Exploring Multiple Methods of Detecting and Classifying Credit Card Fraud

Apurv Manjrekar, Samin Ashik, Kevin Han
May 3rd, 2024

1 Abstract

Attempting to identify fraudulent activity in the realm of financial transactions is becoming increasingly more important as online and contactless methods of payment are more prevalent. Utilizing a dataset that included European cardholders in a two day period in 2013, models were created in order to identify trends and patterns that surround credit card fraud. Initial findings indicated that Logistic Regression performed the poorest on the base data, while the Decision Tree Model performed well with the highest recall. After undergoing undersampling, the Gradient Boosting Model performed the best. Oversampling also led to improvements in recall scores, with Logistic Regression showing good parity with Gradient Boosting. After PCA reduction was applied to the Gradient Boosting Model, training times were able to be reduced without affecting recall scores. This sets up for future improvements where more advanced techniques can be employed to handle imbalance data. This will further enhance the precision and efficiency of credit card fraud detection models.
Code: Apurv-Manjrekar/Exploring-Multiple-Methods-of-Detecting-and-Classifying-Credit-Card-Fraud (github.com)

2 Introduction

2.1 Background

In the rapidly evolving digital economy, the rise of online and mobile payment technologies such as Near Field Communication (NFC) and Radio Frequency Identification (RFID) have dramatically transformed the way that transactions are conducted. While these technologies offer convenience, the vulnerabilities created by these technologies have led to increased credit card fraud. This heavily affects both large businesses and consumers globally.

If left unchecked, this vulnerability can lead to a loss of trust in the integrity of these systems. This could slow down advancement of these technologies as well as potentially reverse some of the widespread adoption of these technologies. To mitigate these risks we propose a model that can effectively identify and notify fraudulent activities. This would help slow down fraud as well as give insight into the patterns that surround fraudulent transactions. This would allow businesses as well as individuals to safeguard their transactions.

2.2 Motivation

Credit card fraud is steadily rising as the digital era continues to grow. Since 2015, the total amount of loss accrued by credit card fraud has almost quintupled. This number seems to be steadily growing, as shown by the various cases where groups of individuals are taking advantage of these vulnerabilities. Groups have been able to utilize new credit cards or stolen credit cards and run transactions of up to 12.7 million USD in just a mere three hours. When attacks like these occur, it is even more important to address these security risks.

As we delve into the specifics, it becomes clear that each attribute provides a piece of the puzzle in understanding and mitigating fraud risks. For instance, examining the transaction amounts might reveal thresholds beyond which fraudulent activities are more likely, thereby enabling preemptive measures. Given the extensive number of attributes associated with each transaction, it is imperative to identify which of these can predict fraudulent activities effectively. By understanding these attributes, it is possible to enhance the security mechanisms of payment systems, ensuring a safer environment for electronic transactions.

3 Methodology

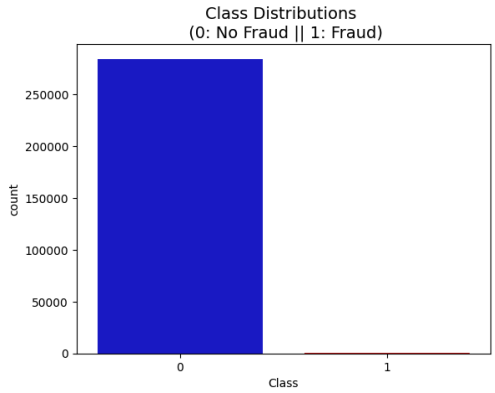
3.1 Dataset

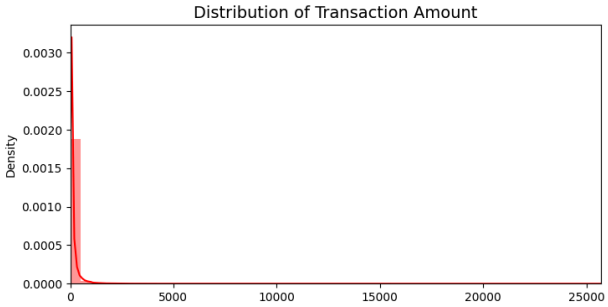
The dataset selected for this project uses credit card transactions from European card holders during a two day period in September 2013. The data was collected from the "Credit Card Fraud Detection" dataset that is available on Kaggle. The transactions have been anonymized for confidentiality reasons. The majority of the dataset was originally prepared and transformed using Principal Component Analysis (PCA), except for the 'time' and 'amount' features.

