Identifying Credit Fraud With a Limited Number of Features

Kieran Hsieh
kth43@txstate.edu

ABSTRACT

One of the main problems with identifying Credit Card fraud using machine learning algorithms is the fact that data for this particular problem is hard to come by due to the privacy rights of credit card holders. This report analyzes how different models perform as they are given a steadily decreasing amount of features to work with in order to identify the model that performs the best with limited data.

Specifically, Logistic Regression, Adaboost, and Random Forest models are analyzed starting with the full number of features, then reduced to two features. These models are then scored by their F1 and Recall scores to determine the best model, which, in terms of an imbalanced data set like Credit Fraud, was determined to be Logistic Regression for most cases, and Adaboost for an extremely small amount of features.

Keywords

Precision, Recall, F1 Score, Logistic Regression, Adaboost, Random Forest, Credit Fraud, Imbalanced Data set

1. INTRODUCTION

Credit Card fraud has become an increasingly large problem as there are more card holders and online shopping becomes an increasingly important part of the economy. One successful method to identify fraud is to train models to identify instances of credit fraud by looking at credit card transactions. However, one of the main problems with this is that credit card data is hard to come by, and usually, the data contains only a few features that have not been transformed to accommodate privacy concerns. Previous work done on this subject has analyzed how different models preform with a full data set, and this report aims to identify which of those models (Logistic Regression, Adaboost, and Random Forest) preform best when the number of features is reduced.

The accuracy of each model was found by analyzing the Recall and F1 scores after they were trained on the data set. These accuracy scores showed that Logistic Regression was the best model to use in most cases, while Adaboost was better in extreme cases, where there were almost no features to use.

2. PROBLEM DESCRIPTION

The main problem working with the data set used [1] was that only the Time, Classification, and Amount were features that weren't changed using a PCA Transformation, while the rest of the features were labeled V1-V28 and the values were adjusted in order to protect the privacy of the card holders. In order to measure the performance of each model as features were reduced, the accuracy was recorded and plotted on a graph using the matplotlib package after they were fitted and predictions were made.

3. METHODOLOGY

In each iteration of the program, the features used were determined by a feature cutoff that removed the two features that were least correlated with the classification, then a constant weight was added to each sample in the full data set that was classified as fraud in order to increase recall accuracy and encourage the model to predict more false positives. After this, the data set was split into training and testing data, and each model was fitted using the same training data and starting weights. After each fit, model accuracy was determined by the recall and F1 scores, which were recorded directly after predictions were made. This process was repeated for 15 iterations, starting each model utilizing every feature, and cutting off two features every iteration until there were only two features being used by each model during the fitting process.

3.1 The Data Set

The data set was pulled from Kaggle.com [4], and contained a total of 31 different features including the classification. The first thing to note about this data set is that, other than Time and Amount, all other features underwent a PCA transformation where they were relabeled as V1-V28. While the transformed values were still visible, this made it difficult to know what you were looking at and limited the level of feature engineering you can introduce. To subvert this and value the importance of each feature, the feature importance was instead determined by its correlation with its classification as fraud or not fraud. This allows us to

[1]The Credit Card Fraud Detection data set from Kaggle.

rank each feature in terms of importance and remove them accordingly.

In addition to the transformation of many of the features in the data set, the overall classification of all of the samples is highly imbalanced.

As seen above, the data set is heavily biased towards cases of no fraud. This imbalance can lead to problems measuring accuracy the usual way, so in order to fix this, recall and f1 scores are used to get an idea of the performance of the model on this data set.

3.2 Determining the Correlation of features

A visual representation of the correlation of each feature with the sample's classification as fraud or not fraud can be seen in table 1. [2]

Table 1: Feature Correlation with Classification

Feature	Correlation
V11	0.154876
V4	0.133447
V2	0.091289
V21	0.040413
V19	0.034783
...	...
V16	-0.196539
V10	-0.216883
V12	-0.260593
V14	-0.302544
V17	-0.326481

In order to use this information to determine which features to cut, the list of features was sorted into ascending order, and in each iteration, the number of features would remove the two features most loosely correlated with classification. This means that the features towards the center of the list were removed, while the rest were used when feeding the data into the train and test splitter.

