CRISP-DM: NYPD SQF Program, 2012.

LOGAN BICKERSTAFFE | 000373798

August 23rd, 2020

Table of Contents

Executive Summary of the SQF Program

Introduced in New York City in 2004, the Stop, Question and Frisk program vas developed with the intent of temporarily detaining, questioning, and frisking individuals on the streets of the city. The police department is in search of contraband of any kind, as well as any weaponry which present obvious dangers to public safety. The scale of this program is very large, as is evident from a single years worth of data comprising over 500,000 records.

Through the data mining process undergone in this series of reports, some biases and shortcomings have been uncovered, as well as some remarkably interesting insights into the nature of the data and program are detailed. Densities of stops are plotted, relationships between features are explored, and how predictions can be made from modeling the data are executed.

Some groups are extremely overrepresented, and others are under represented in the data, specifically Black males are targeted at an exceedingly high rate compared to any other group. This speaks on not only this data set, but also a larger issue in society today, which are preconceived notions of who commits crime, and this data set serves as evidence of systemic racism in this program.

REPORT 1:

Business Understanding

The purpose of the program was to determine the rate of individuals carrying weaponry or contraband through random stop, question, and frisks in New York City. Officers would stop individuals at their own discretion, citing their reasons for suspicion. The program was deployed across the 5 boroughs of New York City, with specific small geographic "hot spots" being targeted. This choice was made in an effort to control crime in areas which are harder hit by poverty, the war on drugs, and other socioeconomic factors. Insight into what areas of the city are most dangerous, what types and quantities of contraband are commonly found, the rates of arrests made throughout all stops, and other important statistics will become available throughout the data mining process. Being able to highlight the areas where illegal activities flourish can help the police department keep the citizens of New York City safe, and aid individuals in need.

To measure the success and effectiveness of this program, a formal review of the statistics and non-biased information must be conducted, where a critical analysis of the data will produce vital insights. Quantitative measures like rate of arrests / stop, rate of arrests made by race, focusing on number of stops and the rate of contraband/weaponry are all useful statistics, as well as predicting whether an individual is armed based on other variables. Whether or not the program is unveiled to be highly biased or not, there is undoubtedly some valuable information that will be uncovered, like the areas of the city which are more dangerous, or what race of people are targeted the most, etc. Another measure for future consideration are changes in legislation to increase public safety, and how the data from this program may influence policy change. Is the force being used by officers warranted in these stops? Are there instances of police brutality? Who is most at risk of being physically harmed by the Police?

Data Understanding

The entire dataset is comprised of 532,911 records/rows, and 112 columns, 0 empty rows, and 106 empty cells. The data types of the columns in this data set are varied, with there being 58 Boolean types, 37 Categorical types, 11 Numerical types, and 5 Unsupported types. There are 98 descriptive type columns, and 14 numeric columns.

	Age	Year	Weight
Range	0-999	2012	0-999
Mean	28.769	2012	169.284
Variance	24.006	0	36.838
Count	532911	532911	532911

Table 1.1: The original data frame statistics for 3 pertinent columns

Table 1.1 offers some interesting insights into the nature of the original dataset, particularly in the Range of values included in these columns. The dataset covers SQF data from 2012, which is why the Year column appears as so, however the Age and Weight ranges from 0-999 are a clear red flag for the presence of errors in the dataset, which will be handled as part of the data preparation and cleaning.

The data set is checked for null values, duplicate values, and NaN values, and all of these errors are removed. To check the data set for null values, the isnull() function is used to return all Boolean values which are True for NaN (Not a Number) values. Checking for duplicate values is done in a similar fashion, using a duplicated() function. And lastly to deal with Null values, the drop.na() function is used.