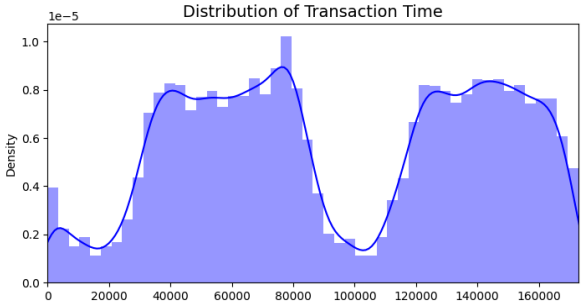
The dataset includes:

- Time: The seconds elapsed between each transaction.
- Amount: The transaction amount, which varies across different transactions.
- #V1 to #V28: Anonymized features resulting from a PCA transformation, representing various characteristics of the transaction that are not explicitly defined due to confidentiality.
- Class: A binary label indicating whether the transaction is fraudulent or genuine with 1 being fraudulent and 0 being genuine.

In total, the dataset contains 284,807 transactions, with 492 (approximately 0.172% of the total transactions) marked as fraudulent. Using this dataset allows for real-world simulations, which is crucial for developing a model for fraud detection. The first figure below helps better highlight the highly imbalanced nature of the dataset. The other figures help describe other characteristics of the dataset such as the distribution of transaction amounts and times.







The distribution of transaction amounts lets us know that there are very few outliers and most transactions lie around the smaller side. Additionally the distribution of transaction time allows us to see when transactions are most likely to occur. Our models will likely leverage these two characteristics in making decisions on whether a transaction should be classified as fraudulent or legitimate.

For this project, we focus on developing a model capable of detecting fraudulent activities among transactions by applying various techniques. Methods such as logistic regression, decision trees, random forest and gradient boosting, chosen for their ability to handle large datasets and complex features. The choice of models and their evaluation hinge on their ability to classify transactions accurately, especially since the dataset is so imbalanced.

3.2 Preprocessing

In our preprocessing process, we encountered variables such as ‘time’ and ‘amount’ that spanned a wide range of values. These values were scaled in order to censure that no single attribute dominated our model. To do this, a robust scaler was used as it is less sensitive to outliers than other methods of scaling. This scaler removed the median and scaled the data according to the quantile range.

Data splitting was then performed to validate the performance of our model. An 80/20 split was utilized, with 80% being used for training and the subsequent 20% for testing. This allowed the model’s performance to be trained on a large subset of data and then tested precisely on a smaller set.

Due to the imbalanced nature of the dataset, various sampling techniques were tested and compared to balance the classes. The RandomUnderSampler() function from imblearn was utilized to randomly select samples from the majority class to reduce its size to match the minority class. This

method also helped improve the performance of the classifier by reducing the influence of the majority class.

RandomOverSampler() was also used in order to avoid loss of important, but rare, patterns that might be overlooked from undersampling. This method randomly picks samples from the minority class, with replacement, to increase its size to match that of the majority class. Using these two techniques allowed the best to be used. Both have their tradeoffs, however, the method selected was undersampling in our Gradient Boosting model.

3.3 Models for Comparison

The nature of this problem is largely determined by the patterns and features within the financial transaction data. The dataset consists of features that are mainly numerical and derived from PCA, making the relationships between them complex and non-linear. The complexity further increases when considering all 28 PCA features alongside ‘Time’ and ‘Amount’. Due to the intricate and hidden relationships in the data, models capable of capturing non-linear patterns are necessary.

Logistic Regression will serve as a baseline model because of its straightforwardness and computational efficiency. Although it generally performs better with linear separable data, it's useful to compare how much better more complex models can handle the intricacies of our dataset. Additionally, it is possible that the key to fraud detection is simplicity and that the Logistic Regression Model performs extremely well.

