DSC680 Project 1: Final Paper

Credit Card Fraud Detection Using Machine Learning

Pragathi Porawakara Arachchige

Bellevue University

DSC680-T301 Applied Data Science (2257-1)

Dr. Matthew Metzger

06/26/2025

Business Problem

Credit card fraud continues to be a major problem costing financial institutions billions of dollars yearly. Fraudulent transactions weaken consumer trust and regulatory compliance as well as cause direct financial losses. Typical systems sometimes miss rare and adaptive fraudulent trends, hence more advanced fraud detection tools are needed. Models based on machine learning provide the chance to find minute patterns in data and strike sensitivity with accuracy, hence resolving the drawbacks of conventional approaches.

Background/History

Cybercrime has grown more and more as digital payments and online shopping have become more common. Fraud detection initially depended on rule-based systems, which were stiff and easily circumvented by changing fraudulent methods. The emphasis changed over time to data driven methods, especially supervised machine learning, which can learn from past transaction data and modify to fit evolving behavior patterns. This project expands upon this change by using predictive modeling methods to identify fraudulent credit card transactions with the aid of a well-known Kaggle dataset.

Methods and Data Explanation

This project's dataset comes from Kaggle and comprises anonymized credit card transactions by European cardholders over a two-day period in 2013. Including 284,807 transactions, only 492 were labeled as fraudulent about 0.172% of the data. The dataset contains 30 numerical features: two are 'Amount' (the transaction value) and 'Time' (the elapsed time in

seconds since the first transaction); twenty-eight of them (V1 through V28) have undergone Principal Component Analysis (PCA). The target variable, 'Class,' denotes whether a transaction is authentic (0) or fraudulent (1). Data preparation included checking for missing values, deleting 1,081 duplicate records, normalizing the numerical features, and using SMOTE to solve the serious class imbalance.

Analysis

Exploratory data analysis (EDA) started the project to help one to grasp the data's distribution and linkages. Visualizations were created to show temporal patterns, transaction amounts, and class discrepancies. Two supervised learning algorithms were chosen for the research: logistic regression and the XGBoost classifier. These models were picked for their ability to manage organized data effectively and their interpretability. Performance was evaluated using metrics including the confusion matrix, precision, recall, F1score, and the ROC AUC curve. Since the dataset was imbalanced, SMOTE (Synthetic Minority Oversampling Technique) was used to improve the representation of fraudulent transactions in the training set.

Conclusion

With fraudulent transactions accounting for less than 0.2% of the total, the original dataset had an extreme class imbalance. Following SMOTE application, the minority class in the training set was balanced, therefore greatly improving model development. With a ROC AUC of around 0.97, the Logistic Regression model showed a good balance between precision and recall; however, the XGBoost model effectively captured more complicated patterns and outperformed

it with a ROC AUC of almost 0.99. Visualizations validating the study included ROC curves for both algorithms, time density plots, transaction amount histograms, and a class distribution bar graph.

Assumptions

The results of this work show that machine learning models, especially XGBoost, are able to accurately identify suspicious transactions in highly imbalanced datasets. Although logistic regression is perfect for detecting nonlinear relationships and sophisticated fraudulent behavior as well as offers a good baseline given its simplicity and clarity of understanding, ensemble methods such XGBoost are more appropriate. SMOTE resampling was essential in boosting recall, hence increasing the probability of identifying actual fraud without too many false positives.

Limitations

The research presupposes that even without the original feature context, the PCA transformed characteristics preserve enough signal to detect fraud. It also presumes that the transaction patterns seen during the two-day data collecting period mirror more general temporal trends. Finally, it presupposes that future or almost-real-time situations will generalize well with model performance on historical data.

Challenges

One major constraint is the PCA characteristics' lack of interpretability, which limits the understanding of the underlying reasons for fraud predictions. Furthermore, the model's exposure to long-term seasonal trends is restricted by the dataset's representation of only

a brief, two-day period. Lastly, synthetic samples created by SMOTE might add noise, especially if the fake specimens don't correctly represent actual fraudulent activity.

Future Uses/Additional Applications

Handling the severe class disparity was the biggest difficulty. The model needed to be specifically calibrated to prevent bias in favor of the dominant group. Particularly with strong classifiers like XGBoost, which may memorize training data if not regularized, overfitting was another problem. Furthermore, making it challenging to understand feature significance, the anonymized PCA characteristics hamper stakeholder communication and model transparency. Realtime fraud detection systems used by financial organizations can be built upon this framework. Furthermore, it can be included in hybrid systems that combine machine learning predictions with rule-based checks for better accuracy. Moreover, by retraining the model on pertinent datasets, the method may be modified to identify additional types of fraud including insurance fraud or identity theft.

