Analysis of Consumer Financial Protection Bureau Complaint Data

Matthew Theisen

Executive Summary

Overall, the identification of companies with the highest rates of compensation for consumer complaints was successful. Additionally, the identification of products whose related complaints were most likely to be disputed, were identified with mortgages having both the highest dispute rate and the largest number of complaints, representing. A predictor for consumer dispute was generated, and the receiver operating curve was plotted.

Data Source Introduction

A database of complaints to the Consumer Financial Protection Bureau is available for download (direct link: https://data.consumerfinance.gov/views/s6ew-h6mp/rows.csv). The data in .csv format has 16 columns. The .csv can be loaded as a dataframe in R and the header names determined with the following lines of code, for example:

```
dat <- read.csv("D:/StatsFun/Consumer_Complaints.csv", header = TRUE)
headers = names(dat)
```

Table 1. Database column headers/features

"Date.received"	"Product"
"Sub.product"	"Issue"
"Sub.issue"	"Consumer.complaint.narrative"
"Company.public.response"	"Company"
"State"	"ZIP.code"
"Submitted.via"	"Date.sent.to.company"
"Company.response.to.consumer"	"Timely.response."
"Consumer.disputed."	"Complaint.ID"

There are roughly ~500,000 rows ("instances"), though new complaints are added continuously. Most columns ("features") exist as factors at different levels. For example, the "Product" product is split into the following factors:

Table 2. Available factors/levels in the "Product" feature

"Bank account or service"	"Mortgage"
"Consumer Loan"	"Other financial service"
"Credit card"	"Payday loan"
"Credit reporting"	"Prepaid card"
"Debt collection"	"Student loan"
"Money transfers"	

The database contains a number of features which could be used to, among other things, look for companies with patterns that indicate systemic fraudulence, look for areas to improve customer experience, look for geographic areas with high or low complaint rates etc.

Initial Data Cleaning

For analysis on this data, some instances are unlikely to be helpful, and can thus be removed. For instance, the feature "Company.response.to.consumer" has levels “In progress” and “Untimely Response”. These are unlikely to be useful in many analyses and can therefore be removed:

```
in_progress <- which(sapply(dat[13], function(x) (x=="In progress")))
untimely <- which(sapply(dat[13], function(x) (x=="Untimely response")))
dat <- dat[-c(in_progress,untimely),]
```

Analyzing Companies’ Responses

Complaints processed by the CFPB are first analyzed by the agency, which then contacts the company on behalf of the complainant to get a response, which is recorded in the “Company.response.to.consumer” feature. Companies whose complaints require significant responses may be more systemically fraudulent. An analysis of this type could be a starting point, but not definitive, in identification of systemic fraud. The exact interpretation may depend on how companies decide to respond, and how much leverage the CFPB has. For example, absence of action toward a complaint could be because of intransigence on the company’s part, rather than that the complaint was baseless to begin with. However, with imperfect business knowledge of the process, I will proceed under the assumption that a higher response rate of action indicates more inappropriate behavior by a company and thus a higher likelihood of systemic fraud.

The “Company.response.to.consumer” feature (may be called ‘response’ for brevity) is separated into six factors, other than the ones already removed. To simplify the analysis, the levels are flattened from six to two, “Yes”, if a response by the company was required, or “No”, for no response:

```
levels(dat$Company.response.to.consumer)[levels(dat$Company.response.to.consu
mer) == "Closed with explanation"] <- "No"
levels(dat$Company.response.to.consumer)[levels(dat$Company.response.to.consu
mer) == "Closed without relief"] <- "No"
levels(dat$Company.response.to.consumer)[levels(dat$Company.response.to.consu
mer) == "Closed"] <- "No"

levels(dat$Company.response.to.consumer)[levels(dat$Company.response.to.consu
mer) == "Closed with monetary relief"] <- "Yes"
levels(dat$Company.response.to.consumer)[levels(dat$Company.response.to.consu
mer) == "Closed with non-monetary relief"] <- "Yes"
levels(dat$Company.response.to.consumer)[levels(dat$Company.response.to.consu
mer) == "Closed with relief"] <- "Yes"
```

The unused factors can now be removed:

```
dat$Company.response.to.consumer <- factor(dat$Company.response.to.consumer)
```

The companies can now be compared on their likelihood of requiring a response. First, it is useful to calculate a “uniform prior”, or the fraction of all complaints with a “Yes”:

Here, dat[13] is the “Company.response.to.consumer” column of dat. Now, we can generate a table

```
Uniform_prior <- table(dat[13])[2]/(table(dat[13])[1] + table(dat[13])[2])
```

which splits the instance space according to company and response:

```
j <- with(dat, table(Company.response.to.consumer,Company))
```

To refine the analysis, eliminate any companies with less than 25 complaints:

```
jj <- j[1,]+j[2,]
jjj <- jj > 25
newj <- j[,jjj]
```

