Complaint Department Assignment

Max Smith and Josh Christensen

12/9/2021

Abstract

When a complaint is sent to a company it is critical that it reaches the correct department and is resolved quickly. We were given a dataset that contained complaints along with the department that the complaint needed to be sent to. We sought to develop a machine learning model that would predict which department a complaint needs to go to be solved. Our final model was a random forest model and we achieved 80.4% predictive accuracy in classifying new complaints. We also classified 10 complaints that were unlabled.

Introduction

Helping customers resolve their concerns quickly and efficiently is one of the things that makes a great company great. This can be a tremendous challenge for companies. When customers reach out to a company with complaints the company must find the right department to resolve each complaint. If a machine learning model can look at this complaint and send it to the right department it will save the company money by eliminating the need for a person to read the complaint and send it to the right department as well as decrease the time in getting the complaint resolved. The data that we have contains complaints as well as the department that the complaint should be sent to. Figure 1 displays a plot with the 9 departments and how many of the 124,896 complaints go to each of them.

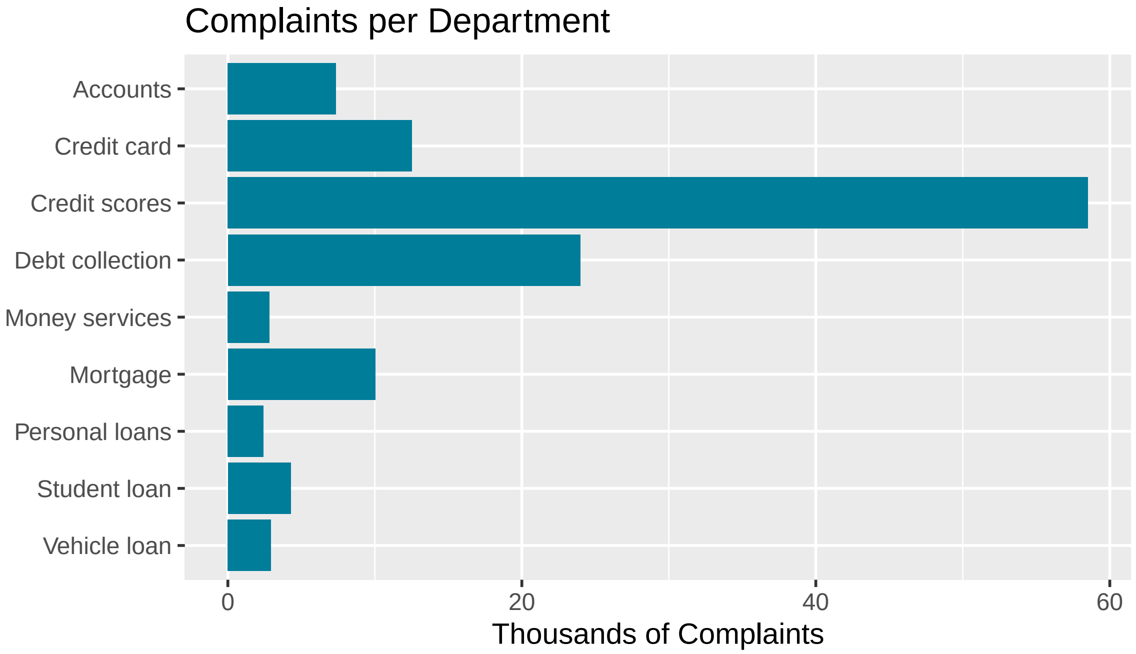

Figure 1: Total complaints sent to each department.

The most important thing to note about this unique dataset is that there are no given explanatory features. Features need to be created using the text of the complaints. Words cannot just be put into most machine learning models. We need to create our features given the complaints that we have in our dataset. We also do not have equal numbers of complaints for each department, which could potentially bias our predictions towards departments that get more complaints. If we do not change our data to get numerical features, we will not be able to fit a model. We also need to look and see if having few observations of complaints in some departments makes it difficult to predict complaints for those departments.

We have several things we want to understand about our data through our analysis. First, we want to see how accurately we can classify the complaints. Second, we want to see what words or other features of the complaints were useful in classifying. Third, we want to see if some of the departments are often mistaken for others. For example, does our model often say that a student loan complaint is a vehicle loan or lease complaint. Finally, we want to classify 10 complaints that we do not have labels for.

Proposed Methods and Models

We fit four models to this data. A gradient boosted tree model, a random forest model, a linear discriminant analysis model and a k nearest neighbors model. The gradient boosted and random forest models performed the best so we will compare these two models primarily, but in-sample and out-of-sample metrics are provided for all models in tables 1 and 2 in the model selection section.

Gradient boosted trees are part of a group of decision-tree-based ensemble methods. In this algorithm the growth of each individual tree is restricted and the final impact of each tree on the model is also scaled down. Each new tree is fit on the residuals of the model up to that point. This allows each tree to discover new aspects of the true function without overpowering the other trees in the model. In our case where we have 47 explanatory variables and potentially complex relationships between them an ensemble method would be expected to perform well. One weakness of gradient boosting is that it requires very careful tuning in order to fit well and quickly without overfitting. Finding the proper balance between learning rate, number of trees and tree depth is not an exact science and can be time-consuming.

