Big Data - Fall 2019

Generic Profiling, Semantic Profiling & Analysis

Palak Raman Patel
NYU Tandon School of Engineering
Brooklyn, New York
prp313@nyu.edu

Soniya Chawla
NYU Tandon School of Engineering
Brooklyn, New York
sc7221@nyu.edu

Serena Lekhrajani
NYU Tandon School of Engineering
Brooklyn, New York
sl6813@nyu.edu

ABSTRACT

NYC Open Data makes the wealth of public data generated by various New York City agencies and other City organizations available for public use. Our project focuses on analyzing these data sets by implementing all steps of the data lifecycle that have become highly feasible due to big data infrastructure. These steps are namely : Generic Profiling, Semantic Profiling and Analysis. By doing so, we have processed 1900 datasets acquired from NYC OpenData. Our objective is to leverage the entirety of the data to extract statistics and interesting patterns over the New York grid. We have attempted to create an elaborate metadata for these 1900 datasets so that the usefulness of these datasets can be increased. Also they are semantically tagged to analyze and understand the patterns in the data. We have also researched deeper into the specialized area of 311 complaints involving some datasets to discover interesting patterns at a more granular level with the help of some questions based on the attributes of these datasets.This analysis evaluates and tries to understand the correlation of complaints with the agency,borough and population type of New York. We have represented these patterns with the use of visualizations to increase human readability and interpretation.

KEYWORDS

big data, data profiling, data analysis

1 INTRODUCTION

NYC OpenData contains records that are raw and untagged. Furthermore, each dataset individually also contains missing values, variety of formats, improper data types etc which classify as noise and mess with the actual patterns to mislead analysts. Since there are so many of them, they qualify to be what is called as ‘Big Data’. Analysing such datasets requires intensive data processing to learn and extract knowledge from the historical repositories contained in NYC Open data. We need solutions that are efficient in both memory utilization and execution time so that the large scale nature of the available data can be fully exploited. In particular, we follow a low-latency, memory-efficient solution to compute statistics to describe New York City from various perspectives.

As mentioned above, we performed three major steps to get through with this analysis : Generic Profiling, Semantic Profiling and finally, Analysis. Generic Profiling refers to deriving metadata that can be used for data discovery, querying, and identification of data quality problems. Semantic Profiling refers to tagging attributes semantically by gaining more information about the data through semantic extraction. Data Analysis as the name says, refers to mining interesting patterns from the processed datasets and representing them using visualizations.

In this sense, our new understanding about Big Data is very important. Not only because all the datasets have a huge amount of data, but analyzing patterns implies looking at creative and complex data structures. We decided to use all the frameworks and methodologies that we learned during the semester.

2 GENERIC PROFILING

The first task of the project was performing generic profiling on the NYC Open Datasets. We had 1900 datasets to process and did not have enough metadata for analyzing them. We extracted the following metadata for each column in each dataset :

- Number of non-empty cells
- Number of empty cells (i.e., cell with no data)
- Number of distinct values
- Top-5 most frequent value(s)

- Data types (a column may contain values belonging to multiple types)

Additionally, we also counted the total number of values as well as the distinct values for each of the data types. For columns that consisted of at least one value of type INTEGER / REAL, we obtained the Maximum, Minimum, Mean, and Standard Deviation For columns that consisted of at least one value of type DATE, we obtained the Maximum, and Minimum values For columns that consisted of least one value of type TEXT, we obtained Top-5 Shortest value(s) (the values with shortest length), Top-5 Longest values(s) (the values with longest length), and Average value length.

2.1 METHODOLOGY

We modularized this whole task into smaller functions in our code. We employed 'UDFs' or User Defined Functions to implement the functionality of each module. Pyspark User Defined Functions (UDFs) are an easy way to turn ordinary python code into something scalable. Since we had 1900 large datasets, scalability was an integral part of our consideration.

For faster processing, we have used spark inbuilt functions. We sorted the datasets according to size to minimize the execution time. The columns from each dataset are not read for every function instead we collected the data once and implemented an aggregate over the entire dataset. We avoided using collect and join operations as they are expensive. Also created random small subsets of data and processed to verify if there are any inefficiencies and errors in those parts. To increase the speed and decrease the processing time, we have used dictionaries.

2.2 CHALLENGES

Some of the challenges we faced while designing and implementing our solution can be summarised as follows.

- **Volume :** In this project, we have been dealing with data that actually qualifies as 'Big Data'. The datasets used for this task range from a few MBs to a huge GBs. This high volume multiplied by the number of users exhausted the capacity of Dumbo HPC many a times due to which it took a lot of time for one single execution to finish. The first 1600 datasets took around 6 hours for execution, next 200 datasets took around 18 hours and the last 100 datasets finished its execution in approximately 40 hours.

