Machine Learning-Powered Fraud Detection in Credit Card Transactions

Erin Ma (em3910), Ruobing Zhang (rz2703), Yili Yu (yy3501), Manas Pange (mmp2248)

I. INTRODUCTION

In today's financial environment, the use of credit cards has become ubiquitous for both consumers and businesses. While credit cards provide a convenient payment method, they also pose a high risk of fraud which is a growing issue for both banks and customers. Identifying fraudulent transactions is a critical challenge for financial institutions as fraud detection plays a pivotal role in protecting consumers, ensuring the credibility of the financial system, and preventing significant financial losses.

Despite this, traditional methods that rely on fixed rules often miss new types of fraud or fraudulent activities are often hidden within massive volumes of transaction data. Thus, many companies are turning to machine learning to find patterns that imply suspicious activity. Our project will develop a machine learning model to detect fraudulent transactions using the Credit Card Transactions Dataset. By spotting suspicious activity early, we aim to improve the safety of financial transactions.

II. DATASET

The "Credit Card Transactions Dataset" was uploaded onto Kaggle by Priyam Choksi but was originally found on Hugging Face, a popular machine learning site. This data is in the form of a csv and provides a comprehensive collection of over 1.85 million records, including detailed information about transaction times, amounts, and associated personal and merchant details. This dataset allows for deep insights into transaction behaviors, including normal patterns and potential anomalies that could indicate fraud. Additionally, the dataset contains the column is_fraud which is the true label of whether such a transaction is fraudulent. Our goal would be to predict this target variable. The large dataset size and rich feature set provide an excellent opportunity to train sophisticated machine learning models for fraud detection.

III. PROPOSED MACHINE LEARNING TECHNIQUES

A. Data Pre-Processing

With over 1.85 million records and 23 columns, this dataset contains ample amounts of data to analyze. However, upon further observation, many aspects seem redundant. To start, we first dropped unnecessary columns such as the transaction number, the customer's first and last name, and their location information (such as street, city, and state).

Next, we split up our test sets into X and y portions by adding the is_fraud column to y and dropping that column from the rest of the dataset to create X. We then split our data into training and test sets with an 80/20 split. We opted for stratified splitting because the dataset is imbalanced. Additionally, we split this data before any more pre-processing was done so there was no data leakage of the test set into the training set.

For the categorical variables, we chose binary encoding for gender while doing target encoding for the others (merchant, category, and job) because of the fact that these categorical variables have a large range of answers. After dropping the highly correlated variables (see section C), we apply scaling to the columns amt, lat, and long.

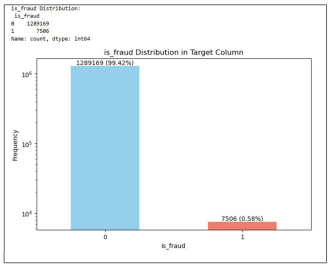

Fig. 1: Visualization

B. Feature Engineering and Visualization

Our dataset came with a lot of variables that included dates or times. To deal with these variables, we split each date/time column into many different columns. The first column we tackled was the 'trans_date_trans_time' column which represented the date and time the transaction occurred. We first converted this column to datetime format and then extracted components such as year, month, day, hour, minute, and second to new columns year, month, day, hour, minute, and second respectively. We then dropped the original variable

The second feature is we changed was the 'dob' variable which represents the date of birth of the user. For this variable, we turned dob into the columns year_born, month_born, day_born and then dropped the dob variable.

Some visualizations were made to look at the distribution of data. As seen in Fig. 1, the target column is largely imbalanced with only around 7500 rows being actual fraudulent transaction. This pattern remains the same when we analyze the distribution across different columns such as splitting between genders and splitting by job.

