Capstone Project 2: Milestone Report 2

Problem Statement

When people require non-emergency municipal services, the number to call is 311. In New York, all the data about 311 calls received in New York city is available online. A potential problem that can be addressed is the number of late resolutions. For each call that gets placed, an estimated due date is made as to when the issue will be resolved. In this project, a model will be built that will predict if a new call that is placed will be resolved on time or not. This information would be useful to various New York public agencies in order to improve their performance and address the needs of the people in a more timely manner.

In this case, the client would be NYC 311. What they could do with the information learned from the project is, they could provide better service to individuals living in areas where wait times are longer than other areas.

Data Wrangling

The dataset that was used is the “NYC Open Data 311 Service Requests from 2010 to Present”. The data was acquired by using the Socrata Open Data API (SODA) which provides programmatic access to the dataset. The original dataset contains 22.1 million rows and 41 columns. To make the project more manageable in python as well as relevant to today, a 10,000 row subset of the data from January 1, 2019 to the present was taken. Because subsets of the data were taken as well as the feed being

updated daily, different versions of the dataset were saved based on the date of access. In the future, the models built on the different days of access could have their performance evaluated and compared.

Once the dataset was read into python, cleaning and transformation of the data was required to prepare the dataset for analysis. With 45 columns in the original dataset, we need to cut down the number of features the model will be built upon. First step, many columns containing redundant information were dropped. For instance, longitude and latitude describes the location of the incident, but so do the following columns: intersection 1, intersection 2, cross street 1, cross street 2, location zip, location address, location, bridge highway name, road ramp, bridge highway direction, bridge highway segment, x coordinate, y coordinate, landmark, street name, incident zip, and incident address. Next, rows containing null values were dropped shrinking out dataset from 50,000 to 16,586.

A new feature called ‘late’ was also created by comparing the”closed_date” of a row with its estimated “due_date”. For those rows where the closed date was greater than the due date, a label of “True” was applied. For those rows that failed the argument above, a “False” label was applied. These labels were then multiplied by “1” in order to get boolean values of “0” or “1”, so that we could use the late feature as the target variable. It was found that approximately 16.36% of all the rows in the dataset were labeled as ‘late’.

Continuing to clean the data, I decided that the day of the week the call was placed on could contain interesting information, so the variable “created_date” was converted to datetime and the day of the week extracted.

In previous version of the project, after converting categorical variables to dummy variables through one hot encoding, a total of 850 columns were produced. As this was too many features to make any conclusions off of, a mapping dictionary was created that transformed 67 different “complaint_types” into 8 groups. Those groups being: water, plants, animals, cleaning, human complaint, maintenance, noise, and traffic. The same method was applied to the “location_types” feature where 53 unique location types was converted into 3 groups of business, residential, and public.

Finally, since the variable late is a combination of the features “closed_date” and “due_date”, both those features were dropped from the dataframe. Thus the final variables selected to go into the predictive model were:

Agency: which New York City agency was assigned to take care of the incident

Borough: which New York City borough the incident took place in

Latitude: latitude of the address where the incident took place

Longitude: longitude of the address where the incident took place

Open_data_channel_type: how the incident was reported

Late: whether or not the incident was resolved on time or not

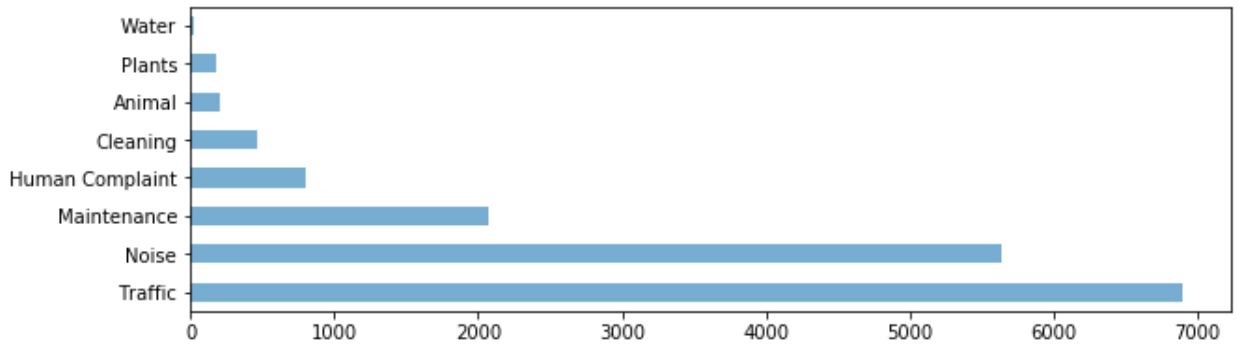
Created_day_of_week: what day of the week the incident was reported

