Fraud Detection in Credit Card Transactions – Project Report

Norbu Gyeltshen – norbugyeltshen2003@gmail.com

Introduction

Credit card fraud is a significant risk to both consumers and financial institutions. It is essential to detect these fraudulent activities as soon as possible to avoid financial losses. This project focuses on developing a machine learning-based fraud detection system from an imbalanced credit card transaction dataset.

Abstract

In this project, anomaly detection and supervised learning are used to detect fraudulent transactions from a significantly imbalanced dataset. As only 0.172% of the transactions are fraudulent, we need to accurately classify the minority class and minimize false positives. We utilized preprocessing, data balancing, anomaly detection (Isolation Forest and Local Outlier Factor), and an XGBoost classifier. Model performance is measured through metrics like the Precision-Recall AUC and confusion matrix. The final model was deployed as a web application with Streamlit.

Tools Used

• Python – Main programming language

• Pandas, NumPy – Data handling

• Scikit-learn – Machine learning models and evaluation

• XGBoost – Gradient boosting classifier

• Matplotlib, Seaborn – Data visualization

• Streamlit – Web app deployment

• Joblib – Model serialization

Steps Involved

1. Data Loading and Exploration

- The dataset was loaded using Pandas. It contains anonymized credit card transactions with only 0.172% transactions classified as fraud (Class = 1).
- We saw feature distributions, observed class imbalance, and checked for missing values.

2. Data Preprocessing

- Features Amount and Time were scaled using StandardScaler to obtain all the features in the same range.
- Data was split into features (X) and target (y), and then split into train and test sets using train_test_split() with stratification to preserve class distribution.

3. Class Imbalance Handling

- We used SMOTE (Synthetic Minority Over-sampling Technique) to generate synthetic fraud cases in the training data to balance the classes for supervised learning, in order to overcome the extremely imbalanced nature of the data.

4. Anomaly Detection (Unsupervised Approach)

- Isolation Forest and Local Outlier Factor (LOF) were employed to detect anomalies in an unsupervised manner. These algorithms provide a score for each transaction of the likelihood of it being an outlier (potential fraud).
- Compared the performance of unsupervised methods with supervised methods for context.

5. Model Training (Supervised Learning)

- Trained an XGBoost Classifier on the resampled data. XGBoost is efficient in dealing with classification and is robust against overfitting.
- Tuned hyperparameters through cross-validation. Saved the trained model with joblib.

6. Threshold Tuning

- Rather than using a default threshold of 0.5, we used predict_proba() to get prediction probabilities and calculated the best threshold on F1 score to trade off precision and recall.

7. Model Evaluation

- Evaluated the model by:
  i. Confusion Matrix to view true/false positives/negatives
  ii. ROC Curve and AUC to assess true positive rate vs. false positive rate
  iii. Precision-Recall Curve, which is more insightful for imbalanced datasets
- Area under PR Curve (AUPRC) was used as the main metric due to imbalance.

8. Model Deployment with Streamlit

- The project was built with a friendly web application in Streamlit for real-time fraud detection.
- Included on the interface are:

i. CSV Upload facility where users can upload new data for transactions.
ii. Threshold slider that allows users to dynamically specify the probability threshold for classifying a transaction as fraudulent. This provides greater control based on business requirements (e.g., optimizing for precision rather than recall).
iii. The app then displays predictions and summary statistics.

- Model and scaler were loaded from saved .pkl files, and uploaded data was preprocessed before prediction.
- The app can be hosted on platforms like Streamlit Cloud, Heroku, or Render.

Conclusion

The project demonstrates an end-to-end pipeline for creating a credit card fraud detection system through machine learning methods. It includes preprocessing, balancing techniques (SMOTE), anomaly detection, supervised learning using XGBoost, and rigorous model testing using AUPRC and ROC curves. The Streamlit integration introduces interactivity where users can upload the data and tune the fraud detection threshold using a slider, providing real-world usability and interpretability. The approach gives a scalable and efficient way of carrying out real-time fraud detection.