C. Correlation and Multicollinearity

As part of simplifying our data we used a correlation matrix to understand the relationships between variables. The correlation matrix showed many variables with high

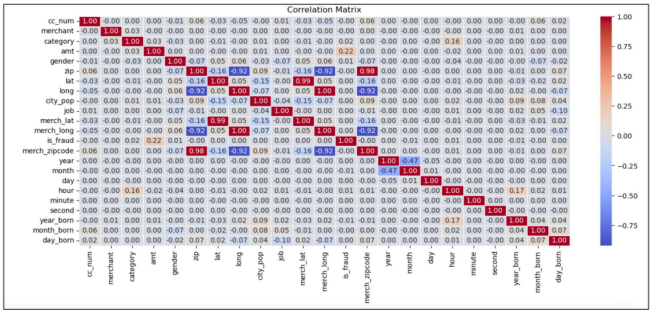

Fig. 2: Dataset

correlations (> 0.8) all of which pertained to locations. Through this, we were able to conclude that many credit card users performed transactions in-person. To avoid multicollinearity we dropped 'merch_zipcode', 'zip', 'merch_lat', 'merch_long'. 'merch_zipcode' was the only variable that had missing values. Although in initial discussions we suggested removing all rows where 'merch_zipcode' had missing data, it became apparent that removing this column would be a more efficient process.

D. Building Models

We started with a basic decision tree model with a max depth of 10 to counter overfitting. To counter the imbalance of data, we planned to use Random Oversampling, Random Undersampling, and Synthetic Minority Oversampling Technique (SMOTE) to either increase the minority class or decrease the majority class. We plan to implement and compare various machine learning models, including XGBoost, Logistic Regression, and Random Forest.

E. Performance Evaluation

1) Accuracy: Accuracy is one of the straightforward ways to test the performance of a model. We checked our accuracy scores with every model that we used.

2) AUC: We scored our models using Area Under the Curve (AUC) which demonstrates a model's ability to distinguish positive and negative classes.

3) Precision: Another metric we utilized was the average precision score to determine the rate of false classifications specifically False Positives.

IV. Observations

A. Model Comparision

The results of our analysis on the Credit Card Transactions dataset reveal interesting insights into the performance of various fraud detection techniques. The Vanilla Decision Tree Classifier did relatively well, achieving an AUC score 0.891 and an average precision of 0.744 on the test set. The Random Forest behaved similarly. Random Oversampling achieved excellent results with both an AUC score and Average Precision of 0.968, highlighting its effectiveness in addressing class imbalance by increasing minority class representation. Random Undersampling, though simpler, showed similar performance with an AUC of 0.966. On the other hand, SMOTE showed slightly decreased performance with an AUC of 0.93. Something to note for the techniques of Oversampling, Undersampling, and SMOTE is that our models yielded low precision rates when deploying our models to the test set. For example SMOTE had a precision of 0.213, Undersampling had a precision of 0.147 and Oversampling had a precision of 0.244. This means that our model seems to be classifying many negatives (non-fraudulent transactions) as False Positives.

Logistic Regression struggled in this highly imbalanced scenario, with an AUC score of 0.500 and Average Precision of only 0.006, emphasizing its limitations without additional balancing methods.Finally, XGBoost demonstrated the highest overall performance with an AUC score of 0.998 and an Average Precision of 0.992, indicating its ability to handle imbalanced data and complex patterns effectively.

B. Confusion Matrix Analysis

From the confusion matrix, Logistic Regression shows a disproportionately high number of true positives (257,834) and false negatives (1,501), which indicates potential overfitting since it seems that the model gave every transaction a negative label. Despite its high accuracy, the model's low AUC score and Average Precision highlight its inability to generalize effectively for fraud detection.

XGBoost, in contrast, demonstrates a more balanced confusion matrix, effectively capturing both positive and negative instances. While it has a slightly higher false positive count (21) compared to Logistic Regression, its overall performance metrics such as AUC (0.998) and Average Precision (0.952) reflect its superior ability to manage imbalanced data and detect fraud accurately.

Random Forest demonstrates exceptional performance in identifying the majority class (Class 0), as evidenced by its nearly perfect confusion matrix for this class. It achieves a minimal false positive count (20) and perfect metrics (Precision, Recall, and F1-Score of 1.00) for Class 0. However, it struggles with the minority class (Class 1), with a lower recall (0.74) and a higher false negative count (391). Despite this, its overall metrics, including an AUC score of 0.870 and an Average Precision of 0.728, highlight its strong discriminative power but reveal room for improvement in managing imbalanced data and enhancing minority class detection.

These results underscore the limitations of Logistic Regression in handling complex imbalanced datasets and the robustness of XGBoost in providing reliable and interpretable fraud detection outcomes.

