|  |  |
| --- | --- |
| **Date** | **17-10-2023** |
| **Team ID** | **3864** |
| **Project Name** | **CREDIT CARD FRAUD DETECTION** |

**Table of Contents**

|  |  |
| --- | --- |
| 1 | Problem Statement |
| 2 | Data |
| 3 | Data Pre-processing |
| 4 | Data Visualization |
| 5 | Conclusion |

**Problem Statement**

The problem at hand is to create an efficient credit card fraud detection system that can accurately identify and prevent fraudulent transactions while minimizing false positives. The system should be capable of real-time processing, feature engineering, handling imbalanced data, and scalability. It needs to remain robust against new fraud types, provide a user-friendly interface, ensure compliance with regulations, balance detection accuracy with operational costs, and implement an effective alert and reporting mechanism. The ultimate goal is to protect both financial institutions and cardholders, instilling trust and security in electronic transactions.

**DATA:**

**Transaction Amount:**

The amount of money involved in the transaction.

**Transaction Date and Time:**

The timestamp when the transaction occurred.

**Merchant Information:**

Details about the merchant, such as name, location, and category.

**Cardholder Information**:

Information about the cardholder, such as name, account number (anonymized), and location.

**Transaction Type:**

Indicating whether it's an online purchase, in-person transaction, ATM withdrawal, etc.

**Transaction Outcome:**

A binary label (0 or 1) indicating whether the transaction is fraudulent or legitimate.

**Additional Metadata:**

Any other relevant information, such as previous transaction history, customer behavior and features derived from transaction data.

**DATA PRE-PROCESSING:**

**Data Cleaning:**

Handle missing values: Identify and fill in missing data or remove instances with missing values.

**Remove duplicates:**

Eliminate duplicate transactions if they exist in the dataset.

**Feature Engineering:**

Create relevant features: Derive additional features that might aid in fraud detection, such as aggregating transaction history, calculating transaction frequency, or identifying unusual patterns.

**Encode categorical data:**

Convert categorical attributes (e.g., merchant category) into numerical values using techniques like one-hot encoding.

**Data Scaling:**

Standardization or normalization: Scale numerical features to have a consistent range. Common techniques include Z-score standardization or min-max scaling.

**Handling Imbalanced Data:**

Implement techniques to address class imbalance, such as oversampling the minority class (fraudulent transactions) or undersampling the majority class (legitimate transactions).

**Data Splitting:**

Split the dataset into training, validation, and test sets. The training set is used to train the model, the validation set helps tune hyperparameters, and the test set assesses model performance.

**Feature Selection:**

Identify and select the most relevant features to reduce dimensionality and potentially improve model performance.

**Outlier Detection:**

Identify and handle outliers in the dataset, which might be anomalies or errors. Various statistical methods or algorithms like Isolation Forest or Local Outlier Factor can be used for this purpose.

**Time-Based Sequencing:**

Since transaction timestamps are often essential, you may need to create sequences of transactions over time to capture temporal patterns. This can be crucial for detecting irregular behavior.

