Analysing NYC 311 Service Requests

CSE 6242: Data and Visual Analytics

Project Final Report

Aishwarya Ramaswamy Govindaraj, Deepak Ravindran, Karthik Kannan, Mahita Mahesh, Seema Suresh

Introduction

'3-1-1' is a special telephone number used to access to non-emergency municipal services in the United States. The service requests are made via telephone or online. Having a system for visualizing and prioritizing these requests would make it possible for them to be handled efficiently by the Municipal Departments. There is no system that we know of, at present, that performs department-wise analysis to prioritize received requests based on level of impact. The existing visualization methods cover complaints received by only a few departments over a short period of time. Also, these are not interactive or dynamic.

Municipal Departments would benefit the most from the implementation of this system, as they can identify the more critical requests and service them first. Visualization methods can be used to identify departments that have poor response time or geographical areas where coverage is inadequate. Overall, the Municipality can use the results to deploy their workforce in such a way that they give satisfaction to the largest number of people in the most timely manner.

Problem Definition

The objective of our project is to analyze 311 service request data in New York City, from 2010 to the present, in order to assign a priority level to requests received by each department, based on level of impact. The level of impact is defined in terms of the number of citizens affected by a particular problem, which is reflected in the number of complaints received regarding that problem. Visualization techniques are used to analyze service requests and success rates by department, precinct, request type etc., to identify patterns, provide search and filter capabilities and analyze trends over the period of time.

Survey

Supervised machine learning is the search for algorithms that reason from externally supplied instances to produce general hypotheses and predicts future instances. A classifier is used to assign class labels to the testing instances with known values of predictor features, but unknown value of class label [1]. The possibility of integrating two or more algorithms together to solve a problem should be investigated. The strengths of one method could complement the weaknesses of another and a single classifier may not perform as well as a good ensemble of classifiers [5]. [11] describes how to integrate existing filtering techniques and show that they should be used together judiciously, since the way of integration can greatly affect performance.

The Apolo [8] system uses a mixed-initiative approach— combining visualization, rich user interaction and machine learning—to guide the user to incrementally and interactively explore large network data and make sense of it. Data should be represented in a way that allows the readers to infer useful information. The survey presented in [2] introduces a new way of looking at comparison using the basic forms of visual design. Yi et. al [3] propose seven general categories of interaction techniques widely used. These categories are organized around a user's intent while interacting with a system.

[4] describes a map-based visualization system which uses animation to convey dynamics in large data sets, preserving the viewer's mental map and offering readable views at all times. [6] describes how to represent a subset of all map areas that are of interest to the user at a given time. The amount of information displayed for each map area is adjusted according to this area of interest by focus & context techniques, displaying it in detail, simplifying other areas. Kapler et al [7] develop a method to visualize the spatial inter-connectedness of information over time and geography within a single, highly interactive three-dimensional (3-D) view. Robertson et al. [9] proposes two trend visualizations that use static depictions of trends : trends overlaid simultaneously in one display and a small multiple display to show the trends side-by-side. [10] presents a new algorithm to explore and visualize multivariate time-varying data sets by identifying important trend relationships among the variables changing over time and how those changes are related to each other in different spatial regions and time intervals. [12] expands on heatmap approaches with novel ways of displaying and interacting with distribution data. [13] describes D3 and demonstrates how D3 enables animation and interaction with dramatic performance improvements. [14] proposes a new technique, a hyperclique-based data cleaner and evaluates in terms of impact on the subsequent data analysis, clustering and association analysis. [15] implements a model which supports interactive response times; generates clutter-free visualization for large results; and shows hidden details in a summary through overlay heat maps.

Proposed Method

The functionality of our system is two-fold. First, the system classifies an incoming service requests as being high priority or not via supervised Machine learning, using random forest classification technique. The incoming request is classified on the basis of similar requests in the past. If multiple complaints were lodged about a similar single issue in the past, we can assume that the incoming complaint would also affect several citizens in the present. The classification model is trained on over a million data points, taken largely from the data from 2015. In the training data, service requests are tagged as high priority if multiple complaints are received regarding the same problem, from the same area, within a 12 hour time frame. This indicates that the problem is affecting many citizens and thus, has larger impact. Using this system, we intend to be able to classify the first occurrence of a service request as high or low priority rather than wait for multiple requests coming in for the same issue before realizing that it is a high impact situation. This would also decrease the total amount of complaints lodged that the responding agencies would have to deal with.

