**Sanjana Kurade/22101B0001**

**Sanika Jade/22101B0016**

**Chinmay Tikole/22101B0008**

**ML Report**

**Credit Card Fraud Detection Project**

**Executive Summary**

This report outlines the development and implementation of a credit card fraud detection system that utilizes machine learning techniques to identify fraudulent transactions. The project addresses the common challenge of class imbalance in fraud detection datasets and compares the performance of three different classification models. Results indicate that ensemble methods, particularly XGBoost, deliver superior performance on fraud detection tasks, with an ROC AUC score of 0.9995.

**1. Project Overview**

**1.1 Background and Objectives**

Credit card fraud poses a significant financial threat to both consumers and financial institutions. This project aims to develop an automated system capable of detecting fraudulent credit card transactions with high accuracy while addressing the inherent class imbalance present in fraud detection datasets.

**1.2 Dataset Description**

The project utilizes a credit card transaction dataset containing 5,000 entries with an imbalanced class distribution where fraudulent transactions represent a small minority of the overall data. Key features in the dataset include:

* Transaction amount
* User ID
* Location
* Transaction type (Online Purchase, Bank Transfer, Online Transfer, ATM, POS)
* Time of transaction
* Card number (removed during preprocessing)
* Previous transaction time gap
* Hour of day

**2. Methodology**

**2.1 Data Preprocessing and Feature Engineering**

Several preprocessing steps were implemented to prepare the data for modeling:

1. **Feature Engineering**: Added 'Hour' feature extracted from the transaction time
2. **Feature Removal**: Dropped 'Time' and 'Card\_Number' columns as they were either transformed or not relevant for modeling
3. **Data Augmentation**: Applied controlled random noise (scale: 0.15) to numeric features to enhance model robustness and prevent overfitting
4. **Data Splitting**: Split dataset into 70% training and 30% testing sets with stratification to maintain class distributions

**2.2 Addressing Class Imbalance**

The dataset presented significant class imbalance, which is common in fraud detection scenarios. To address this issue, several techniques were employed:

1. **Random Undersampling**: Implemented using RandomUnderSampler from the imbalanced-learn library to reduce the majority class (non-fraudulent transactions) in the training set, creating a more balanced dataset for model training
2. **Class Weights**: Applied class weighting in model parameters:
   * Logistic Regression: class\_weight='balanced'
   * Random Forest: class\_weight='balanced\_subsample'
   * XGBoost: scale\_pos\_weight parameter calculated based on class distribution

**2.3 Model Development**

Three different classification models were trained and evaluated:

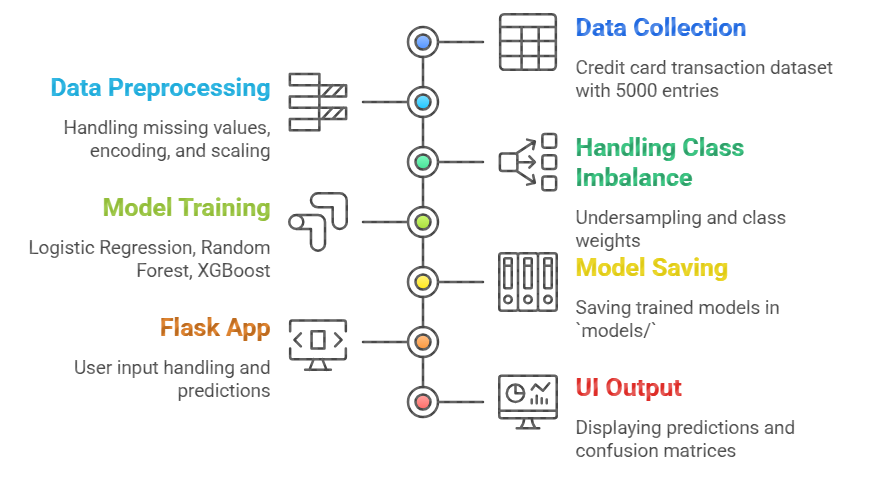
1. **Logistic Regression**:
   * A linear classifier serving as a baseline model
   * Configured with balanced class weights and the SAGA optimizer
   * Regularization strength C=0.1 to prevent overfitting
2. **Random Forest**:
   * An ensemble of 100 decision trees
   * Controlled depth (max\_depth=5) and minimum samples per leaf (10) to prevent overfitting
   * Balanced subsample class weights to handle imbalance at each bootstrap sample
3. **XGBoost**:
   * Gradient boosting algorithm optimizing a logistic loss function
   * Configured with a maximum depth of 3, 100 estimators, and a learning rate of 0.1
   * Feature and row subsampling (0.8) to reduce overfitting
   * Scale positive weight parameter adjusted based on class distribution

