Detecting Credit Card Fraud

Kishan Patel

December 4, 2020

1 Problem Statement

With more purchases being made through credit cards, there is an increased frequency of credit card fraud. With limited resources, credit card companies are not able to examine each transaction manually and have benefited from automated processes of screening transactions. Machine learning models have been developed to process this high volume of transaction data and identify suspicious charges in realtime, protecting consumers from fraud. One unique challenge in this application is the unbalanced number of fraudulent transactions; typically, the number of fraudulent charges are significantly lower than legitimate charges. Most algorithms are not designed to handle unbalanced datasets well as they have high prediction accuracy for the majority class but low accuracy for the minority class [1]. Thus traditional models fail to correctly identify the minority class of fraudulent transactions. Annother unique feature of classifying transactions is the varied costs of different classification error types. The cost of incorrectly classifying a transaction as legitimate is greater than the cost of a false alarm, or incorrectly classifying a transaction as a fraud.

The objective of this project is to build a series of supervised classifiers to predict whether a given credit card transaction is fraudulent or legitimate. The effectiveness of each classifier on the dataset will be assessed. A secondary objective is to see how these classifiers can be improved by mitigating the unbalanced class problem.

2 Data Source

The dataset summarizes credit transactions in September 2013 by European cardholders and can be obtained from Kaggle at the following website: `https://www.kaggle.com/mlg-ulb/creditcardfraud`. The dataset was curated by the Machine Learning Group of Universite Libre de Bruelles and summarizes 284,807 transctions that occurred over two days. Of the 284,807 transactions, 492 transactions, or 0.173%, are fraudulent which makes the dataset very unbalanced.

To preserve confidentiality, the dataset has already undergone dimensionality reduction into 28 numerical principal components via Principal Component Analysis. Untransformed features include the timestamp, the amount, and the class of the transaction. Each transaction's class is either 1 for fraud or 0 for not fraud. Thus the features include time, amount, and 28 numerical components which can hopefully be used to predict the class of the transaction.

Unfortunately, the PCA dimensionality reduction limits the type of modeling strategy that can be used because key characterisitcs are abstracted away into the principal components. For example, since the dataset does not explicitly define the credit card associated with the transaction, a single credit card user's spending habits cannot be utilized to identify an unusual charge as potentially fraudulent. However, it's possible that some patterns in user spending habits are present in the principal components and can still be utilized.

3 Methodology

3.1 Part 1: Basic Classifiers

The project consists of two parts. In part 1, common classifiers are trained on the training dataset to detect fraud using the scikit-learn library in python. The implemented supervised learning models include logistic regression, neural networks, and classification and regression tree (CART). These models assume that historical transactions are accurately classified and that there is an underlying pattern that can be used to predict whether a given transaction is fraudulent. The predictive power of the models is then determined and compared using the test set. In all instances, the train dataset consists of 80% of the obtained data with the remaining 20% forming the test dataset.

Performance of each model is assessed using confusion matrices, Receiver Operating Characteristic curves (ROC), and Precision-Recall Curves (PRC). Table 1 shows a basic confusion matrix with actual classifications along the rows and predicted classifications along the columns. In this context of fraud detection, the negative class (0) is a legitimate transaction and the positive class (1) is a fraudulent transaction. The key performance metric is accuracy, which is defined as $(TP+TN)/(TP+FN+FP+TN)$ or the percentage of records that were properly classified.

Table 1: Confusion Matrix

	Predicted Negative	Predicted Positive
Actual Negative	True Negative (TN)	False Positive (FP)
Actual Positive	False Negative (FN)	True Positive (TP)

For probabilistic models that use a threshold to classify test data, the ROC curve can be used to determine the model's predictive power. The ROC curve is a plot of true positive rate $(TP/(TP+FN))$ against false positive rate $(FP/(FP+TN))$ for various thresholds of classification. A successful model will produce a ROC curve with low false positive rate and high true positive rate. The area under the curve (AUC) can be used as a summary of the model with values closer to 1 having greater predictive power. ROC curves and AUCs will be computed for the probabilistic models. However, it is expected that for this unbalanced dataset, the AUC will be too optimistic.

