Justin Tsao

Improving Customer Service in Finance with Machine Learning

Table of Contents

I. INTRODUCTION

This project will focus on the customer service side of the finance industry. The problem I want to solve is how to increase customer service given the vast amount of data on customer complaints. From experience, I can tell you that there are a lot of complaints from the service of a firm to the products themselves. The clients for this project can be all financial firms who want to improve their responsiveness of their customer service.

The data is coming directly form the database of the Consumer Financial Protection Bureau, or CFPB. The data the project will be working can easily be downloaded and queried here https://data.world/cfpb/consumer-complaints. As of 5/20/2019, the data set is comprised of 1,284,185 individual rows of complaints, updated frequently. I plan to conduct visual and statistical analysis on the entire data set which includes 18 columns. From the data set, I only plan to use the complaints where the customers give a written

response. The reason being, I want to use machine learning and natural language processing techniques on the written customer complaints as well as the categorical columns. Taking only the data with written customer complaints, leaves 383,840 rows of data.

To decide whether a complaint is handle in an 'appropriate' manner I will use whether the complaint was handled in a timely manner or not as a predictor. I use appropriate in quotes because the term 'appropriate' can be ambiguous but I assume it means that the complaint was handled as efficiently as it could. Not responding in time could mean the complaint was not resolved, difficult to handle, or maybe ignored due to lack of concern about the customer's complaint. The goal will be a machine learning model that can correctly identify these non-appropriate response times based on the features of data and the customer complaint. The results could be extremely valuable by giving insights on which complaints are the most difficult to handle. Being able to quickly identify the difficult complaints can be leveraged to improved customer service and satisfaction which will lead to higher customer retention.

II. DATA

As mentioned above, before any pre-processing and wrangling, the data contained 1,284,185 rows of data. The full data set was used for visual data analysis and inferential statistics. A SQL query was performed to only return 383,840 rows of data which had a consumer complaint narrative. The consumer complaint narrative contains a response about the complaint they filed. This narrative will be an important feature for machine learning using natural language processing (NLP). There were 7 columns that contained missing values. Of those seven, two of the features were removed because

they comprised of data that would be given after a complaint is filed. This means the two features, ‘company_public_response’ and ‘company_response_to_consumer’, would qualify for features to make predictions off of. The other columns with missing values were ‘sub_product’, ‘sub_issue’, ‘state’, ‘zip_code’, ‘tags’, and ‘consumer_disputed’. After examining the states feature, I decided to remove the state and zip code data as I did not think it will play a relevant role as a predictor, larger state with big cities, and higher populations had larger proportion of complaints. This could throw off predictions. For the sub issues and sub categories, the missing values were replaced with the string ‘None’. This made sense because not all categories of product and issues will have a sub category. The other columns had over 100k to 300k missing values. I decided to remove those completely without filling in data because they were categorical. The cleaning for these labels was pretty simple.

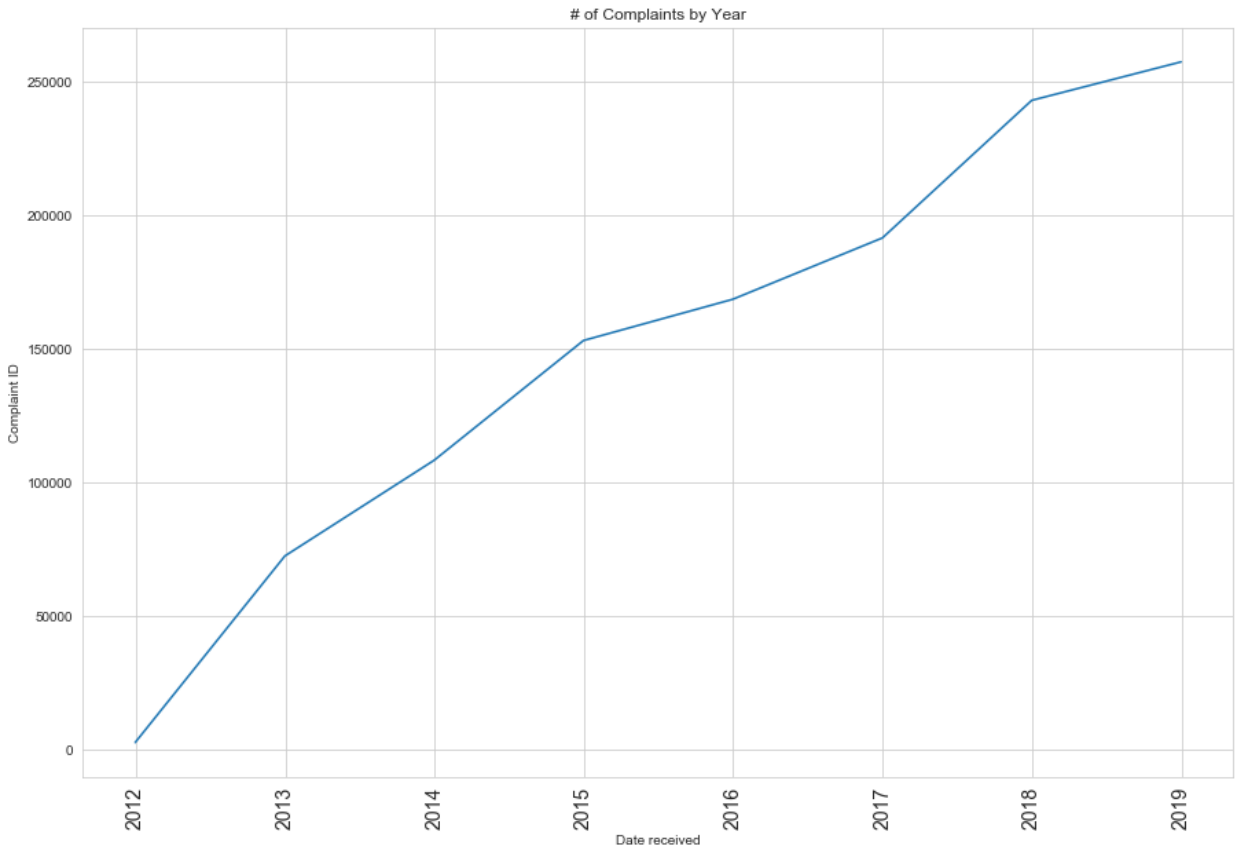
A majority of the cleaning was done to the customer complaint narrative. The first thing to note about the data in this column was that sensitive information like ID numbers or phone numbers were replaced by ‘X’s. Also, monetary values were encased in { }. The narrative data has to be pre-processed for use in NLP and machine learning. All punctuation, '!"#$%&\'()*+,-./:;<=>?@[\\]^_`{|}~', was removed. The ‘X’ values were removed as well as any extra white spaces. Since the narratives were typed in by customers there are chances of typos. One way to ease the number of types was to take care of word lengthening. An example of word lengthening would be ‘realllyyyy’ as way for the consumer to emphasize a point. A function was applied to the column to transform those words into a correct form, ‘realllyyy’ –> ‘really’.