Some alterations will be made to the dataset shortly, including generating some additional columns, such as using “datestop” and “timestop” creating a singular “datetime” column, and changing height from feet and inches into centimeters. There are certainly missing values and outliers in the dataset, as can be viewed through a simple df.describe(). The most significant and clear outliers can be seen in the Age and Weight columns. For instance, the lowest Age value recorded was 0, and the highest Weight value is 999 pounds. It is very unlikely that a newborn child or extremely morbidly obese person were stopped and frisked on the streets of New York, so they certainly could be mistakes, or the data was not available during the time of the SQF stop. An example could be someone being stopped without having valid ID on them, so the officer would have to guess, or ask the individual for these types of statistics. These could also be mistakes which will be dealt with during data preparation.

Using the df.describe() function, it is easy to examine the basic statistics of each column in the dataframe. The pertinent attributes that will be examined at this stage are: Age, Weight, and Height.

	Age (yrs)	Weight (lbs)	Height (cm)
Range	12-90	50-350	91.44-241.3
Mode	19	160	172.72
Mean	28.137	158.95	174.260
Median	27	170	175.26
Variance	11.709	28.753	7.980
Count	499670	499670	499670

Table 1.2: General statistics on 3 pertinent columns

Table 1.2 presents some important details surrounding limitations placed on the dataset throughout the production of these reports. The Age range has been limited to peoples between the ages

of 12 and 90, Weight has been restricted from 50 to 350 pounds, and no restrictions have been placed on Height. The Height column was generated by converting the “ht_feet” and “ht_inches” columns which were originally included in this data set.

To reduce the strain of processing the data, some columns have been removed from the data set. The columns which were chosen to be dropped were those which are derivative, such as “ht_feet” and “ht_inches” as these are no longer needed. To complete this process, the df.drop() function was used with a list of columns no longer needed, one of which is Year because as mentioned above, all data is from the same year, 2012, making this column redundant. After the desired filters have been applied, and null and duplicate values are removed, some visuals can be produced which can provide further insight into the business case. The dataframe at this point consists of 499,670 records, and 79 columns.

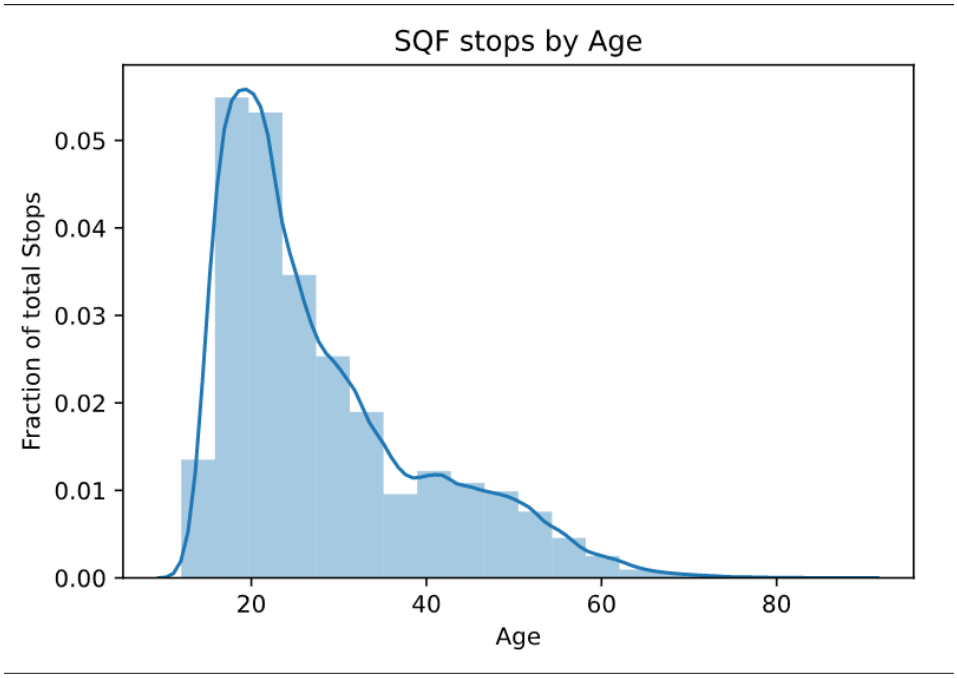

Figure 1.1: Distribution ages of stopped individuals

Figure 1.1 Shows the distribution of stops by age, highlighting that largest number of stops occur on young adults, roughly around the ages of 17-25. The rate at which older people - those above the ages of 40 - are stopped by police as a part of this program is much lower than young adults, and Elderly people - those above the age of 60 - are stopped at an even lower rate than them. This could be the first inclination of bias the data mining process has uncovered in this data set. Could younger adults perhaps be targeted by this program more often than older people, based on preconceived notions of who commits crime? Taking a deeper look into the data set may uncover these truths. The number of bins was set to 20 as the data set was fairly large, at over 490,000 records.

