Analysis of NYC 311 Service Requests

May 13th, 2020

Project By:

Group5:

Diksha Garg(dg3392)
Kshitija Patel(kap676)
Palak Shah(pvs276)
Shruti Yeravadekar(sy2662)

Executive Summary

Data was extracted from Kaggle, and transformation of this data was performed by eliminating null values from the columns. Further, similar values were clubbed categorically to avoid complications in the forthcoming analysis. We then loaded the data for analysis and plotting in Tableau through a spreadsheet. Preprocessed data was split into testing data and training data, used to train the machine learning prediction model through a pipeline. All applications were integrated into a single web application for ease of access.

Problem Statement

When a complaint is registered by 311 Services, users typically do not get an estimate of problem resolution time. The intention of the project is to derive meaningful insights about the type of complaints in various areas of New York City using the 311 NYC Service Request data and to predict complaint resolution time. The analysis and prediction are then integrated into a Web Application where users can be reassured that their complaints will be resolved in the stated time. The extensive analysis can serve various use cases such as helping users select an area to buy a house in, according to the complaints they wish to avoid.

Architecture

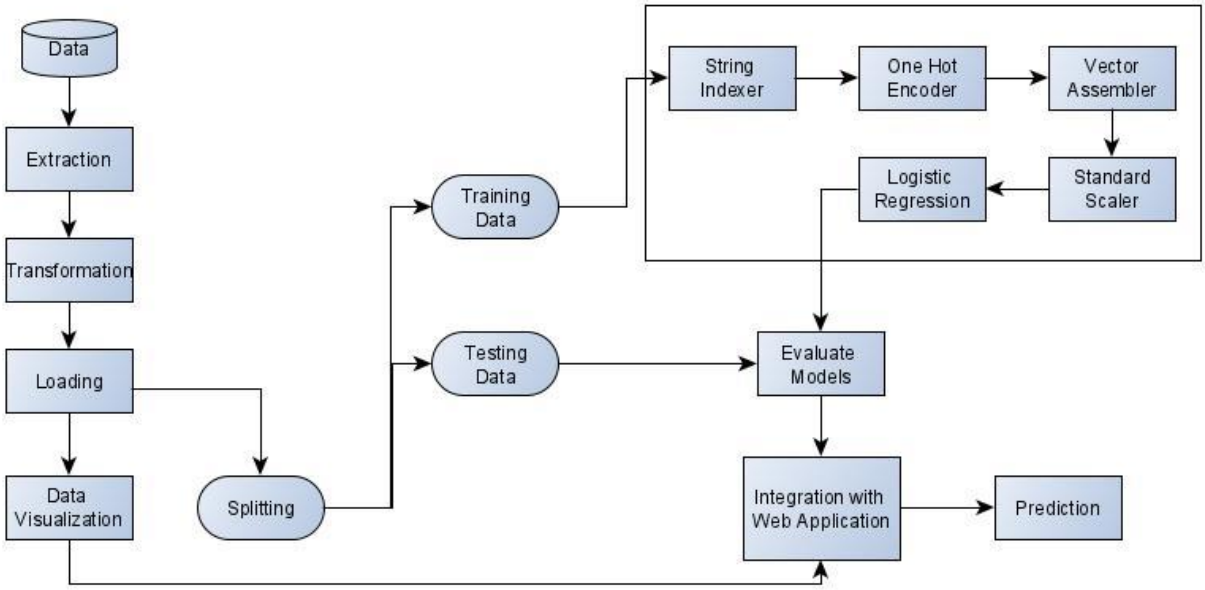

Figure 1

Figure 1 shows the architecture of the project. The data is passed through the ETL pipeline to clean and transform it to the desired format needed for prediction and visualization. It is split into training data which is fed to the Machine Learning pipeline from where predictions are made, and testing data. The evaluation is performed with the output from the pipeline and testing data. The evaluated model is integrated into the Web application.

Preprocessing

NYC open source data for 311 service requests was extracted from Kaggle and then transformed using Apache Spark, Spark SQL and PySpark.

While extracting the data, certain null columns were identified. These columns were removed while transforming the data in the ETL pipeline. Null values were also eliminated from the remaining columns. Values were merged categorically under Complaint Type, Location Type and Resolution Type within a main category. This organized the data effectively and provided better insight into its analysis.