Recommendations

Financial organizations should think about using XGBoost-based models for fraud detection given their great accuracy and resilience based on the findings. In cases where model explainability is paramount, logistic regression can still be employed. Carefully apply SMOTE or other resampling methods to guarantee that models are verified on real-world distributions. To make model results more obvious to analysts, it is also advised to include SHAP or LIME, among other explanatory tools.

Implementation Plan

Starting with daily cleansing and standardization of the transaction data, the suggested implementation plan calls for preprocessing. The trained XGBoost model can then be used in batch or streaming pipelines to rate incoming transactions. Based on precision-recall tradeoffs and ROC analysis, decision thresholds should be set. Regular monitoring of the model's performance should be performed together with quarterly retraining on revised datasets. Furthermore, explainability outputs ought to be incorporated in the warnings sent to fraud analysts to support decision-making.

Ethical Assessment

Several ethical questions arise from the installation of predictive fraud detection technologies. High false positive rates can unfairly flag genuine consumers, leading to service denials or reputational damage. Guaranteeing model transparency is challenging because the PCA-transformed characteristics conceal the underlying transaction characteristics. SHAP among other techniques can help explain model decisions to solve this. Finally, actual systems should abide by privacy laws including GDPR, therefore stressing safe data management and user agreement.

The appendix contains Python code snippets for exploratory data analysis, model training and evaluation, graphical diagrams like ROC curves and transaction histograms, and summary statistics of the dataset. It verifies the results shown and lays the technical groundwork for duplicating the research.

Visualizations

Class Distribution – Bar Chart of Fraud vs. Non-Fraud

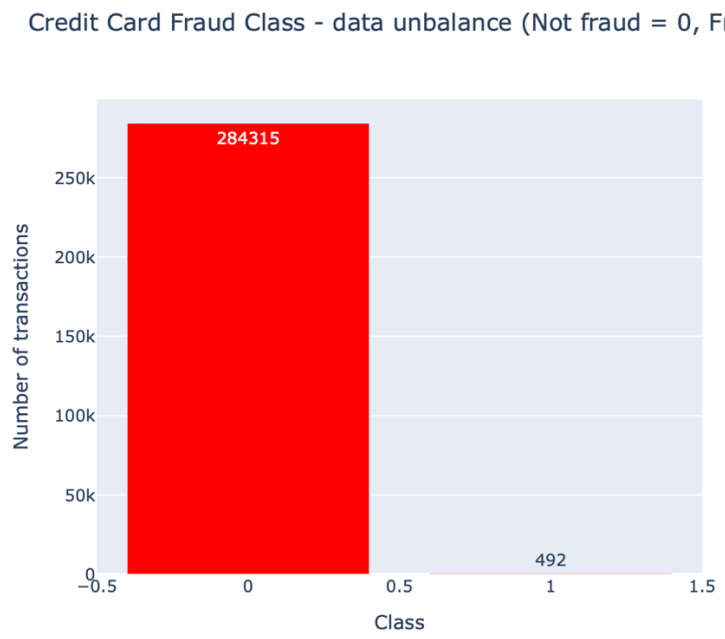

Figure 1. Distribution of fraudulent (Class = 1) vs. non-fraudulent (Class = 0) transactions.

