Final Project Writeup

June 2, 2021

0.0.1 University of Virginia

0.0.2 DS 5559: Big Data Analytics

0.0.3 NYC Housing Complaints Dataset Report

0.0.4 Holden Bruce (HAB6XF), Dara Maguire (DKM2BD), Francisco G. Estrada (FGE8TJ)

0.0.5 1. Abstract

In many ways, New York City is a city of income inequality. As 2/3 of our group are NYC residents, we were motivated to see if we could quantify this inequality, specifically by looking at housing complaints and trends related to the severity of complaints and how long these incidents take to be resolved. After sourcing data on housing complaints and income by zip code and neighborhood, we built a model to predict the time it took for different complaints to be addressed by the City and marked as "Closed" based on the type of complaint, the location of the complaint (specifically the zipcode, including income and population data for each zip code), and the time the complaint was made (specifically year, month). Although our exploraty data analysis did produce interesting results - namely that there was notable variation in the time and number of complaints between different neighborhoods (that seemed to be related to income differences across therse neighborhoods) - our model performance was ultimately lacking. We discuss future tasks to improve our model performance, specifically focusing on address-level data rather than zip-code level data, including average rent, building age, and distance to subways and schools.

0.0.6 2. Data and Methods

Our analysis revolved around two main data sources: complaint data from the New York Department of Housing ((https://data.cityofnewyork.us/Housing-Development/Housing-Maintenance-Code-Complaints/uwyv-629c, https://data.cityofnewyork.us/Housing-Development/Complaint-Problems/a2nx-4u46)), and NY state population and income data, organized by zip code and sourced from the Citizen's Committee for Children of New York (https://data.cccnewyork.org/data/table/66/median-incomes#66/107/62/a/a, https://data.cccnewyork.org/data/table/97/total-population#97/147/62/a/a). We then included neighborhood mappings using UHF data.

"The learning problems that we consider can be roughly categorized as either supervised or unsupervised. In supervised learning, the goal is to predict the value of an outcome measure based on a number of input measures; in unsupervised learning, there is no outcome measure, and the goal is to describe the associations and patterns among a set of input measures." - Preface, ESL

What we were attempting to do is run a supervised learning algorithm to predict the closeTime given

a series of features from the NYCOpenData Housing Complaints dataset. One approach would be to run a statistical regression to predict the value. Another approach would be to think about this more like an unsupervised approach where a new complaint is compared to other complaints like it and then a prediction of that new complaints possible behavior (how long will the NYC government take to close the ticket) would be made based on how other complaints behaved in the past. We tried both.

Luckily, the OpenDataNYC initiative emphasizes good record-keeping, so the data we worked with was relatively clean. Still, we needed to do a bit of cleaning to make sure the data was ready to be fed to our models.

0.0.7 2.1 Code Development

2.1.1 Data Import and Preprocessing Data Import

With our data easily accessible from the sources outlined above, we were able to start our analysis with four main csv files, each respectively containing complaint details, complaint timelines, income data, and population statistics. The City of New York updates the Housing Complaint data every month. The income and population data are census data and are not updated as often.

Specifically, we read the Complaint Problems file and the Housing Maintenance Code Complaints file into Spark and merged them on the ComplaintID column, which joins the two datasets. Then we read in the NY State Population file and merged it with the merged Housing dataset. We repeated that process for the NY State Income file. After that, we consolidated the Zipcodes down to neighborhoods by using the UHF codes, which are the "United Hospital Fund Codes" signifying which Zipcode clusters make up which neighborhoods.

We thought that it made more sense to aggregate at the neighborhood level than at the Zipcode level. We were primarily interested in studying how income and location impact how long it takes a housing complaint to be closed, so grouping by these neighborhoods maintains much of the socioeconomic makeup while allowing us to not be so granular.

Finally, we renamed a few of the columns to improve readability and then wrote that dataframe to the 5_NYC_Merged_Data_May3.csv file for later use. The May 2021 file contained over 3.3 million records and 34 columns/variables.

Preprocessing: closeTime

Our first preprocessing step was to calculate our variable of interest - "closeTime", or the length of time that a complaint took to be filed as resolved. We converted the ReceivedDate and StatusDate columns into a DateType and then calculated the difference between these two columns to generate our response variable. The possible values of our response variables ranged from -180 (clearly a data error) to 6531 days, with an average complaint closing time of 13.6 days and a standard deviation of 37.6 days.

Preprocessing: Missing and errant observations

Our next step in preprocessing was to identify and address missing data. We found that only a few columns had missing data, namely statusDate, closeTime, and familiesWithChildren. We also noticed several observations with negative close times, which was obviously an error. While the percentages of missing and errant data were arguably small enough to drop, we dug deeper into these observations to make sure we were not introducing bias in our sample by dropping them.