Decision trees were notable in the setting for their ability to handle non-linear data. Even with the reduced feature set, decision trees could segment the space into “fraudulent” and “genuine” transactions based on the thresholds. The simplicity of decision trees becomes an advantage in a reduced-dimensional space because it mitigates the risk of overfitting, which is better when working with a higher number of features.

Random Forest is particularly well-suited for this dataset as it builds multiple decision trees and merges them together to get a more accurate and stable prediction. It does not require the dataset to be linearly separable, which is ideal given our PCA-transformed features. A point far from the norm in feature space might be closer to fraudulent transactions, and thus classified accordingly. Challenges might include the interpretation of such a complex model and the computational demands it requires.

Gradient Boosting Machines are also chosen for their ability to handle non-linear data effectively. By focusing on the errors of previous models, gradient boosting models can adaptively refine their predictions, making it robust against the varied and subtle fraud signals in the dataset.

The complexity of fraud detection in a high-dimensional space, influenced by the PCA transformation, makes it hard to predict which model will be most successful. While some fraudulent transactions may be easily visible, and promote simplistic models like Logistic Regression, others may hold more disguised patterns that require a stronger, more in-depth model, such as Gradient Boosting. Due to the ever evolving nature of credit card fraud and the possible number of factors that could

determine a fraudulent transaction, it is likely that a stronger Gradient Boosting Model will perform the best.

3.3 Dimensionality Reduction with PCA

The original dataset comprised 30 features, primarily results of a PCA transformation except for ‘Time’ and ‘Amount’. In the real world, utilizing a large number of dimensions is not practical. As such, after determining the best model, to further enhance model performance and reduce computational complexity, we computed a secondary PCA on the dataset.

The PCA implementation was reapplied to the already transformed datasets to reduce the dimensionality from 30 features to 10 principal components. This reduction was based on the analysis of explained variance, ensuring that the retained components still capture the majority of the information in the dataset. In our specific case, we chose to retain 95% of the variance as to maintain the majority of patterns.

After applying PCA, we trained our calculated best model on this reduced feature set. We hope that this will allow the model to focus on being as efficient as possible, while not losing any important information or patterns about the data. This will thus allow for an increase in the speed and a reduction in complexity. A key goal of the PCA is to not severely impact the accuracy of the model. After completing PCA on our best model, the reduction in information should not greatly affect the ability of the model to identify fraudulent transactions.

If we are able to perform PCA dimensionality reduction without hampering the accuracy of the model, then we should be able to say the PCA was a success. In this case we are able to reduce the number of dimensions and enhance the operational efficiency of the model, while maintaining a high level of fraud detection capability. The application of PCA in our fraud detection system can be a beneficial strategy for managing the trade offs between complexity, efficiency and performance. In such a sensitive topic such as credit card fraud, both efficiency and accuracy are crucial.

3.3 Evaluation Metrics.

For our project, selecting the right evaluation metrics is crucial for assessing the performance of our models. The metrics of concern are AUC-ROC, precision, recall and F1 score. Each model for calculating accuracy provides insights into different aspects of model performance which may or may not be suitable for our imbalanced dataset.

AUC-ROC is the area under the receiver operating characteristic curve. It essentially plots the true positive rate against the false positive rate. This is not ideal in an imbalanced dataset as the number of true positives will be extremely high due to the large number of legitimate cases.

Precision is the ratio of correctly predicted positive observations to the total predicted positives. While precision can be a strong predictor of accuracy, it is similarly not suitable for imbalanced datasets.

This is again because a large number of the transactions will be correctly identified as legitimate, giving a high precision score.

Recall is the ratio of correctly predicted positive observations to all in actual class. It represents the model's ability to catch all the relevant samples in a dataset. That is, the recall score in our case essentially checks how many actual fraudulent transactions were accurately predicted. For our purposes, this metric is the most critical metric because the cost associated with missing a fraudulent transaction can be very high. That is we are okay with some false positives, however we want minimal false negatives (the fraudulent transactions marked as legitimate) in our models as the consequences of each miss can be extremely high.

The F1 score is a weighted average of precision and recall. So, this score takes both false and true into account. It is particularly useful when the class distribution is even. In an imbalanced dataset such as ours, it is likely not that important.