Now, we can use Pearson’s Chi Squared Test to determine the statistical significance of differences from the uniform prior. We first calculate the chi-squared test statistic (chisq_val). In the first column of the chisqval array, we put the p-value for that company (null hypothesis, the population mean of “Yes” fraction for a company is equal the uniform prior). There is one degree of freedom in this chi-squared distribution, since there are two categories (“Yes” and “No”). In the second column, we put the ratio of fraction “Yes” responses for a company to the uniform prior—effectively a “lift” factor for fraction of “Yes” responses.

```
chisqval <- array(0, dim=c(dim(newj)[2],2))

for (i in 1:dim(newj)[2])
{
       Total = newj[1,i]+newj[2,i]
       Expected = Total*Uniform_prior
       Yes = newj[2,i]
       chisq_val = (Yes - Expected)^2/Expected
       chisqval[i,1] = 1-pchisq(chisq_val,1) ## p value
       chisqval[i,2] = (Yes/Total)/Uniform_prior ## “lift ratio”
}

rownames(chisqval) <- names(newj[1,])
```

A p-value less than 0.05 would indicate significance by the usual standard. However, to minimize type I error, we can use a multiple hypothesis testing correction such as the Bonferroni. When testing a hypothesis across 740 companies, Bonferroni indicates the p value for significance should be 0.05/740 = 6.8E-5. This ensures the total type I error rate is 0.05 or less.

We can now rank the companies by the 'lift' coefficient and significance, and for good measure, add the "No" and "Yes" counts as columns 3 & 4, respectively.

```
rownames(chisqval) <- names(newj[1,])
lift_inds <- sort(chisqval[,2],decreasing = TRUE, index.return = TRUE)$ix
sig_inds <- sort(chisqval[,1],decreasing = TRUE, index.return = TRUE)$ix
lift_list <- chisqval[lift_inds,]
sig_list <- chisqval[sig_inds,]

badnames <- names(lift_list[,1])
j[,badnames]
lift_list <- cbind(lift_list, array(0, dim=c(740,2)))
lift_list[,3:4] <- t(j[,badnames])
```

The top 10 companies ranked in order of "lift ratio" for response rate is the following:

```
                                                              [,1]        [,2]  [,3]  [,4]
Lxxxx Pxxxxxxxxx Sxxxxxxx, Lxx.                       0.000000e+00 4.778919312     0    29
Txx Rxxxxxxxxx Mxxxxxxxxx Sxxxxxxx Cxxxxxxxxxx        0.000000e+00 4.778919312     0    76
Wxxxxxx Pxxxxxxxxxxxx                                 0.000000e+00 4.778919312     0   127
Axxxxx Ixxxxxxxxx Lxx                                 0.000000e+00 4.743288973     8  1065
Txxxxxx Pxxxx Sxxxxxxxx Lxx                           0.000000e+00 4.691232719     2   107
Mxxxx Cxxxxxxxxxx, Inc                                0.000000e+00 4.681390347     1    48
Cxxxx Pxxxxxxx, Lxx                                   0.000000e+00 4.673501975     3   133
Exxxxxxxxxx Vxxxxxxx, Lxx                             0.000000e+00 4.642222536    25   849
Txxxxxx Axxxx Mxxxxxxxxx, x.x.x.                      0.000000e+00 4.619622002     7   203
Nxxxxxx Rxxxxxxx Cxxxxx                               0.000000e+00 4.587762540     3    72
Txx Lxx Oxxxxxx ox Mxxxxxxx D. Bxxxx & Axxxxxxxxx     0.000000e+00 4.583861381     2    47
```

Company names have been partially obfuscated to avoid drawing attention to companies who may be doing nothing wrong, note the previous caveat about imperfect business knowledge of how the complaint system works. The maximum lift ratio is 1/uniform prior rate = 1/0.2092523 = 4.778919. In this selection, all p values are effectively 0, indicating significance.

Improving Customer Experience

Another organization goal of the CFPB may be to improve the customer/complainant experience. This would be reflected in decreasing the fraction of consumers disputing the result of the complaint in the "Consumer.disputed." feature. One reasonable feature to compare this against would be “Product”. Using this comparison, we can determine which products are the most likely to generate complaints, the results of which are disputed by consumers. Then, if we can find which types of products generate the most disputes, staff training can be focused on issues related to those products. The following code generates a list of products by disputation rate:

```
Uniform_prior <- table(dat[15])[2]/(table(dat[15])[1] + table(dat[15])[2])

j <- with(dat, table(Product,Consumer.disputed.))
h <- chisq.test(j)
chisqval <- array(0, dim=c(dim(j)[1],2))

for (i in 1:dim(j)[1])
{
       Total = j[i,1]+j[i,2]
       Expected = Total*Uniform_prior
       Yes = j[i,2]
       chisq_val = (Yes - Expected)^2/Expected
       chisqval[i,1] = 1-pchisq(chisq_val,1)
       chisqval[i,2] = (Yes/Total)/Uniform_prior
}

rownames(chisqval) <- names(j[,1])
lift_list <- chisqval
lift_list <- cbind(lift_list, array(0, dim=dim(lift_list)))
lift_list[,3:4] <- j
lift_inds <- sort(chisqval[,2],decreasing = TRUE, index.return = TRUE)$ix
lift_list <- lift_list[lift_inds,]
```

