Credit Card Fraud

Josh Christensen and Jacob Miller

11/29/2021

Abstract

Accurately predicting credit card fraud can save both companies and individuals money. In this analysis, we use a data set of nearly 300,000 credit card transactions and use a SMOTE sampled random forest model to predict fraudulent cases. We were able to identify fraudulent cases over 95% of the time and use the model to predict transactions where fraud is currently unknown.

Problem Statement and Understanding

Predicting and identifying credit card fraud is a necessary and valuable pursuit for banks and credit card companies. However, because of how rarely credit card fraud occurs relative to the amount of non-fraudulent activities, efficiency in prediction and identification can be a difficult task. In this dataset, we have 491 fraudulent transactions and around 280,000 non-fraudulent transactions, along with 28 principle components of each transaction and the amount of each transaction as explanatory variables. This information will allow us to analyze and predict which transactions are fraudulent (or suspicious) given the data. Figure 1 is a jittered scatter plot that shows the relationship between fraud and the only explanatory variable we have that is interpretable. We can see that credit card fraud (1's) become rarer the higher the transaction amount.

The main problem with this data set is the major class imbalance between the fraudulent cases and non-fraudulent cases. We know that only about 0.1% of the data are cases of fraud and that can lead to issues with prediction. One issue is that even a poor model where all of the cases are predicted to be non-fraudulent would give us accuracy and AUC (area under the ROC curve) metrics extremely close to one. If we are not careful, we may choose a model that over or under predicts the fraudulent cases in a way that is no longer helpful.

The goals of this analysis are to 1) accurately identify fraudulent cases. We want to know how well we can identify a transaction when it is fraudulent. And, 2) we want to be able to predict which, if any, of the five unknown cases are fraudulent.

Describe the Methods/Models Proposed

We attempted different models and methods in order to fit the best model using this data. In order to adjust for the class imbalances, two main methods were used in modeling: ROSE (random over sampling examples) and SMOTE (synthetic minority oversampling technique). Summarily, ROSE creates artificial samples from the feature space in the neighborhood of the minority class and SMOTE creates artificial samples by choosing points on the line that connect the minority class to its nearest neighbors. We implemented