Recall metrics stem from the nature of the fraud detection problem, where the consequences of failing to detect fraud are much more severe than incorrect detections. False positives might lead to further investigations, false negatives are worse due to the loss of an individual's finance as well as their loss of trust in their financial institutions. So, recall optimization is essential in this scenario to ensure that as many fraudulent transactions are captured by our models and as such will be the primary metric we will focus on.

4 Results

4.1 Initial Results (on Base Data)

The initial results of each model were promising and showed that further improvements in preprocessing and the model could be achieved. Below we can see the precision, recall, and F1 scores for each model. As stated earlier the key score is the recall score which tells us how many fraudulent transactions were correctly identified. The initial results showed that the Logistic Regression model had the lowest recall score of 0.582 while the Decision Tree Model had the highest recall score of 0.816.

Logistic Regression (without Undersampling or Oversampling)

	Test Set
Precision	0.864
Recall	0.582
F1 Score	0.695

Decision Tree (without Undersampling or Oversampling)

	Test Set
Precision	0.684
Recall	0.816
F1 Score	0.744

Random Forest (without Undersampling or Oversampling)

	Test Set
Precision	0.963
Recall	0.786
F1 Score	0.865

Gradient Boosting (without Undersampling or Oversampling)

	Test Set
Precision	0.738
Recall	0.602
F1 Score	0.663

4.2 Results (on Undersampled Data)

The results after applying undersampling on the dataset showed a massive improvement. Below are the precision, recall, and F1 scores for each model on the undersampled data. Here the Gradient Boosting Model had the highest recall rate of 0.939 while the other three models had a similar recall rate of around 0.929.

Logistic Regression (with Undersampling)

	Test Set
Precision	0.042
Recall	0.929
F1 Score	0.080

Decision Tree (with Undersampling)

	Test Set
Precision	0.015
Recall	0.929
F1 Score	0.030

Random Forest (with Undersampling)

	Test Set
Precision	0.058
Recall	0.929
F1 Score	0.110

Gradient Boosting (with Undersampling)

	Test Set
Precision	0.040
Recall	0.939
F1 Score	0.077

4.3 Results (on Oversampled Data)

The results after applying oversampling on the dataset showed a similar increase in recall scores. Here the Logistic Regression and Gradient Boosting models had a recall score of 0.918. Decision tree had the lowest recall score of 0.724.

Logistic Regression (with Oversampling)

	Test Set
Precision	0.063
Recall	0.918
F1 Score	0.117

Decision Tree (with Oversampling)

