3-1-1 Service Requests Data Cleaning and Profiling

Amani Deepthi Matta
New York University
am10620@nyu.edu

Dennis Fenchenko
New York University
df1911@nyu.edu

Mahima Mehta
New York University
mm11527@nyu.edu

ACM Reference Format:
Amani Deepthi Matta, Dennis Fenchenko, and Mahima Mehta. 2021. 3-1-1 Service Requests Data Cleaning and Profiling. In *Proceedings of ACM Conference (Conference'17).* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 Introduction

311 is a hotline number that is used by people all over United States to make their non-emergency complaints. It facilitates people to directly communication with the government agencies and log their service requests. Since the open government movement in 2010, 311 dataset has gained popularity and is used by multiple Data scientists for analysis and research purposes.

Data cleaning plays an essential role in accurate and efficient data analytics. In this paper, we have used 311 service requests data of the New York City to understand different cleaning techniques and how they ease our processing during analysis. In this paper we are discussing various data cleaning approaches that we have used on the 311 data set to resolve data problems.

Our cleaning strategies include outlier and anomaly detection for different columns, handling invalid 311 requests by filtering data and dealing with missing values. We have also implemented data standarization for correcting the format and misspellings across different columns. In addition to this, to handle enormous types of 311 requests, we have generalised different types of complaints. Apart from this, we have also applied functional dependencies on various dataset columns for correcting dependencies. Finally, we have used the cleaned data to perform analysis on 311 requests by using multivariate visualization methodologies to answer major data questions.

2 Problem Formulation

Conducting analysis or training machine learning models on a poor quality data can significantly undermine the quality of obtained results or even make them useless. Even more dangerous is the fact that it is often difficult to tell whether derived algorithms are producing incorrect results, which can lead to major business and social costs. One of the key attributes of a high-quality data is its accuracy. We have the most control over the data accuracy at the data collection stage. Another important aspect of a good data is data completeness, which means that all data elements we require for the analysis must be present in our dataset. Data consistency also plays a crucial part in constructing a high-quality dataset. We must ensure that there are no conflicts between the data values in different columns of a dataset. We use data cleaning for making our dataset reliable for analysis and model training.

Formally, data cleaning is the process of removing incorrect, corrupted, incorrectly formatted, duplicate, or incomplete data within a dataset. Cleaning strategies the one needs to apply to the dataset depend on the data in question as well as the kind of analysis to be conducted on that data. Some of the most common cleaning strategies include removal of duplicates, fixing structural errors within the dataset, filtering of unwanted outliers, handling missing data, and manually inspecting the data to ensure the correctness of our cleaning strategies.

In our project we focused on "311 Service Requests from 2010 to Present" dataset taken from NYC Open Data. We have developed our techniques based on this dataset and then extended those techniques to 10 other data sets which have some of the overlapping columns with 311 data. After pragmatically implementing the cleaning strategies, we manually inspected the output and ensured effectiveness and correctness of our methods by measuring precision and recall.

First, we would like to discuss which of the general cleaning strategies were applicable to 311 dataset. In order to decide which strategy to apply to our dataset we started by manually inspecting the dataset using NYC Open Data search and visualization tools. This gave us a general idea of which columns might contain poor quality data that might undermine the quality of the overall dataset.

Some of the problems that we have discovered using this approach are: missing values in various columns of our dataset, incorrect values, such as city or borough names being misspelled, and conflicting data in between multiple columns of a dataset (e.g. zip code not matching possible zip codes for a given borough or Park Borough being different from City Borough).

Conference'17, July 2017, Washington, DC, USA
© 2021 Association for Computing Machinery.
ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

After manual identification of problems in our dataset we proceeded to using *OpenClean* framework for profiling the dataset and discovering problems within the dataset automatically on a large scale. Problems that we have discovered using this approach were similar to the ones we discovered manually, but on a larger scale. One of the main bottlenecks we encountered when trying to profile the whole dataset using *OpenClean* was scalability. As a result our algorithms were taking too long to complete. We have solved this problem by rewriting our algorithms in Spark and parallelizing them over multiple nodes on the computing cluster.

Below we will describe the related work on Data Cleaning that we have found.