For the following data pre-processing steps on consumer narrative, I have decided that it may easier to implement in SciKitLearn. Tokenization needs to separate each word into a list of words. Next, we can remove stop words like ‘is’ or ‘of’ that give little meaning to the narrative. We can also decide to stem and lemmatize words. Stemming is essentially removing suffixes like ‘-ly’ or ‘-ing’ from words. Lemmatization is changing words into their root form. For instance, ‘ran’ -> ‘run’ and ‘caring’ -> ‘care’.

I have decided on one feature set to include [product, sub_product, issue, sub_issue, consumer_complaint_narrative, company, submitted_via]. The predictor label will be ‘timely_response’. For a second feature set, I have decided to include a feature based on sentiment score. The second feature set is the same as the first with the added sentiment score feature which is a number between -1 and 1.

III. EXPLORATORY DATA ANALYSIS

The first thing I explored was trend of number of complaints per year. Also, keep in mind these number reflect the number of complaints submitted to the CFPB, there could easily be more complaints unsubmitted, or submitted directly to the financial firms themselves. I actuality, the number plotted below could be many times higher. Another thing to note is that most of data is categorical.



From 2013 to 2014 the complaints submitted to the CFPB increased by 149.53%
From 2014 to 2015 the complaints submitted to the CFPB increased by 141.42%
From 2015 to 2016 the complaints submitted to the CFPB increased by 110.09%
From 2016 to 2017 the complaints submitted to the CFPB increased by 113.64%
From 2017 to 2018 the complaints submitted to the CFPB increased by 126.90%
From 2018 to 2019 the complaints submitted to the CFPB increased by 105.93%

As you can see The CPFB takes in complaints from consumers and sends the complaints to firm the complaint was filed against. What we can infer from this is that the number of complaints has at least double each year. This could be due to increased adoption of technology and the ease to report complaints. Worse yet, we could also infer that customer service could be falling or that consumers are increasingly unhappy with their products. Most of the complaints come from products regarding mortgages, debt, and credit reporting.

There were a total of 18 categories of products:

Justin Tsao

Mortgage: 278249

Debt collection: 245218

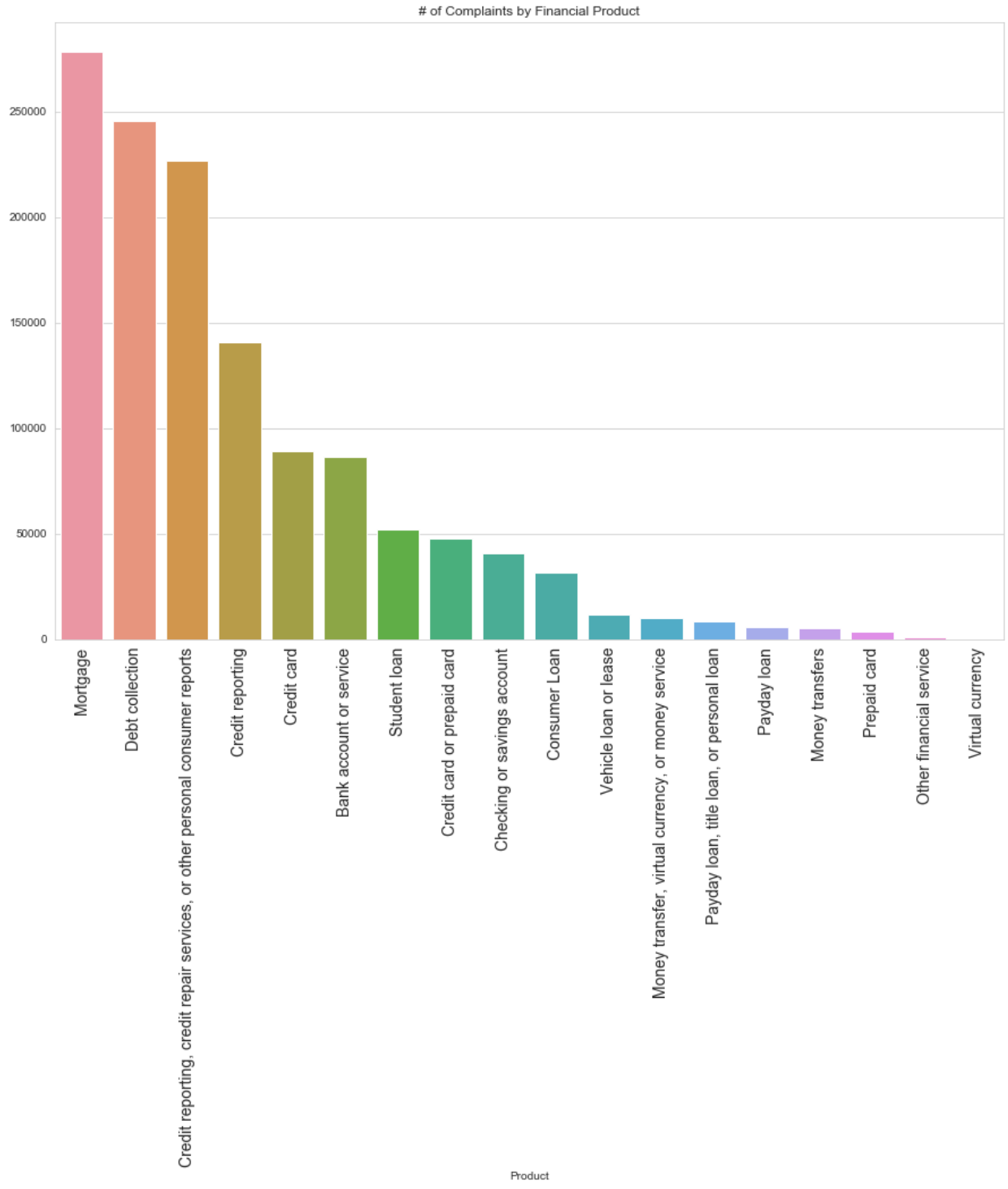
Credit reporting, credit repair services, or other personal consumer reports: 226781