For imbalanced datasets, the ROC curve can be misleading as it focuses more on the accuracy of classifying the majority class when often the interest is in classifying the minority class. The precision-recall curve (PRC) can be more useful in assessing the accuracy of classifying the minority class [3]. Precision and recall do not make use of the true negatives and are only concerned with the correct prediction of the minority class. Specifically, precision is a measure of the positive prediction power given by $TP/(TP+FP)$ while recall is another name for true positive rate $(TP/(TP+FN))$. The PRC is a plot of precision against recall with high-performing models producing curves with high values of both recall and precision. Measures associated with the PRC include the Area Under the Precision Recall Curve (AUPRC) and the F-measure, which is the harmonic mean of precision and recall and is computed as $F = 2 \cdot \dfrac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$. Better performing models will have larger values of AUPRC and F-measure. PRCs, AUPRCs, and F-measures are computed for the probabilistic models.

3.2 Part 2: Improving Classifiers

Based on the performance of these basic models, part 2 of the project involves exploring options to improve model performance. The key method involves a resampling process to handle the unbalanced issue. One resampling method that has proven useful is the Synthetic Minority Over-sampling Technique (SMOTE). This method over-samples the minority class in the dataset to increase its representation in the dataset and improve the accuracy

of traditional classifiers [2]. Over-sampling in SMOTE is achieved by creating synthetic examples in the feature space via interpolation. Within the subspace of fraudulent charges, for each fraudulent charge, k nearest neighbors are identified and synthetic charges are created along the line segments joining the current fradulent charge to its neighbors at random distances. In this way, extra fraudulent transactions are generated in the same region of the feature space as the existing fraudulent transactions. The SMOTE algorithm is implemented in python and used on the train dataset. The three classifiers are retrained on the modified train set using the feature space of all 28 principal components and the transaction amount. For further experimentation, SMOTE is combined with under-sampling the majority class to determine if the combination of under-sampling and over-sampling improves performance. Specifically, the legitimate transactions in the train set are randomly removed to reduce the class imbalance. It is expected the the combination of over-sampling the fraudulent transactions via SMOTE and under-sampling the legitimate charges will improve the models' predictive power on the test set.

Following this, the performance of the improved models from part 2 is compared with the performance of the basic classifiers from part 1 to determine the degree of improvement.

4 Evaluation and Final Results

Before implementing the two-part methodology, the data was visually inspected. Plotting pairs of principal components together revealed that classifying fraudulent charges is a difficult task. The figure below shows how there is substantial overlap between the fraudulent and legitimate classes with no clear boundary between them.

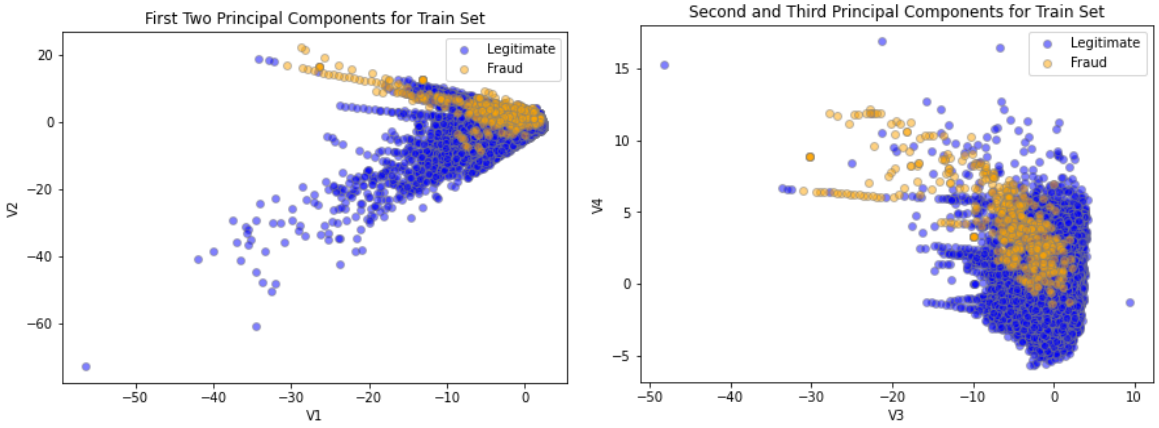

Figure 1: Scatter plot of first two (left) and second two (right) principal components.

After fitting the logistic regression, neural network, and decision tree models on the train dataset, the models were assessed on the test data using the standard 50% threshold to delineate between fraudulent and legitimate charges. For the logistic regression and neural network models, this meant that each test datapoint was assigned the class with the highest probability given by the model. For the decision tree, the test datapoint's class was determined by the majority vote in the terminal leaf. The models produced classifications on the test as shown in the confusion matrices below. The three models have similar reasonable performance with the decision tree producing the least misclassifications of 25 false negatives and 8 false positives.

