[[1]](#footnote-1)

Binary Classification: A Study on Credit Card Fraud

Katherine Yu

*Abstract*—Credit card fraud is a growing issue as digitalization makes it easier for criminals to illegally mine credit card information. Identifying fraudulent transaction correctly and effectively will benefit both consumers and credit card companies and other stakeholders. In this project I study the classification of credit card transactions of European consumer. Findings on a randomly under-sampled dataset indicate that both logistic regression and SVM both predict well with high accuracy scores. However, logistic regression ultimately is more efficient over SVM due to computational practicality and well as ease of threshold change.

# INTRODUCTION

Credit card fraud is an issue of increasing importance as our digitalized world makes it easier for fraudsters to steal credit information. With over 270,000 reports, credit card fraud was the most common type of identity theft last year and more than doubled from 2017 to 2019.[[2]](#footnote-2)

The interest in this data and problem was prompted by an unfortunate situation of my own where I lost my credit card and had multiple unknown items charged to the card. Fortunately, I was able to recover my card without paying any of the foreign charges, but it made me consider the issue of who was fronting the charges. If the credit card company and/or place of unauthorized purchase has to be responsible, then detecting these charges and notifying the card holder in a timely manner would be of utmost importance in order to recover at least some of these charges. Furthermore, in my situation I was notified only because one the foreign transactions had been over $100, a threshold I had set up previously. This threshold may have contributed to the notification of fraud I received from the credit card company (I’m not sure of the details of the company’s fraud detection policy). However, in fact there had been another foreign charge to my card that had been under $100 that I did not receive either notification for, which brings another issue of interest to light - the amount of the charge that fraud detection systems flag. It seems that usually large amounts of money are flagged easily but that these smaller amounts may not be detected in a timely enough manner. Therefore, in this project, in addition to exploring different machine learning techniques, I also wanted to examine the effect of the amount of transaction in determining fraud as well.

# Task Description

This project intends to study the attributes of credit card charges and identify charges that were not made legally. As a binary classification problem, logistic regression and SVM are good models for this problem. Due to the nature of the dataset and the privacy-protected variable names, this project focuses more on the actual categorization algorithm rather than what variables are contributing to credit card fraud or identify credit card fraud.

# Data

The data that I am using contains credit card transaction data from European consumers in September 2013 for the purposes of identifying fraudulent transactions. The dataset was collected for research purposes by the Worldline and the Machine Learning Group of ULB (Université Libre de Bruxelles). It is important to note that the data has already undergone PCA transformation in order to reduce the amount of variables. Currently, there are 31 variables total (after applying PCA), 28 of which are encrypted, or more precisely made anonymous, to protect the privacy of the consumers. The only variables not “encrypted” are time since the first transaction, the amount of the transaction, and the target variable/class. Amount and Time have not undergone PCA transformation.

There are no null values for any of the variables and the data doesn’t need much cleaning. All the variables are integers or floats. It is unclear whether some of these were previously string/categorical variables that had been assigned a numerical value. I decided to drop the Time variable as we don’t have user identification information, i.e. we don’t know if the transactions came from the same card or not, so Time isn’t very useful. I applied standardization to Amount since the other variables have already been standardized somewhat from the PCA transformation. Amount, which had not undergone PCA, had a very different scale from the other variables so I had to scale Amount or else the logistic regression would not converge.

# Major Challenges And Solutions

## Imbalanced Dataset

There are very little positive cases in this dataset. Positive cases, meaning the transaction was a fraud, were very rare. One possible solution is to create “fake” datasets that have a more balanced amount of each class. This can be done by hand manually for under-sampling, reducing one class to the number of instances of the other, or through SMOTE for oversampling, creating fake new points for the class with a lower number of instances based on distance, in Python. I used under-sampling as my method to balance the dataset as to reduce the computational load of SVM. There were 492 positive cases out of over 200,000 samples. After removing duplicates there were 473. Therefore, I randomly sampled 472 from class 0 and combined those 2 sets into a new under-sampled dataset.

## Outlier Detection

Detecting outliers was hard because there is not much economic interpretation due to the PCA transformation already implemented and not being able to know what the variables are. Without knowing the context, it’s quite risky to remove variables, even if they don’t have desirable distributions, since they might still be important to the classification problem. My solution was to take as few variables out as possible and it is likely the variable doesn’t have much additional information to offer. I started by creating at the histograms plot below.

## Diagram, engineering drawing Description automatically generated

From this we can see that some variables have really skewed distributions, like V8, V5, V7, V27, V28, and Amount. This shows Amount before it has been scaled. The scaling only affected the distribution a little bit and it still has many outliers but as Amount is a variable of interest, I chose to keep it in. Next to further look at the relationships within the data I created a heatmap using correlations. The one using the full dataset is shown below.