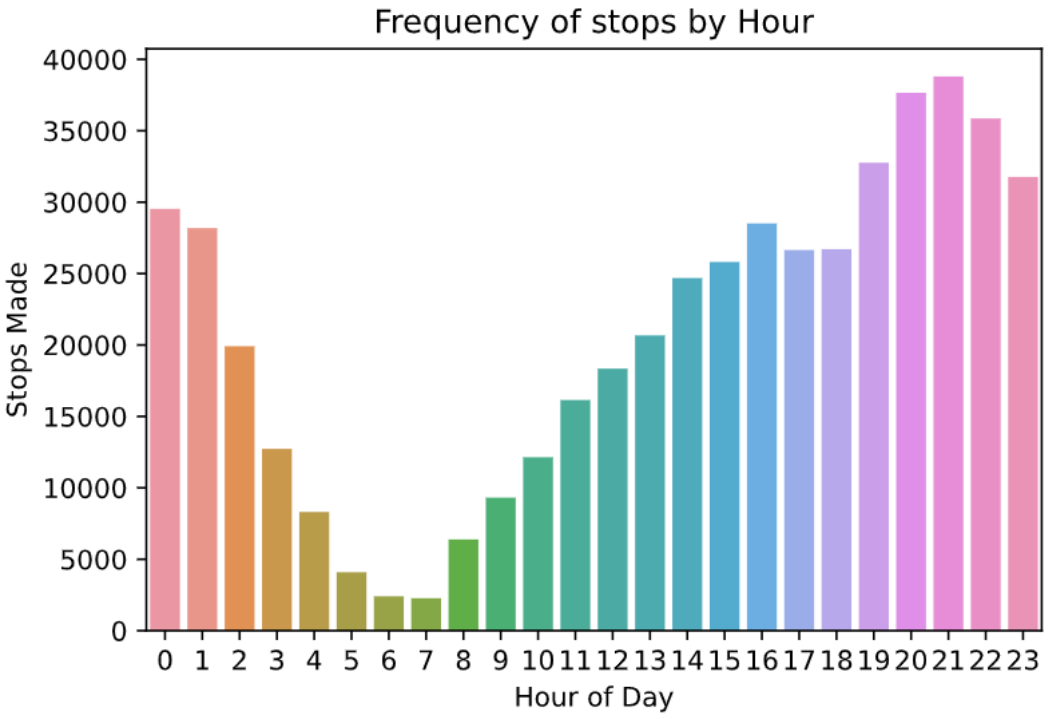

Figure 1.2: The count of stops made per hour of day

The information presented in Figure 1.2 is interesting as it shows the hours of the day when the majority of stops were made, and the hours where minimal stops were made. Stops were made at increasing rates during the evening hours, mostly after sunset, after the 19th hour (7pm). This could be an indication that the police force has increased activities during the evening hours, or alternatively that the majority of crimes are committed during the night hours. As is expected, the morning hours of 5-8 shows minimal activity as the city is waking up and people begin to work, and increases from here into the early afternoon, around the 16th hour (4pm). At this time it plateaus for a few hours, as we can expect to happen during dinnertime hours. The stops begin to increase again after sundown around the 19th hour (7pm) and reach there peak at 21st hour (9pm). The rate of stops remains high for the majority of the evening time.