Random forests are also tree-based ensemble models. random forests use bootstrapping and random variable consideration to create many different trees that together estimate the true relationship between the data and the response. Random forests handle interactions and non-linear relationships well, without extra specification in the model. One weakness of random forests is that they are prone to overfitting if not carefully tuned through cross-validation. Random forests also do not provide interpretable parameter estimates as they are mostly focused on prediction. In our scenario our primary goal is prediction accuracy and variable importance which makes the random forest a good choice for this dataset.

The random forest model and boosting models do not have any formal mathematical assumptions. Becuase both are based on simply dividing explanatory space, as long as there are explanatory variables the algorithms can be used.

Model Selection, Justification & Performance Evaluation

For our final model we selected the random forest model. We begin with a comparison of in-sample model fit metrics. Metrics for all four models are provided in table 1 We can see in table 1 that the ensemble methods outperformed the other methods in terms of model fit.

Table 1: In-sample Performance

	Fitted Accuracy
Random Forest	0.961
Boosting	0.757
KNN	0.770
LDA	0.644

While in-sample metrics give us a good baseline for evaluation we care much more about the cross-validated metrics since they provide information on how well the model is likely to predict. Table 2 summarizes the cross-validated prediction metrics. Table 2 confirms that the ensemble methods outperform other methods and that the random forest model outperforms boosting.

Table 2: Predictive Performance

	CV Accuracy
Random Forest	0.804
Boosting	0.755
KNN	0.635
LDA	0.643

The random forest algorithm has 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 new data down each individual tree to get a predicted calss 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 and x_i represents the explanatory variables of observation i.

In our case we used $B = 200$, $m = round\left(\sqrt{P}\right) = 7$. The value for m was the default from the package, but has also consistently been shown to be a good choice for m. B was chosen based on the data size and cross-validation.

As mentioned above, 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 generated two types of explanatory variables from the text of each complaint. The first were keyword count variables. We determined the 10 most common words for complaints sent to each department after excluding censored words and stop words. With the overlapping words this gave us 46 keywords. We then tallied how many times each keyword was used in each complaint and used those tallies as explanatory variables. The second kind of explanatory variable was the word count of the complaint. We simply iterated through each set of word tokens and counted the words included. We included all generated variables. Since random forest models split on the most impactful variables, the only detriment to including extra variables is an increase in computation time. Becuase of the minimal downside and the potential benefits we included all keywords and length of the complaint as explanatory variables.

Results

There are no true parameters for a random forest model. In other applications of random forests approximate partial effects plots can be useful, but in the multiclassification setting even these lack interpretability. Instead of using parameters to interpret results we extract variable importance and prediction metrics below.

The first question we set out to answer was how well we can classify complaints to the correct department. We found that with our random forest model we were able to correctly classify new complaints 80.4% of the time. This is, of course, significantly better than just randomly assigning complaints to each department and would save resources compared to having a person classify each complaint themselves. Next we looked at which of the explanatory variables helped us classify complaints the most. We found that the top 5 variables were the counts of the following 5 words for each complaint; report, debt, credit, card, and pay. We also found that the length of the complaint was the 7th most important feature we included. Below in Figure 2 is the variable importance score for each of the top 20 in our random forest model.