Credit reporting: 140432

Credit card: 89190

Bank account or service: 86206

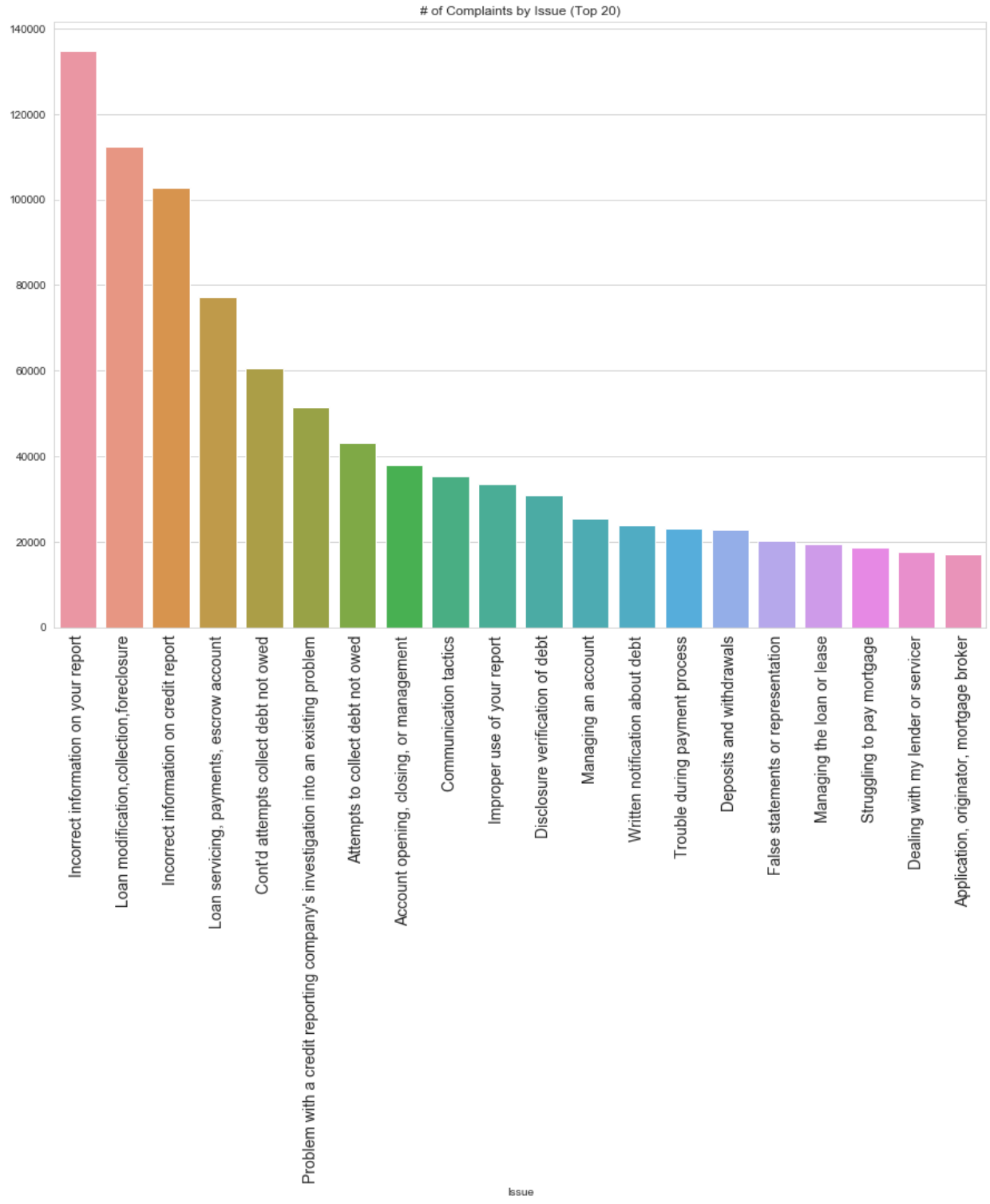
Justin Tsao


# of Complaints by Financial Product
250000
200000
150000
100000
50000
0
Mortgage
Debt collection
Credit reporting, credit repair services, or other personal consumer reports
Credit reporting
Credit card
Bank account or service
Student loan
Credit card or prepaid card
Checking or savings account
Consumer Loan
Vehicle loan or lease
Money transfer, virtual currency, or money service
Payday loan, title loan, or personal loan
Payday loan
Money transfers
Prepaid card
Other financial service
Virtual currency
Product

There are 76 categories of sub products. Other than credit related products (debt, loans, credit cards etc.), checking accounts stand out as the 2nd highest sub products with complaints against it.

The plot below shows the top issues resulting from those complaints.