Complaint resolution time was computed in seconds, minutes and hours into new columns that were integrated into the data. This would help in the prediction of resolution times with respect to complaint types and thereby, transformed data was loaded into a spreadsheet for analysis.

Analysis

Complaint patterns with respect to area of registration of the complaints was analyzed along with their Average Resolution Time and plotted using Tableau as shown below. Percentage of the types of complaints are plotted in Figure 2.

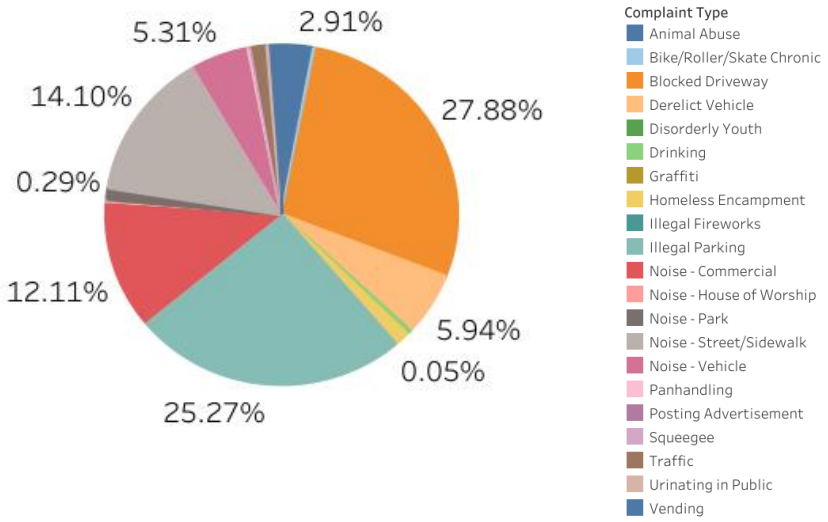

Figure 2

From Figure 2, it can be observed that Blocked Driveways received the highest number of complaints whereas issues like Park Noise received significantly lower number of registered complaints.

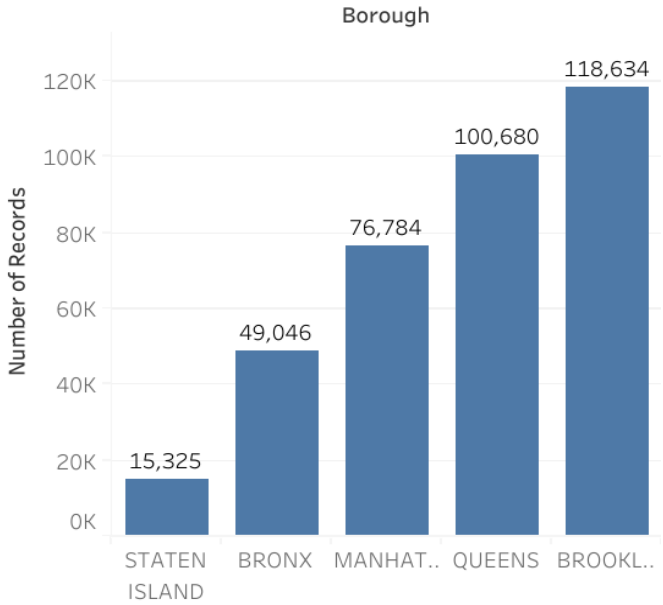

Figure 3

Figure 3 shows the number of records plotted against boroughs. The analysis shows that the number of total complaints from Brooklyn is the highest followed by Manhattan, with Staten Island having the least number of complaints. Similarly, this distribution was plotted against different complaint types as shown in Figure 4.