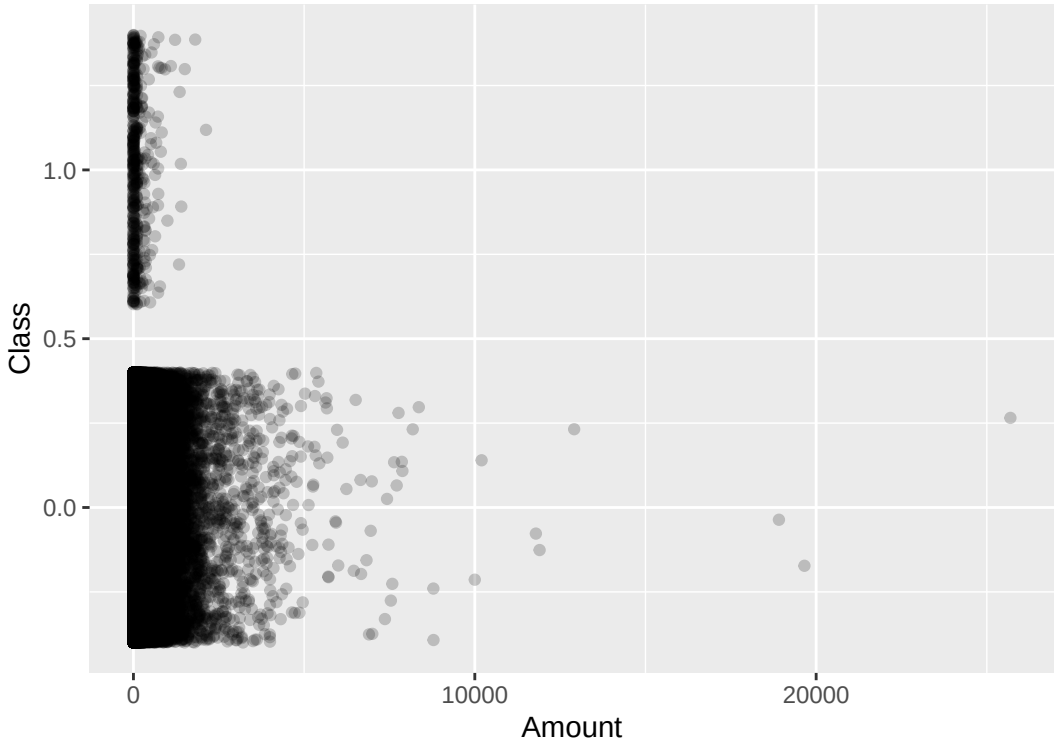

Figure 1: Fraudulent cases appear to mostly have lower transaction values

SMOTE as opposed to ROSE because it created better models. With the sampling techniques, we fit random forest and linear discriminant analysis models. LDA is a method that estimates the (normal) distribution that the predictors come from for each class. If the data comes from approximately normal distributions, there are small sample sizes, and the classes are well defined from each other, LDA tends to be a strong model. However, if we try to estimate the distribution that the data is coming from incorrectly, this model can perform very poorly. This model tends to not do as well as other techniques, including the random forest model, so we decided not to use it for our final model. Random forests are tree based models that implement bagging in order to create many different trees in order to estimate the true relationship between the data and outcome. Random forests are useful because they can deal well with interactions and non-linear relationships. One weakness of random forests is that, if we're not careful in tuning, they can overfit the training data. Another weakness is that there is very little interpretability because it does not estimate model parameters. Because the goals of this study are directed towards prediction (and the explanatory variables are mostly unlabeled principle components), the random forest model makes a very good candidate for the final model. The random forest model does not have any explicit or implicit assumptions we need to check.

Model Selection, Justification, & Performance Evaluation

After performing in-sample cross validation and comparing the results, we decided to use the SMOTE sampled random forest model because it performed better than the ROSE sampled LDA model. The ROSE sampled LDA model actually performed worse overall than the non-ROSE-sampled LDA model. In table 1 we report the cross-validated results from both models.

Table 1: Out-of-sample Metrics

	Random Forest	LDA
AUC	0.998	0.979
Accuracy	0.985	0.996
Sensitivity	0.953	0.996
Specificity	0.995	0.799
PPV	0.985	0.999
NPV	0.985	0.274

After observing the ROSE sampled LDA model metrics, it was obvious that it was altering the data in a way that made the model significantly worse. We suspect that it oversampled too far, causing most of the questionable explanatory space to be identified as fraud. In light of this problem we elected to use the SMOTE-trained random forest as our final model. In-sample metrics from the random forest model are provided in table 2.

Table 2: In-sample Metrics

	Random Forest
AUC	0.999
Accuracy	0.999
Sensitivity	0.999
Specificity	0.999
PPV	0.997
NPV	0.999

The random forest algorithm is an ensemble tree method with two tuning parameters, B and m. B is the number of trees to be grown and m is the number of explanatory variables to be considered at each split. The algorithm proceeds by taking a bootstrapped sample of the data and then growing a tree while only considering m variables at each split. Predictions are then created by passing the observation down each individual tree to get classifications and taking the class that receives a majority vote. This can be written formally as $\hat{y}(x_i) = \text{mode}(\hat{y}^1(x_i), \ldots, \hat{y}^B(x_i))$ where $\hat{y}^b(x_i)$ indicates the class predicted by the the b^{th} tree. Our trees actually create probability predictions so we have the flexibility to set a probability threshold for predicting fraud. We chose 0.4 as our threshold based on what gave the desired sensitivity in cross-validation. The SMOTE algorithm is used to create a more balanced training set when the original data is heavily imbalanced. The original credit card transaction dataset had approximately 0.1% representation of the fraud class, while in the SMOTE dataset there is 23.8% representation. The SMOTE algorithm generates synthetic fraud cases by finding the nearest neighbors in euclidean distance and then generating new observations along the connecting space between them.