Chart

Description automatically generated

In the full dataset there is not much correlation between most of the variables, which is probably due to the imbalance in samples in each class as well. It does justify keeping Amount in however. Amount is the column right before Class (the column isn’t labeled because the labels wouldn’t have fit) but we can see that the line of squares just for Class does show slight correlation with some variables.

To solve this issue of barely any correlation I also did a heatmap of the under-sampled dataset as well. Shown below.

Chart

Description automatically generated

The correlation plot shows a large difference. The variables up to V20 have drastically changed correlation by changing the dataset. This implies that these variables are most likely affected by the different classes. We can still see though that the variables toward the end still don’t have much correlation. At this point I was leaning towards leaving out some of the variables beyond V20. As a last step, I also used the IQR range to calculate the number of outliers for each variable. The counts are shown below:

{'Time': 0, 'V1': 6948, 'V2': 13390, 'V3': 3306, 'V4': 11094, 'V5': 12221, 'V6': 22886, 'V7': 8839, 'V8': 23904, 'V9': 8199, 'V10': 9345, 'V11': 735, 'V12': 15282, 'V13': 3362, 'V14': 14060, 'V15': 2884, 'V16': 8180, 'V17': 7353, 'V18': 7468, 'V19': 10150, 'V20': 27553, 'V21': 14401, 'V22': 1298, 'V23': 18467, 'V24': 4758, 'V25': 5333, 'V26': 5665, 'V27': 38799, 'V28': 30094, 'Amount': 31685, 'Class': 473}

Looking the outlier counts for the IQR range, V27 and V28 have the most outliers out of all the variables. The other variables at the end don’t have nearly as much by a long shot, except for maybe V20. Since I wanted to be cautious about removing variables without knowing the context, V27 and V28 consistently showed not much relationship with the predictions or the other variables, while V20 still showed some relation to other variables. Ultimately, I chose to exclude V27 and V28. (And Time as well, since it isn’t a relevant variable).

# Logistic Regression

The first model I chose to examine was logistic regression. I ran a logistic regression first on the original data, using a 0.5/0.5 train and test subset split. After running the fit on the model, the accuracy score ends up being 1.0 (rounded) on the test set. Checking with cross validation yields scores of :

[0.99915413

0.99936559

0.99911888

0.9991541

0.99929508]

They are very close to 1.0. Since it’s not very informative to look at a confusion matrix for an imbalanced dataset, we turn to the ROC curve instead. Shown below.

Chart

Description automatically generated with low confidence

The ROC curve actually looks pretty good and has a high AUC score as well. Next we look at the precision recall curve. Shown below.

Chart, line chart

Description automatically generated

The precision recall curve isn’t as good as the ROC curve.

Since using an imbalanced dataset, I run the same type of model with the under-sampled data. After running the model, the accuracy of logistic regression classifier on test set is 0.93. It’s lower than the model using the whole dataset but it could also be a better, more general model since the underlying data isn’t unbalanced. The cross validation set yielded the following:

[0.92405063

0.93670886

0.94904459]

All of the cross validation sets yield an accuracy score of close to 0.93 as well. Since seeing if Amount contributes to the accuracy of the prediction of credit card fraud was one of my motivations in this project. I printed out the coefficients and their standard errors, as well as their p-values shown in the table further down below. Amount turned out to be significant in classifying fraud. Surprisingly, the coefficient was negative, which is indicating that the higher the amount is the less likely it is to be fraud. This could support what happened when I had my card used by someone else. In other words, it may imply that people usually charge lesser transactions since they are less likely to get caught right away. However, it’s hard to confirm with no other context in the data.

## Table Description automatically generated

The ROC curve for the under-sampled model is as follows:

A picture containing chart

Description automatically generated

It actually looks really similar to the previous ROC curve and has the same AUC as well, besides a little less smooth. Since the curve is in the upper left corner the model is performing well. Below is the precision recall curve as well.

Chart

Description automatically generated

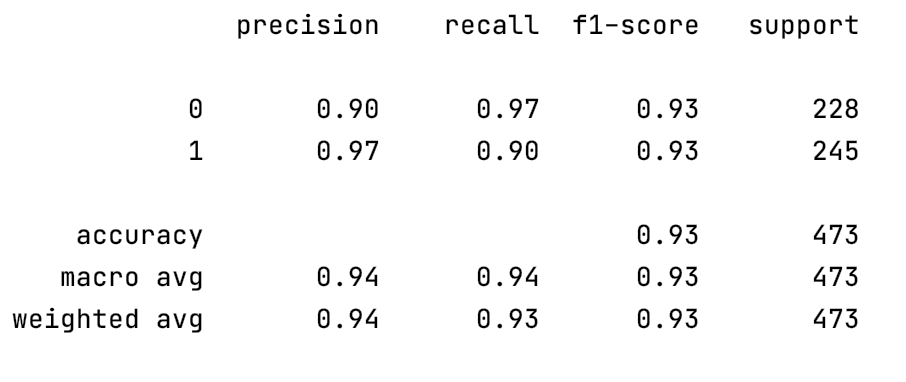
This precision recall looks better than the previous one over the whole dataset and also has a higher average precision metric as well. Since the curve is in the upper right hand corner, this means the model is performing well in terms of precision and recall.