The other functionality of our system is to use visualization techniques to derive insights from the service request data to understand patterns and trends. Map-based techniques make it possible for responding agencies to see the geographic distribution of complaints. This would make it easy to allocate specific complaint instances to the regional office closest to the site of the complaint. Having filtering capabilities for the data displayed ensures that a particular agency can view complaints relevant to them, without having to deal with irrelevant data. Complaints displayed on the map can also be filtered by date using interactive menus and any time frame can be selected from 2010 to the present.

Some of the features of the map-based visualization include:

- Grouping of requests by location and type of complaint
- Greater spatial resolution in terms of location accuracy (by borough, community board, street and building level) and at the finer stages by latitude, depending on the level of zoom used
- Fields displayed in the popup:
    - Complaint type
    - Agency to which the complaint is addressed
    - Street address where the complaint was lodged

A heatmap visualization was also developed as a more continuous form of representation for the data. Agencies can easily identify the parts of the city from which the highest number of complaints are received. This information can potentially be used to ensure that agencies deploy their limited workforce in different areas in the most efficient way possible.

Development Process, Experiments and Evaluation

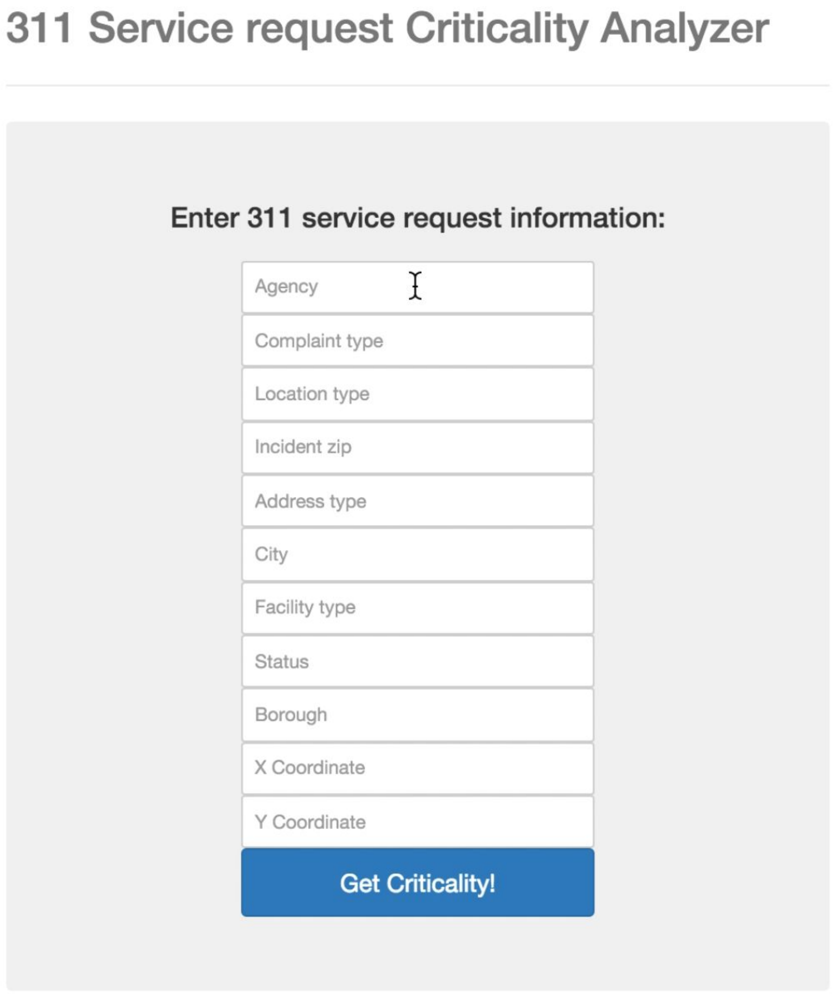
The data needed to be cleaned before any sort of analytics could be performed on the same. OpenRefine was used for cleaning the data. A number of cases were observed where similar types of complaints were registered in different ways and not categorized properly. Such data was transformed with the help of the *Text Facet* feature of Google OpenRefine using 'Descriptor' and 'Complaint Type' attributes of the data. Some complaints did not have any Descriptor field entry, but there were similar instances which were mentioned under a different Complaint Type with the Descriptor identical to the Complaint Type of the previous complaints. In addition, the *Transform Columns* feature of OpenRefine was used to properly label the Descriptor and Complaint Type in cases where complaints did not have a descriptor, or when two similar complaints were stored under two slightly different Complaint Types, which could be transformed to a single Complaint Type with different Descriptors for each. The analytics were performed after cleaning the data.

