Credit Card Fraud Detection Investigation

Kenny Kang

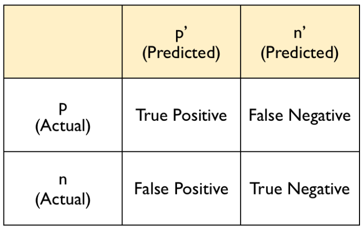
May 1, 2017

1 Problem Statement and Goal of Analysis

Credit card fraud has become a common problem in the past decades as the primary form of payment in first world countries shift from physical to digital. In recent years, data of fraudulent transactions have become more accessible to the public. The purpose of this investigation was to find a way to accurately detect future fraud using machine learning algorithms. The dataset that was used was released by Kaggle and contains transactions made by Europeans in September 2013.

1.1 Metrics for Analysis

The raw data used had an imbalance between the number of positive and negative cases. Therefore accuracy was not the best metric to measure the performance of our models. Instead, better metrics for analysis would be the Confusion Matrix and the Area Under the Precision-Recall Curve (AUPRC). In a binary classification problem, the Confusion Matrix shows exactly how many of the positive and negative classes were classified correctly and incorrectly. Figure 2.1 generalizes how this metric works. In addition, the Area under the Precision-Recall Curve (AUPRC) will also be used to measure the relationship between the precision and the recall. The precision describes how much of the positive predictions are accurate (TP / (TP + FP). The recall shows how many of the actual positive cases were predicted (TP) / (TP + FN). In the case of catching fraud, making sure all of the positive cases were detected (recall) was prioritized over the number of positive predictions that were correct(precision). This is due to the fact that it would be better to wrongly accuse a transaction of being fraud than to ignore an actual case of fraud and let the criminal free.



	p' (Predicted)	n' (Predicted)
p (Actual)	True Positive	False Negative
n (Actual)	False Positive	True Negative

Figure 2.1: Confusion Matrix

2 Data Preprocessing

When working with the raw data, there was very little work to be done with featurization, however most of the preprocessing had to do with balancing out sample sizes.

2.1 Featurization

Due to confidentiality issues, most labels for features in this dataset were unknown aside from their values. The only two pieces of data known to the public are "Time", and "Amount". The "Time" feature was simply a time-stamp and seemed irrelevant to the classification of cases of fraud, therefore it was dropped from the dataset.

2.2 Undersampling & Oversampling

Out of the 284,807 transactions recorded in the dataset, only 492 were classified as fraud (positive class). This would account for only 0.172% of all transactions making the dataset extremely unbalanced. While training the original data would result in a high accuracy, it would fail to accurately predict cases of actual fraud. As an example, training a simple logistic classifier with this data resulted in a 99.9% accuracy, but only accurately predicted 50% of the fraud cases. In order to balance out the proportions, either over-sampling or under-sampling can be used. Over-sampling would entail repeatedly sampling from the minority class until both classes were the same size. In under-sampling, a small sample would be taken from the majority class that would match the size of the minority class. During this investigation, data from both methods of sampling were used in order to compare and ultimately determine the best solution in the given situation.

3 Data Modeling

3.1 Training, Validation, and Test Set

In between dropping features and sampling, the raw dataset was split into a training set (50%), validation set (25%), and a test set (25%). Both the validation and test sets were left untouched until analysis while the training set was sampled for under-sampling and over-sampling. The raw train data, under-sampled data, and over-sampled data were all used to train a classifier

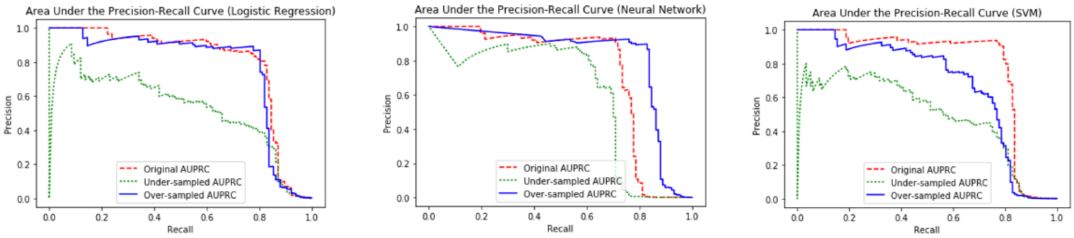


Figure 3.1: AUPRC charts for the three models

with the exact same hyper-parameters. However those parameters were biased and were adjusted for the over-sampled training set. All models used the same validation data for adjusting hyper-parameters as well as the same test set for the final prediction.

3.2 Logistic Regression

The first model that was tested was a Logistic Regression classifier using the sci-kit learn python library. In order to prevent over-fitting, an l2 loss function was applied. Of the three datasets tested, both the over-sampled classifier and the original classifier outperformed the under-sampled classifier. The original training model received the highest AUPRC score of .798, unfortunately while its precision was extremely high, the model classified 30% of the fraud cases as non-fraud. Meanwhile the over-sampled model classified over 90% of the fraud cases, however it's precision was far behind the original training set.

3.3 Fully Connected Neural Network

The second model was a fully connected Neural Network using the Keras framework. The network itself consisted of three layers: an input layer with 29 nodes, a hidden layer with 58 nodes, and an output node a single node. The first two layers had RELUs as activation functions while the output layer used a sigmoid function. A Dropout with a frequency of 25% was applied after the hidden layer in order to counter over-fitting. The best performing classifier was the one using over-fitted data with an AUPRC score of .815. It accurately predicted over 90% of the total fraud cases as well as only misclassifying 385 non fraud cases compared to it's 103 correctly labeled positive predictions.

3.4 Support Vector Machines

The final model used were Support Vector Machines (SVM) using the sci-kit learn library. There were no adjustments to hyper-parameters in this model. The classifier using the original training data was the only one to surpass a .7 AUPRC score. Unfortunately it suffered the same problems as it's Logistic Regression counterpart with a high precision but low recall, missing 30% of the total fraud cases.

4 Comparison of models

Overall, the model that performed the best is the neural network paired with the over-sampled dataset. There are many factors that played into this result. The under-sampled datasets were subjected to hyper-parameters that were intended to prevent over-fitting. However using a small dataset, over-fitting isn't likely. As a result those models were typically extremely under-fitted. In addition, while classifiers using the original training data performed well on every model using the AUPRC, the confusion matrix for these classifiers showed low recall in every case. Since the priority of this investigation was to maximize recall, these models were unusable in comparison.

5 Discussion of results

The over-sampled neural network received an AUPRC score of .815, the highest of any model tested in this investigation. In addition, the classifier had a high recall and a satisfactory precision. If this model were to be applied to more transactions in the future, it would ideally detect 10 out of 11 cases of fraud. It would also only mis-classify 4 out of 5 non-fraud cases, which is far less than most of the test models.

6 Conclusions

This project was been one of the most helpful projects I have done since entering school. I loved how it was open-ended and made use of interesting data. If I had more time I would have definitely tried using cross-validation as well as trying out more models. My biggest challenge, but also my favorite part of the project, was dealing with the unbalanced dataset. It forced me to learn how to deal with classification problems that weren't cleaned up for me. It put me in a situation where I needed to research into different metrics to measure performance other than accuracy which I had no idea existed before this project. Also another problem I initially had was mixing up data from my validation, test, and training sets when sampling from them. I eventually realized I should separate them in advance and only sample from the training set. Overall it was an amazing experience, I only wish I had more time to put into this project.