We found out that the observations with negative closeTimes were related to a single building-level complaint in Rockaway, and we therefore dropped those observations. For the columns whose closeTime was missing as a result of the complaint still being outstanding, we considered replacing the data with the current date so that we could maximize closeTime without excluding these datapoints altogether. However, we chose to drop the small number of remaining rows with null values because we felt that extrapolating the response variable would add bias to our model.

Preprocessing: Data cleaning

An additional step in preprocessing was to ensure that all variables were of the right type. For example, we cast income from a string to an integer so that we could treat it as a quantitative variable. We also added some variables from the data we already had, such as month and year that the complaint were issued.

Preprocessing: OneHotEncoding and categorical variables

With the remaining categorical variables we used OneHotEncoder to convert the categorical features into binary vectors and assemble into a single feature vector using VectorAssembler.

Preprocessing: Variable reduction

Among the variables available to us from this complaint table, several were descriptive IDs which we dropped in our analysis (ComplaintID, problemID, UnitTypeID, SpaceTypeID, typeID, etc.). The variables from the complaint data that we focused on in our regression analysis instead included those such as: SpaceType, Type, MajorCategory, MinorCategory, Code, and Zip.

2.1.2 Data Splitting and Sampling We used randomSplit() to split the train_final_pipe dataset into a 70/30 split: 70% allocated to the training set and 30% allocated to the test set.

2.1.3 Exploratory Data Analysis with two Graphs Location exploration using spark sql:

One of our thoughts when starting this project was that there is likely a disparity between rich and poor neighborhoods. If you have money, you probably won't be calling 311 to complain about your housing situation because you could either move to a new location or hire someone to fix the problem for you. On the other hand, if you do not have money, moving or hiring someone are not options you have available to you. Which means that you are reliant on your landlord to help you or, when the landlords are delinquent and won't fix the problem, you are reliant on the State for support. 311 is the last line of defense for people who have housing problems and no way to solve them.

In this section we cherry picked a few zipcodes; some rich, some poor. While this is not a rigorous test and shouldn't be considered statistically significant, we thought it was interesting to see how the number of 311 Housing Complaints varies depending on location.

With our cleaned data in hand, we first took a look at a couple anecdotal samples. Selecting a couple zip codes that we thought were representative, we found that there were 44,488 complaints from the Bedford-Stuyvesant zipcode of 11216 while the West Village 10014 zipcode had only 9,108.

Of complaints from the same West Village zipcode, only 408 took longer than 60 days to be closed, while the same BedStuy zipcode had 1,294 complaints taking longer than 60 days to be closed.

A more extreme comparison can be seen between Morris Heights in the Bronx and TriBeCa in Manhattan. TriBeCa had only 22 complaints in the past 14 years take longer than 60 days to be closed while Morris Heights had 3,852 complaints taking longer than 60 days to be closed.

It was alarming seeing these numbers at the start of our data exploration. We had barely begun to scratch the surface and we were already seeing drastic differences in how the City responds to housing complaints, and the biggest factor so far seemed to be wealth. The question of whether this truly was an example of income inequality stayed in our minds throughout the course of this project.

Correlations In the beginning of our exploratory data analysis, one of our first steps was calculating simple correlations between the response variable and some of the potential numerical predictors. However, none of these correlations were significant.

Next in our preliminary data exploration, we dug into the potential categorical predictors. One of the predictors we used in our model was "spaceType", which described the location of the complaint. There were 50 different spaceTypes in our data, with the two most common spaceTypes being Entire Apartment and Building-Wide.

We then looked at the MajorCategory variable, which described the complaint type. In our data there were 16 distinct MajorCategory values for housing complaints, with heat/hot water comprising 34% of all complaints and taking an average of only 3 days to be resolved.

Our subsequent analysis centered around trends by zipcode. We created a heat map (Complaints Heat Map), using zip code ("Zip") and complaint id ("ComplaintID") counts to visually compare where the density of complaints were greatest. The shapes of the outlined areas were automatically generated based on zip code and the heat map density color was selected such that hot spots easily stand out. This created an easy-to-comprehend image of where complaints were concentrated across the city, with the darker regions indicating higher concentrations.

The second visualization (Complaints Bar Graph) was created from the same dataset using multiple variables. The bar graph visualization provided additional information regarding the type of issue in addition to the count of complaints by zip code. The integration of the zip code and ComplaintID variables, with the addition of a third "MajorCategory" variable, into one graphic allowed us to focus on trends and assisted us in creating clearer projections of building lifecycles.

The Complaints Bubble Chart visual was an attempt to determine if the number of complaints increased with the age of the building. The bubble chart did not reveal an increase of complaints correlated to an increase in the age of the building. However, it did indicate that the number of complaints was not decreasing but rather holding steady each year since the 311 system was established.

We felt that the three visualization strategies shown below were the most appropriate way to investigate the question: "What is the lapse in time from complaint origination to resolution?" Each visualization organized and compared the data in a manner that gave more insight into the location of the complaint concentrations, the reason for the complaints, and categorical causes of the complaint. Although the bubble visualizations did not show a steady growth in complaints as the age of the buildings progressed, it did reveal that the number of complaints did not decrease over time either.