The classifier model was implemented using the Random Forest algorithm. Apache Spark's MLlib machine learning library incorporates various classification algorithms, Random Forest being among them. This library was specifically chosen because of the scalability advantages that Spark provides if the algorithm were later to be run on a cluster. The current training dataset (of over 1 million data points) can now easily be increased by a large multiple and the implementation would still work if deployed on a cluster (such as AWS EMR).

Since supervised machine learning techniques were used, the dataset had to be tagged - individual requests were tagged as critical or not. This was done by setting an arbitrary heuristic that if a specific complaint instance had multiple other service requests lodged in a short period of time (12 hours), around the same location, with the same complaint type, then all these requests refer to the same incident and they would all be tagged as critical. For each request, this test was performed to identify if it was critical or not. The resulting tagged dataset was used to train the Random Forest model. Random forest is a notion of the general technique of random decision forests. This is an ensemble learning method, which can be used for classification, regression and other tasks. For classification, a multitude of decision trees are constructed at training time and the output class is the mode of the classes predicted by the individual trees.

The features used for training the classifier model were: Agency Name, Complaint Type, Location Type, Incident Zip, Address Type, City, Facility Type, Status, Borough, and (x,y) Coordinates of the incident. Different tuning parameters were experimented with for the Random Forest classifier. A forest of 3 trees, with a maximum depth of 20 and the "Gini" impurity gave an average accuracy of 82.47% over 5 runs with random 70-30 split of the dataset into training and testing sets.

A User interface was developed to query the model, consisting of a form into which a user can enter information about a service request and receive a prediction on whether the request is potentially critical or not. This will help Municipal agencies to identify critical issues when the first complaint is lodged rather than having to wait for multiple complaints to come in.