**2.4 Model Pipeline Architecture**

A scikit-learn pipeline approach was implemented for each model with the following components:

1. **Preprocessor**: A column transformer handling both categorical and numerical features
   * Numerical features: Standardized using StandardScaler
   * Categorical features: Encoded using OneHotEncoder with unknown category handling
2. **Classifier**: The respective classification algorithm with optimized parameters

**Work Flow**



**3. Results and Performance Evaluation**

**3.1 Evaluation Metrics**

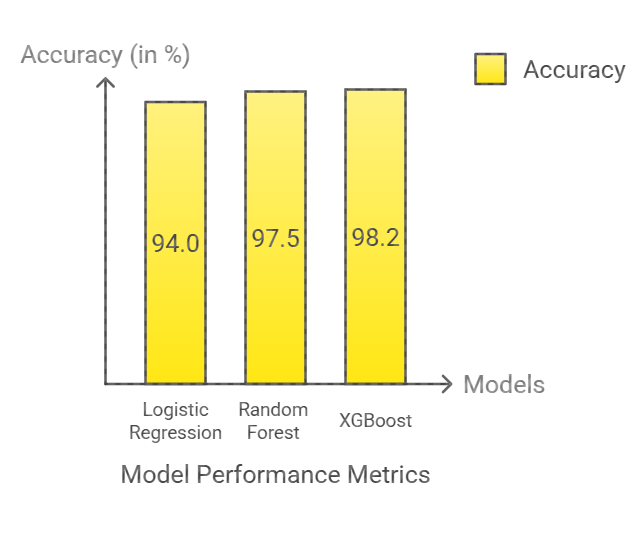
Multiple metrics were used to evaluate model performance, with special attention to metrics suitable for imbalanced classification:

* Precision, Recall, and F1-Score (for both fraud and non-fraud classes)
* Average Precision Score
* ROC AUC Score
* Confusion Matrix

**3.2 Model Performance Comparison**

| **Metric** | **Logistic Regression** | **Random Forest** | **XGBoost** |
| --- | --- | --- | --- |
| **Fraud F1-Score** | 0.6982 | 0.9524 | **0.9624** |
| **Non-Fraud F1-Score** | 0.9695 | 0.9936 | **0.9947** |
| **Weighted Avg F1-Score** | 0.9372 | 0.9887 | **0.9908** |
| **Avg Precision** | 0.6204 | 0.9767 | **0.9957** |
| **ROC AUC** | 0.6768 | 0.9934 | **0.9995** |

The XGBoost model demonstrated superior performance across all metrics, particularly in identifying fraudulent transactions (the minority class) with high precision and recall. The Random Forest model also performed well, while Logistic Regression showed acceptable but significantly lower performance compared to the ensemble methods.



**4. Conclusions**

1. **Model Performance**: Ensemble methods (Random Forest and XGBoost) significantly outperform the linear model (Logistic Regression) for fraud detection tasks.
2. **XGBoost Superiority**: XGBoost demonstrated the best performance with a near-perfect ROC AUC score of 0.9995 and excellent F1-scores for both fraud (0.9624) and non-fraud (0.9947) classes.
3. **Undersampling Effectiveness**: Random undersampling proved effective in addressing class imbalance, particularly when combined with algorithm-specific imbalance handling techniques.