	Test Set
Precision	0.763
Recall	0.724
F1 Score	0.743

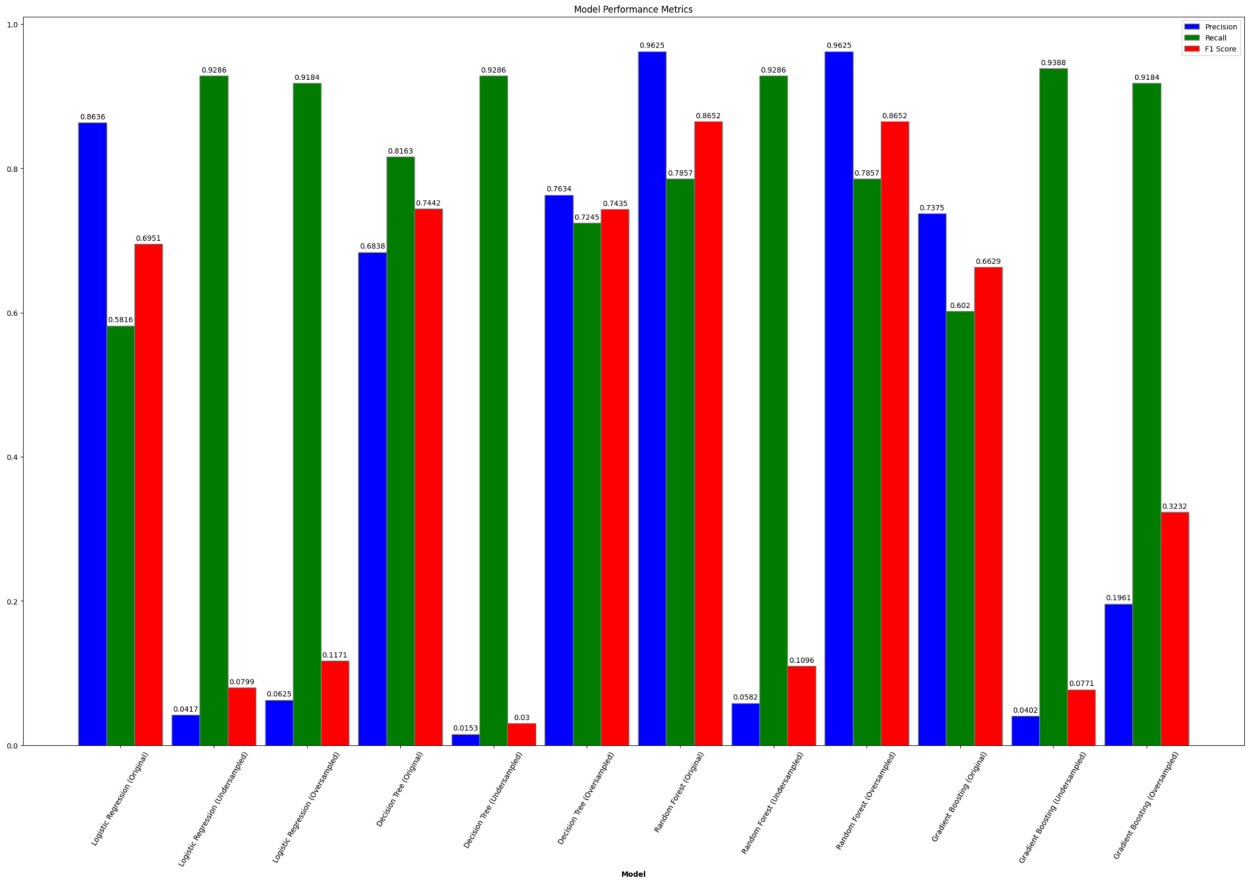
Random Forest (with Oversampling)

	Test Set
Precision	0.963
Recall	0.786
F1 Score	0.865

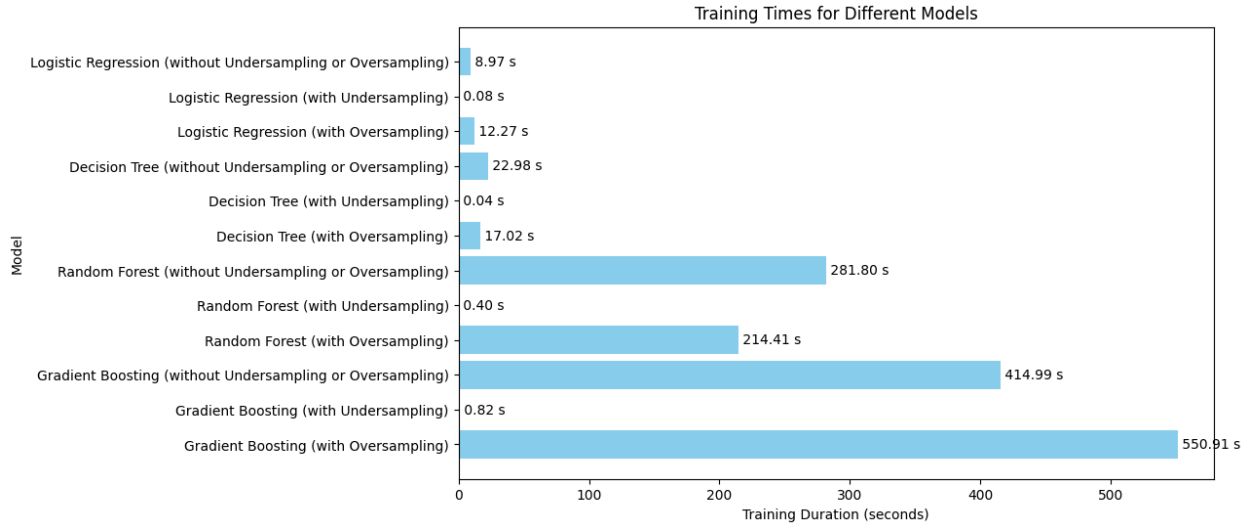
Gradient Boosting (with Oversampling)

	Test Set
Precision	0.196
Recall	0.918
F1 Score	0.323

4.4 Combined Results



The above bar graph shows the combined Precision, Recall, and F1 scores for each of the four models on all three data types. Here it is clear to see the significant improvement of the recall score on both the undersampled and oversampled data versus the base data. Gradient Boosting on undersampled data has the highest recall rate of all the models of 0.939.