2.1.4 Model Construction with three models What three models were selected for construction and why? What are the pro's and con's of each?

We began our model construction by narrowing down our features. Specifically, we implemented a backward-selection process where we began with all the features, and then based on a simple linear regression, dropped features one-by-one until we had a model with fewer features that had similar performance to that with all of them. We ended up using three variables: complaint code, year of complaint, and household income for the zipcode.

First, we ran a simple linear regression as our benchmark model. We were interested to see which kernel-based models performed better on our data, so we then ran decision tree and k-means clustering models.

0.0.8 2.2 Modeling using Machine Learning

2.2.1 Benchmark Model Our benchmark model was a simple linear regression model with the Code, MinorCategory, and All Household Income columns as the three variables.

2.2.2 Champion Model While we tested many models, our benchmark model ultimately performed better than anything else. This was disappointing to us since our benchmark, despite it being the best performing of the models we ran, was still a pretty terrible model for predicting closeTime.

0.0.9 2.3 Model Evaluation

2.3.1 Benchmark Model Our benchmark Linear Regression model with 3 predictors had an RMSE of 34.913799, a R^2 of 0.164723, and an adjusted R^2 of 0.164647. Those numbers were extracted from the summary after running the model on the entire dataset. When using a train/test split, the Apache Spark LinearRegression module failed to run an adjusted R^2 but returned a Root Mean Squared Error (RMSE) on the test data of 34.8297 and a R^2 of 0.184046, which is comparable if not slightly better than the performance of the simple linear regression model without using a test/train split.

2.3.2 Champion Model Interestingly, our Champion Model ended up being the simple Linear Regression model with Code, MinorCategory, and All Household Income as the three features. This surprised us because different complaints likely take longer to complete than others and certain locations might have more complaints than others, so we expected a clustering or kernel-based model to perform better on our data than a simple linear regression.

0.0.10 3. Results

We expected a kernel-based approach like Random Forest or Gradient Boosted Trees to perform better on our data (given the clustering we saw in our initial data exploration), but that ended up not being the case. I'm curious to hear feedback on what direction we could take this project and how to move forward to improve the model and predictions for our regression.

0.0.11 4. Conclusion

Different Models Our biggest hope for the future of this project would be to work with models that are better suited for the clustered data we were working with. Partially, we felt bound to

the limitations of PySpark's available Machine Learning algorithms for working with DataFrames. The available Regression models built out in the Apache Spark Documentation are not as robust as those available through NLTK and in an effort to use the tools available to us, we ultimately settled for models that did not perform as well as we would have liked to see.

More granular is better One of the top future tasks for our analysis would be to focus on address-level data rather than zip-code level data. We think we could have much more success looking at income/rent on the address-level, alongside Building Age and specific location (distance to subways, schools) as a way to capture more variability between observations and improve our model. For example, looking at observations in Rockaway (which is near the beach, but also home to housing projects). We had no way of differentiating between different observations within this neighborhood. We had the data to have address-level data but we chose to use zipcode level.

Synchronization with job posting app (TaskRabbit, Angie's List, HomeAdvisor, etc.) Although we didn't go this route, there are useful applications to this information that would be a wonderful topic for future exploration.

OpenDataNYC is great but the files are uploaded online only once every 30 days. When these reports are filed to 311, they should be analyzed immediately to determine their category (are they electrical issues? Plumbing issues? etc.) and then a file or case should immediately be created for that housing complaint problem. In an ideal world, that housing complaint problem would be pushed out to a network of users (contractors, plumbers, electricians, etc.) who can then bid on the project and get it done in a timely fashion. There is no reason for people to wait 30+ days for the city to make sure that their heat is working.

What's great about this idea is that it supersedes the landlord and no longer makes the tenant reliant on their landlord to fix all of their problems. It might make sense to have landlords create accounts on these job posting sites where their profiles specify a range they are willing to spend for certain things. Some landlords have more disposable income than others and some are more inclined to put money towards giving their tenants a good living experience than others. That's fine. Having them set bounds for what they are able to spend makes it such that the contractors can bid on the project when it is posted.

Blight is a terrible thing and this problem is solveable. There is no reason for NYC to look so run down.

Streaming Twitter Data Since the OpenDataNYC information is only available monthly, pulling in the 311 information every hour would be a great improvement. In parallel, streaming Twitter data might be a nice way to capture some frustrated civilians who don't know to call 311. Streaming Twitter data would also be helpful to see which locations are having certain problems (maybe BedStuy has a power outage) that can then be used to notify the relevant people (media, ConEd Electrical, etc.). Especially if this idea were to scale outside of just NYC, the ability to categorize and manage data in real time sounds like a problem well-suited for streaming.