3 Related Work

We have started developing our strategy for cleaning 311 dataset by exploring *OpenClean* framework for data profiling and cleaning and applying some of the functions provided by *OpenClean* on our dataset. We will discuss the results that we have obtained in greater depth later in the paper.

Our next step was to explore research papers on data cleaning and 311 dataset in particular. Our primary source of research papers was Google Scholar repository.

The first reference was a paper called Data Cleaning: Overview and Emerging Methods. The paper mentions two types of data cleaning approaches: quantitative and qualitative. Quantitative data cleaning techniques employ statistical methods for identification of abnormal behaviours and errors, such as outliers detection algorithms. The other approach, which paper actually focuses on, is qualitative data cleaning which is using constraints, rules, and patterns to detect errors. The paper discusses both qualitative identification and subsequent repair techniques that can be used on a data.

For the purposes of out project, one of the most useful contributions of the paper were the questions that the authors presented in it. These questions shaped our vision of what to look for. For example, for the qualitative error detection the authors focus on three main questions: Error Type (What to detect?), Automation (How to detect?), and Business Intelligence (Where to detect?). The first question helped us to think about which columns are likely to be problematic within our dataset and in what ways. The second question helped us to think about the tools we can use to automate error detection, such as *OpenClean* and *Spark* frameworks. And lastly the third question guided our thinking in terms of who could have made a mistake and the given dataset and how. For example, in our dataset there is a unique complaint ID which is likely automatically generated once the complaint is being placed. Based on this we can assume that there are likely no errors in this column of a dataset. On the other hand, complaint description and address were likely entered manually by a human and is thus are much more error prone.

Another question considered in this paper was regarding the choice of a Repair Model. The authors suggested that there are two primary possibilities: a repair model that would describe where the problems occur, but would not modify the original dataset or a repair model that would actually change the original data set. In our project we applied both of these models. Before cleaning the data, we ran functions that would identify and display problems in the data set so that they could be manually inspected by us. Then, after the manual inspection, we would clean the original data set and write it to a file so that it could be used for further research.

Another research paper that we found immensely useful in guiding our thinking was the paper named "Quantitative Data Cleaning for Large Databases" written by Joseph M. Hellerstein.

Hellerstein's paper starts by discussing possible sources of error in data. The main distinction stressed in the paper is between the data that was manually entered by humans versus automatically generated data. Namely, the author suggests that many of the sources of error fall into one or more of the following categories: data entry errors, measurement errors, distillation errors, and data integration errors. While we cannot know for sure, while inspecting our data set, we concluded that most of the errors in the data are likely data entry errors (e.g. people who were manually entering the data were making mistakes)

Another section that we found immensely important for our project was the section on approaches to improving data quality. The paper suggests the following approaches for improving the data quality: data entry interface design, organizational management, automated data auditing and cleaning, and exploratory data analysis and cleaning. Since we don't have any control over the data acquisition stage, only last two approaches were applicable in our case.

A third paper which greatly influenced our approaches and techniques is the paper titled "311 service requests as indicators of neighborhood distress and opioid use disorder".

Our motivation for exploring this paper was that the choice of cleaning strategies depends on the data analysis techniques that one wants to apply on a dataset.

The main idea of this paper is to establish connection between the levels of substance abuse and 311 service calls across the US.

For the purposes of our project, we are mainly interested in the analysis tools and techniques that the authors applied on 311 data set. Knowing the techniques that need to be applied to this data set, we can better customize our cleaning strategies to suit these techniques.

By examining the data snippets included in the paper, we concluded that one of the most important data cleaning techniques we need to apply on our dataset is data standardization. To obtain meaningful results we need to combine various spellings and ways of conveying the same information into one.

Also, by observing analysis techniques used in the paper, we concluded that we don't need all the columns in the data set and many of the data entries in the original data set contain duplicate data

4 Methods, Architecture and Design

Big data refers to large "volume" and "variety" of data, which essentially means that the data will be in large volume coming from multiple sources with lots of different formats. Since the data is in large volume and gathered from multiple sources, it have a lot of "veracity". The data received could be incomplete, biased and most importantly noisy.In order to process such a huge data, data cleaning plays an essential role in "velocity" of data processing.