Table 2: Confusion Matrices with Standard Thresholds

(a) Logistic Regression

		Predicted	
		0	1
Actual	0	56,865	10
	1	39	48

(b) Neural Network

		Predicted	
		0	1
Actual	0	56,864	11
	1	29	58

(c) Decision Tree

		Predicted	
		0	1
Actual	0	56,867	8
	1	25	62

The plots of each model's ROC curve and PRC can be seen in the Figure 2. The corresponding area under the curve is listed in the legend of each graph. The effect of the imbalanced dataset in assessing model performance can be seen when comparing these curves. As expected, the AUC is much higher with values over 0.91 compared to the AUPRC with values below 0.72. This shows how the traditional ROC is deceptively optimistic when used to predict the minority fradulent class. This pattern can also be seen when examining the accuracy and F-measure. Table 3 below shows how all three models show almost perfect accuracy with values over 0.999. This high score is due to the models' ability to correctly classify the majority class of legitimate charges. The F-measure gives a more realistic estimate of the models' ability to correctly classify the fraudulent charges. With values between 0.66 and 0.25, the F-measure shows that all three models are only moderately capable of classifying fraudulent charges.

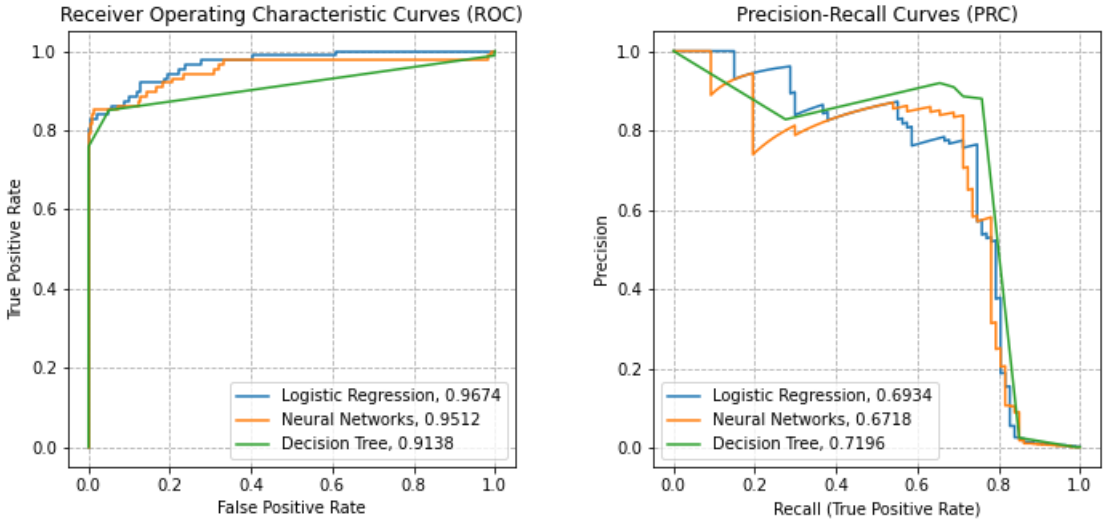

Figure 2: ROC (left) and PRC (right) for three models.

Table 3: Metrics of Model Performance

Metric	Logistic Regression	Neural Network	Decision Tree
Accuracy	0.9991	0.9993	0.9994
F-Measure	0.6621	0.7436	0.7196

In the second part of the methodology, the SMOTE algorithm with 5 nearest neighbors was used to resample

the train dataset by synthetically creating new records of the minority fraudulent charges. While the original train dataset has 227,440 legitimate charges and 405 fraudulent charges, the SMOTE resampled train set has 227,440 charges of each class, bringing the percentage of fraudulent charges from 0.17% to 50%. Figure 3 presents the synthetic fraudulent charges in the same region as the original fraudulent charges.