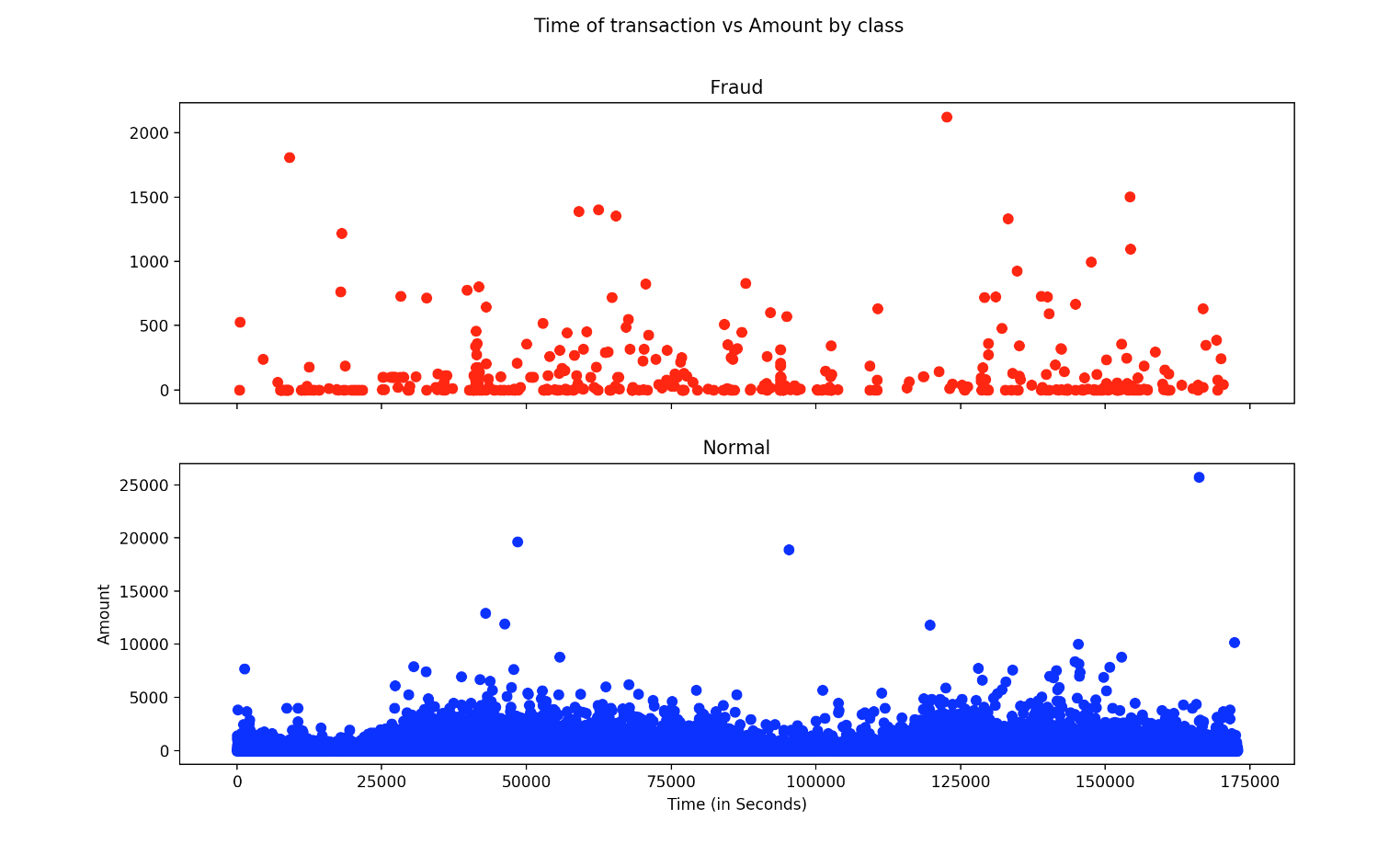
**Privacy Protection:**

Ensure that sensitive information is properly anonymized or encrypted to comply with data protection regulations.