In this section we will discuss various methods involved in cleaning the data. We will also throw light on the Architecture and design of different cleaning strategies that we have used to clean 311 service requests data.Here are the steps and methods for data cleaning and processing.

4.1 Data Profiling

The first step for developing a successful data cleaning methodology is to get the insights of the data. In order to dive deeper into the data, we did *Data Profiling*. Data profiling is the primary of data cleaning to know the data set. It helps us to understand the type of data, its properties and detect flaws from a larger perspective.

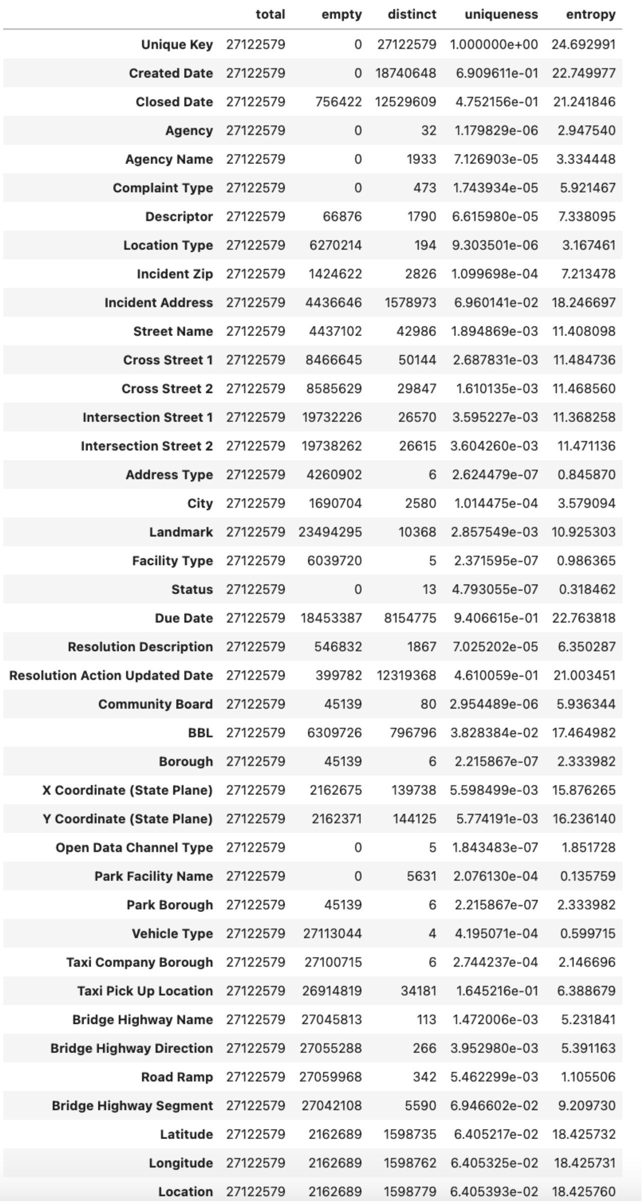
In our project, we have used openclean, which is a python library to perform data profiling and basic data cleaning. openclean comes with a profiler which helped us to profile the data and know the data types of different columns in the dataset.

As shown in Fig. 1, openclean for data profiling helped us by using existing libraries to get data stats including the uniqueness of columns. In addition to that, we also get the total and distinct counts for data in each column. In order to understand the range and boundaries of our data we used minimum and maximum frequencies of different columns(e.g. min-max year range). Data profiling also answers questions related to most frequent data types in the data-set which makes it easy to understand which columns in the data can take a backseat in data processing.

4.2 Functional Dependencies

One of the key aspect of data cleaning is correctness. Violations of functional constraints indicates set of incorrect value combinations in the data set. For this, we take into consideration the mapping of zip codes with the boroughs.