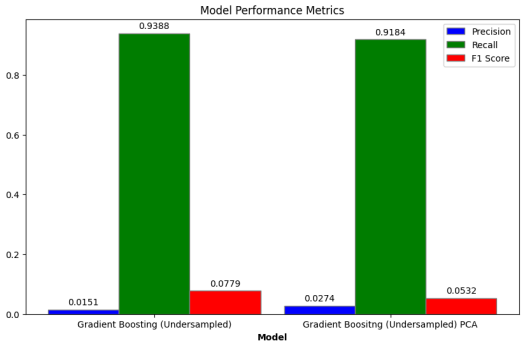
This chart shows the various training times for all the models. As expected there is a rise in time as the model complexity increases. While Gradient Boosting seems to outperform all other models it also takes the longest amount of time. It is also visible that training times for the oversampled dataset take a similar length of time to training times on the base dataset while undersampling seems significantly faster.

4.5 Results of PCA on Best Model

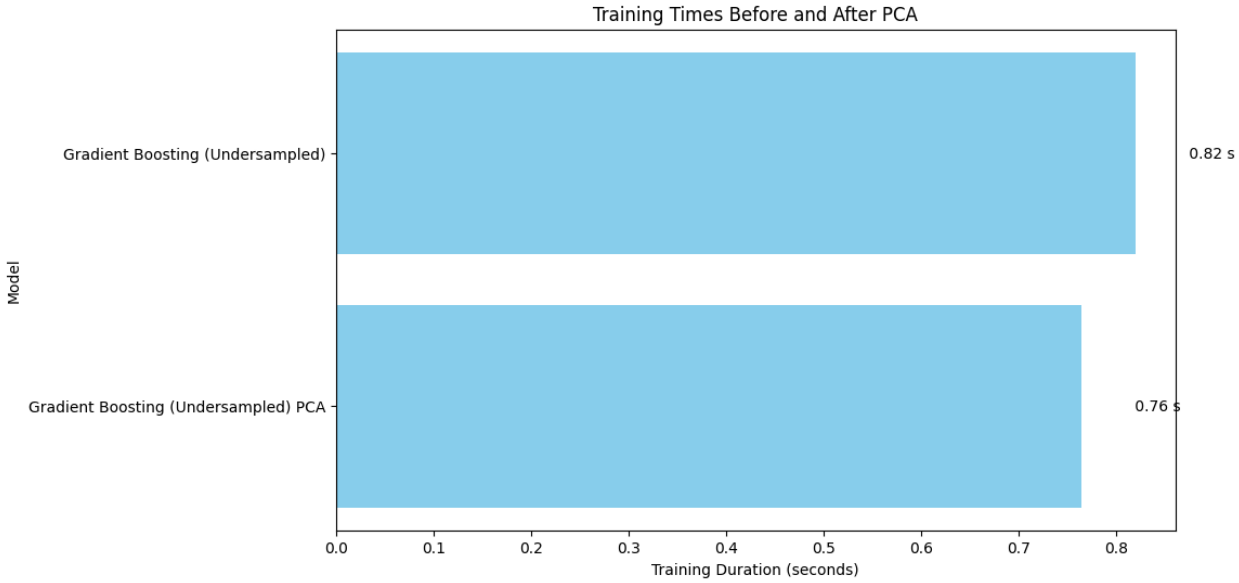
In order to reduce the training time of these models we performed PCA dimensionality reduction on the undersampled dataset. We hoped that this will help make our best model, Gradient Boosting (with Undersampling), train faster. The PCA dimensionality reduction helped reduce the number of dimensions from 30 to 10.