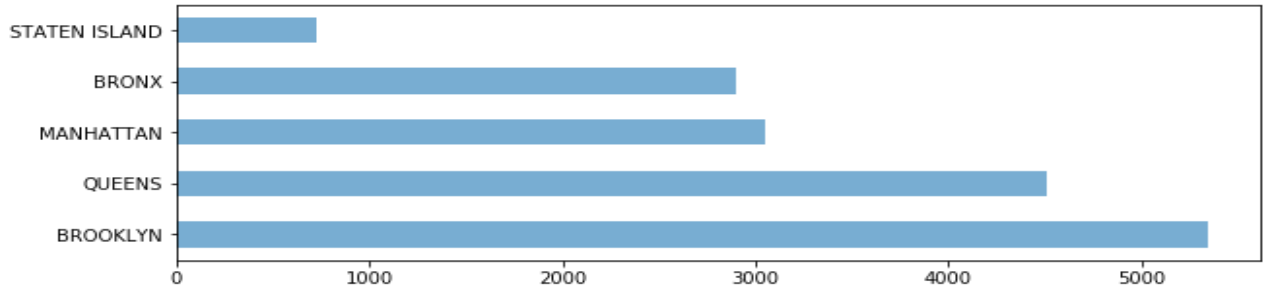
Complaint Grouping: the 8 different types of complaints

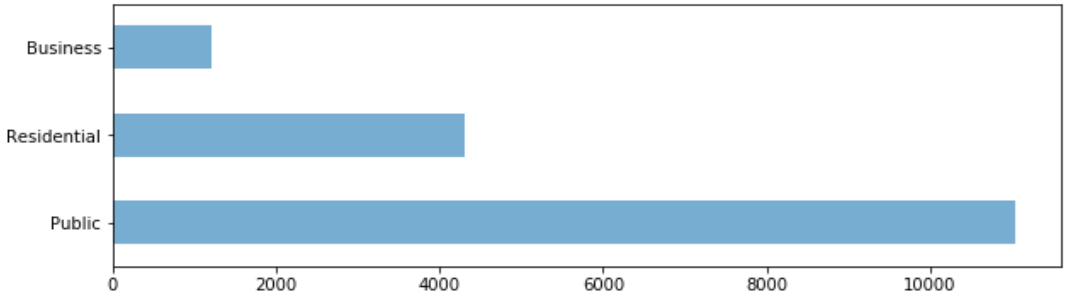
Location_Grouping: the 3 different types of location groups