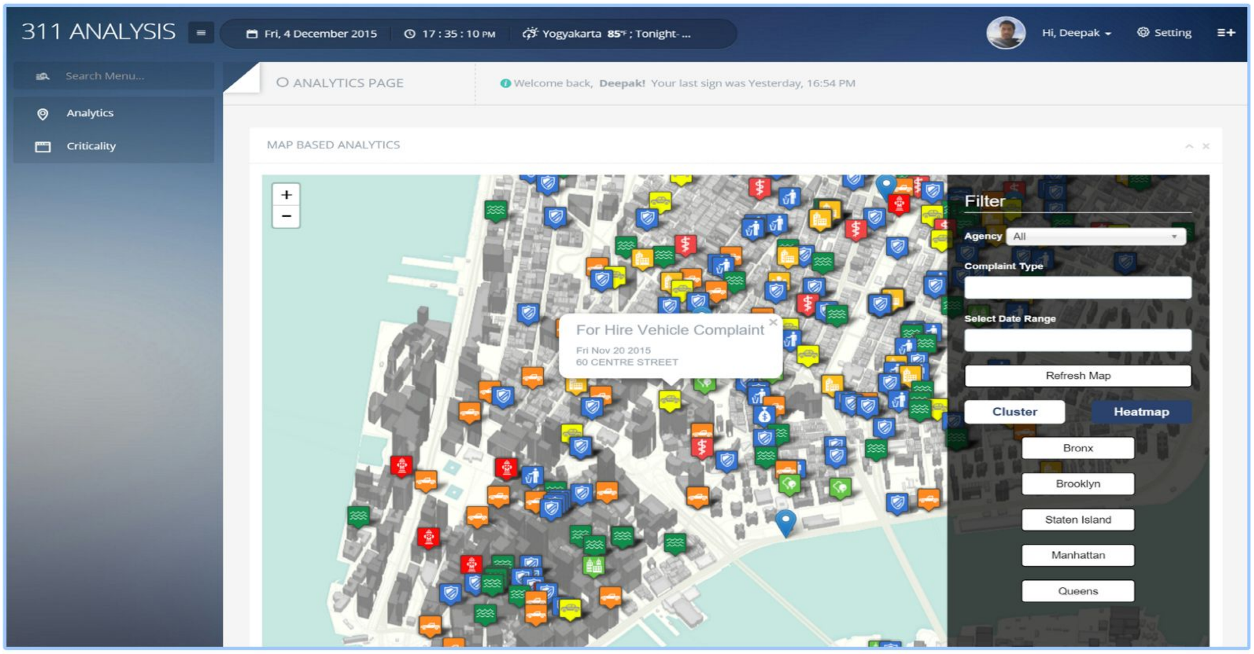


311 Service request Criticality Analyzer

Enter 311 service request information:

Figure: Classification User Interface

The data visualization user interface is a web application consisting of a dashboard built in such a way that it can be easily plugged in with newer features in the future. In the main section of the dashboard is the interactive map visualization. The map was created using Leaflet, Mapbox and OpenStreetMap. OpenStreetMap provides the actual map data(in the form of tiled web maps) and is used by Mapbox to create custom designs as per the requirement. This includes highlighting specific cities/regions, styling the map(dark, light, etc.) and creating custom symbol markers on the map. This Mapbox data is then fed to Leaflet which is used to create the actual interactions for the user. Leaflet uses layering to create the various elements. The map is the base layer and markers/shapes are drawn on newer layers. It can load feature data from GeoJSON files, style it and create interactive layers, such as markers with popups when clicked. It also provided various events for zooming and panning which were then used to dynamically alter the data displayed.