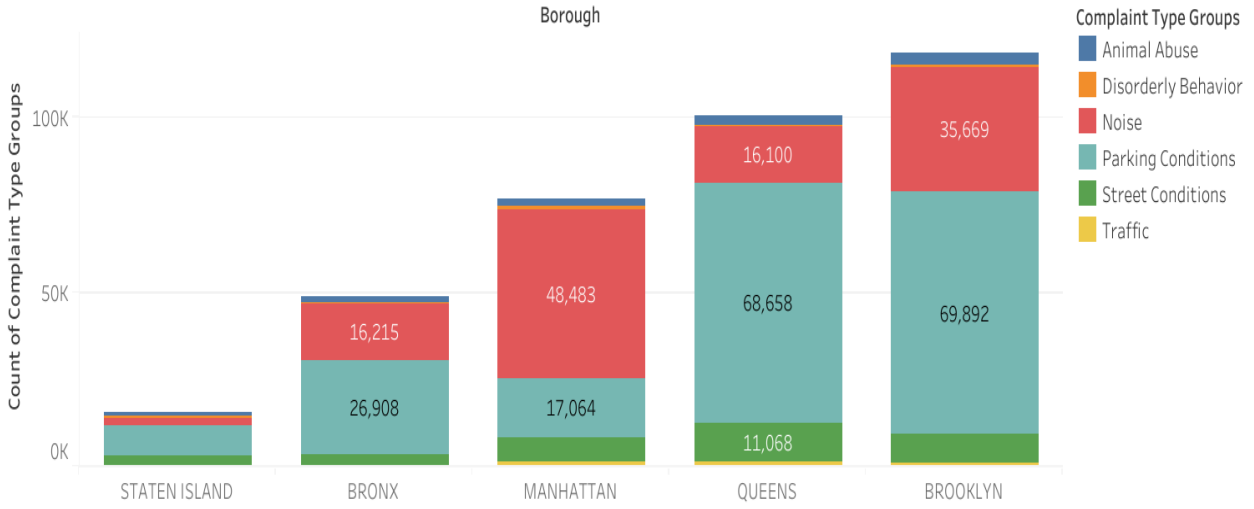

Figure 4

From Figure 4, it can be seen that Brooklyn has highest number of complaints with 'Parking Conditions' being the highest. Figure 5 shows the comparison of different complaint types with respect to incident zip.

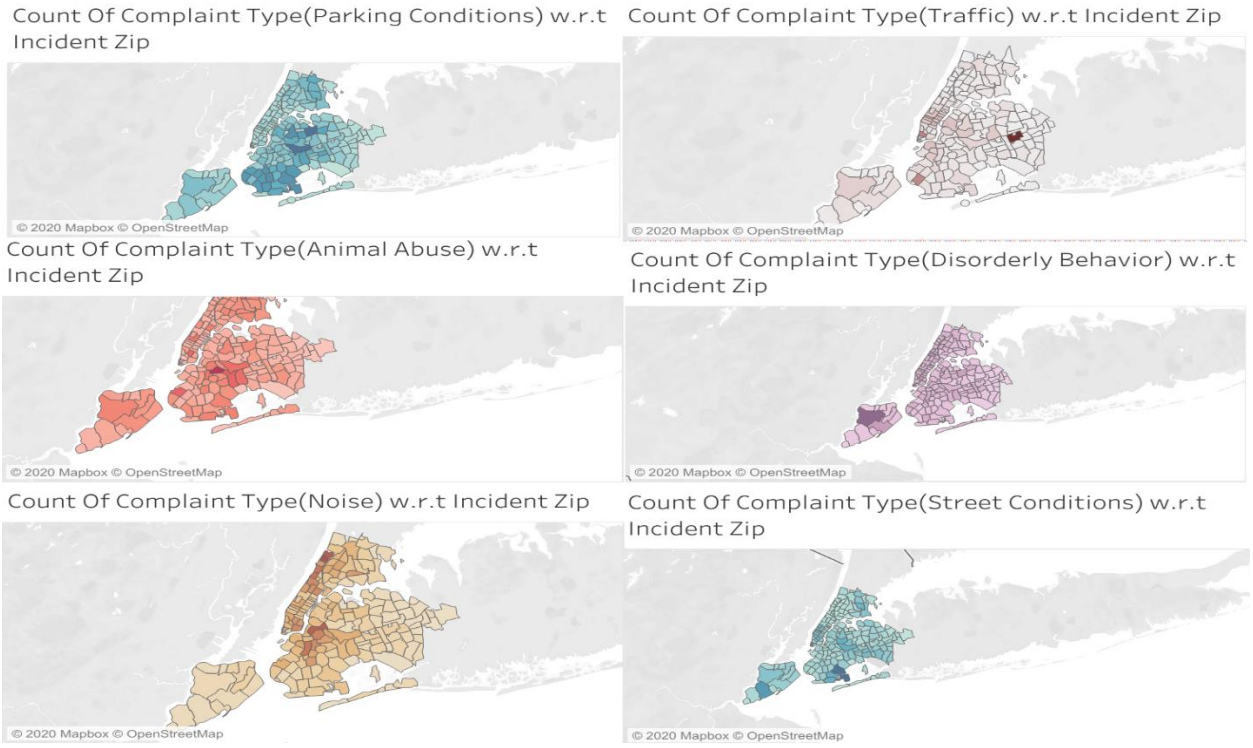

Figure 5

In Figure 5, the darkest shade of the highlighted colors on the areas of the map show the maximum number of complaints in that zip code in New York City. Figure 6 depicts the analysis of the average resolution time (in hours) with respect to Complaint Type.

