Complaints Classification

Carly Lundgreen and McKay Christensen

April 14, 2021

Introduction

In order to quickly, and accurately, assess the needs of their customers, companies need to be able to understand and address concerns and complaints that arise from use of their products. The saying goes that "the customer is always right". If a company take this to heart, they will likely do everything in their power to assure that their customers' needs are taken care of. The difficulty then arises of how to assess and address these concerns in the quickest way possible. Part of this process is understanding exactly to which department inside a company a customer's complaints need to be sent. In this analysis, the researchers will demonstrate the ability of two models to accurately predict the correct department to which each complaint should be sent.

Data and Problem

The data from this analysis comes from a set of ∼125,000 complaints sent to companies via email. The data set also contains one of the nine departments in the company to which the complaints were sent. The purpose of this analysis is to build a model that will ease this process. If the company can accurately assign each complaint to a specific department without having a middleman manually read and assign each complaint, the lag time for complaint responses can be dramatically reduced.

The difficulty of this problem is to identify possible predictors to understand the complaints. Due to the simplicity of the data set, feature engineering was necessary to extract useful information from each complaint. The researchers decided to identify the most common words in the complaints, grouping by each department. This resulted in ∼65,000 unique words (including grammar errors, symbols, etc.) in the complaints and ∼168,000 word-department combinations.

Identifying the frequency of words used in each department's complaints is only the first step. With thousands of words that could be used to pair departments and complaints, it would be unreasonable to assume that all words could be useful. After collaborating, the researchers decided that no more than 750 words would be computationally reasonable to classify the complaints. To account for the unbalanced amount of complaints being sent to each department, more of the common words were taken from the departments with more words in their complaints. For example, the largest proportion of the indicator columns in the final data set comes from the Credit Reporting department because Credit Reporting had the most observations in the original data set. Table 1 shows the number of most common words selected from each department. Before these words were selected, stopwords (words that contribute only to grammar or add no concrete value) were removed. These words were selected from a database of stopwords stored in R. This prevented words such as "and" or "the" from being considered as predictors.

Department	# of Words Used
Checking/Savings	45
Credit/Prepaid Card	76
Credit Reporting	352
Debt collection	144
Money Transfer	18
Mortgage	61
Payday/Title/Personal Loans	15
Student Loans	26
Vehicle Loans/Leases	18

Table 1: Number of words sent for feature engineering

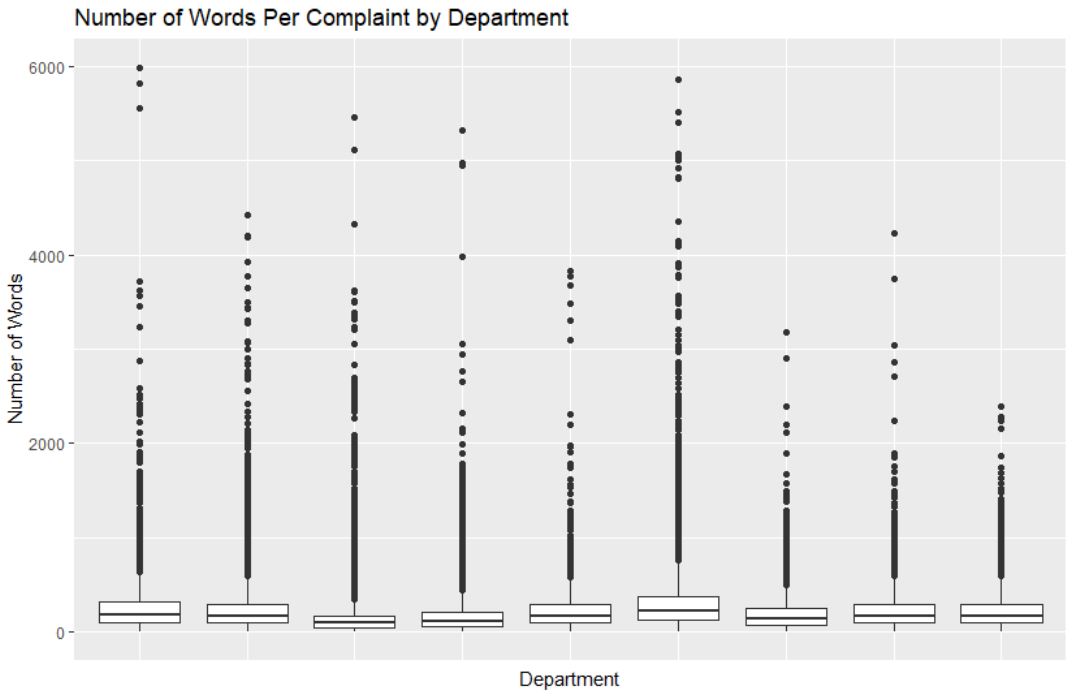
Department	Mean of Words Used
Checking/Savings	257.75
Credit/Prepaid Card	240.98
Credit Reporting	154.09
Debt collection	168.87
Money Transfer	240.04
Mortgage	324.61
Payday/Title/Personal Loans	206.02
Student Loans	240.33
Vehicle Loans/Leases	240.99

Table 2: Mean number of words per complaint

After removing these stopwords, the researchers noticed that a large proportion of the 750 selected words were appearing in multiple departments. Including these words as predictors would decrease the relative 'importance' of each word because the algorithm may have a difficult time identifying to which department each word should be assigned. To combat this, any word that was identified as being important to more than two departments' complaints was removed from consideration as a predictor variable. After removing these words, 401 words remained. The data set was then expanded so that each of the 401 words became an indicator column that denoted whether that particular word appeared in the complaint. After creating these columns, a final variable was added that indicated length (i.e. number of words) in each complaint.