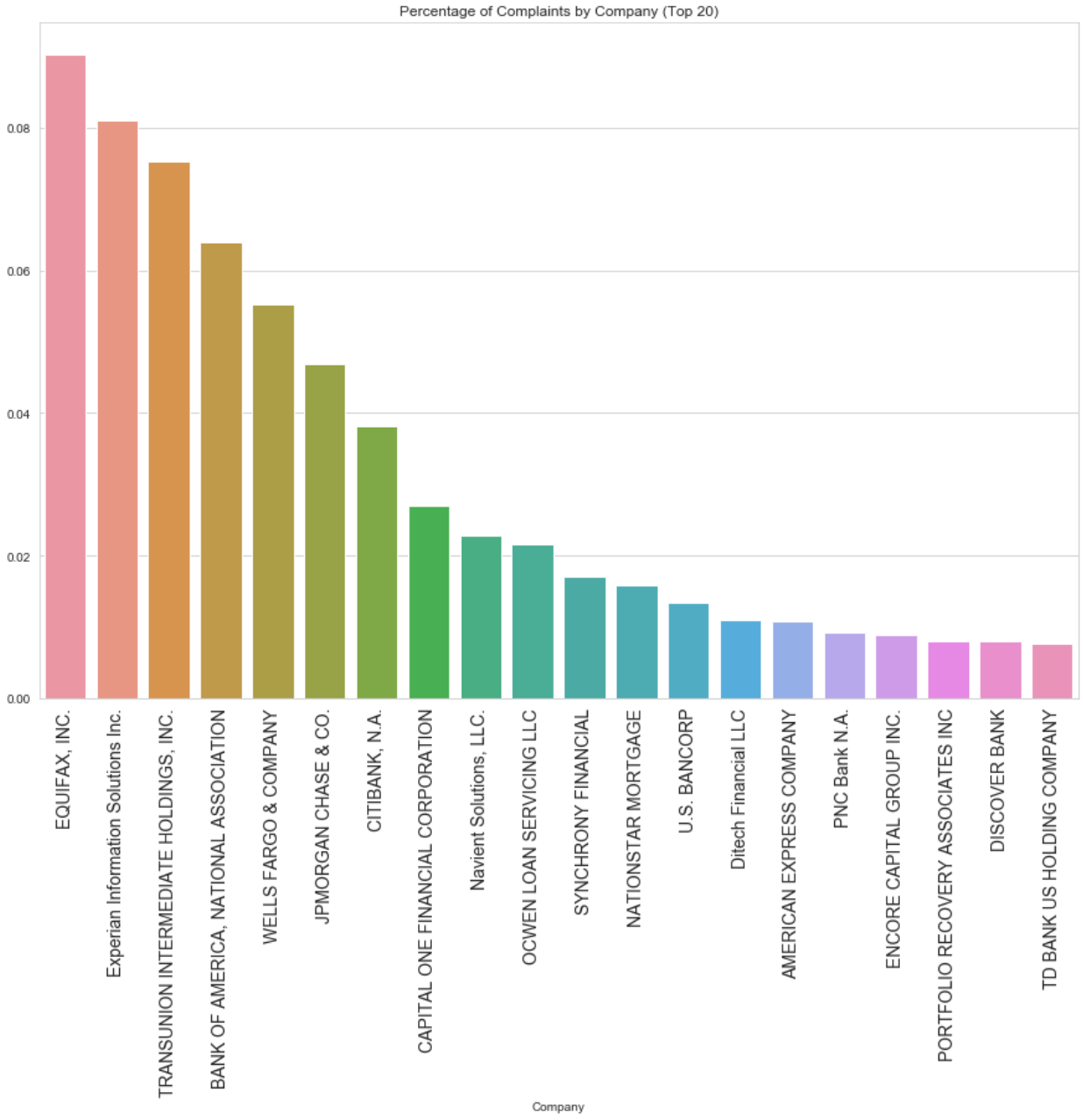
Issue
Incorrect information on your report: 134809
Loan modification, collection, foreclosure: 112311
Incorrect information on credit report: 102686
Loan servicing, payments, escrow account: 77333
Cont'd attempts collect debt not owed: 60687
Problem with a credit reporting company's investigation into an existing problem: 51498
Attempts to collect debt not owed: 43181
Account opening, closing, or management: 37961
Communication tactics: 35449
Improper use of your report: 33441
Disclosure verification of debt: 30800
Managing an account: 25535
Written notification about debt: 23766
Trouble during payment process: 23188
Deposits and withdrawals: 22851
False statements or representation: 20278
Managing the loan or lease: 19438
Struggling to pay mortgage: 18682
Dealing with my lender or servicer: 17630
Application, originator, mortgage broker: 17229



Many of these issues seem to stem from how consumer information is handled and reported. Perhaps this is a red flag that employees need to be trained to handle

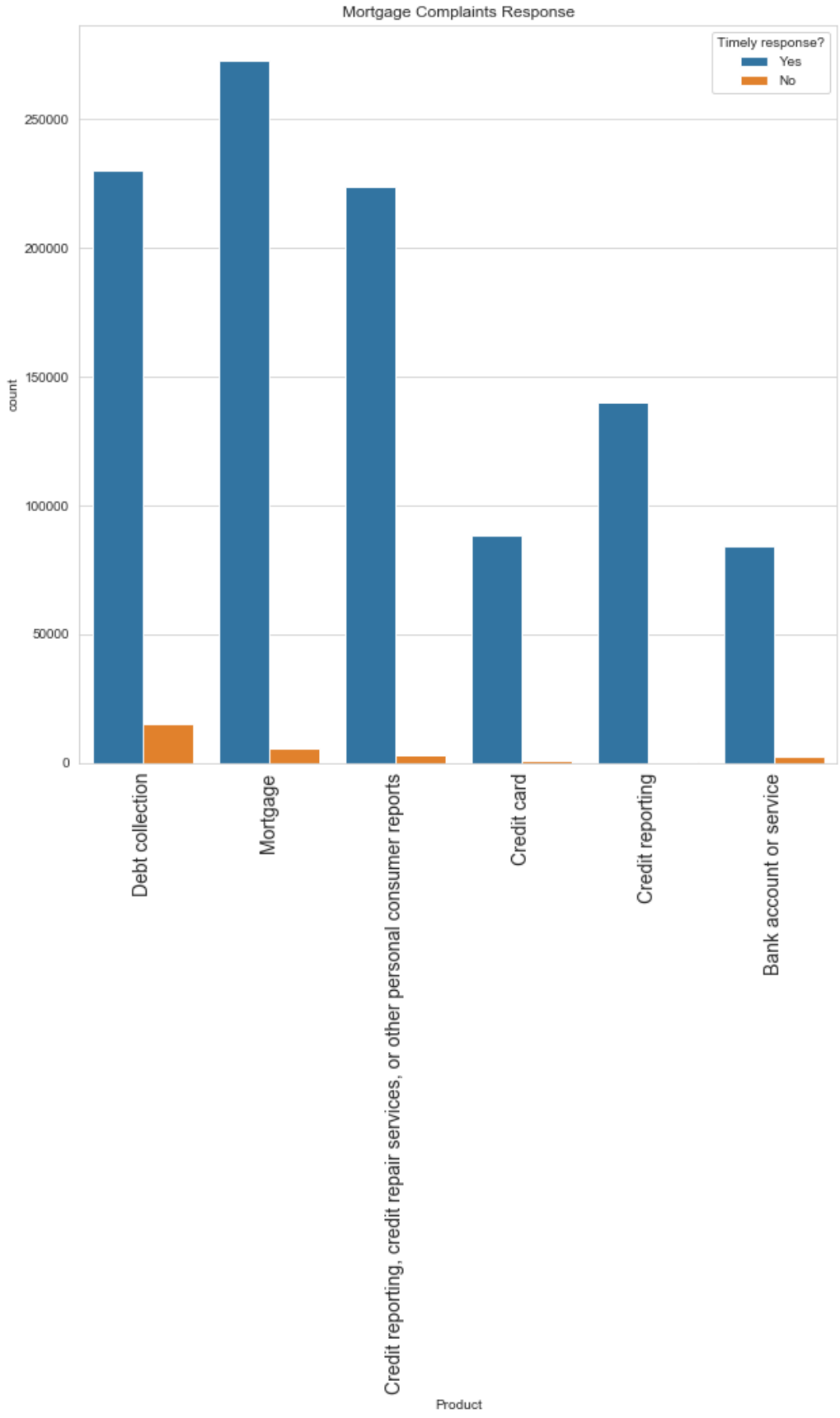
consumer information with better care both in use and when entering in data. From my experience working both at a bank and a mortgage lender, there are plenty of applications taken on a daily basis. Much of the information entered is sensitive information like ID numbers, SSN's, DOB, etc. Not properly communicating a product is also seen a lot in finance. At least in retail, there is always the rush of getting the next client in and out of the seat as fast as possible. After all, more sales means more commissions.