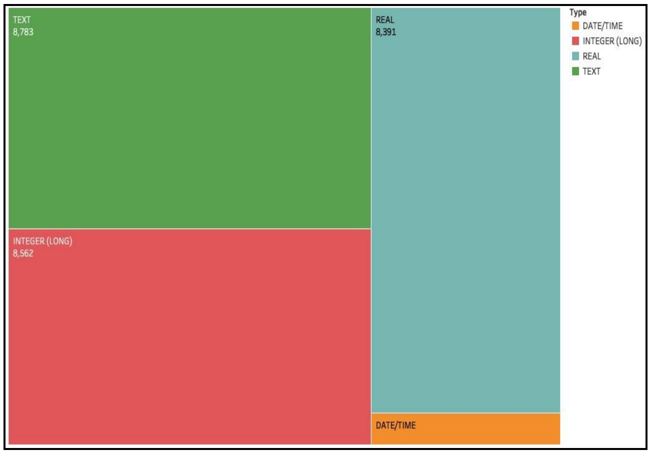

Figure 1 : Data Type Distribution

- **Efficiency :** We needed an algorithm that was scalable, efficient and accurate at the same time. Having the right trade off was crucial because if a code was too accurate, it would take several days to run, and making a code too efficient (running within an hour) would compromise the accuracy to a significant amount leading to incorrect results in the analysis step. Therefore, we repeatedly revised the code which involved quite some brainstorming and research.
- **Variety** : The data and its formats were not consistent across all datasets. In some cases, one single column of a single dataset also contained values of multiple formats and data types. This required thorough scrutinizing each column in order to obtain the correct metadata. Some of the columns had diverse values which interfered with clean execution of the code.

- **Missing Values :** Missing values treatment is difficult.To understand the column datatypes we need sufficient values to mark it. Also it makes much more sense to calculate minimum or maximum values and do the rest of the operations on it. But there were columns having missing values and inappropriately handling these missing values would have lead to poor knowledge extracted and also wrong conclusions. As missing values have been reported to cause loss of efficiency in the knowledge extraction process, strong biases if the missingness introduction mechanism is mishandled and severe complications in data handling,this was a major challenge while doing the base task.
- **Noise** : Data mining algorithms tend to assume that any data set is a sample of an underlying distribution with no disturbances. Data gathering is rarely perfect, and corruption often appear. Since the quality of the results obtained by a data mining technique is dependent on the quality of the data, tackling the problem of noise data is mandatory. There were values in columns which created complexities determining the types of the column and thus made the entire part ahead difficult to achieve.

3 SEMANTIC PROFILING

Big data profiling is about creating a semantic understanding of the data, so the data can be used to solve a business problem. For this task we extracted more detailed information about the semantics of columns.

3.1 TECHNIQUES

Some of the strategies we used to perform the semantic profiling can be summarised as follows.

- **LAT/LON Coordinates, Phone Number, Zip Code, Address, Person name,Website, Building Classification** : We used a regular expression to match the general structure of the commonly occurring entities.We thought this is a good strategy as values like phone number, zip code, websites have a common structure which we can exploit to get it's semantic label from regular expressions.
- **City Names, Borough, Street Name, Neighbourhood:** As the data belonged to New York,we got a compiled list of cities, streets and neighbourhoods.We also utilised the commonly occurring terms in neighbourhoods and addresses like avenue, street, boulevard along with abbreviations like St, Av, Blvd to match maximum specified values.
- **Car make,Vehicle Types, Color, CityAgency School Subjects, School Level :** Tagging was done by an extended version of strategy described above. There was a list of values and this list was used with a similarity measure for the column values to be tagged. Different similarity measures like Jaccard similarity and Levenshtein distance were used to get the values which might be similar but not exact same as the list of values provided by us.
- **School name, Business name** : For columns like these, utilising and getting a list like above didn't seem reasonable enough as there are many such schools and getting a match from that search was expensive computationally. So the strategy that we utilised was using some common words for school name detection like school, academy, high, children etc. which occur is most of the school names and a similar strategy was used for business names where words like corp, inc, LTD, LLC which occur in most of the business names.