Exploratory Analysis

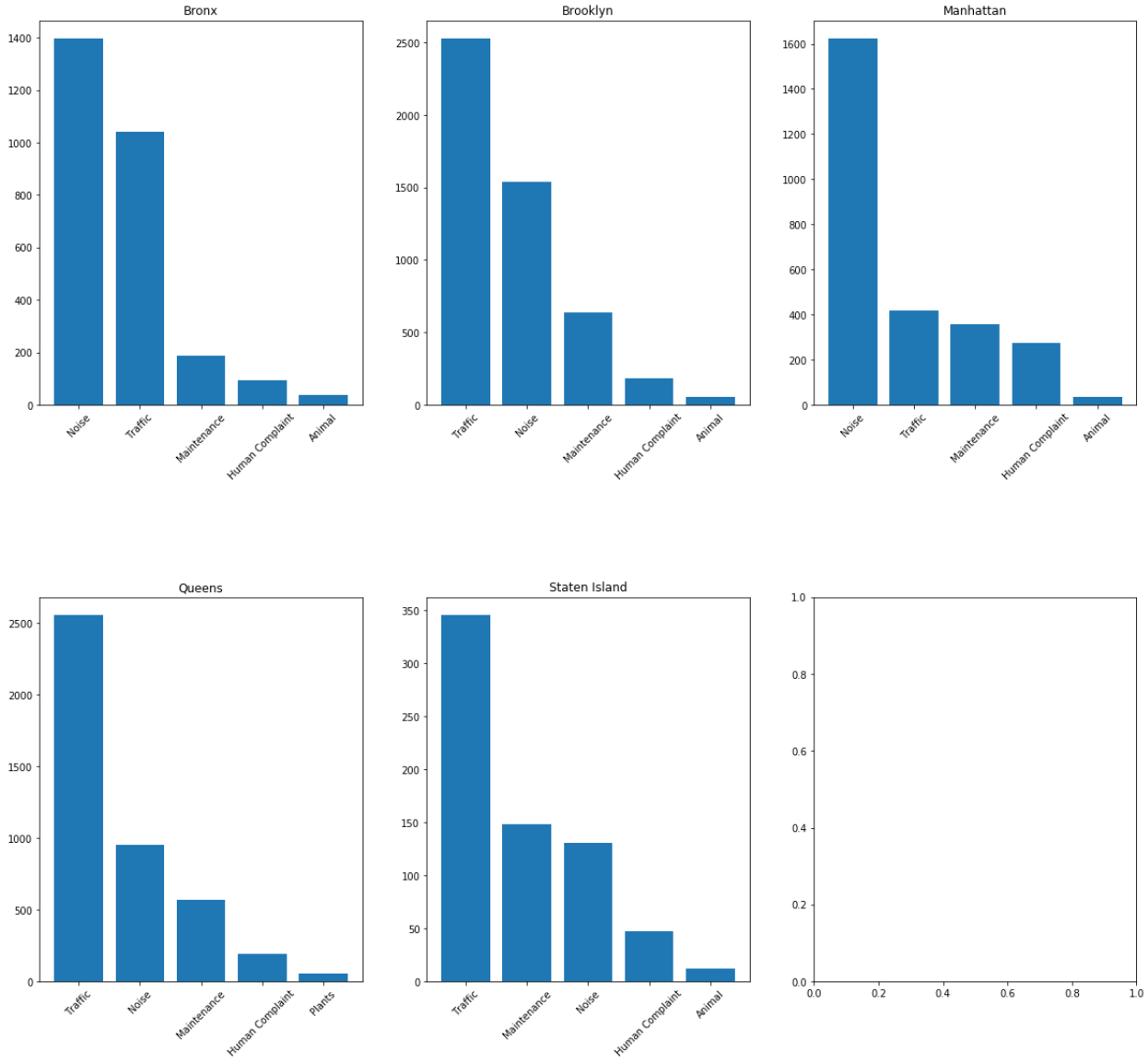
With data preprocessing finished, exploratory analysis could be conducted to see how our data looked. Some bar charts were created to look at counts of the data by feature variables in order to look at the spread of the data. Some examples are shown below:



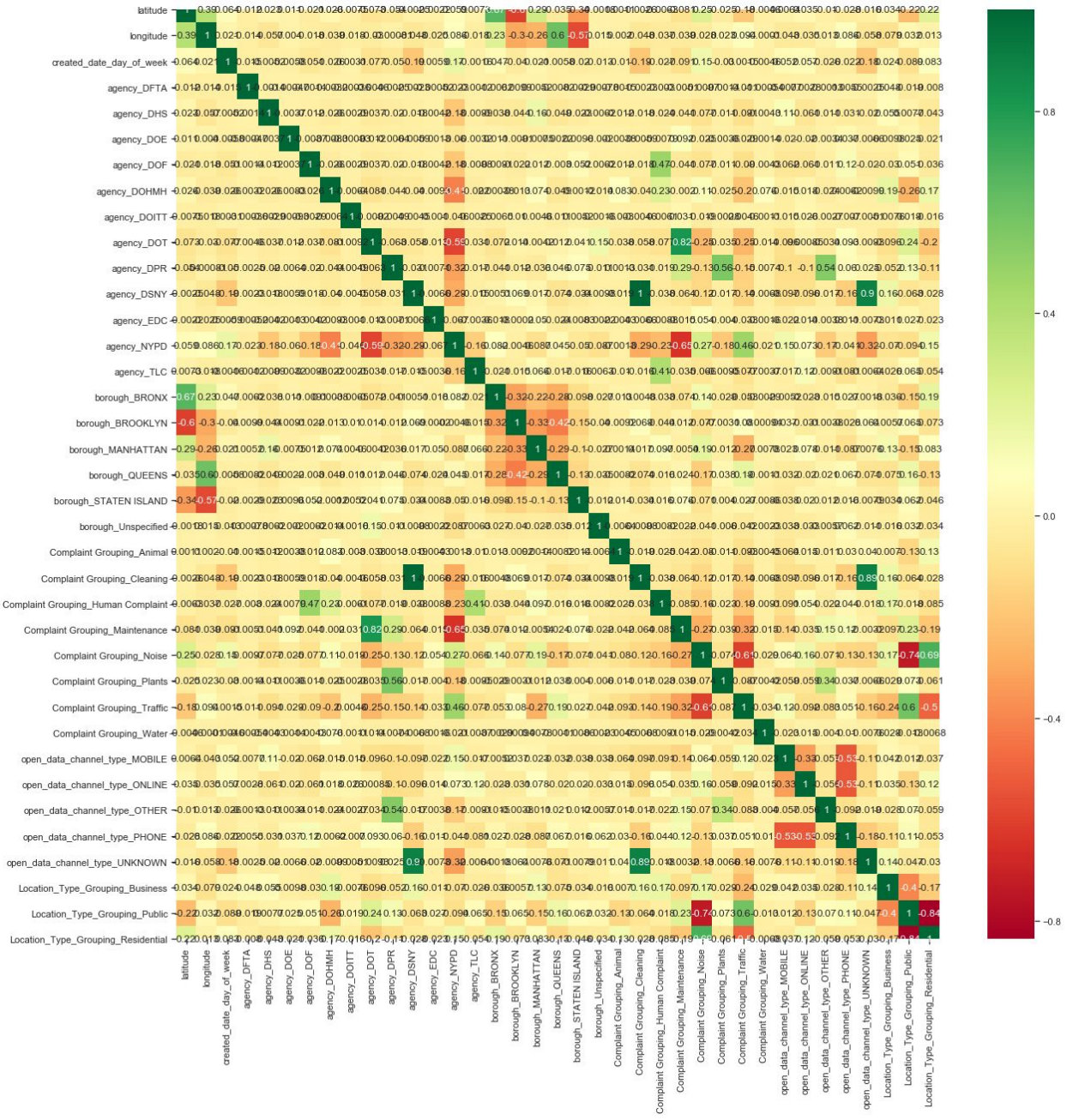




Bar charts were also created for the complaint types per borough.



Dummy variables were then created to allow the use of machine learning algorithms from sci-kit learn. The dummy variables created were then placed in a correlation matrix to show correlation coefficients between variables.



Although a few of the features seem to be correlated, overall there is not much correlation between the features. Now the dataset is ready to be used for predictive modeling.

Predictive Modeling

The target variable selected was the ‘late’ column which indicates that an incident report was closed after the estimated due date. The features that will contribute to the model are the other columns of the dataframe. The first machine learning algorithm applied to the dataset was the random forest regressor. As this is a classification problem, this algorithm is applicable. After splitting the dataset into a training and test set, the random forest regressor was fit to the data to create a model. When we looked at the score of the model created, only about 27% of the variability in the data was explained by the model. This score is on the lower end, and implies the model is not very strong.

A logistic regression model was then applied to the data, but this time we wanted to select for the more important features. Recursive feature elimination for logistic regression was conducted to select the features to build the model. Out of the original 37 features, recursive feature elimination recommended dropping 17 features to leave 20 columns to build the model with.

