Camilla Handley
Carly Lundgreen
Stat 536

Credit Card Fraud

Introduction

It is reported that credit card fraud costs consumers around $22 billion every year. This is a significant issue for credit card companies (and consumers), so it is important to be able to identify when credit card fraud is occurring. According to Fidelity National Information Services, a banking fraud detection company, the dollar amount of attempted fraudulent credit card transactions rose 35% in April 2020--during a high point of the Covid-19 pandemic. All these figures showcase the issue of credit card fraud and its recent increase. The aim of this analysis is to accurately identify fraudulent transactions using machine learning techniques so that credit card companies can freeze accounts or take any necessary action against the fraud.

The provided dataset contains information on 284,807 credit card transactions. While credit card fraud is certainly a major issue, it is still a relatively rare event considering the sheer number of transactions performed by consumers every day. We can see this in our dataset, which has only 492 known fraudulent transactions. This is about 0.1% of the observations in the data. This class imbalance will be important to account for in our predictive model. If the class imbalance is disregarded, then the metric of predictive accuracy becomes meaningless. For example, if a fitted model correctly classifies 99.9% of transactions, this appears to be extremely accurate. However, a model with this accuracy may have simply identified every transaction as non-fraudulent, in which case it would be mostly correct because the majority of the observations are non-fraudulent. This would be a useless model, however, because it is not identifying any fraudulent transactions. Other metrics, such as precision or recall, may also be affected by this class imbalance. We discuss a solution for this issue in the following section.

In addition to a classification of legitimate vs. fraudulent transactions, the dataset contains information on 29 different characteristics of the transactions. Banking information, however, is highly confidential, so the explanatory variables are instead reported as principal component scores of the original explanatory variables. A final covariate, the amount of the transaction, is also included.

Statistical Method

For this analysis, we built a Naive Bayes model to classify the transactions. Naive Bayes is a special case of discriminant analysis. In this process, we assume that each explanatory variable is independent (which is never really true - hence the name “Naive”), so the joint distribution can then be factored as seen below:

$$f_{\mathbf{X}}(\mathbf{x}) = \prod_{p=1}^{P} f_{X\,p}(x_p)$$

If a certain x_p is quantitative, we assume that its distribution will be Gaussian and if it is categorical, then it will be multinomial. In order to classify observations, we will apply the Bayes Theorem to the joint distribution of Y given X to get the following:

$$\mathrm{f}_Y(y|\mathbf{x}) = f_X(\mathbf{x}|y)f_Y(y)$$

We can then factor the joint distribution as follows:

$$f_X(\mathbf{x}|y=0) \sim \mathcal{N}_p(\mu_0, \Sigma_0) \qquad \mathrm{f}_X(\mathbf{x}|y=1) \sim \mathcal{N}_p(\mu_1, \Sigma_1)$$

Together with the prior distributions (π_0 and π_1) for Y=0 and Y=1, which are simply the proportion of non-fraudulent transactions and fraudulent transactions, we can then calculate a conditional distribution for being fraudulent or not. $\widehat{\mu_0}$ and $\widehat{\Sigma_0}$ are defined as the following:

$$\hat{\mu}_0 = \tfrac{1}{N_0}\textstyle\sum_{i:y_i=0}\mathbf{X}_i \quad \hat{\Sigma}_0 = \tfrac{1}{N_0-1}\textstyle\sum_{i:y_i=0}(\mathbf{X}_i - \hat{\mu}_0)(\mathbf{X}_i - \hat{\mu}_0)'$$

$\widehat{\mu_1}$ and $\widehat{\Sigma_1}$ are defined similarly, but with the sums being over $y_i = 1$ and N_1 instead of N_0. For example, the first number in the $\widehat{\mu_0}$ vector would represent the average V1 value for all non-fraudulent observations. The second number would represent the average V2 value for all the non-fraudulent transactions, and likewise for each explanatory variable. X_i is a matrix with 29 columns (one for each explanatory variable) and the number of rows equal to the number of non-fraudulent transactions in the data set. Then using these calculations, each observation is assigned to the class that maximizes the following:

$$\max_{i\in\{0,1\}} \texttt{dmvnorm}(\mathbf{X} \mid \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)\pi_i$$

$$\max_{i\in\{0,1\}} -\frac{1}{2}(\mathbf{X} - \boldsymbol{\mu}_i)'\boldsymbol{\Sigma}_i^{-1}(\mathbf{X} - \boldsymbol{\mu}_i)' + \log(\pi_i)$$

Classification Tree

We also looked into using a classification tree to identify credit card fraud. In general, classification trees divide the data into non-overlapping regions and classify each observation by looking at the majority class of the observations in that region. However, due to the unbalance in this dataset, we suspected that this method would not be as accurate, because the fraudulent transactions are so few and far between and would rarely compose the majority of a region.