Welcome back, **Deepak!** Your last sign was Yesterday, 16:54 PM

For Hire Vehicle Complaint
Fri Nov 20 2015
60 CENTRE STREET

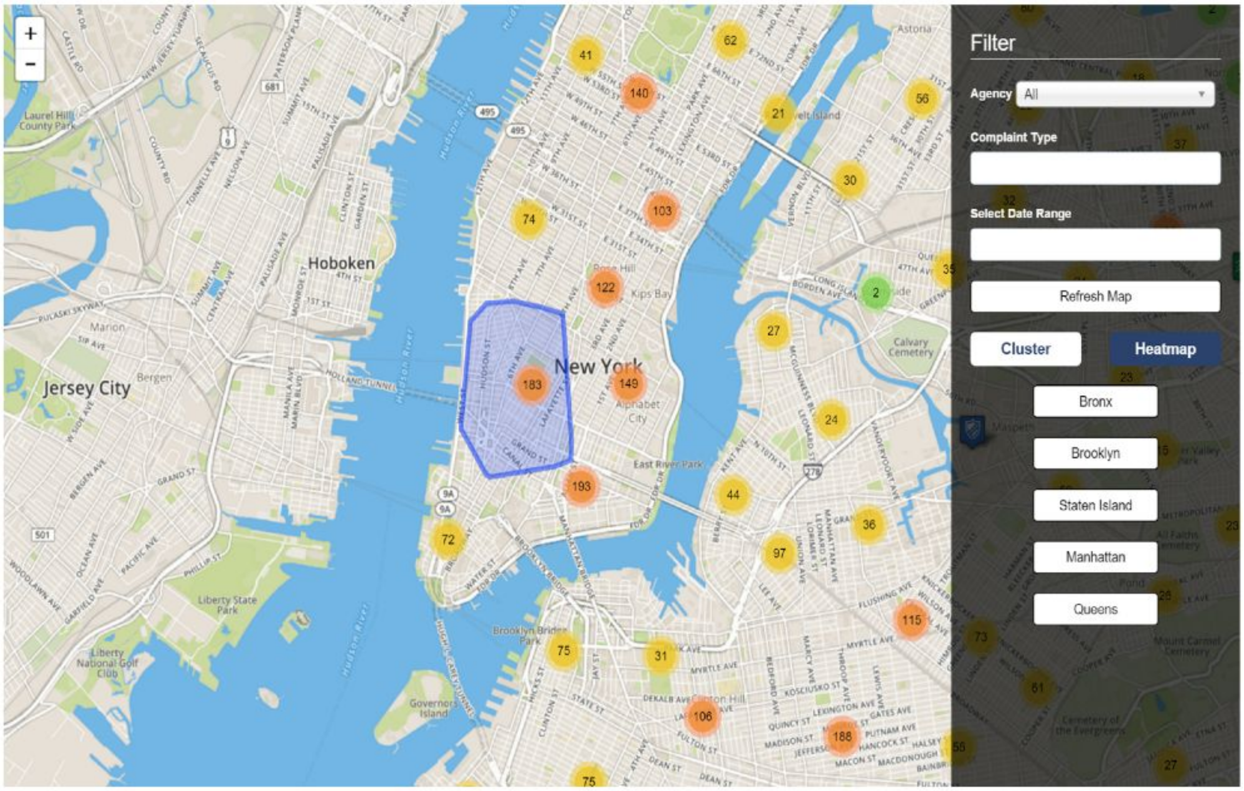
Agency: All

MAIN DASHBOARD WITH ICONS REPRESENTING DIFFERENT COMPLAINT TYPES

Figure: Data Visualization User Interface

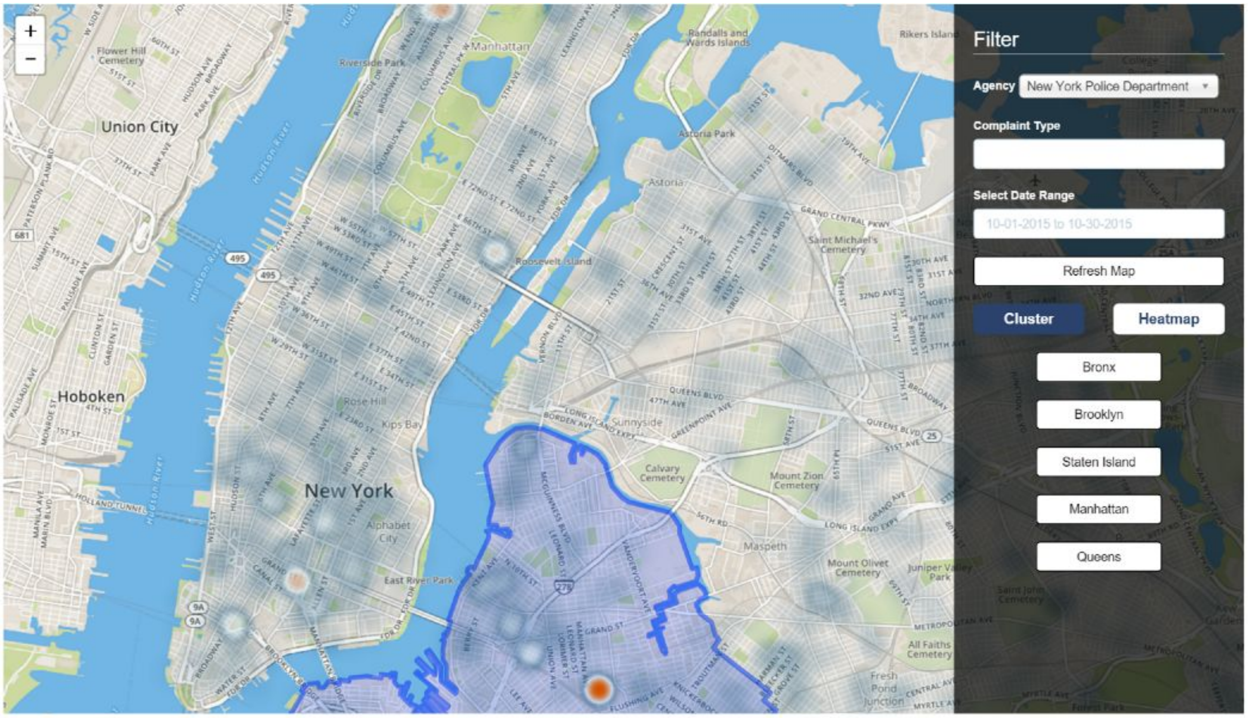
The map based visualization was developed with the goal of making it easy to view and draw insights from a very large amount of complaint information. Clustering is done based on location in order to improve readability and to reduce clutter due to a large number of points being marked on the map. Depending on the granularity of the zoom, the amount of clustering changes. As the user zooms into a particular area, the data points split into smaller clusters. At the closest level, individual icons can be seen for each complaint at the correct address. Information related to the individual complaint is displayed in a pop-up bubble, when the icon is clicked.

