Credit Card Fraud Detection Report

Indulekha K P

November 13, 2024

Abstract

This report presents a machine learning-based approach to detect fraudulent transactions in credit card data. We implemented and compared three classification models: Random Forest, Logistic Regression, and XGBoost. Each model was evaluated using various performance metrics, including accuracy, precision, recall, and F1-score. This report provides an overview of the methods used, data preprocessing steps, model implementations, results, and a final comparison to identify the best-performing model.

1 Introduction

Credit card fraud is a significant issue in the financial industry, leading to considerable financial losses for both consumers and institutions. This project aims to develop a machine learning-based solution to detect fraudulent credit card transactions. We use a dataset containing transactions made by European cardholders in September 2013, with a focus on identifying patterns associated with fraudulent activity.

2 Dataset Description

The dataset consists of 284,807 transactions, with 492 labeled as fraudulent. The features were anonymized for privacy, with 28 principal components labeled as V1 through V28. Additional features include the transaction amount, transaction time, and the class label (0 for non-fraud, 1 for fraud). Due to the imbalance in class distribution, specialized techniques were used to handle this challenge in model training.

2.1 Data Preprocessing

Before model training, we conducted data preprocessing, which included:

- Handling missing values and outliers
- Scaling numerical features for models sensitive to feature scale
- Balancing the dataset using oversampling techniques (such as SMOTE)

3 Models Implemented

In this project, we implemented three different classification algorithms: Random Forest, Logistic Regression, and XGBoost.

3.1 Random Forest

Random Forest is an ensemble learning method that combines multiple decision trees to improve accuracy and reduce overfitting. It is suitable for handling imbalanced datasets and provides insights into feature importance.

3.2 Logistic Regression

Logistic Regression is a simple and interpretable model used as a baseline in this project. It calculates probabilities for each class and is effective for binary classification problems, even in imbalanced datasets when class weights are adjusted.

3.3 XGBoost

XGBoost, or Extreme Gradient Boosting, is a powerful boosting algorithm known for its high accuracy and efficiency. It is particularly effective for structured/tabular data, such as the dataset used in this project.

4 Evaluation Metrics

Given the highly imbalanced nature of the dataset, we used the following evaluation metrics:

- **Accuracy:** Overall correctness of the model.
- **Precision:** Proportion of correctly identified fraud cases among all predicted fraud cases.
- **Recall:** Proportion of correctly identified fraud cases among all actual fraud cases.
- **F1-Score:** Harmonic mean of precision and recall, balancing both metrics.

5 Results

The following table summarizes the performance of each model on the test set:

Model	Accuracy	Precision	Recall	F1-Score
Random Forest	99.98%	0.96	0.89	0.92
Logistic Regression	97.64%	0.06	0.92	0.12
XGBoost	99.99%	0.98	0.90	0.94

Table 1: Performance Comparison of Models

6 Conclusion

Among the three models tested, **XGBoost** performed the best overall in terms of precision, recall, and F1-score. This model is recommended for real-world deployment due to its high accuracy and ability to correctly identify fraudulent transactions while minimizing false positives. Future work could include exploring more advanced balancing techniques or implementing additional models, such as neural networks, to further improve performance.

7 Future Work

Potential improvements to this project include:

- Further hyperparameter tuning of the models to enhance performance.
- Experimenting with additional balancing techniques, such as undersampling or using synthetic data.
- Implementing more complex models, such as deep learning algorithms, to explore if they offer any performance gains.