**Project 2: Handling Imbalanced Data to Detect Credit Card Fraud**

Christina Vosnak

Data Science, Bellevue University

DSC 680

Professor Iranitalab

May 4th, 2025

**Business Problem**

Credit card fraud is a major threat worldwide that impacts both institutions and consumers. Each year, billions of dollars are lost in fraudulent transactions which affect businesses’ profitability and consumer accounts. This project aims to develop a predictive model capable of identifying fraudulent transactions with high precision and recall, while minimizing the inconvenience of false positives.

**Background/History**

With the growth of online transactions, mobile banking, and digital payment platforms, financial fraud has become a growing global concern (Nguyen et al., 2022). Fraudsters are increasingly leveraging advanced technologies to exploit vulnerabilities in traditional security systems, making it more challenging for conventional rule-based methods to keep up. Static fraud prevention tools often fail to detect novel or subtle fraudulent patterns, especially as fraud tactics continuously evolve to bypass new safeguards (Bhattacharyya et al., 2011). In this context, machine learning has emerged as a powerful solution for enhancing fraud detection systems. Building a well-trained model that can learn from historical transaction data, identify hidden patterns, and detect suspicious behavior in real time will be helpful to mitigate fraudulent activities. By adapting to new forms of fraud, these models help reduce financial loss, minimize false alarms, and most importantly, increase consumer trust in digital financial systems (Sahin & Duman, 2011).

**Data Explanation (Data Prep/Data Dictionary/etc.)**

The dataset used for this project was sourced from Kaggle and contains credit card transactions made by European cardholders over a two-day period. It contains 284,807 transactions, with 492 labeled as fraudulent (a highly imbalanced dataset). Features are numerical results of Principal Component Analysis transformation, except for 'Time' and 'Amount'. The 'Class' variable indicates fraud (1) or non-fraud (0). To begin looking at the data, I checked for any missing values and found none. I also checked for duplicates and out of the 284,807 transactions, 1,081 transactions were considered duplicated. I removed these rows of data from the dataset. Next, I checked the balance of the dataset to find that 283,253 transactions were marked as non-fraud and only 473 transactions were marked as fraud and created a bar chart to display the class A blue rectangular object with red bars

AI-generated content may be incorrect.distribution.