EDA

Performing exploratory data analysis on text strings can be difficult and, oftentimes, non-informative. However, in addition to Table 2 which compares the average number of words per complaint by department, we humbly offer a simple boxplot to appease the yearning statistical spirits of all who read this report.



This boxplot compares the number of words per complaint and illustrates just how much variation there is within each comment. Most of the departments appear to have similar medians and IQR values. Each department also has many outlier data points, representing abnormally long complaints.

Methods and Models

Random Forest

The first machine learning algorithm presented here is the random forest. To take a step back, decision trees are the initial building blocks. A decision tree is a way to classify, or categorize, data into bins. Each bin of a decision tree is built from "branches" of a tree where a certain "branch" is created after data is split based on a criteria from a single predictor. The predictor used at each split of a branch is decided by the greatest reduction in classification error.

A random forest is built on the backs of many decision trees built from many training and testing data sets. After each tree is built, new training and testing sets are formed. However, it is reasonable to assume that the compilation of training and testing data sets built from the original data set would be very similar, having come from the data original full data set. This leads to correlated trees (random training/testing data sets should have similar features to the full data set). To combat this, only a selection of m predictors is considered at each split of a branch, where `m` $<$ `P` (number of total predictors). At each stage in the splitting process, the `m` predictors are randomly selected from the `P` total predictors. By forcing the model to make selections based on subsets of the predcitors at each split, each tree in the forest is formed in a decorrelated manner from the others.

The beauty of a single decision tree is that it is very interpretable. When each observation is separated into bins, it makes sense that it is similar to the other observations contained in that bin. However, when random forests are built, interpretability is mostly lost because interpreting hundreds of trees all at once is not feasible. Because of this predictions are the most useful output from random forest models.

Gradient Boosting Classification

Gradient boosting is another useful ensemble machine learning method for classification. Models are fit sequentially on residuals from previous models, with the first model being fit to the data set as a whole. The first model, which can be denoted $f_0(x)$, which is a weak learner that classifies $\mathbf{y}$ given a set of predictors $\mathbf{x}$. The model is then adjusted according to the direction of the gradient (which is the direction of the negative log likelihood) to determine an optimal model that will return $\hat{\mathbf{y}}$, or the final classifications. Very generally, this relationship can be represented as follows:

$$F_0(\mathbf{x}) = f_0(\mathbf{x})$$

$$F_m(\mathbf{x}) = F_{m-1}(\mathbf{x}) + \Delta_m(\mathbf{x})$$

Where $\Delta_m(\mathbf{x})$ can be represented more specifically as $\gamma\frac{dL}{dx}$, where L is the log likelihood and γ is a learning rate parameter that determines how large of a "jump" to take in the direction of the negative gradient. $\Delta_m(\mathbf{x})$ is trained on the residuals from the previous model, where 'residuals' in the classification paradigm for a given model are the negative gradients of some differentiable loss function.

The negative gradient can be estimated by the depth, d that minimizes logarithmic loss at each iteration. A particularly helpful variant of generalized gradient boosting is stochastic gradient boosting, which only utilizes a certain proportion of data at each iteration of the algorithm. This is implemented automatically via `xgboost` in `R`.

Justification - Tuning the Models

In machine learning, it is common for the algorithms used to not have model assumptions. Speaking specifically at random forests, this makes sense because each decision tree in a forest classifies observations based on similar predictor characteristics, and not on any overarching distribution. While assumptions are not important for this algorithm, justification of tuning parameters in the chosen model is important.

In random forests, the researchers chose to focus on the tuning parameters of number of trees (`n.trees`) in each forest and number of predictors (`mtry`) to randomly select at each branch split. A grid was selected on `n.trees` per forest between (50, 250) and `mtry` between (25, 75). From the search, the optimal combination to minimize misclassification rate was `n.trees` of 150 and `mtry` of 50.

Similar to random forests, the gradient boosting algorithm does not have any implicit assumptions that often accompany statistical learning (rather than machine learning) algorithms. For the gradient boosting model, the learning rate parameter was tuned, along with the tree depth (`max_depth`), and number of trees (`nrounds`). Due to the size of the data set, which led to a considerably heavy computational load, a portion of the data (33%) was selected as a training set, and then a 5-fold cross-validation procedure was conducted on 20% of that training subset. The selected tuning parameters were a `max_depth` of 8, `nrounds` of 30, and learning rate parameter of 0.03.