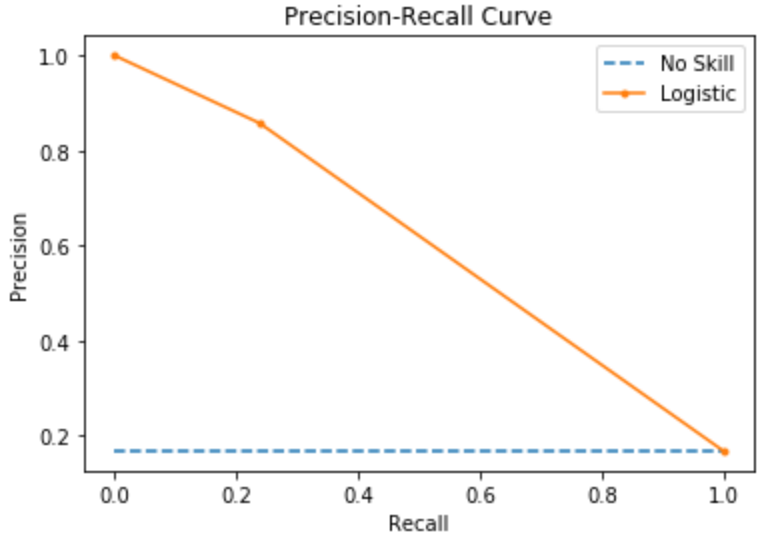
Using these twenty features, a logistic regression model was built on the training set. The performance of the model was then evaluated on the test set. Accuracy of the model was reported to be 87%. The confusion matrix of predicted vs actual values is as follows:

```
[[6850   55]
[1058  331]]
```

Next, a receiver operating characteristic curve was graphed:

The area under the curve was reported as 0.62, and the f1 score was 0.37.

Finally, a precision recall curve was graphed in comparison to a no-skill classifier showing the improvement predictions the model made.



The Odds Ratios for the features is as follows:

'agency_DFTA' =0.48460976

'agency_DOF' = 0.41817013

'agency_DOHMH' = 0.28076319

'agency_DOITT' = 0.51824128

'agency_DOT' = 0.09518558

'agency_DPR', = 0.26289544

'agency_DSNY', = 14.62876731

'agency_EDC', = 2.45378257

'agency_NYPD' = 0.22250191

'agency_TLC', = 3.10634805

'borough_BRONX' = 2.69486611

'borough_STATEN ISLAND'= 2.03279121

'borough_Unspecified' = 0.82362529

'Complaint Grouping_Animal' = 0.38543842

'Complaint Grouping_Cleaning' = 7.62616487

'Complaint Grouping_Human Complaint' = 2.41194631

'Complaint Grouping_Noise' = 0.43356204

'Complaint Grouping_Plants' = 3.07854911

'open_data_channel_type_OTHER' = 0.40464371

'Location_Type_Grouping_Business' = 1.9252957

Conclusion and Recommendations

Based on this study, the following are concluded:

- Approximately 16% of all incidents called into New York City 311 are resolved after the estimated time of resolution historically
- The model built on the training set had 87% accuracy in predicting the correct labels on the true set, as can be seen from the confusion matrix.

```
[[6850   55]
[1058  331]]
```

- Incidents reported in Manhattan, Queens, and Brooklyn had no effect on the outcome of the prediction, but incidents reported in Staten Island and the Bronx did with 0.71 and 0.99 coefficients respectively. This implies that living in Staten Island and the Bronx had a positive effect on the probability a call would come in late. Better service in those areas is recommended.
- Using the exponential function on the coefficients, we can compare the increase in probability of lateness for the two affected boroughs as well. EXP(0.76)= 2.03 which implies that a call originating in Staten Island has a 103% increased likelihood of being labeled as late according to my model, if all other features are held constant. The same method can be applied to the Bronx. EXP(0.99)= 2.69 implying that there is a 169% increase in likelihood of being labeled as late according to the model if other features are held constant.
- Calls handled by the NYPD had a 78% decrease in the likelihood of being labeled as late. (EXP(-1.5)=0.22) Good work NYPD!
- The largest odds ratio seen belonged to calls handled by the New York Department of Sanitation with a 1462% increase in probability of a call being labeled as late according to the model. This seems like an outlier,

but it is possible that the agency is understaffed and requires more help to fulfill its responsibilities.

- The feature with the lowest odds ratio belong to calls handled by the New York Department of Transportation. These cases were 90% less likely to result in a label of late according to the model. With 5,060 employees, this agency seems to be addressing the concerns of the public well.
- Although discarding 17 of 37 features during recursive feature elimination while building the model seems excessive; when the model is run on all 37 features together, no change in the confusion matrix of predicted versus actual values was seen.