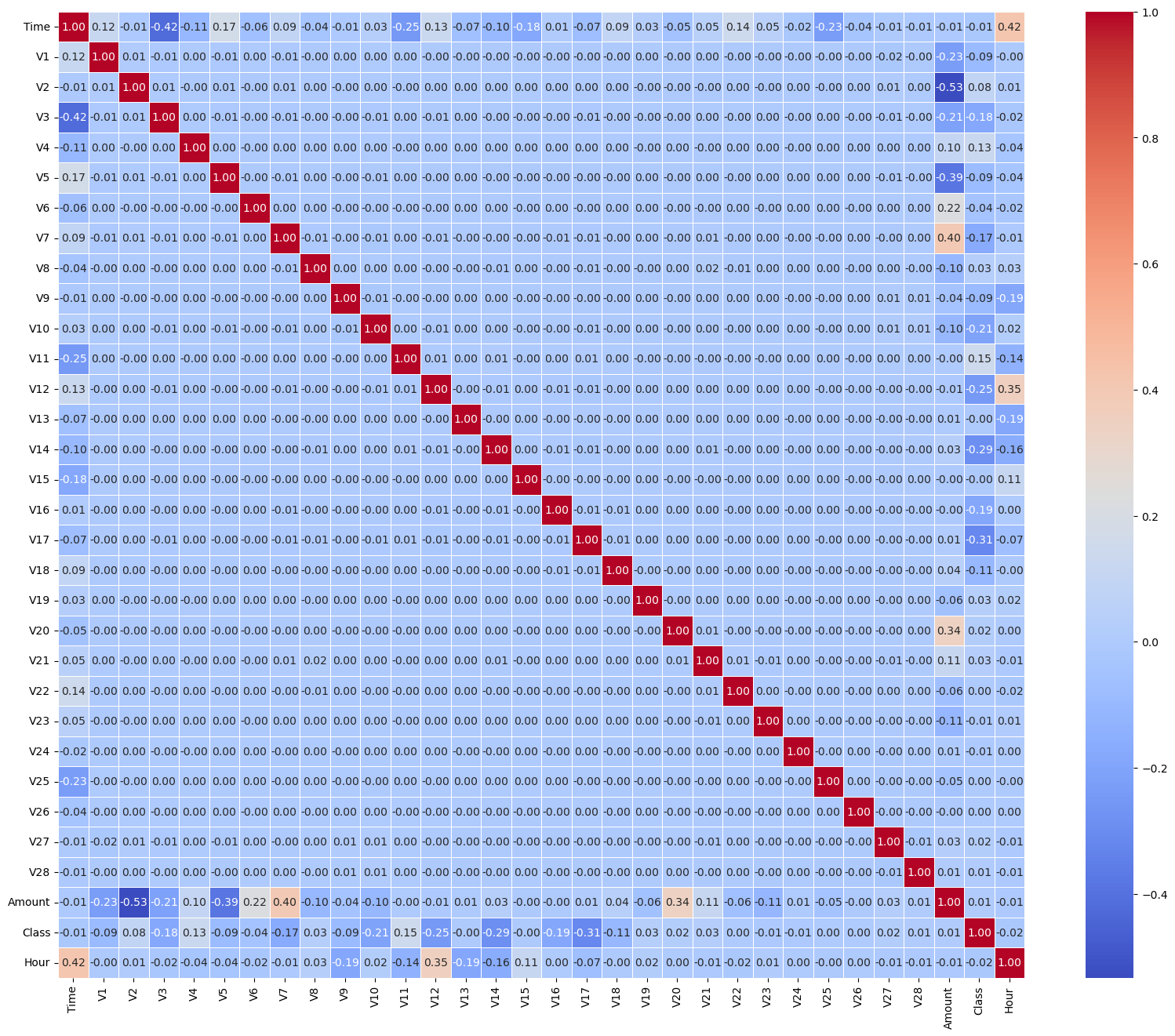
A graph with numbers and lines

AI-generated content may be incorrect.With the scale of the imbalance in mind, I created boxplots of the amount of transaction, separating fraud and non-fraud to get a preliminary understanding of the dataset. I found that the distribution of fraudulent transactions ranged between $0-$3,000 where anything over that number was considered non-fraud in the dataset.

A graph of different colored bars

AI-generated content may be incorrect. I also looked at time against fraudulent and non-fraudulent tractions and created two histograms to show the timing of the transactions. In the blue histogram, we see that most transactions occurred between 9AM-9PM with transactions slowing down after those after those times and increasing before those times. This makes sense as typically, most businesses are closed overnight, and individuals are sleeping during the late-night hours. However, the red chart shows the timing of fraudulent transactions and most notably shows that the most fraudulent transactions occur at 2AM and 11AM.

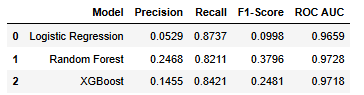
Finally, I created a correlation heatmap to see if any other inferences can be made about the data prior to model development. It is important to note that all of the variables except Time, Amount and Class have been transformed due to a Principal Component Analysis so even if distinct inferences can be made from a correlation heatmap, it will be impossible to trace back those inferences to specific values in the original dataset prior to PCA. The results of the correlation heatmap are below and while there are notable points to look at, nothing truly stands out as significant factors.



**Methods**

For this project, it is important to manage the imbalance within the data. For that reason, I will apply SMOTE to oversample the minority class (fraudulent transactions) within this set. This is a great technique to balance the dataset without losing valuable information. I will also test three predictive models to understand which may work best for this information. I will apply logistic regression, a random forest classifier and an XGBoost Classifier to the data and evaluate the results using several metrics. These metrics include Precision, Recall, F1-Score and AUC-ROC.

From the model results below, we see that Random Forest delivered the best overall performance, achieving the highest F1-Score (0.3796) and ROC AUC (0.9728), indicating a strong balance between precision and recall, and high discriminative ability. XGBoost also performed well, with slightly lower F1-Score (0.2481) and ROC AUC (0.9718), suggesting it is competitive but less balanced in handling false positives and false negatives. Logistic Regression, while showing excellent recall (0.8737), struggled with precision (0.0529), meaning it correctly identified many frauds but at the cost of a large number of false positives. Its low F1-Score (0.0998) reflects this imbalance. Overall, tree-based models significantly outperformed logistic regression in precision and F1-score, making them more suitable for this imbalanced fraud detection task.



Comparing the Precision-Recall (PR) curves of all three models—Logistic Regression, Random Forest, and XGBoost—XGBoost clearly demonstrates the most consistent and reliable performance. Its curve remains elevated for a longer stretch of recall, indicating strong precision across a broader range of fraud detection thresholds, and its Average Precision (AP) of 0.7736 surpasses both Logistic Regression (0.6779) and Random Forest (0.6564). While Logistic Regression achieves a high recall, its precision drops steeply as recall increases, suggesting a high rate of false positives. Random Forest shows slightly better balance than Logistic Regression but still exhibits a noticeable precision decline with higher recall. In contrast, XGBoost maintains high precision until the very end, making it the most stable and effective model in handling the precision-recall trade-off, especially critical in highly imbalanced fraud detection tasks.

A graph with a line

AI-generated content may be incorrect.