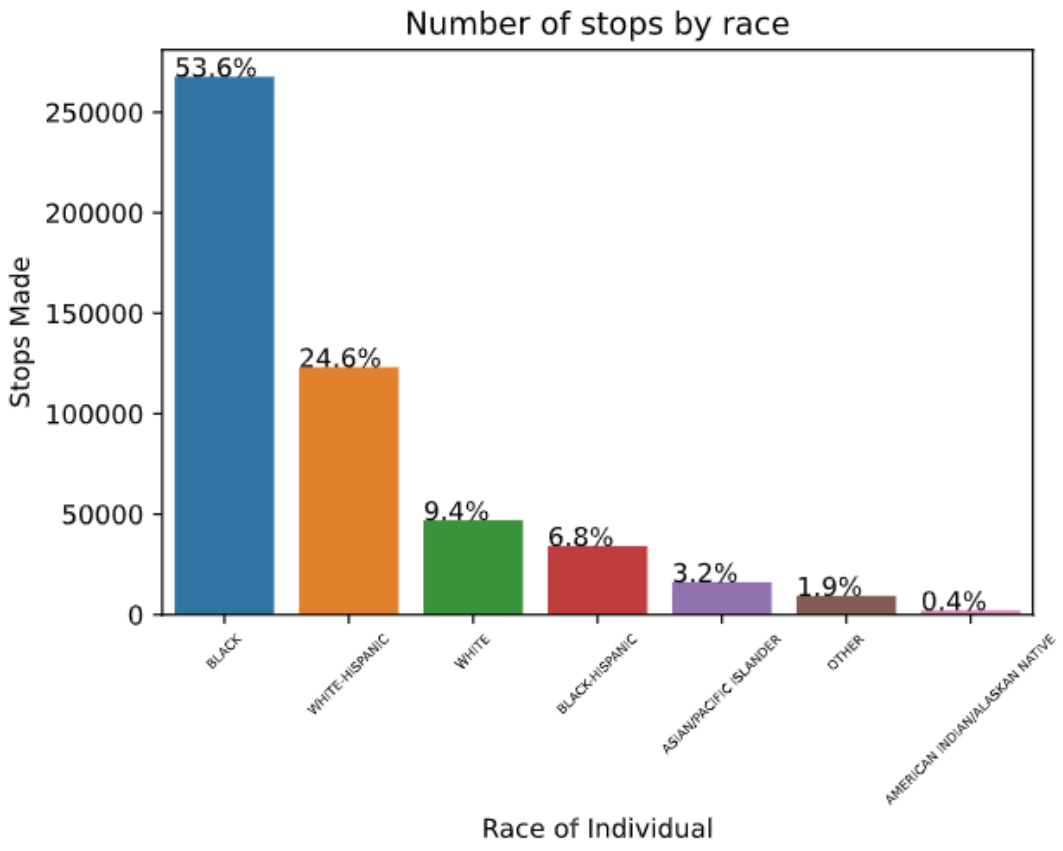

Figure 1.3: Stops by race

As evident from the generated bar chart of counts of stops made by race presented in Figure 1.3, the police department had stopped Black individuals at more than 2x the rate as any other group of people in 2012S. We know from Census data that roughly 24.1% of the population of New York City identifies as Black or African American, and because of this statistic it has become apparent through the data mining process that the program targeted Black individuals at a disproportionate rate compared to every other race of people. More Black individuals were stopped in the name of the SQF program than all other stops made on non-Black individuals combined. A glaring statistic such as this puts the programs legitimacy into question, however there are still important nuggets of information which are yet to be uncovered.