The strategy is based on the basic understanding of boroughs. Each borough can be bounded by a pair of min-max zip codes. We apply this functional constraint and perform zip code to borough mapping for the columns in our data set.



	total	empty	distinct	uniqueness	entropy
Unique Key	27122579	0	27122579	1.000000e+00	24.692991
Created Date	27122579	0	18740648	6.909611e-01	22.749977
Closed Date	27122579	756422	12529609	4.752156e-01	21.241846
Agency	27122579	0	32	1.179829e-06	2.947540
Agency Name	27122579	0	1933	7.126903e-05	3.334448
Complaint Type	27122579	0	473	1.743934e-05	5.921467
Descriptor	27122579	66876	1790	6.615980e-05	7.338095
Location Type	27122579	6270214	194	9.303501e-06	3.167461
Incident Zip	27122579	1424622	2826	1.099698e-04	7.213478
Incident Address	27122579	4436646	1578973	6.960141e-02	18.246697
Street Name	27122579	4437102	42986	1.894869e-03	11.408098
Cross Street 1	27122579	8466645	50144	2.687831e-03	11.484736
Cross Street 2	27122579	8585629	29847	1.610135e-03	11.468560
Intersection Street 1	27122579	19732226	26570	3.595227e-03	11.368258
Intersection Street 2	27122579	19738262	26615	3.604260e-03	11.471136
Address Type	27122579	4260902	6	2.624479e-07	0.845870
City	27122579	1690704	2580	1.014475e-04	3.579094
Landmark	27122579	23494295	10368	2.857549e-03	10.925303
Facility Type	27122579	6039720	5	2.371595e-07	0.986365
Status	27122579	0	13	4.793055e-07	0.318462
Due Date	27122579	18453387	8154775	9.406615e-01	22.763818
Resolution Description	27122579	546832	1867	7.025202e-05	6.350287
Resolution Action Updated Date	27122579	399782	12319368	4.610059e-01	21.003451
Community Board	27122579	45139	80	2.954489e-06	5.936344
BBL	27122579	6309726	796796	3.828384e-02	17.464982
Borough	27122579	45139	6	2.215867e-07	2.333982
X Coordinate (State Plane)	27122579	2162675	139738	5.598499e-03	15.876265
Y Coordinate (State Plane)	27122579	2162371	144125	5.774191e-03	16.236140
Open Data Channel Type	27122579	0	5	1.843483e-07	1.851728
Park Facility Name	27122579	0	5631	2.076130e-04	0.135759
Park Borough	27122579	45139	6	2.215867e-07	2.333982
Vehicle Type	27122579	27113044	4	4.195071e-04	0.599715
Taxi Company Borough	27122579	27100715	6	2.744237e-04	2.146696
Taxi Pick Up Location	27122579	26914819	34181	1.645216e-01	6.388679
Bridge Highway Name	27122579	27045813	113	1.472006e-03	5.231841
Bridge Highway Direction	27122579	27055288	266	3.952980e-03	5.391163
Road Ramp	27122579	27059968	342	5.462299e-03	1.105506
Bridge Highway Segment	27122579	27042108	5590	6.946602e-02	9.209730
Latitude	27122579	2162689	1598735	6.405217e-02	18.425732
Longitude	27122579	2162689	1598762	6.405325e-02	18.425731
Location	27122579	2162689	1598779	6.405393e-02	18.425760

Figure 1. Stats of the 311 service requests data set.

The implementation of this strategy corrects all the boroughs that are incorrectly mapped to the zip code.

Another strategy to correct the data is to look at the functional constraint aspect of geographic attributes in the data set. The idea here is get the correct borough, based on the latitude and longitude coordinates of the incidents. Essentially, the approaches counts the occurrences of a particular combination latitude and longitude against the borough names. The probability of correct borough is directly proportional to maximum count. Hence, we update all the incorrect boroughs with the most frequent borough value for that combination of longitude and latitude.

4.3 Missing data cleaning and standardization

As we discussed in the above section, after applying functional dependency we get a clean borough mapping. Once we have the cleaned data for all boroughs in place, we apply the cleaning on the city column.