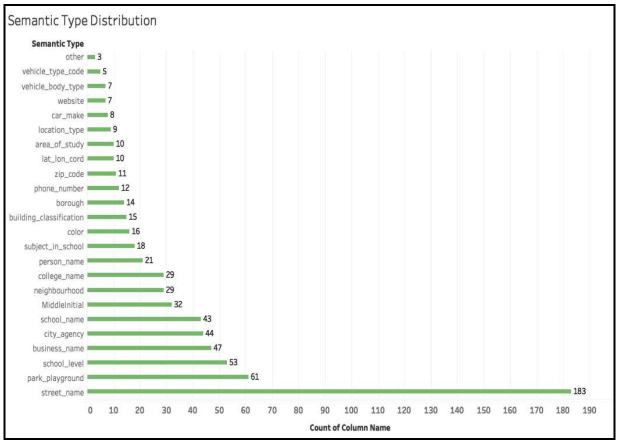

Figure 2 : Count of Columns for Each Semantic Type

Some of the benefits we observed while implementing the above strategies can be listed as below.

- **Regular Expression :** Harnessing the structure of certain values like phone numbers, websites, zip code etc. made the matching successful as only valid values were parsed.
- **List Matching:** Data belonged only to New York City, this was an advantage because we were able to match the names with the list of specified cities.This made the tagging accurate as we were assured that the tagging is correct based on the real time value.
- **Similarity** : This strategy was beneficial as the data that we were dealing with was incomplete and inconsistent, the distance measure made the matching more accurate by tagging the columns whichever were similar based on the list. The right data couldn't fall far from the actual list of values.
- **Common Words**: This strategy benefitted us as the column values had most of the keywords mentioned in the set. This made it easier to detect such columns and get the right tag. For every such column, we had a large set of values. This made it easier to tag the column to its correct type.

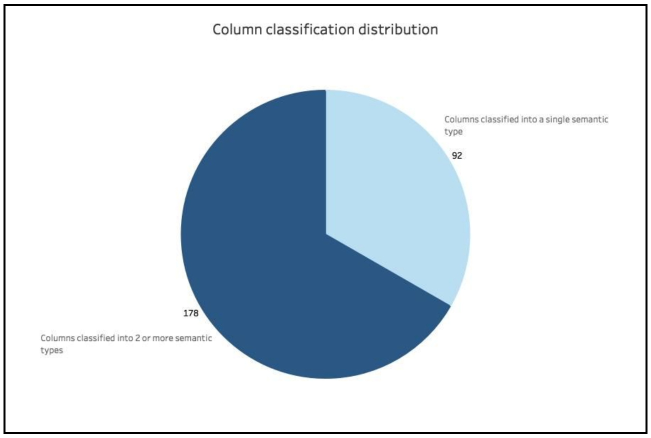

Figure 3 : Count of Columns Classified into Single & Multiple Semantic Types

While there were some benefits of using the various techniques, there were also quite a few limitations associated with it.

- **Regular Expression:** As the values in the data are collected from various agencies in New York, getting only the right values which will match with the regular expressions properly was a big challenge. Incomplete Values/Abbreviations got matched with wrong semantic types as well, junk values were matched with random columns which had a similar regular expression.
- **List Matching:** The memory consumption by such exhaustive lists along with matching with each and every value tag made the processing slower even though it was accurate. This approach will fail when we are dealing with data which is not just of New York but any state of city anywhere. Assembling and comparing such lists in that case will be infeasible.
- **Similarity:** The variety of data was humongous so matching and finding similarity to a small subset(comparatively) was inefficient. As we are dealing with "Big Data", getting all possible list of values and finding appropriate similarity measure was a limitation. As certain measures perform better with some data than others, this approach was limited in a way to get the most appropriate similarity measure.
- **Common Words:** This strategy is very limited and will fail when the business names will not contain certain keywords that we specified to match.To tag it as the correct type,this approach will just match certain values but values which don't have any of those keywords will make the task impossible to achieve. It was limited in the way that it worked with the current data but won't generalize.

3.2. PRECISION AND RECALL

Using the manually labelled semantic types and the predicted labels, we calculated the precision and recall.

$$precision = \frac{\text{number of columns correctly predicted as type}}{\text{all columns predicted as type}}$$

$$recall = \frac{\text{number of columns correctly predicted as type}}{\text{number of actual columns of type.}}$$

Semantic Type	Precision	Recall
Person Name	0.956	0.57
Middle Initial	0.340	1
Business Name	0.580	0.875
Phone number	0.909	0.909
Address	0.617	0.717
Street name	0.415	0.727
City	0.650	0.710
Neighborhood	0.814	0.687
LAT/LON coordinates	1	1
Zip code	0.800	0.900
Borough	0.882	0.750
School name	0.520	1
Color	0.583	0.875
Car make	0.800	0.857
City agency	0.857	1
Areas of study	1	1
Subjects in school	1	1
School Levels	0.928	1
College/University names	0.300	0.750
Websites	1	0.750
Building Classification	0.909	1
Vehicle Type	0.600	0.800
Type of location	0.710	1
Parks/Playgrounds	0.330	1

Table 1 : Precision and Recall For each Semantic Label

3.3 CHALLENGES

Some of the challenges we faced while performing the semantic profiling could be as follows.