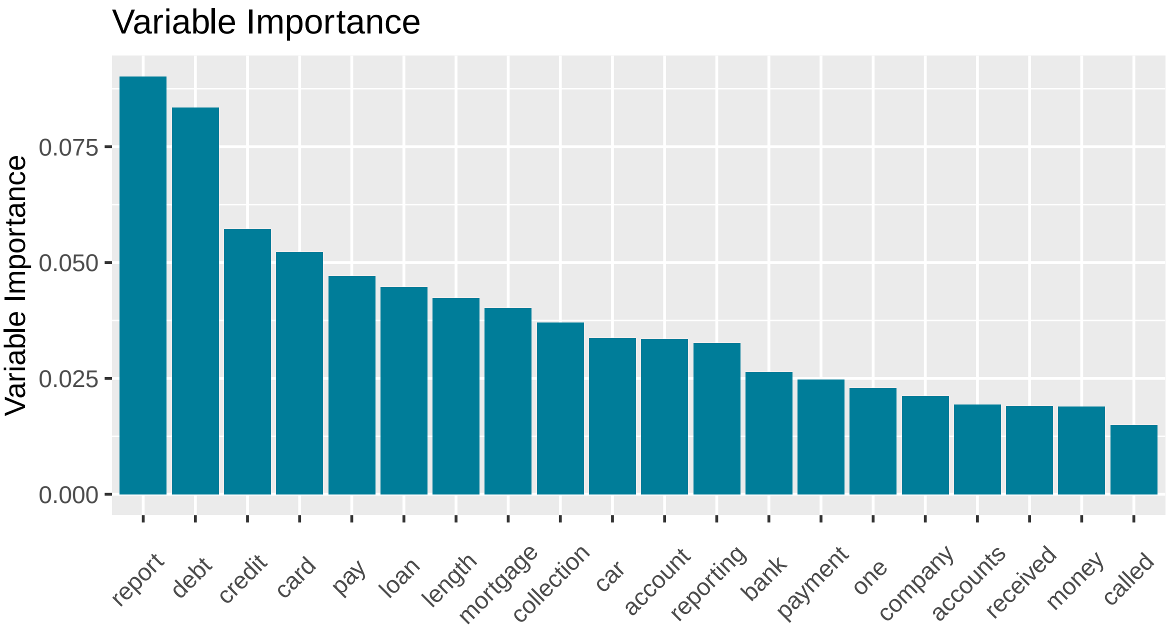

Figure 2: Random forest variable importance

Next, we looked at which departments were often confused with one another when being classified. We found that the top three departments that were miss-classified with others are as follows. First, when the true department for the complaint was Vehicle loans the model miss-classified it as Credit scores 31.9% of the time. Second, when the true department was Money services our modeled misclassified it as Credit or prepaid card 29.5% of the time. Third, when the true department was Personal loans the model misclassified it as Credit scores 21.4% of the time. After these three the remaining frequent misclassifications fell below 20%. The rest of the misclassification is summarized in figure 3. Values indicate percentage of complaints that were sent to a given department that were classified in each bin. Darker values indicate higher percentage.

Our final objective was to make predictions for 10 complaints that were received. The predictions for which department our model thinks those 10 complaints should be directed to are in order in the table 3. For example we think that the first complaint should be directed to the credit reporting department.

Table 3: New complaint predictions

	New predictions
1	Credit reporting
2	Credit reporting
3	Credit reporting
4	Mortgage
5	Credit reporting
6	Credit reporting
7	Mortgage
8	Credit card
9	Debt collection
10	Student Loan

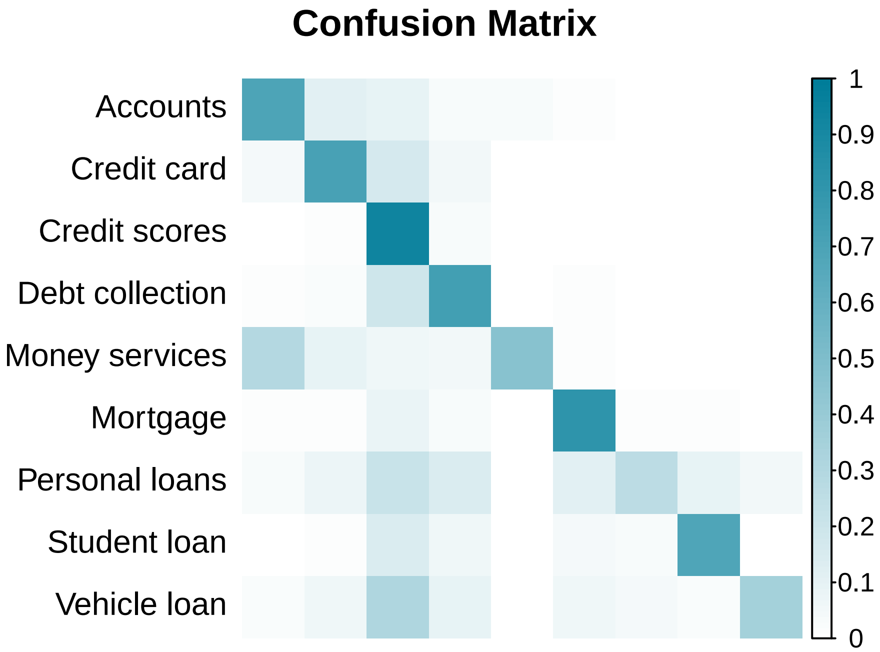

Figure 3: Confusion matrix. True values are in the rows and predicted values in the columns

Conclusion

As companies grow and receive more complaints getting them to the correct department quickly and accurately is crucial. We found that with our final random forest model we were able to correctly classify 80.4% of the complaints. We also found that the top words that helped us classify a complaint were report, debt, credit, card, and pay. We found that the department that received the most complaints, Credit reporting, was often predicted in place of some of the smallest departments such as Vehicle loans and Personal loans. We also used our model to predict the departments for the 10 new complaints.

A shortcoming of the approach we used here is that we did not fully explore other potential models for multiclassificiation. Ideally we would have a comparison against neural nets, quadratic discriminant analysis, logistic regression and other methods. Another major shortcoming is that we did not experiment with balancing classes. If we had under or oversampled our data to balance it we may have avoided the prediction bias towards the Credit reporting department.

For future analyses we would suggest using sampling methods to balance the dataset, trying more models and tuning the used models more. We would also suggest further exploration of the effect of adding more words to the keyword bank and of adding other variables like quantity of censoring and quantity of numerics to the explanatory variables.

Teamwork

Max: Abstract, Problem Statement and Understanding, and Conclusion
Josh: Describe Proposed Methods and Model Selection/Justification/Evaluation
Both: Results, Fit models to compare, and Feature selection