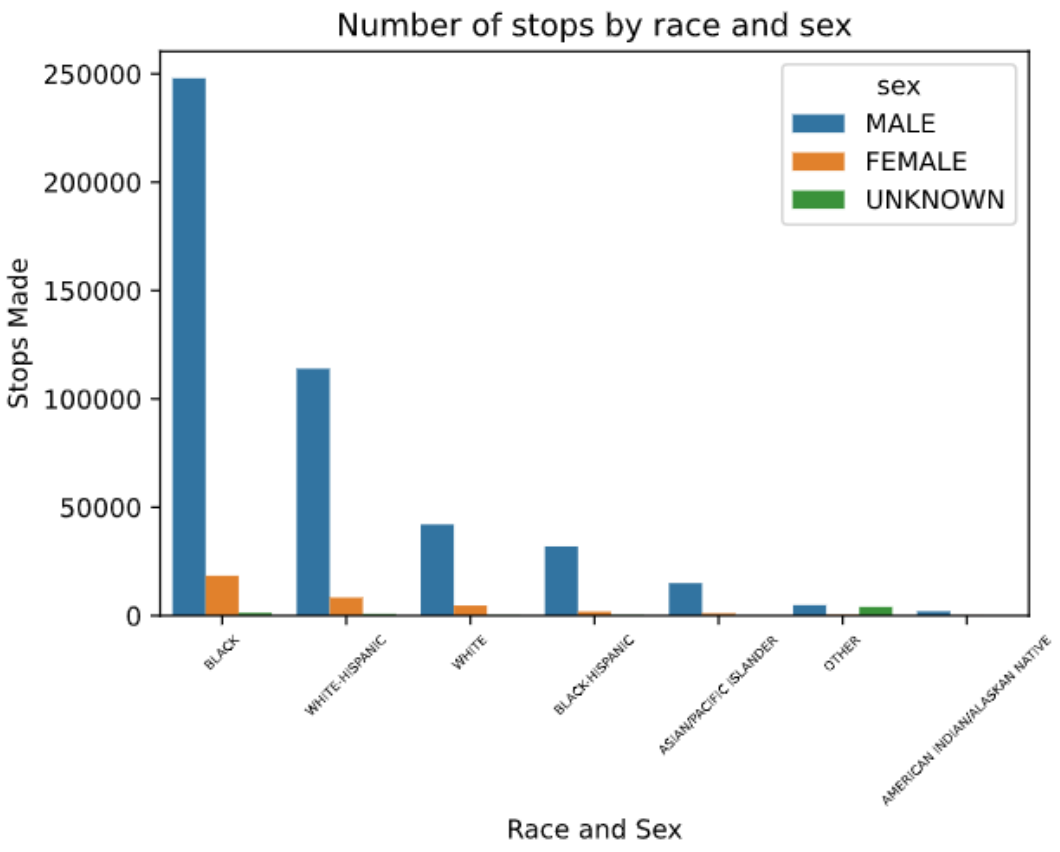

Figure 1.4: Count of stops by race and sex

This count plot shows the stark contrast between how many males and females were stopped in 2012, as a part of this program. It is evident from this data that males were stopped at a many times greater magnitude than females and that regardless of race, this is a uniform statistic. According to New York City Census data, roughly 52.3% of the population is female, which would make them severely underrepresented in this study. It is possible that this is due to reconceived notions of who commits crime on behalf of the police department, or simply due to the approved reasons for stopping and frisking an individual.