Results

Recalling to the research questions, in this analysis the researchers were hoping to build a model to accurately classify complaints and send them to the department to which the complaints pertain. In Table 3, a comparison of the two models can be seen of the model accuracy on training data sets. The random forest model was able to correctly identify ∼78% of the complaints and the gradient boosting model was correctly categorizing complaints at ∼75%. Although not completely satisfied with these results, the researchers were content knowing that they are able to identify approximately 80% of complaints. This will dramatically reduce the amount of time needed to sift through complaints by hand and classify them one at a time.

The `xgboost` package in `R` has implementation that calculates accuracy for each class in addition to reporting the overall accuracy. Table 4 compares the accuracies for each of the nine departments.

Model	Accuracy
Random Forest	0.782
Gradient Boosting	0.753

Table 3: Comparing accuracy between models

Department	Accuracy
Checking/Savings	0.793
Credit/Prepaid Card	0.795
Credit Reporting	0.841
Debt Collection	0.819
Money Transfer	0.658
Mortgage	0.887
Payday/Title/Personal Loans	0.525
Student Loans	0.789
Vehicle Loans/Leases	0.641

Table 4: Accuracies from each of the departments for gradient boosting model

The model performs quite well in classifying complaints going to the Mortgage, Credit Reporting, and Debt Collection departments. It breaks down in attempting to classify complaints to the Payday/Title/Personal loans and Money Transfer departments. This could be occurring because payday/title/personal loan department complaints have many common words that overlap with the money transfer department, although a further analysis would be needed to confirm this.

Figure 1 below illustrates some of the most important words, where importance is measured by the word's frequency of appearance in the raw data, that aided these models in identifying the department to which a complaint should be sent. A discussion of these features can be found in the following section.

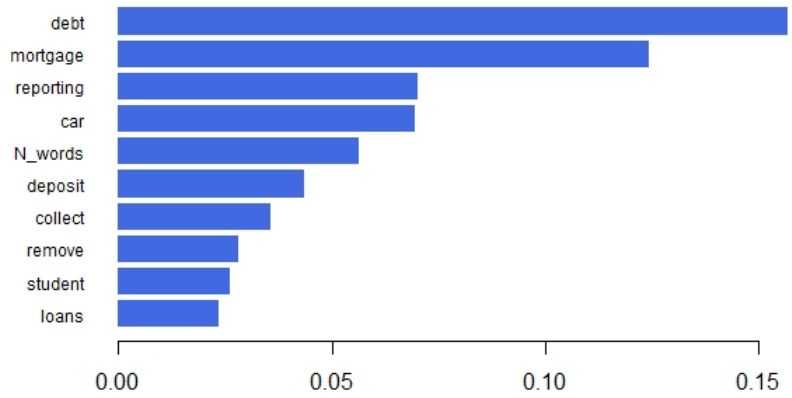


Figure 1: Top 10 Most Important Features

With the 125,000 complaints used to train and test the models, the researchers were also provided with 10

observations upon which they will test the accuracy of their models. Due to the complexity of the complaints, the full data will not be shown, but rather the researchers decided to number these complaints from 1-10 in the order which they were received (seen in column "Obs" of Table 5). Predictions for these observations can be seen as well in the table. As a note, both the random forest and gradient boosting models predicted the same departments for each of the 10 test observations.

Obs.	Model Prediction
1	Credit Reporting
2	Credit Reporting
3	Credit Reporting
4	Mortgage
5	Debt Collection
6	Credit Reporting
7	Mortgage
8	Credit Card or Prepaid Card
9	Debt Collection
10	Student Loan

Table 5: Predictions for 10 new observations

Conclusions

Classifying text strings through processes such as the ones presented is no simple task. It is not always clear which department in a company will handle every complaint. Certain departments have different specialties that make them uniquely qualified to answer various customer complaints. That being said, the models explained in this analysis were able to correctly predict between 70-80% of the same departments that had previously been identified. Errors in the model prediction are inherent and we feel that the prediction accuracy speaks to the dramatic improvements that modeling the complaints can make on time efficiency.

The two most important predictors for these models were the words 'debt' and 'mortgage.' This aligns with the fact that the credit reporting, debt collection, and mortgage departments had the highest accuracies as seen in Table 4. Another important feature was the number of words in the complaint. It seems that some departments tend to receive more wordy complaints than others.

In order to classify the department to which a particular complaint should be sent, the predictors that were identified and created in the data cleaning portion of this analysis were used. Indicator columns were then created from all the words that were identified as being the most important words from each of the nine departments.

Feature engineering using text strings can be taken in many directions. Here the researchers suggested a method of constructing data features based on the most common words from each department's complaints. In future studies, feature engineering, such as TF-IDF, could improve the accuracy of the model presented here. Along with TF-IDF, the researchers would consider building out other models such as neural networks that could potentially improve on current work.