In our case we used $B = 200$, $m = round\left(\sqrt{P}\right) = 5$, and a threshold of 0.4. The value for m was the default from the package, but has also consistently been shown to be a good choice for m. B and the

threshold were both chosen based on cross-validation metrics.

One of the benefits of tree-based models is that they do not require the data to meet any assumptions. Trees simply split explanatory variable space to reduce variability and ensemble methods like random forest are built on the basic tree structure. We are therefore safe to use this method on this data set. We used all available variables to fit our model. Tree-based models split on whichever variables reduce variance the most. Therefore if there are variables that are not relevant to the response variable they will simply be ignored by the algorithm.

Results

The random forest algorithm has no assumptions and therefore does not estimate model parameters. It simply produces predicted classes. While we do not have any formally estimated parameters we can still get information about variable importance and approximate effects using predictions. Figure 2 displays a variable importance plot generated by our model.

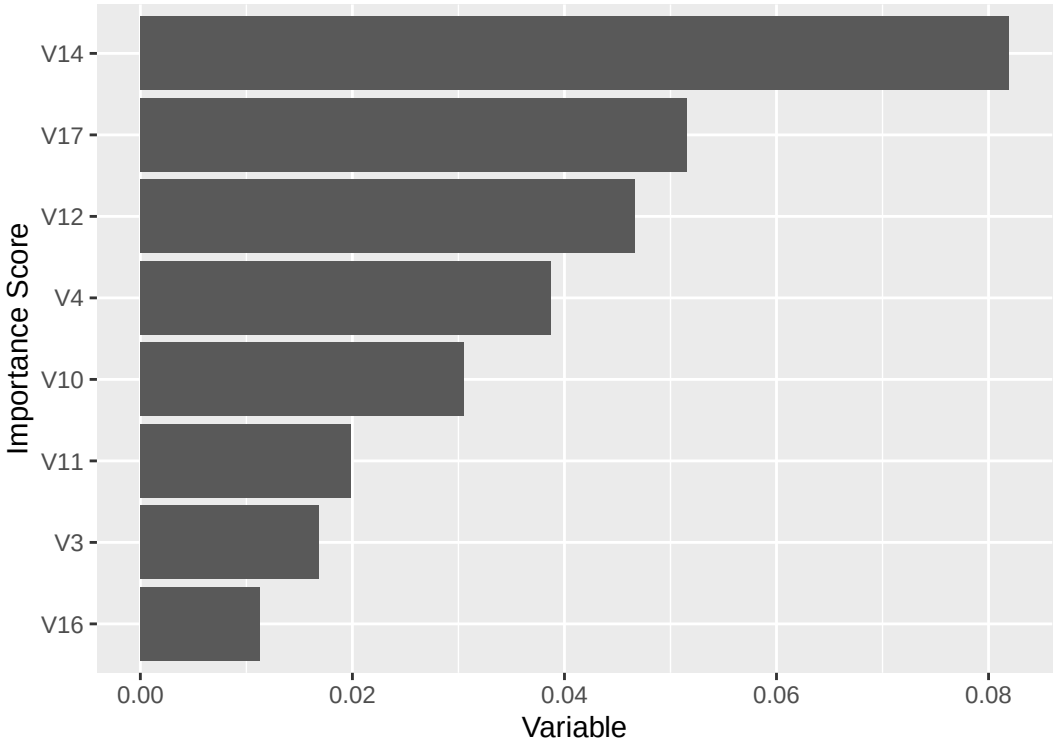

Figure 2: Variable importance plot from the Random Forest algorithm

In order to approximate the effect of transaction amount on the probability of fraud we will predict for various transaction amounts. All other variables will remain constant at the mean. The resulting plot is displayed in Figure 3.

