Part 3: Finalizing Analysis and Preparing Model

Ion Barbus

The third phase of the project began with some preparation of a supplemental dataset that includes dates and descriptions of special events permits granted in Baltimore. The location format in the dataset was mixed, containing both coordinates with notes, and street addresses. Using regex, I collected all the street addresses and by utilizing Geopy I was able to retrieve coordinates for the data. The coordinates already used in the dataset were also expunged of additional notes, such as ‘(closest intersection)’ and prepared for use with Geopoints and Geopandas. I also created datetime objects out of the date strings to prepare the dataset to be used for time forecasting. Events were listed with a start and end date, as well as number of days the event would go on for. In these instances, I replicated the rows and changed the start date so that an event that has gone on for ten days, would have ten instances and be accounted on each day the event moves forward. After this data was prepared, I conducted some initial analysis, plotting the events on a map, along with crime cameras to see how the two correlates.

As part of my project proposal, I was very interested in seeing how each neighborhood is different from one another so that we can gain insight based on individual communities and propose solutions for them as well. As such, the first modeling tool I used was DBSCAN with the intent to find crime hotspots in the city. DBSCAN, which stands for Density-Based spatial clustering of applications with noise, is a relatively common data clustering algorithm in data mining and machine learning. DBSCAN groups together points close to each other based on a metric and can be used to separate high density areas from lower density areas. This should give us certain hotspots for whatever we’re looking at. The size of the dataset was taxing to my system, as such a sample of the data was used when looking at all crimes. The algorithm used was Ball Tree as it provides the function to calculate Euclidian distance between neighbors, as well as haversine distance. Haversine distance is ideal to use when dealing with coordinates, such as in this dataset, as it factors in the circle distance, such as seen on the spherical earth. The initial search yielded 11 hotspots but encompassed the entirety of Baltimore. When the data was broken down into subsections for shootings, rape I was able to get interesting hot spot clusters that resembled the Baltimore neighborhoods.

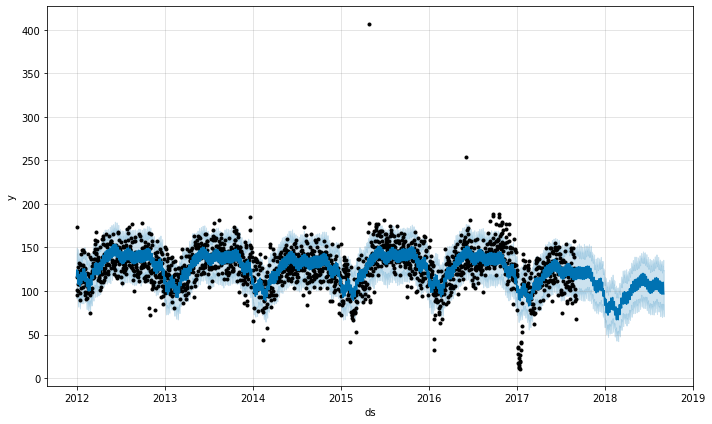
Running the DBSCAN model for Baltimore shootings has yielded 43 clusters that I was then able to plot onto a map of Baltimore. The same model yielded 51 clusters for rape, with a major cluster seen around Patterson park and the Mt Vernon as well as other regions. This data could potentially be used to adjust policing practices in certain regions of Baltimore, or to change how and which social resources are expended in those areas. The results from the model can also be used to focus special attention to certain neighborhoods and adjust my crime forecasting to fit the special characteristics of each area.

In preparation for a time series forecasting model I have also performed some time series analysis on the data to better understand how to alter the models. First, I indexed the dataset by the Date Time objects and aggregated all the instances of an event by day, week and month. Right away I notice some trends emerging that could be investigated. I also noticed a sharp outlier with a spike in crime April 27, 2015. A quick google search shows us that this was the first day when the infamous Baltimore riots turned violent. There is also a steep decline in January of 2017 that is unexplained, google searches only yielded an article confirming the drop off according to FBI data, which is based on the Baltimore city data as well. Next I wanted to see if there was any initial correlation between the events dataset I have and the crime dataset. Visually there seems to be some correlation with a spike in May on both datasets, but this needs to be confirmed. Finally, I began my initial time series modeling forecasting using Facebook’s Prophet.

According to the Facebook GitHub account, Facebook’s Prophet is an open source procedure for forecasting time series data based on an additive model where non-linear trends are fit with yearly, weekly, and daily seasonality, plus holiday effects. It handles outliers well and can be easily tuned. The initial model used takes only the date object and the y variable. Using these variables, I was able to make a future data frame for the next year and predict what crime occurrences will look like. Prophet returns the estimate, and the lower and upper bounds estimates as seen below:

ds yhat yhat\_lower yhat\_upper

2018-08-29 104.280489 78.613682 128.743013

2018-08-30 102.961917 78.495364 129.230602 …

I will tune the model to include several regressors such as holidays, events as seen in the special permit dataset, and seasonality. The quick forecasting model also has yielded some surprising results in showing reduced forecasting for crime in the future. While looking at the analysis and the model there seems to be a drastic drop off in crimes towards the end of 2017. I will investigate this to see if its due to unreliable data or missing data.