The confusion matrix is:

Text

Description automatically generated

And the precision and recall values:



# SVM

SVM is another highly popular model used for classification problems. Since it has a different algorithm for classification for logistic regression, it could yield different results. I also only worked with the under-sampled data set as the computation using all the data and doing cross validation and fine-tuning the parameters was consuming too much time.

I chose to use an RBF kernel for my analysis because I think that with so many variables, a linear SVM model won’t work quite well as there is bound to be some nonlinearity in the data. There is a weight linear SVM option that would have been interesting to analyze, where one can specific the importance of different classes so it’s somewhat of a solution to an imbalanced dataset problem, but as mentioned the computational time would have taken too long if I had used the full dataset.

Using a grid search CV algorithm, the best parameters for C and gamma were found to be:

{'C': 10, 'gamma': 0.01}

Since we have a high C value that means that we allow a decent amount of errors in the model. This is alright since it helps to generalize the model, and seen below we still get a decent accuracy rate. Since we have a low gamma value, that means that we aren’t using much curvature in the model. This actually surprised me because of the amount of variables that are present in the data. However, I suspect that

The testing accuracy of the model using the above best parameters turned out to be 0.93, same as the logistic regression.

In comparison to logistic regression, it is known that SVM performs better when the classes are clearly separated while logistic regression performs better in the opposite situation. Since out models performed relatively the same, for this data the separation in the classes is probably normal or average.

The confusion matrix for the testing data is shown below:

Text

Description automatically generated with medium confidence

The precision and recall for each class are shown in the table below:

Table

Description automatically generated

The two models predicted very similar to each other. SVM only misclassified 1 sample differently from logistic regression. The sample is a false positive. Between false negatives and false positives, for this problem we will prefer false positives. It is most likely better to flag a credit card transaction as fraud than to have one missed. However, flagging too many false positives would also be bad as you would have to notify the consumers each time, and inundating consumers with transaction notifications might be annoying to the consumer. Below we try adjusting the threshold to see if the amount of false negatives will go down while keeping false positives in a reasonable range.

Implementing a threshold of 0.4 for logistic regression results in the following:

Text

Description automatically generated with medium confidenceTable

Description automatically generated

As seem above, having 0.4 as the threshold reduce the number of false negatives while keeping the same number of false positives. If we further reduce the threshold to 0.3:

Text

Description automatically generated with medium confidence

Table, calendar

Description automatically generated

We see that at 0.3 we start gaining for false positives but also lessen the number of false negatives by some as well. 0.3 is most likely a good threshold value for this dataset.

# Conclusion and Future Work

In conclusion, logistic regression and SVM are suitable classifiers for this problem. However, due to the nature of amount of data points are collected to analyze credit card transactions, logistic regression would be the more efficient choice. Training an SVM model on almost 300, 000 rows in this dataset was not feasible on a regular laptop. Of course, there are better resources at companies, but there is also more data as well. Additionally, it is quite simple to adjust the threshold in logistic regression as opposed to fine-tuning multiple parameters in SVM.

In terms of the dataset, I think it would be really interesting to find data on credit card transactions that have not yet undergone a PCA transformation in order to better understand the economic implication behind what the model tells us. In addition, this data was collected on European consumers, who may have different consumption patterns from American consumers or consumers from any other country. What makes this issue more complicated is that fraudsters can easily digitally mine credit card data from another country and some criminal hackers will also mine credit card data to sell to someone else. Ultimately it’s probably useful to get a dataset that captures all of the activity mentioned above.

Additionally in regards to the model, I would have liked to explore online learning using a neural network and the complete imbalanced dataset. Credit card fraud identification relies on timely detection in order to bring the most benefit to credit card companies. Credit card transactions are live data and the credit card companies database is probably being constantly updated; therefore, being able to construct a model that captures this requirement seems like an online learning model.

Overall, I find that logistic regression and SVM are good predictors of credit card fraud. However, logistic regression is preferred due to ease of computation and implementation.

References and Footnotes

## Data

The data file was very large even compressed so I am not able to upload it to CCLE. The link to the data is below:

<https://www.kaggle.com/mlg-ulb/creditcardfraud>

1. [↑](#footnote-ref-1)
2. The Motley Fool. https://www.fool.com/the-ascent/research/identity-theft-credit-card-fraud-statistics/#:~:text=Key%20findings,of%20identity%20theft%20in%202019.&text=Georgia%2C%20Nevada%2C%20and%20California%20were,doubled%20from%202017%20to%202019. [↑](#footnote-ref-2)