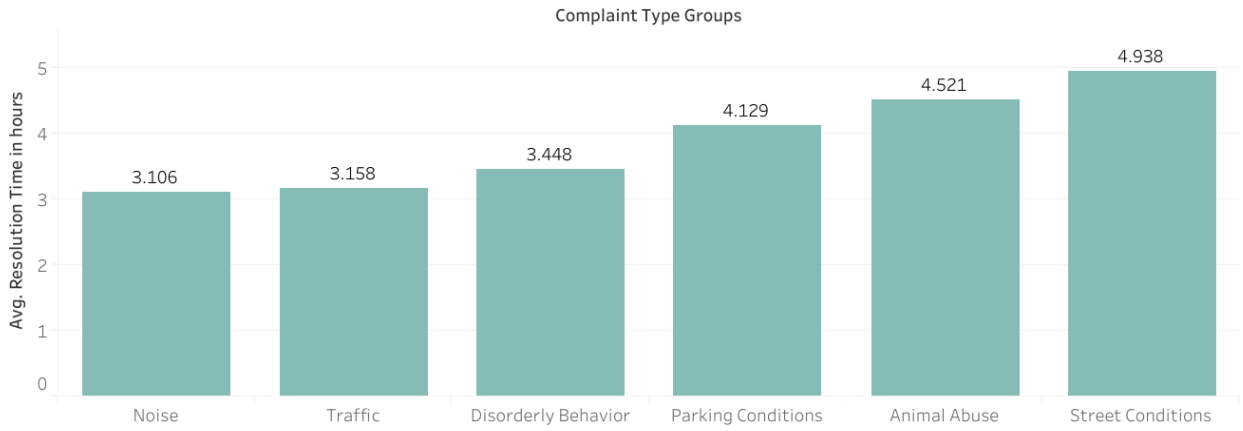

Figure 6

From the graph, it can be seen that street conditions take the highest amount of time to be resolved with noise problems taking the least.

Prediction

Resolution time was predicted using independent variables from the data. The variables used were Complaint Type, Zip code and Day of the week. This prediction helps the user in acquiring an estimate of resolution time of their complaint.

Multinomial Logistic Regression was used as a classifier to predict the resolution times within three categories:

1. Within 2 hours
2. Within 6 hours
3. More than 6 hours

If a complaint can be resolved within 2 hours in a zip code, it can be expressed that the resolution time to handle that complaint is excellent. If it is within 6 hours, it is satisfactory and if it is more than 6 hours, it can be expressed that the complaint needs more time to get resolved.

The training data is fed to the Machine Learning pipeline from where we get the value for predictions. The evaluation is done with the output from the pipeline and the output for the testing data. Thereafter, we get the predictions and integrate them into a Web Application as shown below.