Below is a plot of financial firms and number of complaints. As a past financial professional, the graph above is not surprising at all. The biggest culprits are the credit reporting agencies and big banks. With the larger firms, it does make sense that there would be more over all complaints due to the size of their customer base. It is also harder to manage customer service and sales reps as a company becomes larger and the customer base increases.



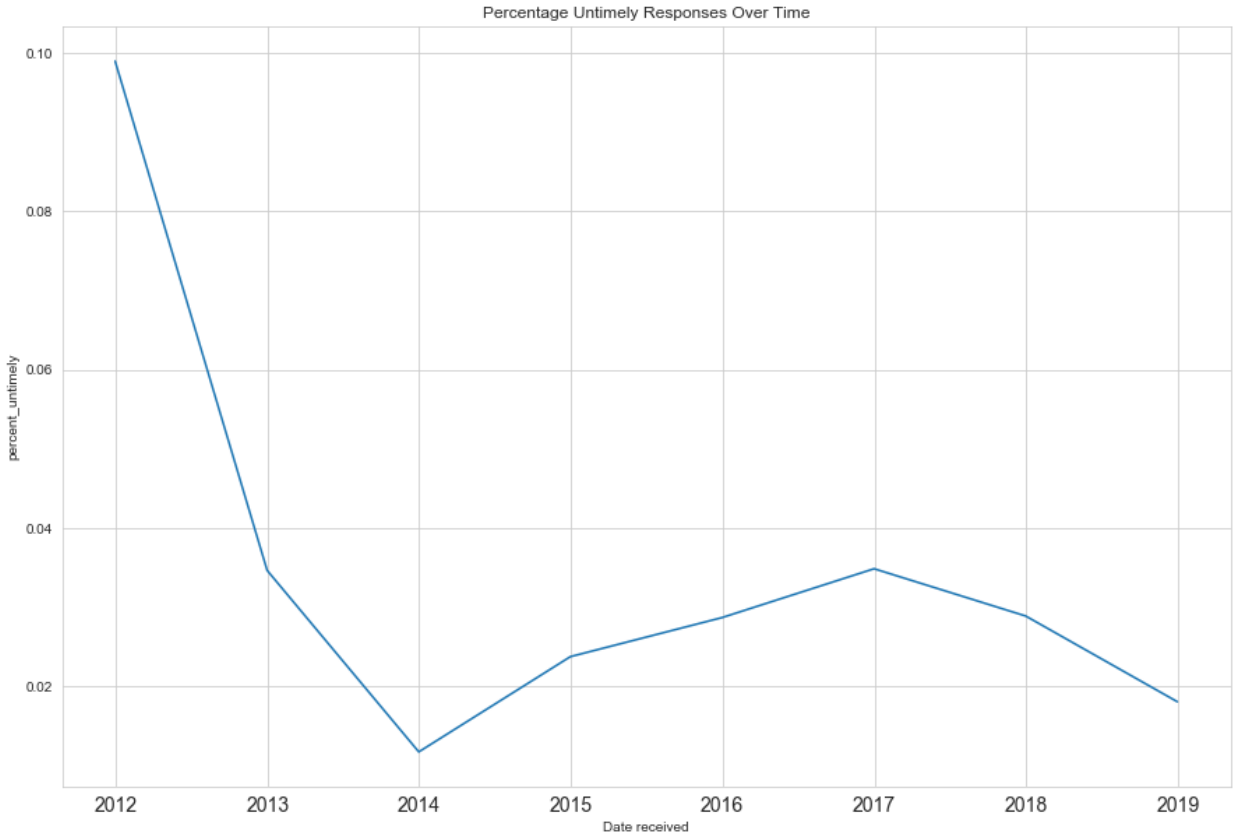
Looking at the full data, there were 1,251,987 responses which were answered in a timely manner. 32,198 were not handled appropriately. That number seems pretty low, but just because a complaint was handled in a timely manner does not mean it was handled in the best way. Also, 2.51% is reflected of complaints reported to the CPFB and is still around 32198 customers.

Justin Tsao


Mortgage Complaints Response
Timely response?
Yes
No
250000
200000
150000
100000
50000
0
count
Debt collection
Mortgage
Credit reporting, credit repair services, or other personal consumer reports
Credit card
Credit reporting
Bank account or service
Product

```
Percentage of untimely responses
Debt Collection: 6.22%
Bank Account: 2.58%
Mortgage: 1.99%
Credit Services: 1.41%
Credit Card: 1.10%
Credit Reporting: 0.21%
```

Though most of the complaints were against credit reporting/services products, the companies responded in a timely manner relative to companies responding to complaints about debt collection and bank accounts.