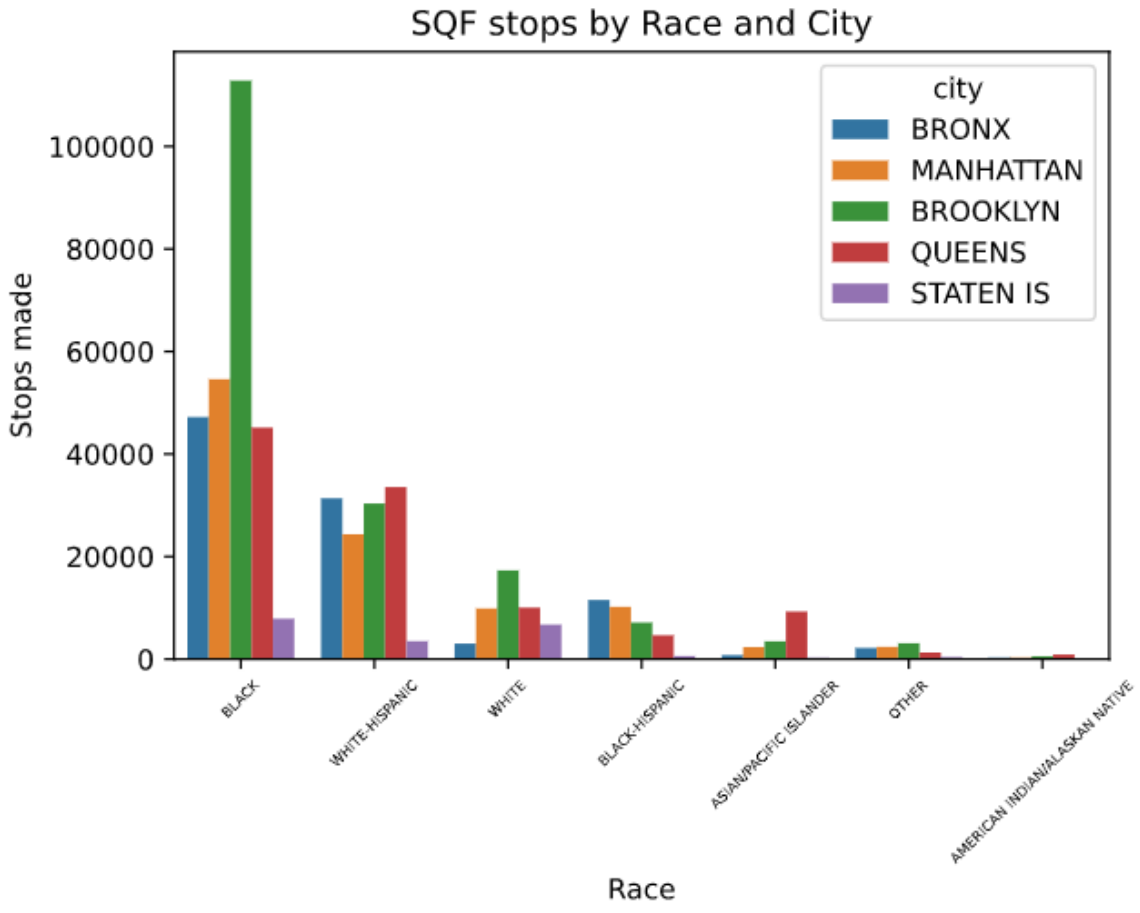

Figure 1.5: SQF stops by Race and City

Further mining of the data provides insights into SQF stops by borough, all of which share a common characteristic – more Black individuals are stopped in every single borough compared to other races. The only borough in which the statistics for counts of individuals stopped being close to even was in Staten Island, where Black people were stopped at only a slightly higher rate than White people. Figure 1.5 also shows that the greatest number of stops on Black individuals occurs in Brooklyn, where well over 100,000 stops were made. Other trends become evident through this figure, although they may have less overall bearing on gauging the effectiveness of the program, however they are interesting and valuable nonetheless. These include that the races excluding "Black" see an extremely varied amount of stops in each borough, none of which are constant. This is a clear indication of true randomness in the data set – no overwhelming or consistent statistics on a single group or place. The White race experiences an interestingly even distribution of stops made across the 5 Boroughs, as do both the Black-Hispanics and White-Hispanics. This specific characteristic of the data shows the fairness and equality of stops made across the 5 Boroughs; however this statistic is of much lesser importance.

To take a closer look at the geographic locations of where a specific crime is likely to occur, the "detailcm" column can be used to isolate a specific crime, cluster it, and map that across New York City, using the folium library. The df.value_counts(df["detailcm"]) command was used to explore the counts of each crime within the data set, which produced some statistics that are noteworthy. The maximum count of any crime belongs to Criminal Possession of a Weapon (CPW), at 122,125 instances, followed by Robbery at 109,772 instances, and Burglary at 56825 instances. The lowest instances are for obscure crimes and have a count of one, amongst these are things like Issue a false certificate, and killing/injuring a police dog. For the purposes of mapping, a crime with a relatively low crime count is chosen to limit the time required for processing. The crime from the "detailcm" column which will be mapped and examined was chosen to be Riot, which has 51 instances in the data set.