A heatmap visualization was also developed to get a more continuous representation of the complaint data. At a high level of zoom, this shows the actual complaint locations. When zoomed out, it also allows users, especially agency personnel, to quickly identify areas in the city experiencing high complaint traffic, which could be critical complaints, requiring immediate action.



CLUSTERED REPRESENTATION OF COMPLAINTS

The visualization includes many filtering options for the user to view specific data. The time range menu allows any subset of data from 2010 to the present day to be selected. The map automatically displays the data for complaints that have been lodged over the most recent week. Complaints can also be filtered according to the agency to which the complaint is addressed and according to the type of complaint (eg. Noise, fallen tree, parking issues, etc.). There is also an option to display the boundaries of the boroughs into which New York is divided geographically, to get a more area-specific visualization. For example, in the displayed image, the borders of the borough of Brooklyn are highlighted in blue.



Agency: New York Police Department

Select Date Range: 10-01-2015 to 10-30-2015

HEAT MAP REPRESENTATION OF COMPLAINTS

A vertical stacked bar chart was created to illustrate the distribution of complaints among the 40 different agencies and to analyse the performance of the agencies based on the distribution of the 'Status' field in the complaint database. There were eight possible statuses for the complaints - open, closed, pending, assigned, unassigned, started, draft and email sent. We were able to infer several useful conclusions from the barchart. For example, the number of complaints directed to the top two agencies, the Department of Housing Preservation (HPD) and the New York Police Department (NYPD) was greater than the sum of the complaints received by the remaining agencies. There were also a number of agencies such as the Community Hall (CHALL) where the status for all received complaints is 'Email Sent'. This type of analysis also gives us an idea about the size of the workforce that each of these agencies has to employ to deal with the complaints they receive and whether this workforce is mobile or perform desk jobs. The barchart is shown below.