Based on CFPB's data, untimely responses have started to decrease over all from 2017 for financial firms over all. The last thing I want to remind readers, is that the numbers represent only complaints filed to the CFPB. Therefore, the actual numbers are higher. But the decrease in untimely responses is a good sign! Think about how many of these

claims are mortgage or credit based. Errors in customer service could be millions of dollars in combined losses for consumers and companies. Bad credit reporting could prevent customers form getting a loan from their dream house and thus a mortgage firm would lose out on this sale. Even worse, what if an error in customer service caused funds to not be released from a bank account to a consumer. This means bills go unpaid or late. Late fees rack up, and this can spiral out of control. From experience, I can tell you of a horror story about a teller who forgot to check the endorsement on the back of a check for a college student. Because the teller deposited the check with out seeing if there was an endorsement, the college student was with out money to pay for books, rent, or food for weeks. They had to wait for the check to be return and re deposited.

IV. FEATURE ENGINEERING

As I started training the machine learning pipelines and fitting the data, I tried to brainstorm ways to improve the results of identifying ‘un-timely’ responses. More on the results of the machine learning is in the section below. Basically, the models were classifying the ‘timey’ complaints with high accuracy but had a hard time classifying the ‘un-timely’ complaints. I first started tuning the hyper parameters to my models. There were improved results, but still wanted to find ways to improve results. I decided on using sentiment analysis.

Sentiment analysis attempts to analyze the written complaint as input and output a value between -1 and 1. Sentiment can be thought of as the view or attitude toward something. For this project, a negative sentiment score means the comment was viewed as having a negative connotation. A positive score reflects a positive attitude. I

want to note that sentiment analysis is useful in many cases but is not an exact science. You would expect, because these are complaints, that all the complaints would have a negative sentiment score. However, that was not the case. There were many complaints, both those handled appropriately and those not, that had positive sentiment scores. Overall, the 'un-timely' responses had an average score of -0.167 and the 'timely' responses had an average score of -0.03. So, based on that statistic, the sentiment analysis on average was producing a lower score for the 'un-timely' complaints, which was what I was hoping for.

To create the sentiment score feature, I ran all the complaints through the python package NLTK's SentimentIntensityAnalyer(). This sentiment module uses VADER:

"If you use the VADER sentiment analysis tools, please cite:

Hutto, C.J. & Gilbert, E.E. (2014). VADER: A Parsimonious Rule-based Model for Sentiment Analysis of Social Media Text. Eighth International Conference on Weblogs and Social Media (ICWSM-14). Ann Arbor, MI, June 2014.

"A new column would be created called 'sentiment_score' in the feature set.

V. MACHINE LEARNING

This project will be using the Python package scikit-learn for machine learning. The goal of the project will be to correctly predict and classify a complaint as 'un-timely' for predicting the 'timely_repsponse' label as False. As mentioned in the EDA above, the 'un-timely' responses were only about 2.5% of the responses. Therefore, accuracy would be a bad performance metric for this model. A better metric to gauge model performance would be log loss. Log loss is a better metric because, if the model predicts True for all 'timely_responses', then the accuracy would still be around 97%.

Log loss on the other hand, measures error. So we are looking for a log loss score as low as possible. One way to think about log loss intuitively is that it is better for the model to be less confident than confident and wrong about a prediction. To clarify, let's say the model predicts a True label with 90% probability but the label is actually False. The log loss score would be around 2.30. If the model predicts the label as True with 50% probability when the label is actually True, then the log loss is only 0.69. A lower log loss is better. As you can see, log loss punishes incorrect predictions. That being said, I have also included other performance metrics like classification reports, accuracy, and AUC in additional to log loss to evaluate model performance.

There were additional preprocessing steps that were included directly into the machine learning pipe line. The categorical features were one hot encoded. The written customer complaints, 'consumer_complaint_narrative', were adjusted for lemmatization or stemming. The written complaints were vectorized using stop words and ngram range of (1,2).

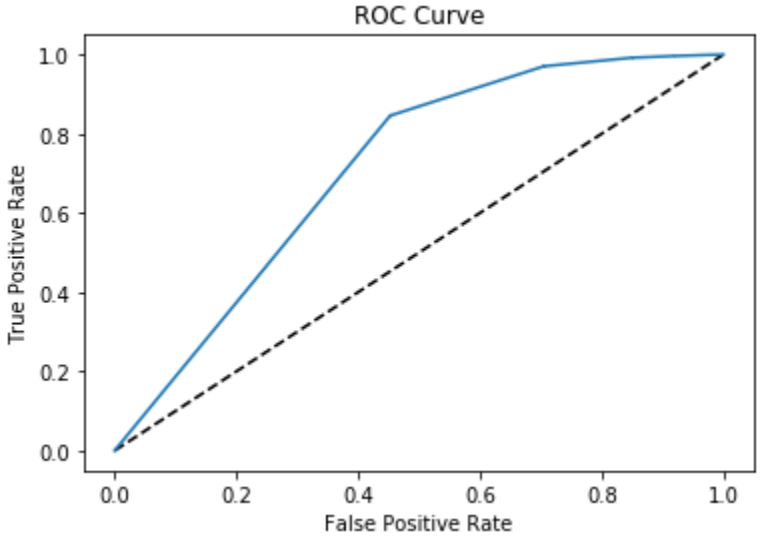
Base Model used gauge performance:

Since log loss is harder to gauge than something like accuracy, I decided to run untuned versions of a random forest model and use the log loss as a base to measure improvements to model performance.

Justin Tsao

Random Forest (Un-tuned)				
	Precision	Recall	F1-Score	Support
FALSE	0.44	0.05	0.08	2843
TRUE	0.97	1	0.98	93117