V. Limitations

A. Methodological Limitation

One notable limitation is the presence of missing values and noise in certain features, such as location-based variables (longitude and latitude), which impact the overall quality of the dataset. Furthermore, the dataset mostly represents

Model	AUC Scores	Average Precision
Decision Tree	0.891	0.744
Random Oversampling	0.968	0.244
Random Undersampling	0.966	0.147
Logistic Regression	0.500	0.005
SMOTE	0.930	0.213
XGBoost	0.998	0.952
Random Forest	0.870	0.728

TABLE I: Observations

Model	TP	FP	TN	FN	Accuracy
Decision Tree	1174	62	257772	327	0.99
Random Oversampling	1430	4161	253673	71	0.98
Random Undersampling	1444	7998	249836	57	0.96
Logistic Regression	0	0	257834	1501	0.99
SMOTE	1316	4123	253711	185	0.98
XGBoost	1256	21	257813	245	0.99
Random Forest	1110	20	275814	391	0.99

TABLE II: Comparison of Confusion Matrices for Different Models

transactions from a single source, reducing its diversity and potentially overlooking significant fraud patterns that may occur in other contexts. To solve this issue, future work should aim to collect data from multiple sources.

B. Empirical Limitation

One major empirical limitation of this project is the imbalance in the dataset. The number of fraudulent transactions is significantly smaller compared to legitimate ones, which pushes the model to favor predicting non-fraudulent outcomes. Although techniques like SMOTE and oversampling were applied, these approaches may not fully resolve the issue.

In addition, fraud patterns change over time, but the dataset lacks enough variety over time to reflect these changes. These issues limit the model's flexibility, as it may have trouble identifying other types of fraud or performing well on datasets with different structures or patterns.

C. Ethical Limitation

Deploying the model in a real-world environment may raise privacy concerns due to the use of sensitive user data, such as address and transaction details. Furthermore, the model may unintentionally introduce biases against specific groups, such as gender, job, or location, which could result in unfair treatment of individuals by credit card companies.

To address these risks, privacy-preserving techniques, such as data anonymization or secure sharing, can be applied to future study to protect user privacy without compromising the model's effectiveness.

VI. Future Work

While the current models present promising results, there are several areas that future research could explore to address the limitations and further improve the model's performance.

One important area for improvement is enhancing the dataset's diversity and quality. Future work may involve collecting data from multiple sources, such as different industries, financial institutions, and even different countries. This could help uncover more common and universal fraud patterns that are applicable across various contexts. Addressing the issue of data imbalance remains a critical challenge. Advanced techniques like Class-Balanced Loss could be explored, which differ from methods like SMOTE and resampling by directly adjusting the model's loss function.

Another direction is exploring models beyond the scope of what we covered in this course, such as EasyEnsemble and time-series models like Long Short-Term Memory (LSTM) networks or Transformer-based architectures. These models have the potential to improve the detection of minority class instances and capture temporal dependencies in transaction data, making the model more effective at identifying evolving fraud patterns over time.

VII. Conclusion

The analysis of the Credit Card Transactions dataset highlights the significance of employing advanced techniques for fraud detection, particularly when dealing with highly imbalanced data. Through our experiments, it became clear that simpler models like Logistic Regression, while useful in some scenarios, are insufficient for accurately identifying fraudulent transactions in such datasets. The model's low AUC and Average Precision metrics, despite its high accuracy, indicate a tendency to overfit to the majority class, rendering it ineffective for fraud detection tasks.

On the other hand, approaches like SMOTE and Random Oversampling demonstrated considerable improvements by addressing the class imbalance issue. Among the methods evaluated, XGBoost emerged as the most robust and effective, achieving near-perfect AUC and Average Precision scores. This performance underscores its ability to capture complex patterns in the data while maintaining high sensitivity to both fraudulent and legitimate transactions.

Ultimately, this study underscores the importance of leveraging both pre-processing techniques, such as stratified sampling, target encoding, and scaling, and advanced modeling approaches to achieve reliable and interpretable results in fraud detection. The findings provide a roadmap for future work in developing and deploying efficient, scalable machine learning solutions to tackle fraud in real-world financial systems.

References

[1] Priyam Choksi. Credit Card Transactions Dataset. Kaggle. https://www.kaggle.com/datasets/priyamchoksi/credit-card-transactions-dataset/data