- Certain column names(even during manual tagging) were not descriptive enough and did not make much sense as to which semantic tag they belong to. For instance, Middle Initial in certain columns had some random single alphabets and Borough columns also had values which were single alphabet values. In such situations, columns were tagged with multiple types which were ambiguous for the code to match.
- Many columns had dirty data therefore matching them was very difficult to achieve.

4 ANALYSIS

Each day, NYC311 receives thousands of requests related to several hundred types of non-emergency services,including noise complaints, plumbing issues, and illegally parked cars. These requests are received by NYC311 and forwarded to the relevant agencies, such as the Police, Buildings or Transportation. The agency responds to the request, addresses it and the request is then closed.

At the beginning of the project we had a very broad idea of the analysis we wanted to perform. We realize that urban interactions are really complex and hard to interpret without much domain knowledge. This drill-down approach from more general ideas to more specific concepts while we were getting acquainted with the data let us define a more concrete and relevant problem.

4.1 METHODOLOGY

The data set used for this part of the analysis is “NYC Open Data 311 Service Requests from 2010 to Present”.

Spark introduces a programming module for structured data processing called Spark SQL. It provides a programming abstraction called DataFrame and can act as distributed SQL query engine. Our algorithm required multiple SQL queries to process the raw datasets. Various SparkSQL queries involving aggregation functions such as Count, Min, Max, typecasting to convert the required columns into their detected data types for correct results (String to Date). Converted this data set to a temporary view and extracted query results into a dataframe and further into a CSV. We used Tableau to create visualizations for better understanding of the results obtained.

4.2 RESULTS

We started analyzing this dataset by looking for the frequent complaint types in each of the boroughs. The graphs in the image below shows some interesting findings.

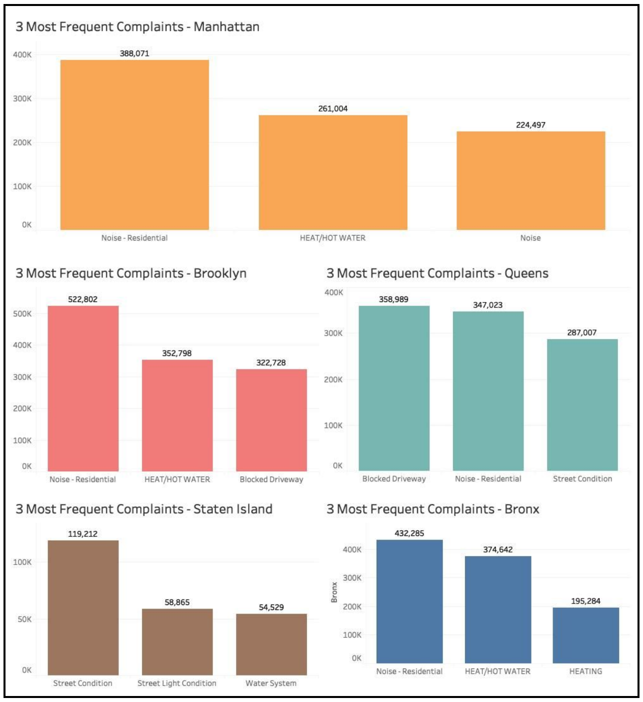

Figure: Frequent Complaint Types By Borough

We can observe through this analysis that the complain types of each borough align with actual situation of NYC. Manhattan is extremely populated and is also home to 70% of the city's traffic. This fact justifies that the majority of complaints from Manhattan were for Noise(both residential and other). Heat/Hot Water being the second. Heat/Hot Water complaints are also consistent across Brooklyn and Bronx. Noise complaints in general have been consistent over Bronx, Brooklyn as well as Queens. As far as Queens and Brooklyn is concerned, Blocked Driveway complaints have been pretty common. It can be seen that the complaints in Staten Island are completely different, Since it is across the river and less populated, Noise complaints are not many. Additionally, we know that Staten Island is less developed than the rest of NYC, it can be seen that their top 3 complaints are for Street Condition, Street Light Condition and Water Systems which are basic living necessities.

After observing the different compliant types, the next step in the analysis was to find if the same complaint types are frequent in all the five boroughs of the city.