To grow the classification tree, a method called recursive binary splitting is used. For each predictor, this method will identify a cut point such that splitting the predictor space into two regions (above and below the cut point) leads to greatest reduction in error. This process is repeated for all previously defined regions. To avoid overfitting to the data, regions are identified that minimize $\text{Error}(\mathbf{y}, \hat{\mathbf{y}}) + \lambda T$, where λ is the complexity parameter to control for overfitting, T is the size of the tree. The error can be measured in various ways, but for this analysis, we used the misclassification rate. The complexity parameter is chosen via cross-validation to minimize the chosen measure of error.

In order to account for the imbalance between the fraudulent and non-fraudulent transactions, we performed a random over-sampling procedure on the minority (fraudulent) class and a random under-sampling procedure on the majority (non-fraudulent) class. Over-sampling implies random sampling with replacement until the desired proportion of fraudulent transactions is reached. Under-sampling samples from the majority class without replacement until the desired proportion of legitimate transactions is reached. While simultaneously over and under-sampling, we were able to create a dataset that had approximately 50% fraudulent and 50% non-fraudulent classifications. We fit the two models discussed above on both the balanced and unbalanced dataset for comparison, and these results are given in the following section.

Model Justification

Aside from considerations of class imbalance, the classification tree does not have distributional or independence assumptions. The Naive Bayes model assumes independence between explanatory variables, a Gaussian distribution for the quantitative variables and a multinomial distribution for the categorical variables. Given that we don’t have any categorical variables, we don’t have to worry about that assumption. After centering and scaling the data, most variables appeared to have a somewhat normal distribution, with a few outliers. However, due to the considerable size of our data set, we are not worried about extreme outliers. There were still a few variables that appeared to have skewed distributions however, so we decided to use kernel density estimates for the distributions of the explanatory variables. Kernel density estimation is a non-parametric method to estimate the probability density function (pdf) of a random variable. To illustrate the need for a kernel density estimate, we can see that Figure 1 shows the skewed distribution of V1, so this variable obviously does not conform to the Naive Bayes assumption of normality. Figure 2 shows the approximately normal distribution of V19 for comparison.

Figure 1: Histogram of V1

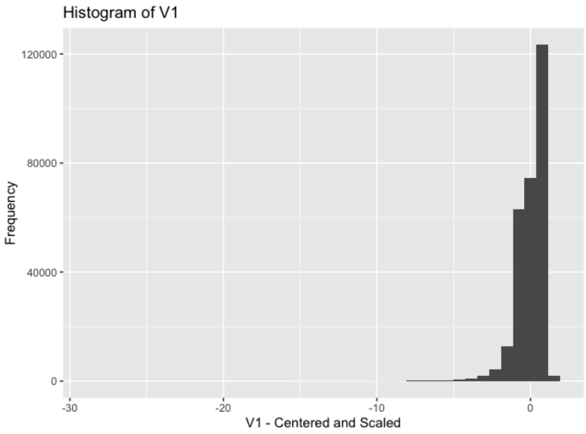
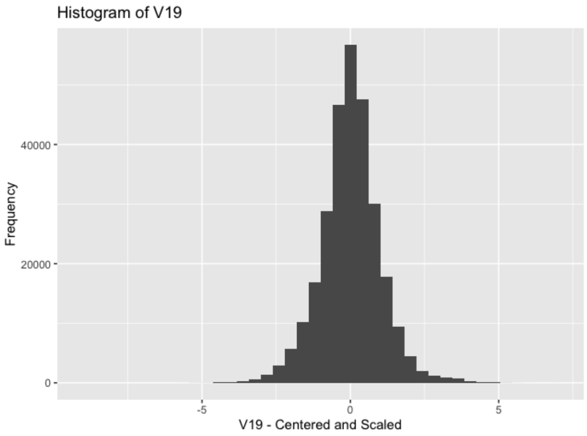


Figure 2: Histogram of V19



In kernel density estimation, kernels are used as a basis function expansion in the design matrix, X. Weights are applied based on where the majority of observations are located. Using a kernel K(x), which integrates to 1, the pdf of each explanatory variable is estimated according to the following formula:

$$\hat{f}_X(x) = \frac{1}{\sigma n} \sum_{i=1}^{n} K\left(\frac{x - x_i}{\sigma}\right)$$

Where σ is referred to as a bandwidth parameter and controls the smoothing of the estimated pdf to prevent overfitting. This parameter is selected through cross-validation to select the bandwidth that provides the best fit. This parameter is not of interest in the analysis however, so we will not report on it in the next section.