A graph with a line

AI-generated content may be incorrect.

A graph of a graph

AI-generated content may be incorrect.The confusion matrices for Logistic Regression, Random Forest, and XGBoost illustrate how each model balances fraud detection (true positives) and false alarms (false positives). Logistic Regression detects the most fraud cases (83 true positives) with only twelve missed (false negatives), but it does so at the cost of an extremely high number of false positives—1,486 non-fraud cases incorrectly flagged as fraud—which severely limits its precision. Random Forest achieves a more balanced outcome with seventy-eight true positives and only 238 false positives, reducing unnecessary alerts while still catching most fraud cases. XGBoost strikes a similar balance, correctly identifying eighty frauds with just 470 false positives. While Random Forest slightly outperforms XGBoost in reducing false positives, XGBoost edges out with fewer missed frauds (15 false negatives vs. 17). Overall, these confusion matrices show that while Logistic Regression maximizes recall, Random Forest and XGBoost offer a stronger trade-off between accuracy and alert burden, making them more practical for real-world fraud detection systems.

A screenshot of a computer screen

AI-generated content may be incorrect.

**Conclusion**

In conclusion, this fraud detection analysis compared three machine learning models—Logistic Regression, Random Forest, and XGBoost—on a highly imbalanced credit card dataset. While Logistic Regression achieved the highest recall, it struggled significantly with precision, resulting in a large number of false positives. Random Forest and XGBoost both offered a stronger balance between detecting fraud and minimizing false alarms. XGBoost demonstrated the highest average precision on the Precision-Recall curve, indicating it consistently maintained strong precision across various thresholds. Random Forest, on the other hand, delivered the highest F1-Score and the lowest number of false positives, showcasing its practical reliability. Ultimately, both tree-based models significantly outperformed Logistic Regression, with Random Forest being slightly more conservative and XGBoost offering a more aggressive but still controlled fraud detection strategy. Because these two models performed the best, Because these two models performed the best, I also found the top three features of each model. To reiterate, a Principal Component Analysis has been applied to the features within this dataset, so they cannot be specifically interpreted or tied to real-world variables. However, both Random Forest and XGBoost identified a consistently dominant component—**V14**—as the most influential feature in predicting fraudulent transactions. This suggests that whatever original variables contributed most to V14 hold critical predictive power for identifying fraud. The charts below highlight this consistency, showing that V14 had the highest feature importance score in both models, reinforcing its role as a key indicator in this task. Although the exact meaning of V14 remains obscured due to PCA, its impact underscores the value of feature importance analysis when model interpretability is limited.

A graph with blue rectangles

AI-generated content may be incorrect. A graph with red and white lines

AI-generated content may be incorrect.

**Assumptions**

It is assumed that the behaviors during the two-day window represented in this dataset are representative of a larger window of fraudulent patterns. It is also assumed that the data preprocessing of a principal component analysis preserved significant insights related to the feature structure of the dataset. However, a feature importance chart was still create to evaluate the results.

**Limitations**

The PCA limits this model’s capability to determine what factors are considered features within the data to understand what exactly can provide the most insight into when a transaction will flag as fraudulent. Since the features are anonymized, it is difficult to interpret the model for real-world investigative actions. As mentioned previously, it is also important to note that fraud tactics evolve as detection technologies evolve so this model will continuously have to evolve to keep up with changing tactics if used in a real-world environment.

**Challenges**

The most significant challenge in the project is managing the data imbalance where only .2% of the transactions within the dataset have been marked as fraudulent. While SMOTE helps to mitigate this challenge, it is important to note that other as data is introduced to the model over time, it may need to be adjusted. It is also challenging to interpret the results of this model given the anonymized features within the dataset.

Additionally, it is important to note that one of the primary challenges encountered during this project was the computational cost associated with training models on the full dataset. Due to the size of the credit card fraud dataset—containing 285,000 transactions—and the application of techniques like SMOTE to address class imbalance, training times became significantly prolonged, particularly for ensemble models such as Random Forest and XGBoost. As a result, the analysis and evaluation had to be conducted on a representative sample of the dataset to ensure timely experimentation and comparison across models. While this approach allowed for faster iterations and model tuning, it also introduced limitations in terms of generalizability and potential performance differences when applied to the full dataset. The reduced sample may not capture the full variance of transaction behavior, which could affect the robustness of the models in real-world deployment. Future iterations would benefit from running the models on the complete dataset using more optimized infrastructure or distributed computing resources to improve accuracy and scalability.

**Future Uses/Additional Applications**

While this model is not quite ready for real world deployment as it should be tested with a full dataset, financial institutions can use an tree based models with their specific data to build an adaptive version for their consumers to help predetermine fraudulent transactions before they process in customer accounts.