The steps for handling missing city data include:

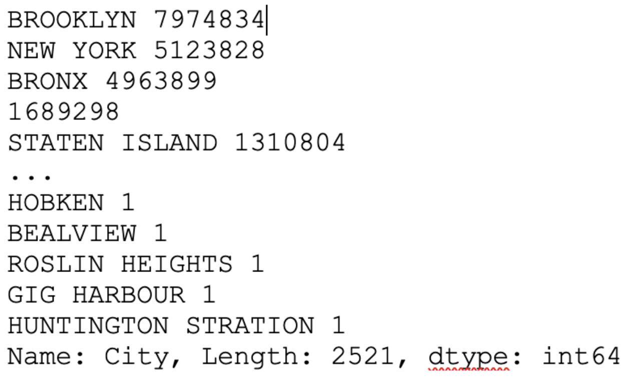
- `Check for missing data`: The first step is to check if city column has missing data or if it has values like "`null`", "`NA`", "`Unspecified`" or "`N/A`".
- `Check for borough`: The second step is to check if there is data in Borough.
- `Standardize`: The third step is to perform standardization. If there is data in borough we update the city column with Borough name (since the data set uses them inter-changeably) else if data is missing in borough as well then we can generalize the city and set it to "`New York`".

4.4 Outliers and Anamolies detection

Outliers and Anamolies detection plays a significant role in data cleaning. It is a process by which we can find noise in the data-set which is due to veracity of the data. By finding the outliers and anamolies in the data, we can improve the quality of data by removing unnecessary values.

In our analysis we are using 311 data set for New York City, so in order to find out outlier, we have used city column from our data set. As shown in Fig 3, by counting the values corresponding to each city, we were able to get all the requests that did not belong to New York City and hence they should not be considered in our analysis.

Therefore, by detecting all the outliers, we cleaned all the city column by removing all requests that were not a part of New York City.



```
BROOKLYN 7974834
NEW YORK 5123828
BRONX 4963899
1689298
STATEN ISLAND 1310804
...
HOBKEN 1
BEALVIEW 1
ROSLIN HEIGHTS 1
GIG HARBOUR 1
HUNTINGTON STRATION 1
Name: City, Length: 2521, dtype: int64
```

Figure 2. Detecting outliers and anomalies in the city column for 311 service requests data set.

4.5 Data cleaning using Union architecture

A major concern that comes with the Variety of big data is redundancy. Identifying redundant data and perform cleaning is a crucial part of data cleaning. To implement this cleaning technique, we came up with the union architecture.

The union architecture that we propose is as follows:

- `Identify similar columns`: Firstly we identified similar columns in the data set. After careful observation of the data, we understood that column combination of "`Cross Street 1`" and "`Cross Street 2`" is same as the combination of "`Intersection`" "`Street 1`" and "`Intersection Street 2`". Similar is true for "`Borough`" and "`Park borough`" columns as well.
- `Identify patterns`: We observed that if data is present in "`Cross Street 1`" and "`Cross Street 2`", then the data in "`Intersection Street 1`" and "`Intersection`" "`Street 2`" columns is missing, and the vice versa is true as well. This holds true for "`Borough`" and "`Park borough`" columns as well.
- `Perform Union and remove redundant columns`: After identifying the pattern, we can perform union on these similar column and hence reduce the redundancy in the columns of the data set. Therefore, after performing the union, we drop the redundant columns.

4.6 Data Mapping

In the 311 request data set, we have Agency column that represents the acronym of the agency. We also have Agency Name column which represents the full name of that agency. In order to have correct names of the agency mapped with their acronym, we implemented the mapping for all 32 distinct agencies acronyms with corresponding agency names.

4.7 Data Generalization

The 311 request data set has many different types of non-emergency service requests. We worked on implementing technique which generalizes similar types of complaints under one umbrella. This includes one to many mapping to complaint type with appropriate clustering similar complaints together.

4.8 Data Standardization using Key collision clustering techniques

The 311 requests data is of huge volume. It is prone to have multiple values indicating the same data interpretation. The idea is to standardize data by using key collision clustering techniques in order to get inculcate similarity in terms of values.

We have applied the data standardization over multiple columns in the data set like agencies and street names. A very interesting observation was made based on key collision clustering for Agency column. The agency column in the data set came out as the most clean data. There were zero clusters formed with no conflicts in the data.

Adding on, we applied the same strategy to the Street Name column. As a part of data standardization, our design helped us to understand that multiple clusters for same street name showed there was a lot of scope for data cleaning for that column.