Transaction Amount Distribution

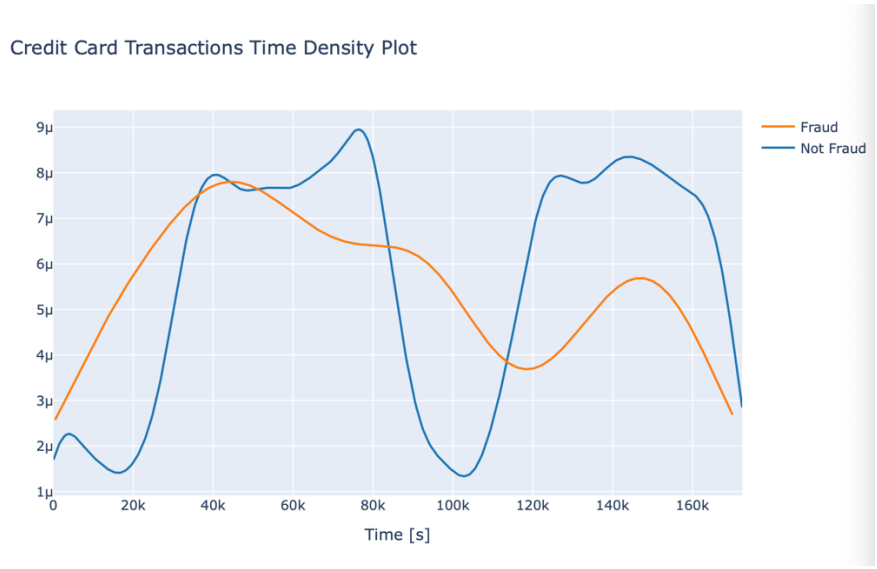

Figure 2. Distribution of transaction amounts for both fraud and non-fraud cases.

Time of Transaction vs. Fraud – Line/Density Plot

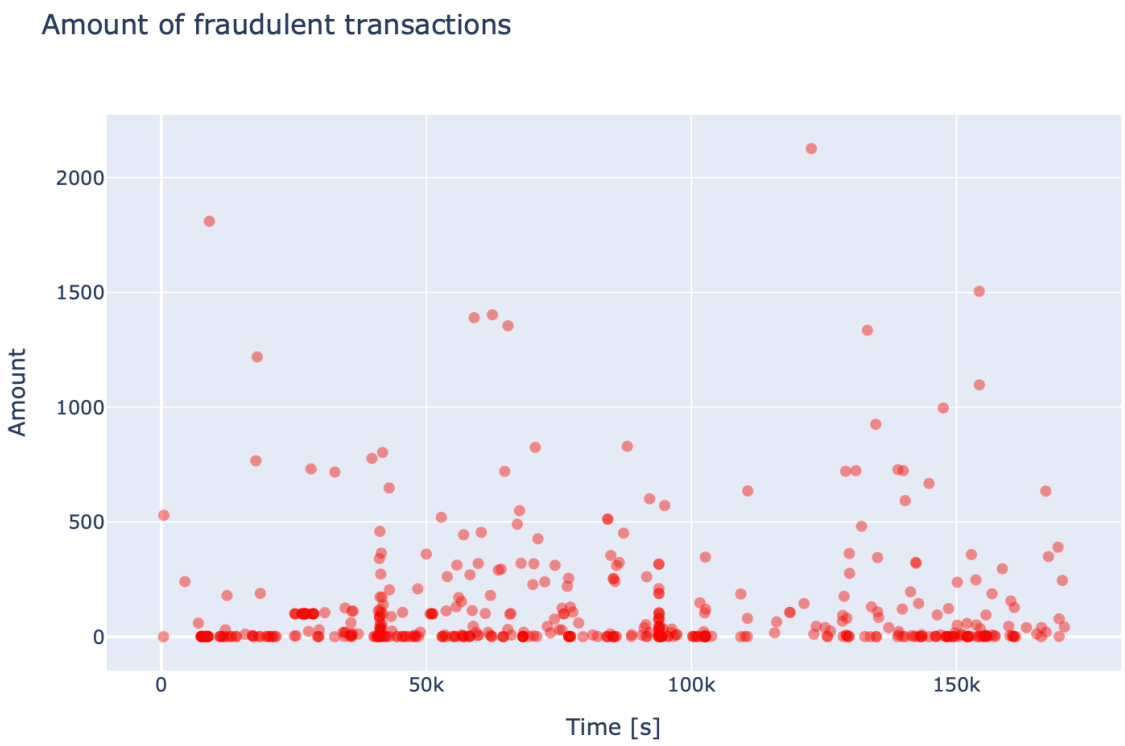

Figure 3. Time-of-day distribution of transactions

ROC Curve – Logistic Regression vs. XGBoost

Logistic Regression	**XGBoost**
Area under curve In [60]: `#Let's calculate ROC-AUC.` `roc_auc_score(test_df[target].values, preds)` Out[60]: `0.9698724983292885` The AUC score for the prediction of fresh data (test set) is **0.97**.	Area under curve In [40]: `#Let's calculate the ROC-AUC score` `roc_auc_score(valid_df[target].values, preds)` Out[40]: `0.8528641975628091` The ROC-AUC score obtained with RandomForrestClassifie

Figure 4. ROC Curve comparison between Logistic Regression and XGBoost.

