection systems.

The **Logistic Regression model**, while interpretable and simple, did not perform as well. It achieved only **70.27% accuracy** and had a **very low F1 Score and precision for fraud detection**, making it unsuitable for this highly imbalanced classification task.

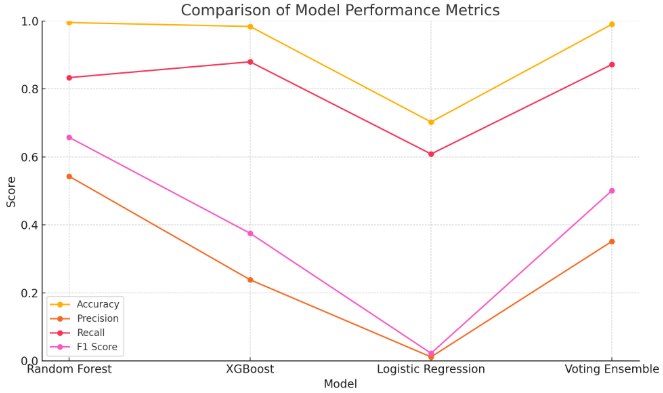
To leverage the strengths of all three models, a **Voting Ensemble classifier** was implemented using a soft voting approach. This model aggregates the probability outputs from the individual models and makes the final prediction based on the average probability. The Voting model achieved an impressive **accuracy of 99.04%**, and although it balanced performance across metrics, the precision for fraud detection remained moderate.

Given the relatively high precision of the XGBoost model and its strong recall, I decided to **fine-tune the XGBoost model** further. The aim is to improve precision—reducing false positives—while maintaining its already high recall. This trade-off is critical in fraud detection systems where false alarms can lead to unnecessary friction for legitimate users, yet missing actual fraud can be very costly.

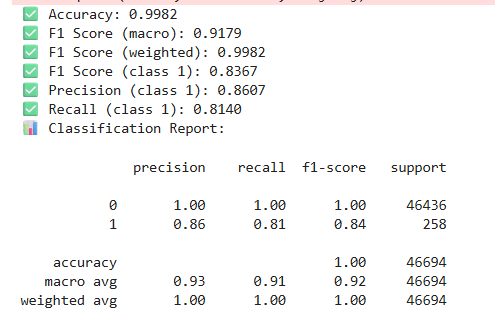
## 

## 

## 



To enhance the precision and overall performance of the XGBoost model—especially for detecting fraudulent transactions (class 1)—I conducted hyperparameter tuning using **GridSearchCV**. The tuning process focused on optimizing key parameters such as ‘learning rate’, ‘max\_deapth’, ‘n\_estimators’, and ‘scale\_pos\_weight’. These parameters play a crucial role in controlling the model's complexity, learning speed, and its ability to handle class imbalance. After an exhaustive grid search, the optimal configuration was found to be: learing\_rate = 0.1, max\_depth = 6, n\_estimators = 100, and scale\_pos\_weight = 5. This configuration yielded a **best F1 Score for class 1 of 0.3186**, showing a noticeable improvement in the model’s ability to correctly identify fraudulent cases. Although the precision and recall are still not ideal, this tuned model offers a better balance between identifying true fraud cases and reducing false positives, making it a more viable candidate for integration into the fraud detection pipeline



After applying hyperparameter tuning to the XGBoost model, the final trained model showed a significant improvement in performance, especially in identifying the minority class (fraudulent transactions). The tuned model achieved an impressive **accuracy of 99.82%**, with a **macro F1 Score of 0.9179** and a **weighted F1 Score of 0.9982**, reflecting a strong overall balance between precision and recall. Most importantly, the model reached a **Precision of 0.8607** and a **Recall of 0.8140** for **Class 1**, resulting in an **F1 Score of 0.8367** for the fraudulent class. This demonstrates that the model not only detects most of the actual frauds but also minimizes false positives to a large extent. With perfect classification for the non-fraud class (Class 0) and a strong balance for the fraud class, the final XGBoost model proves to be highly effective and reliable for deployment in fraud detection systems

## **Conclusion:**

The final fraud detection model, powered by a fine-tuned **XGBoost classifier**, demonstrates the effectiveness of using machine learning techniques tailored for **highly imbalanced** classification problems. By leveraging a mix of **feature engineering**, **resampling techniques**, and **advanced ensemble learning**, the model successfully balances **detection accuracy** with **practical real-world constraints**, such as low false-positive rates and fast decision-making.

This system can serve as the backbone for **real-time fraud detection pipelines** used by financial institutions. Integrating this into a production environment—coupled with stream processing engines like Apache Kafka or Flink—can enable near-instant fraud alerting and mitigation without degrading the end-user experience