**Recommendations**

Financial institutions should implement fraud detection models similar to the Random Forest Classifier and XGBoost developed in this study, with a strong emphasis on maximizing recall to ensure that as many fraudulent transactions as possible are identified. Maintaining a continuously updated dataset of flagged and verified transactions will be essential to retrain and adapt the model over time as fraud patterns evolve. Furthermore, institutions should consider adopting a multi-layered detection system that integrates machine learning models with traditional rule-based methods to provide a comprehensive defense against both known and emerging types of fraud.

**Implementation Plan**

To implement this model, I’d recommend that financial institutions adopt the information from this analysis to their own data over a longer period of time and a full and complete dataset. I’d also recommend that they use raw data that includes feature names to better evaluate the results. Finally, I would recommend that institutions use data that spans at least a month and continue to update and evaluate the model over time as fraud techniques change.

**Ethical Assessment**

It is critical to ensure that any analysis that includes sensitive information protects the privacy of each transaction within the dataset. This ethical consideration has already been mitigated by the dataset developer who performed a Principal Component Analysis to remove any sensitive or personal information. This ensures that the data truly is anonymous and doesn’t create any bias against specific groups with demographic or geographical information.

**Audience Questions**

1. How did you address the severe class imbalance in the dataset?
   1. To address the imbalance, I applied SMOTE (Synthetic Minority Oversampling Technique) to the training data, which generates synthetic examples of the minority class (fraud) to balance the dataset. This helped the models learn patterns from both classes more effectively without discarding valuable non-fraud data.
2. How does undersampling or oversampling affect the model’s ability to generalize to real-world data?
   1. Oversampling, like SMOTE, may improve model sensitivity to the minority class but risks overfitting if synthetic data doesn't represent real-world variation. Undersampling reduces training time and class imbalance but may result in the loss of useful majority class information, which can hurt generalization. Both methods must be carefully validated to avoid misleading performance.
3. Why did you choose the models you used for this project?
   1. I selected Logistic Regression as a baseline for interpretability, Random Forest for its robustness and ability to manage non-linear relationships, and XGBoost for its strong performance on imbalanced and tabular datasets. These models provide a good mix of simplicity, power, and flexibility for fraud detection.
4. How did you evaluate the model performance given the imbalance within the data?
   1. I used metrics that focus on the minority class, including Precision, Recall, F1-Score, and Average Precision (AP) from the Precision-Recall curve. I also analyzed confusion matrices to assess how many frauds were correctly detected versus how many false positives were triggered.
5. What were the most important features driving fraud detections according to your model results?
   1. Although the features were anonymized through PCA, both Random Forest and XGBoost consistently identified V14 as the most influential feature in detecting fraud. This suggests that the components contributing to V14 strongly correlate with fraudulent activity in the dataset.
6. How would your model handle new types of fraud not represented in the training data as fraud techniques change over time?
   1. The model may struggle with entirely new fraud patterns not seen during training. To adapt, I would implement continuous model monitoring, retraining updated data periodically, and possibly include anomaly detection models or online learning approaches to capture emerging fraud behaviors.
7. What steps would you take to prevent model overfitting?
   1. For this analysis, I used SMOTE to balance the dataset which mitigated risks of overfitting due to the class imbalance presented in the original dataset.
8. How did you address ethical concerns when evaluating such sensitive information?
   1. Since the dataset was pre-anonymized and sourced for academic use, direct privacy risks were minimal. However, in a real-world dataset, I would recommend removing potential sensitive information prior to analysis to protect transaction identities.
9. If deployed, how often would you retrain this model?
   1. I would recommend retraining the model monthly or quarterly, depending on the transaction volume and observed shifts in fraud trends. More frequent retraining may be needed if fraud behaviors evolve rapidly or new types of attacks emerge.
10. If false positives become too frequent, how would you adjust the model or thresholds?
    1. I would first review the prediction threshold and increase it to favor precision. If that is insufficient, I would recalibrate the model, retrain with adjusted class weights or improved sampling, or consider post-model rule-based filtering to reduce unnecessary alerts without missing true frauds.

**Works Cited**

Bhattacharyya, S., Jha, S., Tharakunnel, K., & Westland, J. C. (2011). Data mining for credit card

fraud: A comparative study. Decision Support Systems, 50(3), 602–613. https://doi.org/10.1016/j.dss.2010.08.008

Nguyen, T. T., Ngo, T. D., & Le, H. T. (2022). An overview of machine learning in fraud detection.

Applied Sciences, 12(3), 1123. https://doi.org/10.3390/app12031123

Sahin, Y., & Duman, E. (2011). Detecting credit card fraud by ANN and logistic regression. Expert

Systems with Applications, 38(10), 13305–13310. https://doi.org/10.1016/j.eswa.2011.04.032