4.9 Reference Data Creation

The information extracted from any data will be accurate only if there is enough data to perform analysis. As the last step of data cleaning, we drop all the the columns which have more than 75 percent null values. In addition to that, we also drop multiple columns which provide the same information. In the 311 data set there are multiple columns like X and Y coordinates, Longitude and Latitude which essentially gives us the location information of the incident. Hence we keep Location column and drop all the redundant columns to reduce the Volume of the data by extracting only meaningful data.

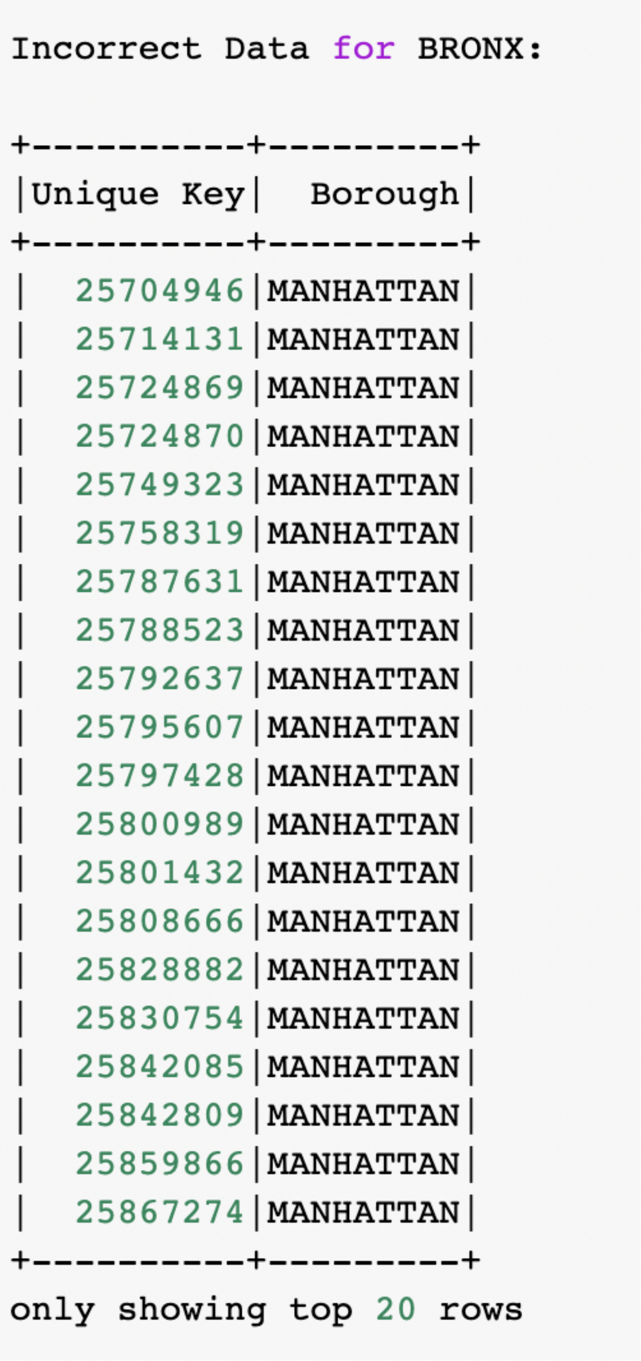
5 Results

The techniques we have mentioned above have produced remarkable results. The quality of these results have been calculated using the performance metrics, precision and recall. Precision (also called positive predictive value) is the fraction of relevant instances among the retrieved instances, while recall (also known as sensitivity) is the fraction of relevant instances that were retrieved. Both precision and recall are therefore based on relevance. The formulae for precision and recall is as follows:

$$Precision = TP/(TP + FP)$$

$$Recall = TP/(TP + FN)$$

For the first method discussed above, that is, Zip code to Borough mapping, from the below output we understand that there are significant amount of records that are incorrect in the data.



```
Incorrect Data for BRONX:

+----------+---------+
|Unique Key|  Borough|
+----------+---------+
|  25704946|MANHATTAN|
|  25714131|MANHATTAN|
|  25724869|MANHATTAN|
|  25724870|MANHATTAN|
|  25749323|MANHATTAN|
|  25758319|MANHATTAN|
|  25787631|MANHATTAN|
|  25788523|MANHATTAN|
|  25792637|MANHATTAN|
|  25795607|MANHATTAN|
|  25797428|MANHATTAN|
|  25800989|MANHATTAN|
|  25801432|MANHATTAN|
|  25808666|MANHATTAN|
|  25828882|MANHATTAN|
|  25830754|MANHATTAN|
|  25842085|MANHATTAN|
|  25842809|MANHATTAN|
|  25859866|MANHATTAN|
|  25867274|MANHATTAN|
+----------+---------+
only showing top 20 rows
```

Figure 3. Incorrect Data for BRONX

After the method runs, there are no more faulty rows. Since, precision needs to be calculated, the number of true positives and false positives are calculated over 100 rows of the same data.