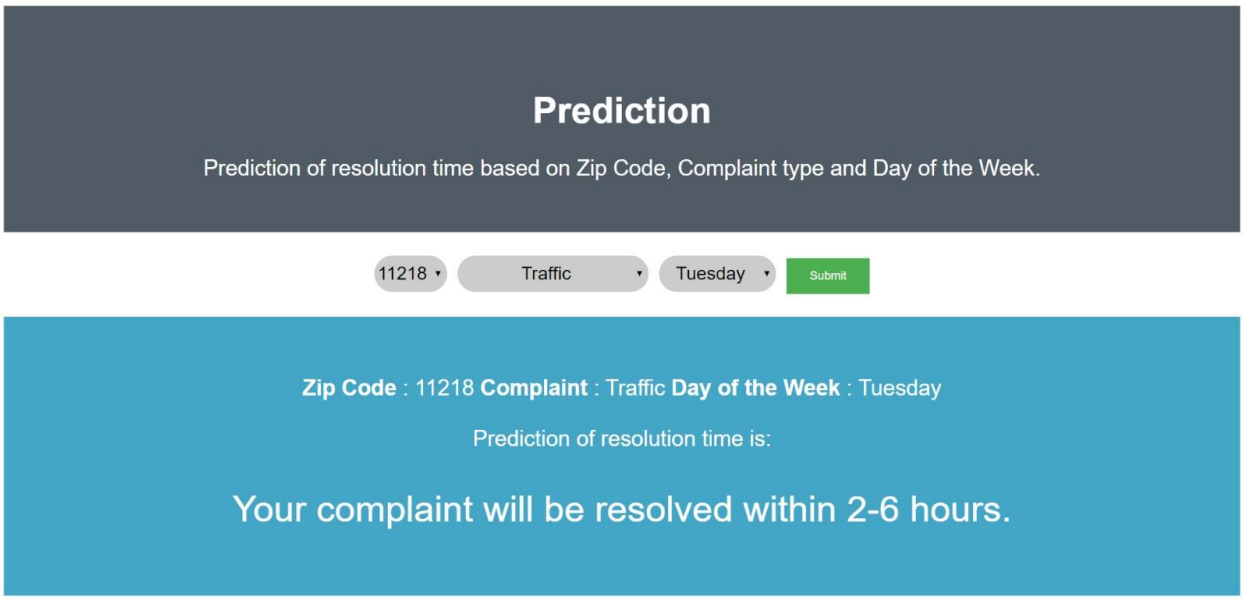

Figure 7

Figure 7 shows the web application page for the prediction. Selecting the zip code, Complaint Type and the day of the week followed by enabling the submit button, fetches the user the result of their prediction query. For example, in the above image, the user enters '11218' as the zip code, 'Traffic' as the complaint type and 'Tuesday' as the day of the week for which a prediction that the complaint will be resolved within '2-6 hours' is generated.

Web Application

The web application was developed using HTML, CSS and JavaScript. To display the visualizations, Tableau was integrated using Tableau online. Finally, the machine learning model was incorporated as the back end using Flask. The Web Application has four sections:

1. Header - Displays the title of the project and consists of a navigation bar that links to other sections in the web page as shown in Figure 8.
2. About - Tells the user about the problem statement and links to the analysis section.
3. Analysis - A click of a button that enables the user to see interesting insights about the problem. This was made possible by integrating Tableau with the front end as shown in Figure 9.
4. Prediction - The button is linked to localhost:5000 where Flask runs on another page developed specifically for the prediction. This page is shown in Figure 7.

Figure 8

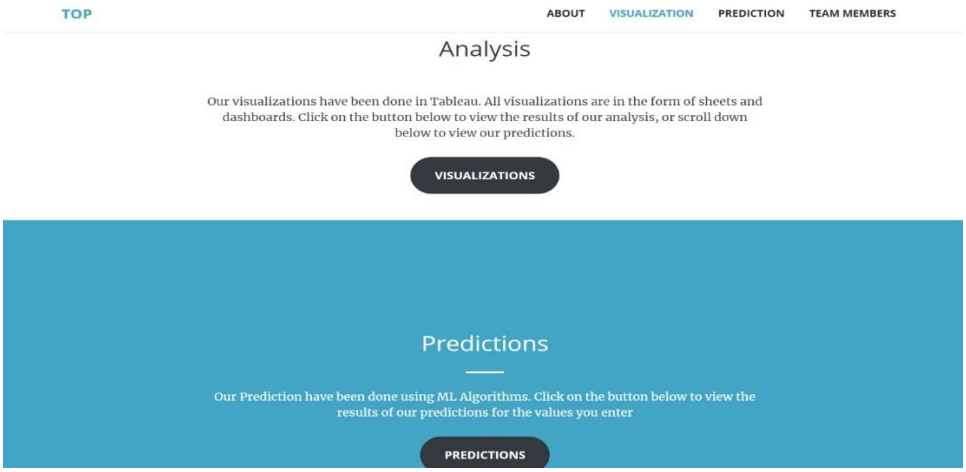

Figure 9

Conclusion

NYC 311 receives thousands of requests related to several hundred types of non-emergency services, including noise complaints, parking conditions and many more. These requests are received and forwarded to the relevant agencies, such as the Police, Department of Buildings or Transportation. The agencies then respond to the request, addresses it, before closing the request. Prediction of the resolution time (in hours) was successfully performed to address those complaints. Interactive dashboards were also created to provide various insights on 311 complaints from the last 10 years.