Gradient Boosting (with Undersampling) PCA

	Test Set
Precision	0.027
Recall	0.918
F1 Score	0.053



Above are the Precision, Recall, and F1 Scores of the Gradient Boosting model on the undersampled dataset after PCA as well as the comparison with the same scores before PCA. There is roughly a 0.0204 drop in recall score from before to after PCA.
Below we can see how the training times were affected by the PCA transformation. As we can see the PCA transformation did make the model slightly faster, going from 0.82 to 0.76 seconds.



Hence, the PCA transformation was able to successfully lower the training time of the model without greatly impacting its accuracy.

5 Discussion

5.1 Analysis of Results

The overall results were promising and gave a strong indication of the best method for classifying fraudulent transactions. The initial results on the base dataset were especially encouraging, with the Decision Tree Model boasting a recall score of 0.816. The other models ranged from 0.58 to 0.79. With all models having a starting recall score of greater than 0.5, we were optimistic about fine tuning our models to become perfect.

As initially predicted, Logistic Regression on the base set performed the worst with a recall score of 0.582. Similarly, it was also the fastest model throughout our experimentation. We were disappointed with Logistic Regression as we hoped simplicity could be key. However, the complexity of the problem and the imbalanced nature of the dataset made Logistic Regression similar to a random classifier.

After applying random undersampling on the dataset, we saw a sharp increase in recall scores. The Gradient Boosting Model now showed the greatest promise with a recall score of 0.939, a 0.337, increase. All the other models also showed a large increase, hovering around the 0.929 range. We found this jump quite surprising as any model over 0.90 is relatively strong.

Then we followed up by applying oversampling on the base dataset. Similar to the models on undersampled data, the models on oversampled data showed strong growth from their base models. Once again Gradient Boosting had the highest recall score in this category of 0.918. Interestingly, Logistic Regression had a similar recall score. The Decision Tree and Random Forest Models didn't do as well and hovered around the 0.7 to 0.8 range. Comparatively, undersampling seemed to outperform oversampling. This could be the result of a variety of reasons including the fact that undersampling removes noisy and redundant data. It is likely that the transactions classified as legitimate had a lot of redundancy and the large sample size was not necessary and as such reducing this sample size did not lead to the loss of significant information. On the other hand, oversampling the minority classes likely kept redundancy in the legitimate class and also added redundancy to the fraudulent class. The relative simplicity of undersampling seemed to have removed redundancy and imbalance within the train set.

Interestingly, it was earlier mentioned that Logistic Regression performed poorly on the base dataset. However, this was not the case on the undersampled and oversampled dataset with it boasting recall scores of 0.929 and 0.918. This indicates that the Logistic Regression model was not able to successfully identify patterns within the minority class which was overwhelmed by the noisy majority class. Once this imbalance was removed, the model's stability was improved and its bias towards the majority class reduced, allowing it to better gauge patterns that dictate a fraudulent transaction.

From the results thus far we were able to see a clear winner in Gradient Boosting which (on undersampled data) performed the best out of all models with a recall score of 0.939. One major drawback of Gradient Boosting is the inherently large time cost that comes with its heightened accuracy when compared to models such as Logistic Regression. In fact the Gradient Boosting Model was the

slowest out of all the models. For example on the base dataset, Gradient Boosting was roughly 46 times slower than Logistic Regression and for our best performing dataset, the undersampled dataset, Gradient Boosting was 10 times slower than Logistic Regression, 20 times slower than Decision Tree, and 2 times slower than Random Forest.

This heavy time cost of Gradient Boosting, our best model, showed a clear problem and as such we attempted to use PCA dimensionality reduction in order to reduce the number of dimensions within the dataset and improve the model's time. After doing PCA, we were able to reduce the dimensionality from 30 dimensions to only 10 dimensions. This did end up reducing the speed, going from 0.82 to 0.76 seconds. While this may seem insignificant, as the size of the dataset increases, this small change can become very apparent. We were initially concerned that losing a large number of dimensions would plummet our recall score. Fortunately, this was not the case and we only saw a drop of 0.0204 in the recall score going to 0.918. As we were able to improve the time cost and did not see a significant reduction in the recall score, we can conclude that the PCA was a success.