Questions

1. Why was credit card fraud detection chosen as the focus of this project?

For both customers and companies, credit card theft is a high-stakes real-world issue. Financial institutions lose billions in yearly fraud losses as digital payments grow more common. Traditional rule-based systems have a small reach and cannot adjust to emerging fraud trends. Machine learning offers the ability to learn complex behaviors and improve fraud detection accuracy, making it an ideal solution for this problem.

2. What was the biggest challenge in working with this dataset?

The biggest challenge was the extreme class imbalance. Out of 284,807 transactions, only 492 were fraudulent (about 0.172%). This imbalance can cause models to ignore the minority class entirely unless addressed with techniques like SMOTE or cost-sensitive learning. In this project, SMOTE was used to balance the training set.

3. How did you handle the class imbalance in your data?

SMOTE (Synthetic Minority Over-sampling Technique) was applied after splitting the data into training and test sets. This technique generates synthetic fraud samples in feature space, allowing the model to train on balanced data without losing any legitimate transaction records. Post-SMOTE, the model could learn fraud patterns more effectively and improved recall without excessive false positives.

4. What types of models were used, and why?

Two models were used: Logistic Regression and XGBoost. Chosen for its straightforwardness and interpretability, logistic regression provides a starting point. XGBoost was selected for its capacity to manage challenging, nonlinear patterns in structured data. Additionally aiding early stopping and regularization, it helps to minimize overfitting and improve generalization.

5. Which model performed better, and how was performance measured?

XGBoost outperformed Logistic Regression, achieving an AUC of ~0.99 compared to Logistic Regression's ~0.97. Performance was measured using metrics including ROC AUC score, precision, recall, F1-score, and confusion matrix. The ROC curve especially highlighted XGBoost's superior ability to distinguish fraud from non-fraud.

6. How did you ensure the model was not overfitting?

Overfitting was managed through a combination of early stopping (in XGBoost), regularization parameters, and by validating on a separate test set not seen during training. SMOTE was only applied to the training data to prevent data leakage. Model performance was evaluated using AUC and F1-score to ensure generalizability.

7. What patterns were revealed through your exploratory data analysis (EDA)?

EDA showed that Fraudulent transactions are rare and often involve smaller to mid-sized amounts.

Fraud tends to cluster around specific times, possibly indicating automated attacks. Class distribution plots confirmed the imbalance. These patterns were visualized using histograms, density plots, and bar charts to help guide model choice and preprocessing.

8. How many duplicates were found and removed during data cleaning?

Identified and deleted were 1,081 duplicated records overall. Deleting duplicates guaranteed better training data and assisted to lower noise that may have hampered model performance.

9. How does SMOTE impact model training and fraud detection?

By producing fake examples of the minority class, SMOTE improves the model's capacity to identify fraud. This improves recall the speed at which genuine frauds are identified without greatly compromising accuracy. SMOTE, though, has to be utilized judiciously to prevent the presentation of improbable patterns.

10. Can this model be used in a real-time fraud detection system?

Yes. The implementation plan involves cleaning and standardizing transaction logs, applying the trained model in real-time or batch mode, and retraining quarterly. Model thresholds can be tuned to suit business risk tolerance, and tools like SHAP or LIME can help make outputs interpretable for analysts.

11. What ethical considerations were accounted for in this project?

Key ethical concerns include false positives that may unfairly block legitimate customers. Lack of transparency due to PCA-transformed features. Privacy regulations such as GDPR when deploying in production. To address these, the use of explainability tools and secure data handling protocols were recommended.

12. How can this fraud detection framework be adapted for future use?

The approach used here can be extended to other domains like insurance fraud, identity theft, or loan default prediction. With different labeled datasets and retrained models, the same workflow cleaning, feature scaling, resampling, modeling, and evaluation can be replicated for a wide variety of anomaly detection problems.

References

Kaggle Dataset: https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud/data

Sample Project Notebook: https://www.kaggle.com/code/gpreda/credit-card-fraud-detection-predictive-models/notebook

Dal Pozzolo et al. (2015). *Calibrating Probability with Undersampling for Unbalanced Classification*. IEEE Symposium.

Brownlee, J. (2020). *Imbalanced Classification with Python*. Machine Learning Mastery.

Kadam, D., Chiparikar, R. S., Kamble, M., & Attarde, M. H. (2024). *Machine Learning Approaches to Credit Card Fraud Detection. International Journal for Research in Applied Science and Engineering Technology*.https://doi.org/10.22214/ijraset.2024.60531