3.3 Tailoring Data Set

In order to promote an increased recall score, before splitting the data into training and testing sets, a weight of 10 was applied to all samples classified as fraud, and a weight of 1 was applied to the samples classified as not fraud. After this was applied to the data set, the set was then split into training and testing with the test set comprising of 30% of the full data set, and the training set comprising of 70%.

[2]An important note is that most features have a low correlation with the actual Classification, this aims to utilize the correlation that does exist to predict the end classification.

3.4 Evaluating model Performance

3.4.1 Recall

Recall is the preferred metric for measuring model accuracy with an imbalanced data set. This is because the usual method of determining accuracy for a binary classification problem with:

$$Accuracy = \frac{CorrectPredictions}{TotalPredictions} \tag{1}$$

ends up being completely wrong, since even if the model wasn't trained and guessed every sample as not fraud, the the accuracy would still be extremely high and not representative of the model's actual accuracy. Instead, by using recall, which is calculated with:

$$Recall = \frac{TruePositive}{TruePositive + FalseNegative} \tag{2}$$

you are able to take into account how accurate you were getting the classification correct even if the data set is completely imbalanced. This is why weights were added onto samples that were classified as fraud, since it increased the recall accuracy at the expense of precision. Doing this was good for this data set only because finding as many instances of credit fraud as possible is better, since identifying a possible case of credit fraud is more important than getting every prediction correct.

3.4.2 F1 Score

On the other hand, F1 Score is a different metric that takes into account both precision and recall as seen by how you calculate it with:

$$F1 = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \tag{3}$$

The reason this is taken into account is to make sure that precision didn't fall so far that the model was over-adjusting to increase recall. Together, F1 and recall scores were used to measure the overall accuracy of the model.

3.5 Model Selection

The models were selected based on the articles [1, 2]. However, the decision to analyze supervised learning rather than unsupervised was due to how unsupervised models struggled less with a limited amount of features than the supervised models did. [3]

3.6 Models Used

3.6.1 Logistic Regression

Logistic Regression is an algorithm that uses a logistic curve in order to make classifications. In order to do this, Logistic Regression finds the probability of a sample being of one classification or another, then classifies based on a threshold.

$$Probability = \frac{e^{i_0 + i_1 \cdot x_1}}{1 + e^{i_0 + i_1 \cdot x_1}} \tag{4}$$

3.6.2 *Adaboost*

Adaboost is an algorithm that fits by creating many trees with two leaves. These trees, called "stumps", are used to classify the sample, and are weighted differently based on how well each stump classifies the samples in a set. The algorithm eventually makes the final classification based on every stump, taking into account every stump.[3]

Training is preformed with:

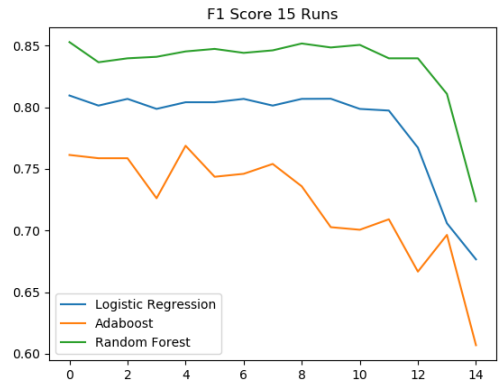
$$E_t = \sum_i E[F_{t-1}(x_i) + \alpha_t h(x_i)] \tag{5}$$

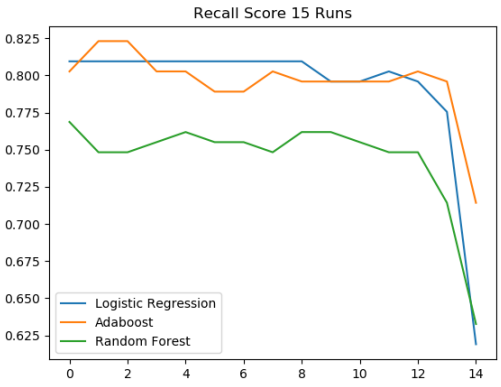
3.6.3 *Random Forest*

Random Forest is an algorithm similar to Adaboost, except instead of stumps, there are full trees of variable size that are used, and during the final classification, every tree is weighted equally.