We will now address the two main goals of the analysis:

1) Our model had a cross-validated sensitivity of 95.3%, which means that given a transaction was fraudulent, we were able to predict it as fraudulent 95% of the time.

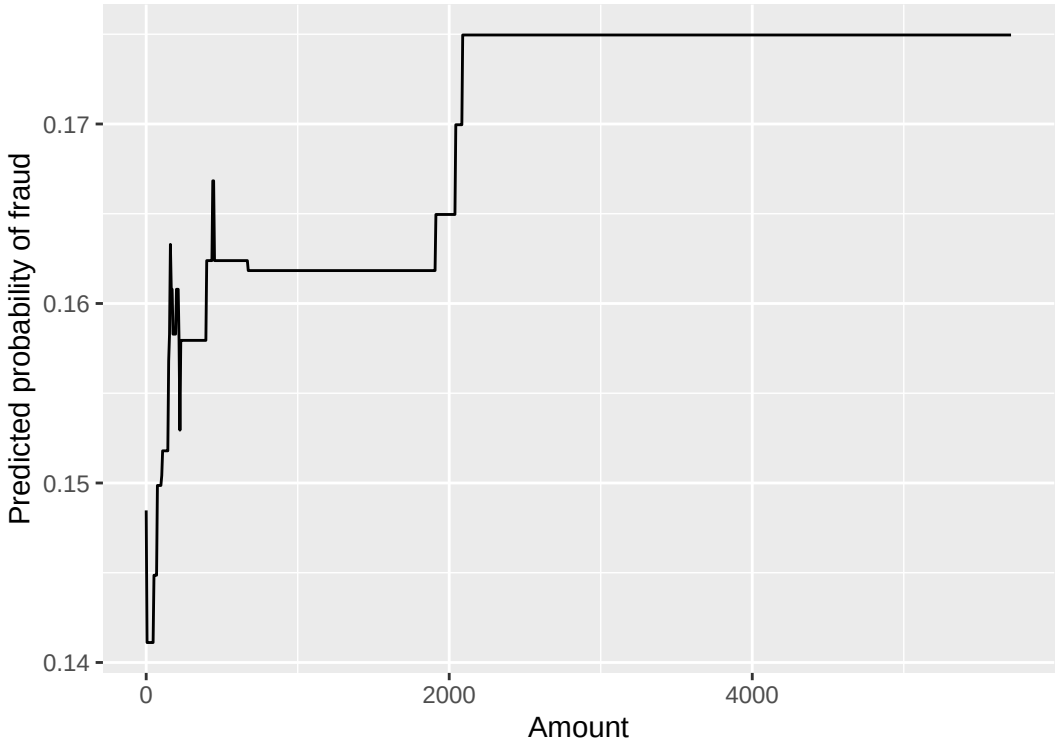

Figure 3: Approximate partial effect of transaction amount

2) Of the five different transactions we were given to predict on, our model determined that the third one was fraudulent and that the remaining four were not.

We found that the random forest algorithm did the best job of identifying fraudulent transactions. In data that was not used for training the model we were able to identify 95% of fraudulent cases.

Conclusion

In this analysis we were able to build a random forest model that accurately identifies fraudulent transactions over 95% of the time. In addition, we predicted that the third of the five test transactions we have was fraudulent (one of the $1.00 transactions).

A shortcoming of our approach would be that we could have tuned more of the hyperparameters of the random forest model instead of mainly using the defaults. In the future, we can adjust the cutoff value in order to increase or decrease the sensitivity and discuss the tradeoffs. Is it worth correctly identifying 1% more fraudulent transactions if that means we misclassify thousands of additional non-fraudulent transactions? How much time does it save the credit card company to reduce the amount of non-fraudulent cases they have to inspect and how much is that time worth? In future analyses we would like to get more information on where the ideal cutoff would be for the bank or be able to quantify the trade off better.

Teamwork

Josh - SMOTE sampling, final random forest model and other models, half of write up and final touch ups and formatting
Jacob - ROSE sampling, LDA model, half of write up, final read through