Log Loss	Accuracy	AUC
0.5128	0.9699	0.7124

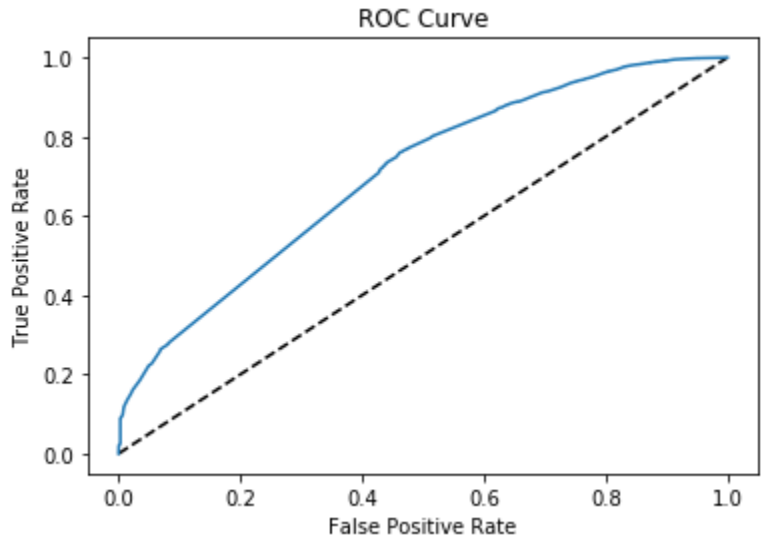


I used those metrics above as by baseline to improve model performance. High accuracy as expected, but want needs to be improved is the precision, recall, and f1-score of classifying the False labels. Also, we look to decrease the log loss and increase AUC. For this project, I will run 2 different machine learning models, random forest and logistic regression. The hyper parameters will be tuned and changes to the feature set will be adjusted. Then, we can look at the performance metrics again to see what allows our model to perform better.

Random Forest (Stemming and Hyper Parameter Tuning):

Random Forest (Stemming, Tuned)				
	Precision	Recall	F1-Score	Support
FALSE	0	0	0	2843
TRUE	0.97	1	0.98	93117

Log Loss	Accuracy	AUC
0.131	0.9704	0.7038



Although the log loss went down and accuracy went up slightly, the model was not able to predict any of the False labels. As a reminder, stemming is the process of removing suffixes like '-ly' from the end of words. Stemming the 'consumer_complaint_narrative' actually caused the model to perform worse. It performed worse because I am the goal is to increase the ability of the model to find False labels accurately. For the random forest model, I used a randomized grid search with cross fold validation.

```
#Hyper Parameters
n_estimators = [int(x) for x in np.linspace(start=10, stop=100, num=10)]
max_features = ['auto', 'sqrt']
max_depth = [int(x) for x in np.linspace(10, 110, num = 11)]
max_depth.append(None)
min_samples_split = [2, 3, 5, 10]
min_samples_leaf = [1, 2, 3, 4]
bootstrap = [True, False]

param_grid = {'clf__n_estimators': n_estimators,
              'clf__max_features': max_features,
              'clf__max_depth': max_depth,
              'clf__min_samples_split': min_samples_split,
              'clf__min_samples_leaf': min_samples_leaf,
              'clf__bootstrap': bootstrap}
```

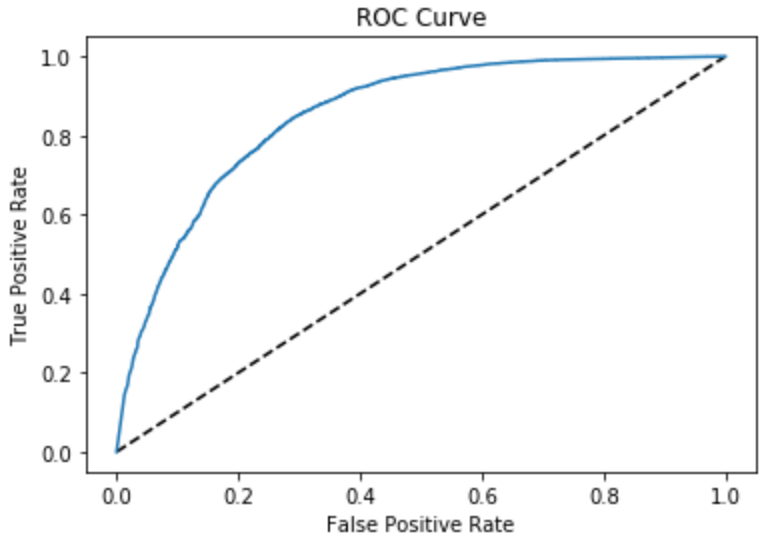
The hyper parameters that performed best were:

```
{'clf__n_estimators': 20,
 'clf__min_samples_split': 5,
 'clf__min_samples_leaf': 2,
 'clf__max_features': 'auto',
 'clf__max_depth': 10,
 'clf__bootstrap': False}
```

Random Forest (Lemmatization and Hyper Parameter Tuning):

Random Forest (Lemmatization, Tuned)				
	Precision	Recall	F1-Score	Support
FALSE	0.71	0.02	0.04	2843
TRUE	0.97	1	0.99	93117

Log Loss	Accuracy	AUC
0.113	0.9707	0.852



Lemmatization performs better on all metrics. Since we are using a bag of words approach, lemmatizing will help in reducing the number of tokenized features but also break down the words into their respective lemmas. This would make a lot of sense to model improvement. The same parameter grid was used for hyper parameter tuning. There was a huge improvement in precision in identifying False labels. So far, based on the precision when a complaint is classified as False for handled in a timely manner, the model predicts a False label is False 71% of the time. However, the recall is still low at 0.02 meaning the model correctly identifies only 0.02% of all complaints not resolved in a timely manner.

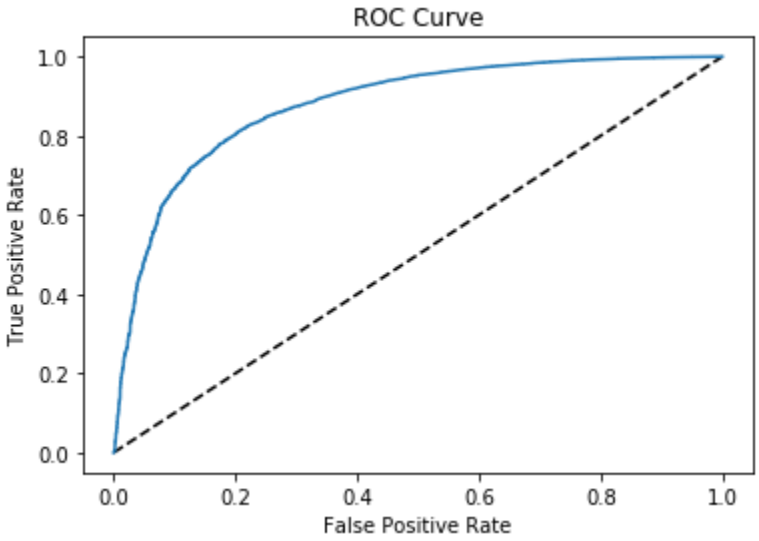
The best parameters were:

```
{'clf__n_estimators': 80,
 'clf__min_samples_split': 10,
 'clf__min_samples_leaf': 1,
 'clf__max_features': 'sqrt',
 'clf__max_depth': None,
 'clf__bootstrap': True}
```