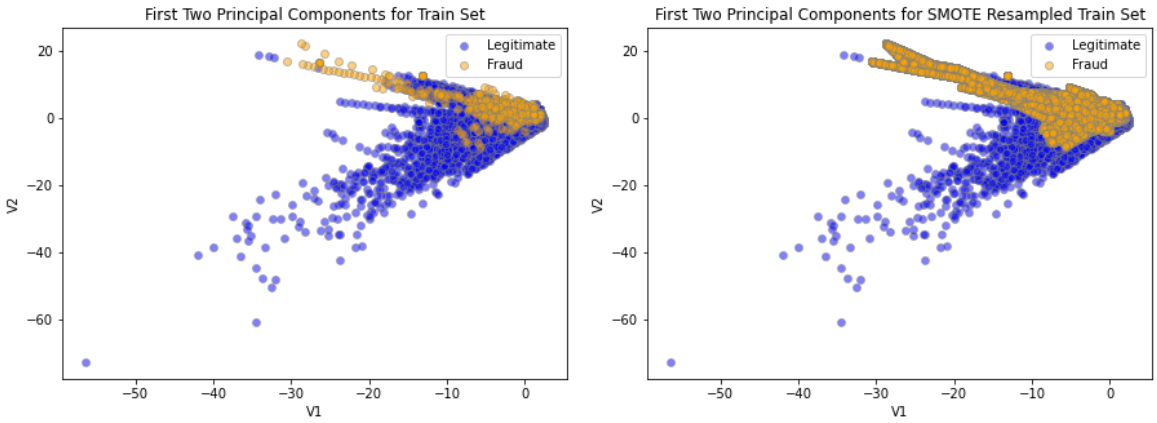

Figure 3: Scatter plot of first two principal components using original train set (left) and SMOTE resampled train set (right).

Unfortunately, since the original train set had a significant overlap between the two classes, the SMOTE resampled train set only extenuates the overlap with more fraudulent charges in the same region as legitimate charges. This led to the hypothesis that SMOTE would actually lower the models' predictive power for this particular dataset. Table 4 shows this to be true as each model fit on the SMOTE train set has a slightly lower accuracy on the test set and a significantly lower F-measure compared to the models fit on the original train set. In this case, the synthetic fraudulent charges made it more difficult for the models to classify fraudulent charges.

Table 4: Metrics of Model Performance with SMOTE

Metric	Logistic Regression	Neural Network	Decision Tree
Accuracy	0.9840	0.9973	0.9805
F-Measure	0.1416	0.4800	0.1177

A further attempt was made to rectify the imbalance issue by under-sampling the majority legitimate class in addition over-sampling the minority fraudulent class. The train set was modified in two steps: First, the fraudulent charges were resampled by using SMOTE so that fraudulent charges made up 10% of the train set. Second, the legitimate charges were resampled so that they only consisted of twice the fraudulent charges. This resulted in a modified train set with 45,488 legitimate charges and 22,744 fraudulent charges. The graphs of the first two principal components can be seen in Figure 4. Unfortunately, this worsened the visual boundary between the two classes as the legitimate charges with distinctively different features from fraudulent charges were reduced. In other words, this resampling technique made the overlap of both classes even worse by removing charges that were not part of the overlap. As expected, the performance of the models fit on this new train set degraded as seen in Table 5. In particular, the neural network F-measure fell from 0.48 with the SMOTE train set to 0.2287 with the SMOTE and under-sampling train set. Surprisingly, however, the logistic regression model produced a higher F-measure of 0.2517 with the SMOTE and under-sampling train set. This seems to indicate that the logistic regression model is most impacted by a reduction in imbalance.

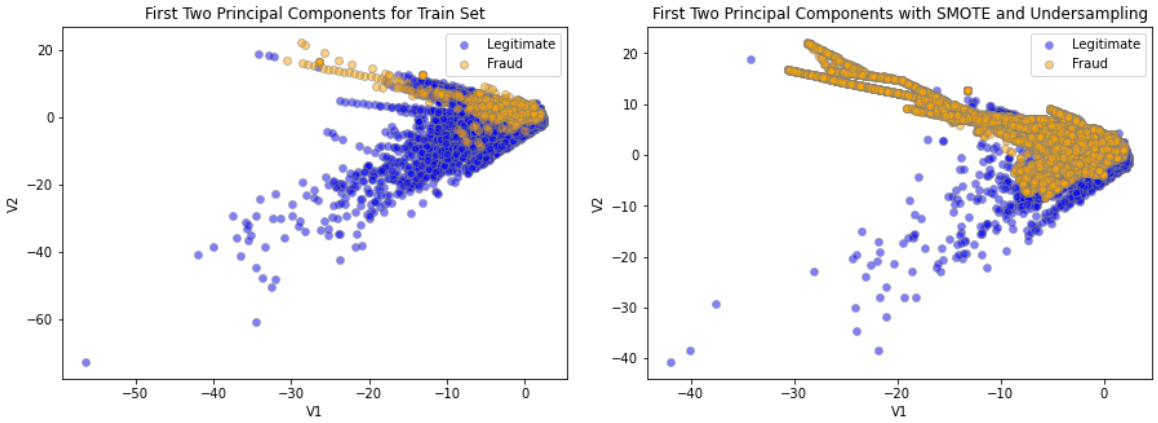

Figure 4: Scatter plot of first two principal components using original train set (left) and SMOTE resampled train set with under-sampling of majority class (right).