4. RESULTS

The following graphs represent the F1 and Recall scores throughout 15 iterations, with each iteration representing 2 features cut per iteration. The y-axis represents the % accuracy of each model





4.1 Model Performance Over Iterations

4.1.1 *Logistic Regression*

As you can see in the graphs, the Logistic Regression model preformed fairly well in both F1 score and recall score. Compared to the Adaboost model, which has a slightly higher recall score, and Random Forest, which has a better F1, Logistic Regression maintains a consistent level of accuracy.

However, it is important to note that at the lowest level of features included, Logistic Regression quickly drops off, and ends up with the lowest recall score of the three. This suggests that logistic is better for every case, except for when you are using only 2-6 features.

4.1.2 *Adaboost*

Unlike Logistic Regression, the Adaboost model preformed better only once the number of features started getting down to the 2-6 range. However, Adaboost also preformed the worst of all 3 models in F1 score, implying a much lower precision score.

It is also important to note that Adaboost had the highest maximum recall value, and wasn't effected as much recall-wise as the other models as the number of features was reduced. Despite the high recall score, Adaboost looks to be outclassed by Logistic Regression, since the ideal is having both high precision and high recall.

4.1.3 *Random Forest*

While the Random Forest model preformed well in F1 score, it did poorly in terms of the end recall score, suggesting a high precision score. This means that, for this imbalanced data set, the model is out preformed by both the Logistic Regression and Adaboost models.

In terms of how the model preforms with a reduced amount of features, the Random Forest model holds up slightly less well than the Adaboost model, but the stability of the model is offset by the overall low recall score. This implies that, in a more balanced model, Random Forest might be the better model for this data set, although this test was not built to measure performance in a balanced data set.

4.2 Conclusions

Analyzing these 3 models reveals that the Logistic Regression algorithm is the best model for classifying credit fraud for most cases, while the Adaboost algorithm is better for when you reach a lower amount of features.

4.3 Future Work

Future analysis of how models preform on imbalanced data sets with a limiting number of features can include models other than the three used here, as well as tailoring the data set specifically for each model. In particular, Adaboost's performance was most likely effected by outliers in the data set, but in the interest of consistency, the outliers were kept so that the data set would be the same for each model. Similarly, the overall model accuracy can be improved with random under and oversampling of the full data set. Implementing these things and testing on more models can provide a more complete picture of the relationship between model performance and the number of features used.

4.4 Lessons Learned

Starting out this project, I initially assumed that the accuracy could be measured normally with a generic right over total accuracy metric. However, after researching more, I

[3]It is important to note that the constant weight set for each algorithm is used as a starting weight for the Adaboost algorithm

discovered why that didn't work and why recall score was a better measurement of the performance of the model. This also lead me to learn how to increase recall accuracy by adding weights to the samples with the classification that had low frequency.

Additionally, I also learned how analyze data sets with encoded features and make decisions based off what the statistics told me, which is something that I had previously struggled with.

5. REFERENCES

[1] Heta Naik, Prashasti Kanikar, *Credit card Fraud Detection based on Machine Learning Algorithms*. Inter. Jrn. Comp. App., Volume 182, No. 44, Pages 8-10, March 2019

[2] Suraj Patil, Varsha Nemade, PiyushKumar Soni, *Predictive Modelling For Credit Card Fraud Detection Using Data Analytics*. Proc. Comp. Sci., Volume 132, Pages 385-395, 2018

[3] Xuetong Niu, Li Wang, Xulei Yang, *A Comparison Study of Credit Card Fraud Detection: Supervised versus Unsupervised*. April 24, 2019

[4] Kaggle Dataset, `https://www.kaggle.com/mlg-ulb/creditcardfraud`