Logistic Regression (Un-tuned with Lemmatization):

Logistic Regression (Un-tuned)				
	Precision	Recall	F1-Score	Support
FALSE	0.56	0.11	0.18	2843
TRUE	0.97	1	0.99	93117

Log Loss	Accuracy	AUC
0.104	0.9711	0.8784



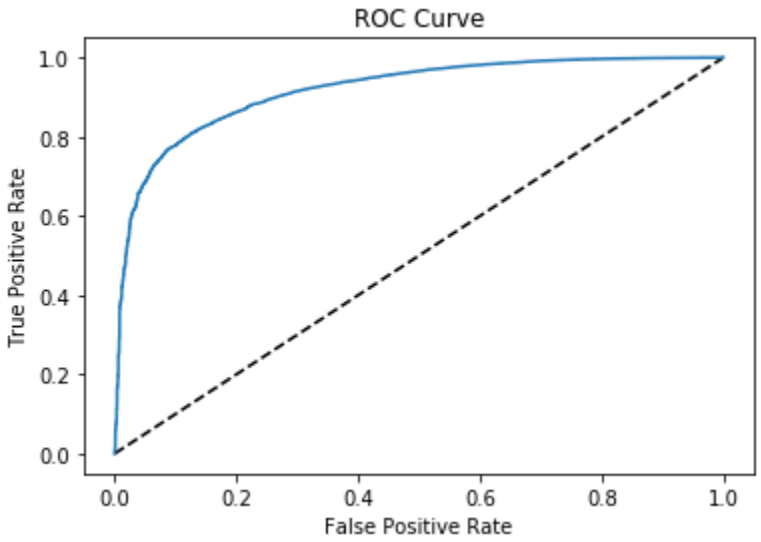
With the out of the box logistic regression model, we get better results overall except in precision of the False labels. But because the recall went up to 11%, I would call that an improvement based on our goal of identifying False labels. Now we have a model that can predict 11% of all complaints with a False label.

Justin Tsao

Logistic Regression (Tuned with Lemmatization):

Logistic Regression (Un-tuned)				
	Precision	Recall	F1-Score	Support
FALSE	0.61	0.21	0.31	2843
TRUE	0.98	1	0.99	93117

Log Loss	Accuracy	AUC
0.0875	0.9731	0.9175



For logistic regression we used a grid search with cross validation using:

```
param_grid_lr = {'clf__penalty': ['l1','l2'],
                 'clf__C': np.logspace(-4, 4, 12)}
```

The best hyper parameters:

```
{'clf__C': 0.43287612810830617, 'clf__penalty': 'l1'}
```

After tuning hyper parameters we get even better results. The loss of precision for the False labels is made up for the increase in recall. We have the lowest log loss highest AUC so far.

Logistic Regression (Tuned with Lemmatization, Sentiment Score):

So far, we have tweaked the models performance using by finding the right way to augment the 'consumer_complaint_narrative' and tuning hyper parameters. Next, I wanted to see if I could improve performance even more by engineering features. The sentiment score was engineered off of the 'consumer_complaint_narrative' and added as a feature of the model. In the exploratory data analysis, the sentiment scores were lower on average. My hope, was that this could add to model performance. However, adding the sentiment score did not improve results in the random forest model, and the logistic regression performed slightly worse. I will not show the full details here since there was no improvement. You can check out: https://github.com/jltsao88/Improving_Customer_Service_Machine_Learning/blob/master/05_Machine_Learning.ipynb

There you can see the full details on the training and results of the grid searches. There is also some analysis on feature importance that support some of the findings in the initial EDA.

VI. CONCLUSION

Using a logistic regression model, the project was able to identify almost all of the complaints that were responded to in an appropriate manner and 21% of complaints not responded to appropriately. The goal was to predict the complaints that were not

responded to appropriately so financial firms can identify these complaints as quickly as possible. The firms, in response, could reach out to customers a prevent churn due to bad service or product. This can also lead to new practices in customer service to be implemented such as a special response team to harder complaints.

The EDA of the data also led to some important findings. A lot of complaints were in large dollar value products like mortgages and loans. Also, a majority of the complaints were due to mis handling of information. Improving upon these mistakes could save time and money for both consumers and financial firms. For instance, if credit reporting is handled incorrectly this could cause a consumer to not be approved for a mortgage. Not only will the consumer not get a chance at their dream home but this will prevent firms from lending to that customer and lose out on a sale. Furthermore, a lot of time can be saved for both the consumer and firms to fix a reporting mistake that should not have happened in the first place.

So, what about the 21% accuracy in predicting 'un-timely' Reponses? It might seem low. Remember the 1.4 million complaints being analyzed were only coming from the CFPB and does not include all other complaints filed directly to the firms. Also, about 2.5% were the untimely responses.

Let's do some math: 1,400,000 complaints X 0.025 X 0.21 = **7350 complaints**

The model should predict about 7350 of those complaints.

On another note, 5528 complaints that were not responded to appropriately were mortgages. The model could possibly predict 21% of those or 1160 complaints. For a mortgage company or big bank that is a lot of customers that could possibly churn.

Looking at https://www.magnifymoney.com/blog/mortgage/u-s-mortgage-market-statistics-2018/, on some mortgage statistics. The average mortgage balance is about $148,060.

$148,060 X 1160 = **$171,749,600**

For a big firm who loses 1160 from churn, that's profit from potential interest on $171,749,600. Think about all the other products like huge trust accounts, personal loans, commercial loans, brokerage accounts, etc.

So, yes, 21% may seem low, but considering the amount of consumers and money involved, that is nothing to bat an eye at. Although, I wish the model could perform better, I am not entirely upset with the results. I believe the bulk of the predictive power of the model comes from the 'consumer_complaint_narrative'. Repeating what I said above, natural language processing is not an exact science. Looking forward, finding the better ways to preprocess the 'consumer_complaint_narrative' will be key in improving future results. Other features from the complaint narrative could also be engineered.