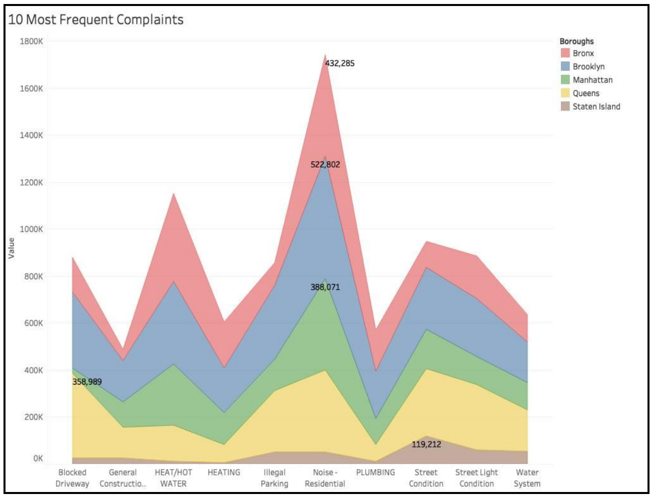

Figure: Comparison of Complaint Types across Boroughs

From the above graph, it can be observed that for Bronx, Brooklyn and Manhattan, the maximum complaints have been recorded for Noise in Residential areas.This can be justified since these boroughs are quiet populous than the others. In Queens, maximum complaints have been recorded for Blocked Driveway and for Staten Island the majority of the complaints have been made for Street Condition.

In the next part, we aim to analyze by borough which agency has recorded the maximum number of complaints. In Manhattan and Queens, majority of the requests have been recorded by NYPD. This is because the overall crime rate in these boroughs is higher than others. Next, in the Bronx and Brooklyn, maximum complaints have been recorded by the Department of Housing Preservation and Development. This reflects that housing quality and overall environment is in a poorer state in these boroughs comparatively. The reason why maximum complaints in Staten Island is recorded by the Department of Transportation is pretty self explanatory. All other boroughs have subways which is a very convenient mode of transport whereas all the residents of Staten Island have to take a ferry to get anywhere into NYC. The ferry may be free of charge but it is slow and inconvenient for commuting on a daily basis.

5 CODE AND OUTPUT FILES

Our code for each task is shared through the Github Repository and the JSON output files are available at the NYU HPC Dumbo HDFS directory /users/sl6813/FinalProject_BigData_prp313_sc7221_sl683.

6 CONCLUSION

The boom of Big Data has allowed the analysis of complex urban patterns that were unmanageable before. This brings opportunity for the discovery of improvement opportunities for more sustainable and better cities. Through this project, we became proficient with techniques, issues and tools to handle, process, analyse and visualize big data.

REFERENCES

[1] Adams, M.N.: Perspectives on Data Mining. International Journal of Market Research 52(1), 11–19 (2010)

[2] Economist Intelligence Unit: The Deciding Factor: Big Data & Decision Making. In: Capgemini Reports, pp. 1–24 (2012)

[3] B. Saha and D. Srivastava, "Data quality: The other face of Big Data," in 2014 IEEE 30th International Conference on Data Engineering (ICDE), 2014, pp. 1294–1297.

[4] M. Janssen, H. van der Voort, and A. Wahyudi, "Factors influencing big data decision-making quality," J. Bus. Res., vol. 70, pp. 338–345, Jan. 2017.

[5] I. Taleb, R. Dssouli, and M. A. Serhani, "Big Data Pre-processing: A Quality Framework," in 2015 IEEE International Congress on Big Data (BigData Congress), 2015, pp. 191–198.

[6] P. Ciancarini, F. Poggi, and D. Russo, "Big Data Quality: A Roadmap for Open Data," in 2016 IEEE Second International Conference on Big Data Computing Service and Applications (BigDataService), 2016, pp.

210–215.

[7] N. Abdullah, S. A. Ismail, S. Sophiayati, and S. M. Sam, "Data quality in big data: a review," Int. J. Adv. Soft Comput. Its Appl., vol. 7, no. 3, 2015.

[8] Li, Y., Krishnamurthy, R., Raghavan, S., Vaithyanathan, S., Jagadish, H.V.: Regular expression learning for information extraction. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 21–30 (2008)

[9] R. Pearson. Mining imperfect data: Dealing with contamination and incomplete records. In Proc. 2005 SIAM Int. Conf. Data Mining, New Port Beach, CA, April 2005.

[10] Pipino, L., Lee, Y., Wang, R.: Data quality assessment. Commun. ACM 4, 211–218 (2002)