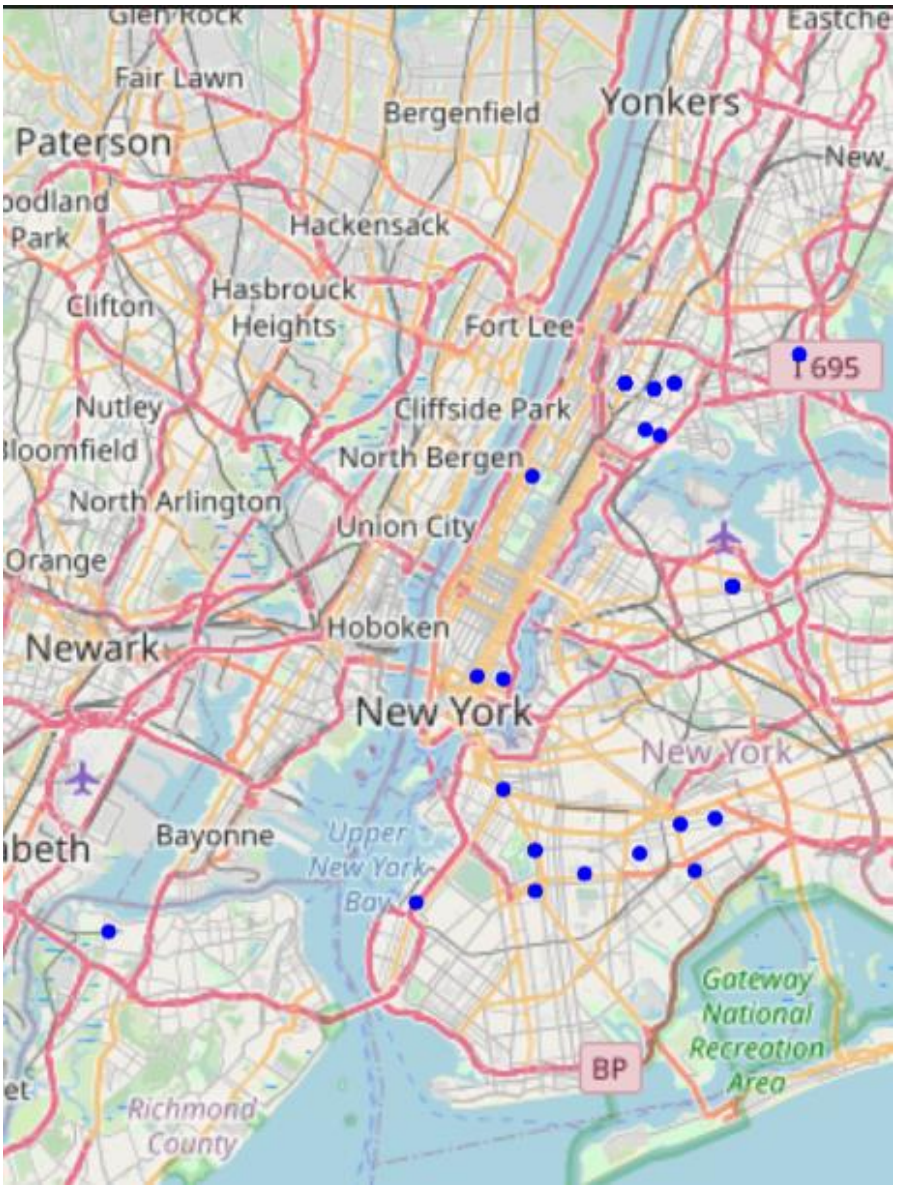

Figure 1.6: Geographical location of Riots in New York City

With the collected data points for instances of Riot plotted on a map, the greatest numbers and concentrations of Riot can be seen in Brooklyn and the Bronx. The 2 data points in southern Manhattan are in the financial district, and could have some relation to the financial protests common in the early 2010's – Occupy Wall Street and the 99% Movement. Adding another crime from the "detailcm" list adds additional complexity to the Map.