Table 5: Metrics of Model Performance with SMOTE and Under-Sampling

Metric	Logistic Regression	Neural Network	Decision Tree
Accuracy	0.9922	0.9911	0.9862
F-Measure	0.2517	0.2287	0.1548

Figure 5 shows a compilation of precision-recall curves for the models trained on the original train set, the SMOTE train set, and the SMOTE with under-sampling train set and assessed on the same test set. By the area under these curves, the logistic regression and neural network models have modestly greater predictive power when trained on the SMOTE train set instead of the original train set. However, these models are weaker when trained on the SMOTE with under-sampling set. Unexpectedly, the decision tree produces an abnormal PRC on both resampled train sets due to the penultimate threshold. Investigation revealed that the sklearn module computed a low precision value for the penultimate threshold due to a rounding issue.

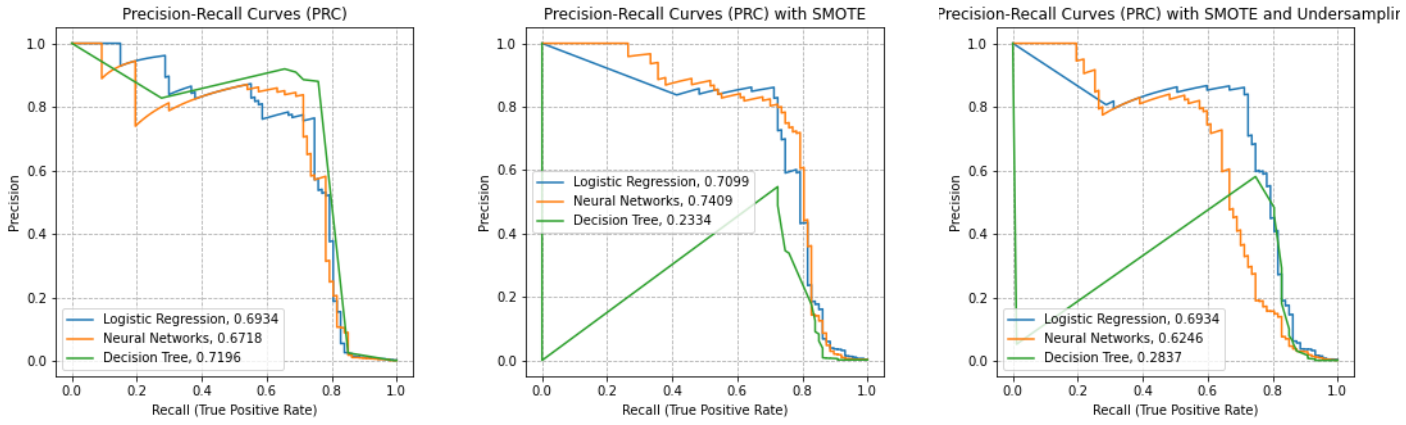

Figure 5: PRCs with standard train set (left), SMOTE train set (center), and SMOTE with under-sampling train set (right).

5 Conclusion

This project was an attempt to classify fraudulent charges among a dataset of credit card transactions using a series of classifiers and resampling methods. Due to the significant overlap between both classes, determining the decision boundary was difficult for all three classifiers. The overlap also limited some benefits of over-sampling the minority fraudulent class and under-sampling the majority legitimate class. While the SMOTE method did not provide significant improvements on this particular dataset, the method has proven successful on other datasets with more distinguishable decision boundaries [2]. Further exploration can include more hyperparameter tuning of the three models as well as the use of other models such as kernel SVM.

References

[1] Batista, G., Carvalho, A., and Monard, M. Applying one-sided selection to unbalanced datasets. *MICAI 2020: Advances in Artificial Intelligence* (2000), 315–325.

[2] Chawla, N., Bowyer, K., hall, L., and Kegelmeyer, W. Smote: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research 16* (2002), 321–357.

[3] Saito, T., and Rehmsmeier, M. The precision-recall plot is more informative than the roc plot when evaluating binary classifiers on imbalanced datasets. *PLoS ONE 10*, 3 (2015).