Using kernel density estimates for the distribution rather than assuming all variables follow a normal distribution will take care of this assumption and we can continue with the Naive Bayes model. As mentioned previously, the dataset consists of the principal component scores of the explanatory variables, so we will assume that all variables in the dataset are significant and should be included. This would account for any possible collinearity or other issues due to high-dimensionality.

Model Evaluation and Results

In order to evaluate how well the models fit the data, we calculated the recall, precision, and F1 score for the in-sample predictions using both balanced and unbalanced data. For the classification tree, we will only report the results from the balanced dataset for comparison to the Naive Bayes results. The tree had an in-sample recall (sensitivity) of 0.805, precision (positive predictive value) of 0.929, and F1 score of 0.863.

These are each lower than the results from the Naive Bayes (although not by much--the classification tree on balanced data still has a considerably good fit). However, the Naive Bayes metrics for the balanced data are higher, which is why we selected the Naive Bayes model to identify fraudulent transactions. The table below displays the in-sample precision, recall, and F1 score for the balanced and unbalanced data using a Naive Bayes classifier. Note that the F1 score is the harmonic mean of precision and recall. Ideally this value is as close to 1 as possible, indicating high precision and recall.

	Precision	Recall	F1 Score
Balanced	0.9958	0.8305	0.9056
Unbalanced	0.3810	0.7967	0.5155

We focused on recall, and then considered precision, because of the imbalance in the dataset and the purpose of this analysis. We calculated these metrics because they measure how well we identified fraudulent transactions. Given that fraudulent transactions make up only 0.1% of the dataset (unbalanced), the fact that we correctly classified 79.67% of the fraudulent transactions in the dataset indicates adequate fit. Out of all the transactions that we classified as fraudulent, 38.1% of them were actually fraudulent. The results from the balanced data indicate that the Naive Bayes classifier works quite well, and when the classifier is applied to the original data, the recall only decreases a slight amount. Considering the small information we have about fraudulent transactions, these numbers indicate adequate fit. Notably, recall is comparable for both the balanced and unbalanced datasets while precision differs significantly between balanced and unbalanced. This is likely because the balanced data simply has more opportunities to classify transactions as fraudulent because there were more fraudulent cases in the data. The fact that recall is comparable between balanced and unbalanced is promising because we know that the naive bayes classifier is working well and identifying truly fraudulent cases about 80% of the time.

In order to evaluate how well the model can predict we calculated the recall, precision, and F1 score for the out-of-sample predictions. These are listed in the table below.

	Precision	Recall	F1 Score
Balanced	0.9943	0.8313	0.9055

Unbalanced	0.3179	0.7997	0.4541

These values are very similar to the in-sample predictions. Using similar interpretations as above, the precision and recall are quite high and allow us to conclude that our model has good predictive ability. Note that we are not using predictive accuracy as a measure of predictive ability because of the high class imbalance in the data. The balancing procedure should take care of this issue, but recall (and precision) give a greater insight into the classification ability for only fraudulent transactions. The recall, however, provides better insight into the question at hand. Given the transaction is fraudulent, we identify it as such about 80% of the time using the Naive Bayes classifier. While certainly not perfect, our method has relatively high classification ability.

Conclusion

Using a Naive Bayes classifier, we were able to accurately identify fraudulent transactions about 80% of the time. We recognize that there are many other methods that could be used to classify this data, and that the Naive Bayes may not be the optimal method. As previously mentioned, this dataset is severely unbalanced because of the rarity of a fraudulent transaction in comparison to the high number of transactions that happen. The assumption that the Naive Bayes classifier makes is somewhat impractical, but in this case may be reasonable because our explanatory variables are principal components, which have no collinearity and thus are inherently independent of each other. However, different methods may be beneficial to look at. K-Nearest-Neighbors, Neural Nets or stacked models are just a few ideas of additional methods that could possibly result in more accurate results. As is always the case, more data would be useful. Whether it is simply more explanatory variables, or more observations that are fraudulent, usually more data results in higher accuracy.

Teamwork:

Carly wrote the intro and the stuff on classification trees, Camilla wrote the conclusion and the Naive Bayes methodology. We split the evaluation/results section and code about half and half.