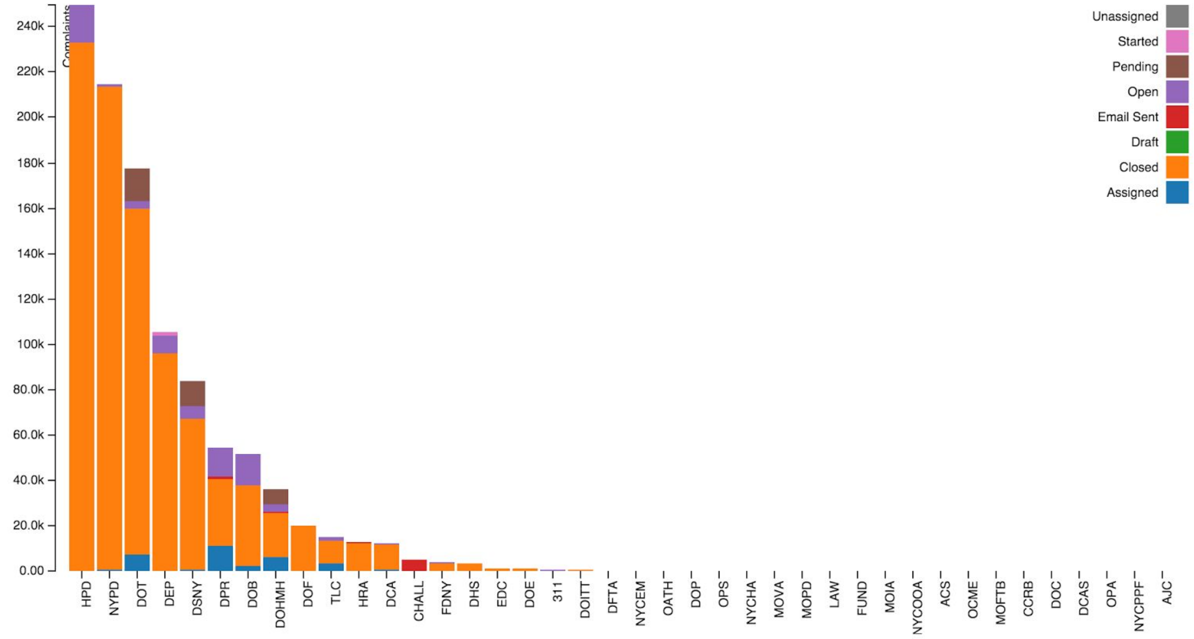


Figure: Stacked Barchart with distribution of complaints across agencies with status

Distribution of Team Member Effort

All team members contributed similar amount of effort.

Future Work

Using visualization, we can analyse how the criticality of various complaint types has varied over the years. This could provide useful insight into identifying future trends and bettering the response system. With this information, we can adjust the classification model dynamically, to be in line with the changing trends. We can also train the classifier with different classification models.

Conclusion

The classification system was able to achieve an accuracy of 82.47% in predicting the criticality of a particular complaint instance. The visualizations provided insights into the nature of the complaints received by different agencies and trends in the complaints received and how they were resolved. The visualizations were also useful in identifying the number of complaints in each locality. Thus, using this system would allow municipal agencies to use manpower and resources to address critical tasks quickly and efficiently. This would help reduce the number of complaints lodged and reduce overhead. We believe the system would also improve the satisfaction levels among the general public with regards to how municipal issues are resolved.

References

[1] S. B. Kotsiantis, ”Supervised Machine Learning: A Review of Classification Techniques”, July, 2007.
[2] Michael Gleicher, Danielle Albers, Rick Walker , Ilir Jusufi , Charles D. Hansen and Jonathan C. Roberts, “Visual comparison for information visualization”, 2011
[3] Ji Soo Yi, Youn ah Kang, John T. Stasko, Member, IEEE, and Julie A. Jacko, “Toward a Deeper Understanding of the Role of Interaction in Information Visualization”, November/December 2007
[4] Daisuke Mashima, Stephen G. Kobourov, and Yifan Hu, “Visualizing Dynamic Data with Maps”, September 2012
[5] S. B. Kotsiantis · I. D. Zaharakis · P. E. Pintelas,”Machine learning: a review of classification and combining techniques”, November 2007
[6] G. Fuchs, H. Schumann, “Visualizing Abstract Data on Maps”, 2004
[7] Thomas Kapler, William Wright, “GeoTime information visualization”, 2005
[8] Duen Horng (Polo) Chau, Aniket Kittur, Jason I. Hong, Christos Faloutsos, “Apolo: Making Sense of Large Network Data by Combining Rich User Interaction and Machine Learning”, 2011
[9] George Robertson, Roland Fernandez, Danyel Fisher, Bongshin Lee, and John Stasko, “Effectiveness of Animation in Trend Visualization”, November / December 2008
[10] Teng-Yok Lee and Han-Wei Shen, “Visualization and Exploration of Temporal Trend Relationships in Multivariate Time-Varying Data”, November/December 2009
[11] Chen Li, Jiaheng Lu, Yiming Lu, “Efficient Merging and Filtering Algorithms for Approximate String Searches”, 2008
[12] Awalin Sopan, Manuel Freier, Meirav Taieb-Maimon, Catherine Plaisant, Jennifer Golbeck & Ben Shneiderman, “Exploring Data Distributions: Visual Design and Evaluation”, 2013
[13] Michael Bostock, Vadim Ogievetsky, and Jeffrey Heer, “D3: Data-Driven Documents”, December 2011
[14] Hui Xiong, Gaurav Pandey, Michael Steinbach and Vipin Kumar, “Enhancing Data Analysis with Noise Removal”, 200X
[15] Nivan Ferreira, Jorge Poco, Huy T. Vo, Juliana Freire, and Claudio T. Silva, “Visual Exploration of Big Spatio-Temporal Urban Data: A Study of New York City Taxi Trips”, December 2013