When scaling this model to a large-scale application that classifies fraudulent transactions, it is necessary that we focus on accuracy as well as time. These two features are given even greater prominence when taking into consideration the sensitivity and importance of the classification to people's daily lives. Analyzing these results it is clear undersampling is the best method when dealing with the detection of fraudulent transactions and an imbalanced dataset such as ours. Furthermore, it can be seen that Gradient Boosting (with undersampling) is the superior model for classifying credit card fraud, having the highest recall score. We were also able to successfully perform PCA on the dataset in order to reduce Gradient Boosting's biggest drawback, its time complexity, making it even more appeasing. Thus we are able to conclude that the strongest model for dealing with the detection and classification of credit card fraud is Gradient Boosting (on undersampled data).

5.2 Future Improvements

Credit card fraud is an ever evolving topic that continues to become more complex as technology advances in this new age. As such it is extremely important that a classification model dealing with such fraud is as accurate as possible. While an accuracy in the low 90s is no doubt the sign of a strong model, it is prudent we seek as close to perfect as possible.

An important way to improve our model could be by utilizing more powerful imbalanced data handling by using tools such as SMOTE, Synthetic Minority Oversampling Technique. This is an extremely useful oversampling technique which synthetically adds samples to the minority class. This ensures that the samples added are the average of the minority class allowing for greater precision. While Random Over Sampler randomly picks from the minority class, with replacement, SMOTE provides a method for more precisely generating the new dataset which could lead to a higher accuracy and recall score.

In this experiment, we only focused on four key models, Logistic Regression, Decision Tree, Random Forest and Gradient Boosting. It could be useful to look at other models such as Isolation Forests. Isolation Forest Models are specifically created for imbalanced datasets. They look for anomalies

in the data using binary trees. Considering the extreme imbalance present within our dataset and fraud in the real world, such a model created to target this imbalance could prove to be strong and more efficient.

Dealing with fraud in the real world, we do not practically have an abundance of time. As such the practical world demands an efficient solution that is not only accurate but also instantaneous. Additionally, with each individual having their own spending needs, it is not viable to use the same model for everyone. This poses another challenge in efficiently creating models that update in real time to align with an individual's unique spending habits. As such we would generally like to continue to find ways to improve our models' accuracies as well as their training times. Doing so will allow our models in the future to be as realistic and applicable as possible.

6 References

[1] "Oversampling-Undersampling for Imbalanced dataset," kaggle.com. https://www.kaggle.com/code/chickooo/oversampling-undersampling-for-imbalanced-dataset (accessed May 04, 2024).

[2] "Credit Fraud || Dealing with Imbalanced Datasets," kaggle.com. https://www.kaggle.com/code/janiobachmann/credit-fraud-dealing-with-imbalanced-datasets/notebook

[3] Kaggle, "Credit Card Fraud Detection," www.kaggle.com, 2018. https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud

[4] "CONSUMER SENTINEL NETWORK," Feb. 2023. https://www.ftc.gov/system/files/ftc_gov/pdf/CSN-Data-Book-2022.pdf

[5] J. Kiernan, "Credit Card Fraud Statistics for 2023," WalletHub, Mar. 04, 2024. https://wallethub.com/edu/cc/credit-card-fraud-statistics/25725

[6] "Special Report: Feds Bust $200 Million Credit Card Fraud Ring," Verifi. https://www.verifi.com/news-and-press-releases/special-report-feds-bust-200-million-credit-card-fraud-ring-2/

[7] J. McCurry, "100 thieves steal $13m in three hours from cash machines across Japan," The Guardian, May 23, 2016. Accessed: May 04, 2024. [Online]. Available: https://www.theguardian.com/world/2016/may/23/japan-cash-machine-100-thieves-steal-13m-dollars-three-hours

[8] IBM, "What is a Decision Tree | IBM," www.ibm.com, 2023. https://www.ibm.com/topics/decision-trees#:~:text=A%20decision%20tree%20is%20a

[9] A. Pant, "Introduction to Logistic Regression," Medium, Jan. 22, 2019. https://towardsdatascience.com/introduction-to-logistic-regression-66248243c148

[10] D. Gunay, "Random Forest," Medium, Sep. 14, 2023. https://medium.com/@denizgunay/random-forest-af5bde5d7e1e

[11] Hemashreekilari, "Understanding Gradient Boosting," Medium, Sep. 05, 2023. https://medium.com/@hemashreekilari9/understanding-gradient-boosting-632939b98764