The resulting product ranking:

```
                                [,1]      [,2]   [,3]  [,4]
Mortgage                0.000000e+00 1.1471127 132962 41009
Consumer Loan           1.047162e-12 1.1155571  14256  4240
Credit card             7.837309e-02 1.0156698  48589 12816
Bank account or service 7.099280e-04 0.9689238  46262 11501
Student loan            1.606972e-02 0.9561518  11786  2882
Other financial service 4.305381e-01 0.9146402    337    78
Debt collection         0.000000e+00 0.8996741  72837 16520
Credit reporting        0.000000e+00 0.8177912  67155 13565
Payday loan             2.654502e-07 0.8037916   2795   553
Money transfers         0.000000e+00 0.6717227   2885   462
Prepaid card            8.437695e-15 0.6293880   1858   276
```

This ranking indicates that complaints based on Mortgage products are most likely to be disputed by the consumer after receiving a response from the company. Thus, to improve consumer experience, focusing staff training on mortgage products is likely to be most meaningful and useful. As a bonus, Mortgage is also the largest category, by number of complaints.

Developing a Predictor of Consummer Dissatisfaction

So far I have used ratios and the chi squared test to analyze this database. Perhaps more interesting would be a predictor which can analyze complaints as they come in, and flag a supervisor on calls which are most likely to result in consumer disputes.

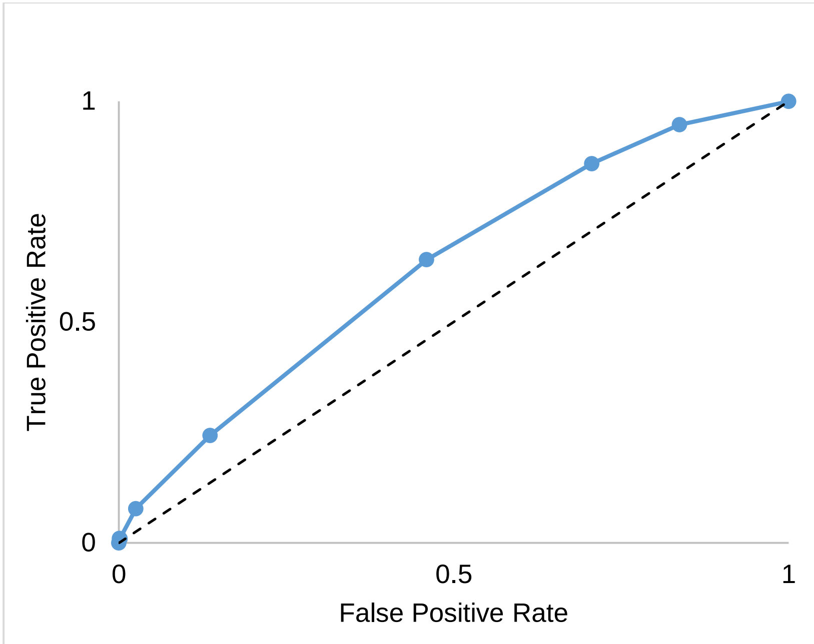
R package randomForest allows us to construct such a predictor, automatically. The randomForest method is very powerful since it is surprisingly resistant to overfitting. This is because its estimates are based on the mode of multiple predictors, each of which is built on only a subset of the instance space. Thus, particularities in the data which are not reflective of general trends are usually outvoted. Here is one method for it:

```
yesrows = which(dat[15] == "Yes")
norows = which(dat[15] == "No")
rows = c(yesrows[1:5000], norows[1:5000])
outcome_ind = 15
toInclude = c(2,3,4,5,9,11)
testdat <- dat[rows,toInclude]
outcome <- dat[rows,outcome_ind]
array <- model.matrix(~., data=testdat[1])

for (j in 2:length(toInclude))
{
       bbb <- model.matrix(~., data=testdat[j])
       array <- cbind(array,bbb)
}

library(randomForest)

forests <- randomForest(x = array, y = outcome)
```

This method uses 10,000 rows of the data set to generate a random forest, and uses equal numbers of yes and no instances. Here is a summary of the forest generated:

```
               Type of random forest: classification
                     Number of trees: 500
No. of variables tried at each split: 17

        OOB estimate of  error rate: 40.63%
Confusion matrix:
      No  Yes class.error
No  2738 2262      0.4524
Yes 1801 3199      0.3602
```

The randomForest algorithm maximizes accuracy, so using a 0.5 prior minimizes the accuracy of any uniform classifier. We can more fully characterize the classifier by varying the prior and generating a receiver operating characteristic (ROC) curve, which plots the true positive rate vs. the false positive rate.



Code

Full code is available at: https://github.com/theis188/consumer-complaints