Therefore, we have the precision and recall for the above data as following:

$$TP = 100, FP = 0, FN = 0$$

Hence,

$$Precision = 100/(100 + 0) = 1$$

$$Recall = 100/(100 + 0) = 1$$

The precision and recall here is 1 which means that, all the records will be corrected to the respective borough names. The precision might be less than 1 if more records are taken into consideration since the only records that will be left out from this process are the records that have both the boroughs and zip codes as null which will be handled later in our cleaning process.

The next method, standardization of the NYC data results in correcting all the data which has Null values in any format. This reduces the data by at least 2% and the precision and recall for this result over a sample size of 100 is as follows:

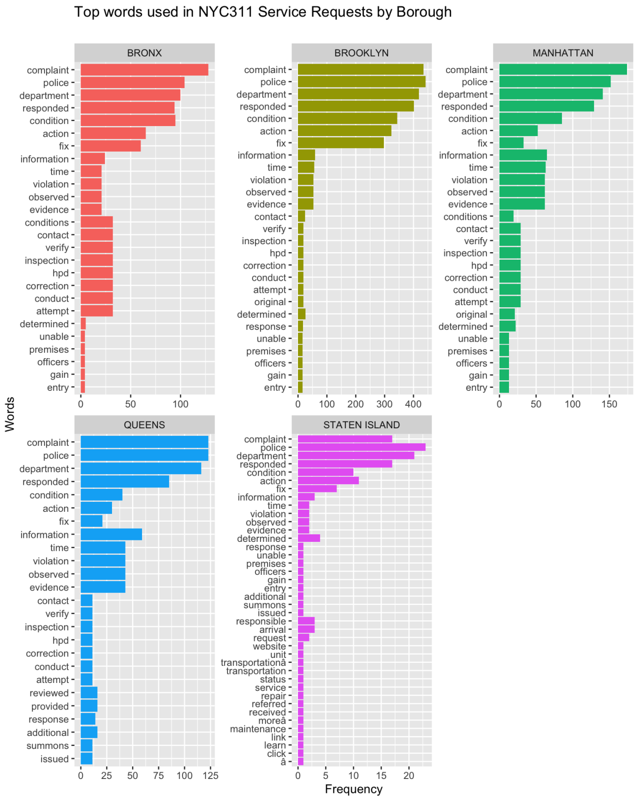


Figure 4. Top words used in service requests

$$TP = 100, FP = 0, FN = 0$$

Hence,

$$Precision = 100/(100 + 0) = 1$$

$$Recall = 100/(100 + 0) = 1$$

The next method, unionizes the cross street 1 and cross street 2 and drops Intersection street 1 and Intersection street 2 and since cross street 1 and cross street 2 are similar to intersection 1 and intersection 2. Dropping the 2 columns and merging cross street 1 and cross street 2 reduces the data size by 3.5%. The precision and recall for this result over a sample size of 100 records is as follows:

$$TP = 100, FP = 0, FN = 0$$

Hence,

$$Precision = 100/(100 + 0) = 1$$
$$Recall = 100/(100 + 0) = 1$$

The next method, agency mapping to agency name corrects all the agency name that are wrongly recorded for the given agency, assuming the agency are 32 distinct and are correct. One of the main reasons behind cleaning the data above using the above techniques and focusing on few particular columns is to answer the data questions to analyse and make a use out of the analysis.
For a better understanding, the data visualization in figure 4. answers our data question; What are the most frequent words used in a 311 service request call?
These kinds of visualizations help in making important decisions by the individual departments.

6 References

1. https://en.wikipedia.org/wiki/Precision_and_recall