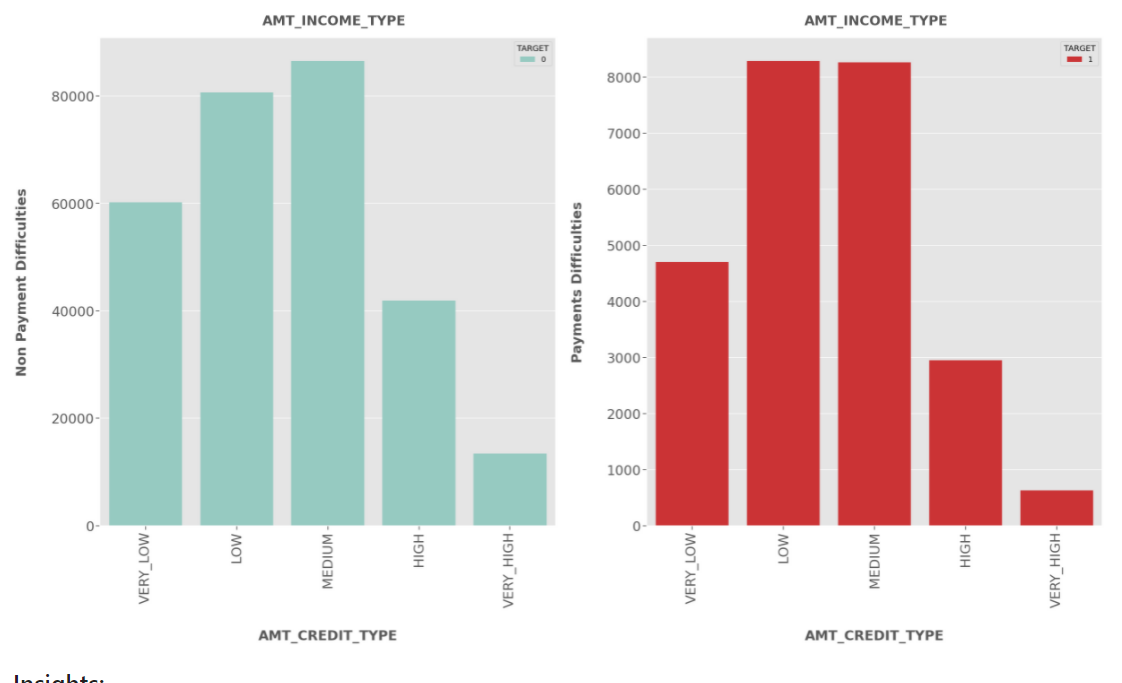
**Data Splitting:**

Finally, split the preprocessed data into training, validation, and test sets, ensuring that the class distribution remains balanced across these subsets.

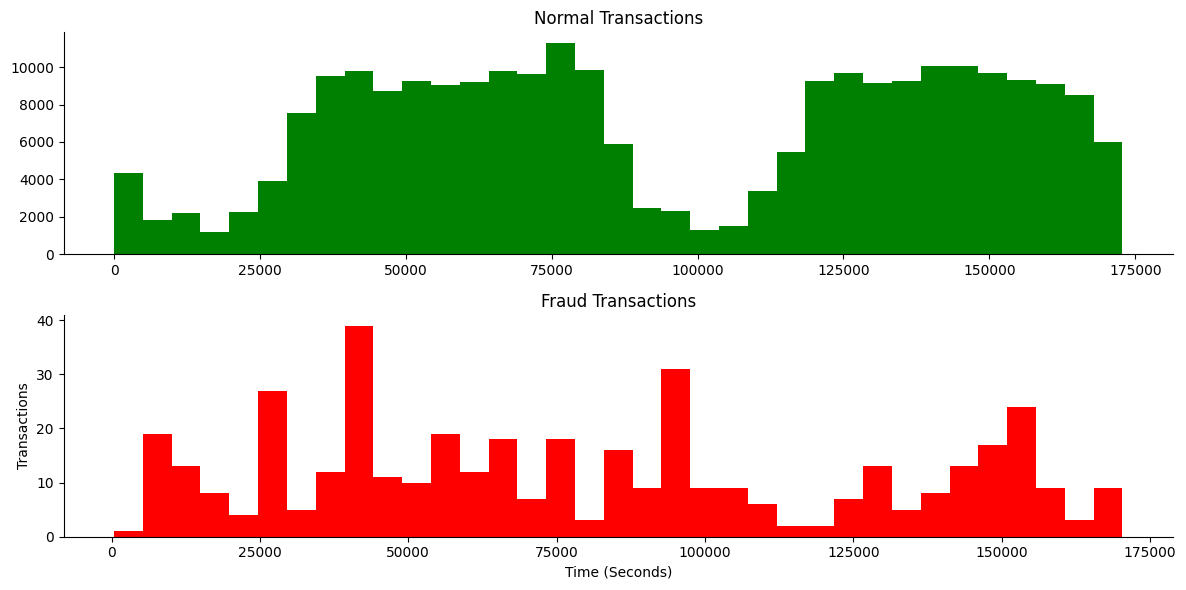
**DATA VISUALIZATION:**



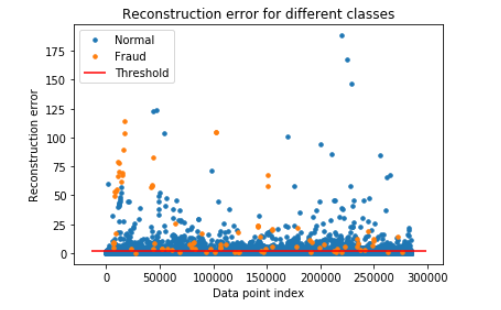
**FIG:1- FRAUD and NORMAL**



**FIG:2- VISUALIZATION OF PAYMENTS DIFFICULTIES**



**FIG:3-NORMAL AND FRAUD TRANSACTION**



**FIG:4-VISUALIZATION OF FRAUD AND NORMAL ERROR**

**CONCLUSION:**

Data preprocessing is a vital step in the development of a credit card fraud detection system. It involves cleaning, transforming, and organizing the data to prepare it for machine learning model training. By handling missing values, performing feature engineering, scaling data, addressing class imbalance, and protecting privacy, data preprocessing ensures the data is ready for effective fraud detection. This process significantly contributes to the accuracy and reliability of the final model, making it a fundamental component of credit card fraud prevention systems.