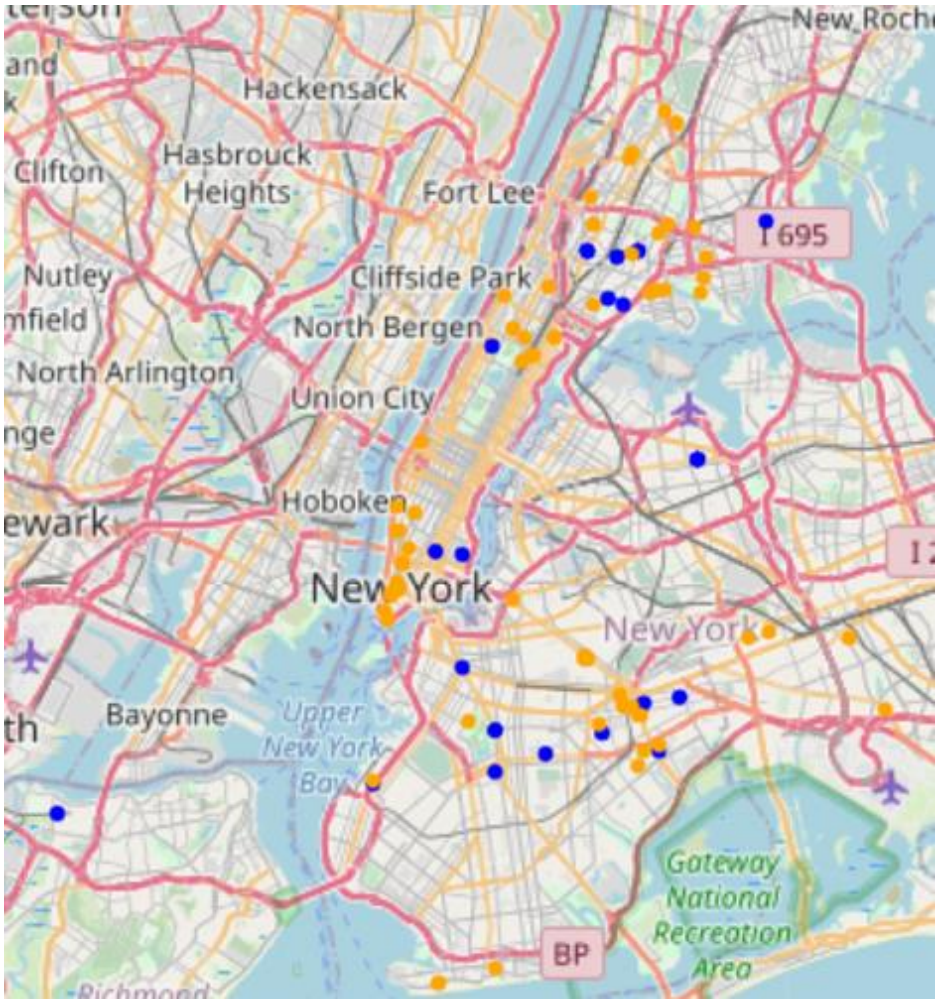

Figure 1.7: Geographical Map showing Instances of Riot and Loitering

Other important insights can be gathered from the data at this stage, including identifying which attributes have the strongest correlations. This is best achieved through a heatmap, which can be created in python by utilizing the 'seaborn' and 'matplotlib' python libraries. Figure 1.7 is the result of the heatmapping operation and shows the strength of the correlation between the columns. From the figure it is evident that the strongest relationship exists between Height and Weight, which logically makes sense, A taller person with a larger frame is more likely to weigh more than a shorter person with a smaller frame. On the opposite side of the same coin, a heavier individual is more likely to be tall, and have a frame, than they are to be short, and with a smaller frame. The second strongest relationship exists between Age and

Weight, which is a little less intuitive. One way to understand this feature of the data set is to think that a longer lifespan equals more time to accrue mass, however it Is more likely that the younger age ranges in the data set, or those younger than 28 generally have lower masses, especially those in their teenage years. As noted previously, the younger age ranges are stopped at a higher rate than older folk, and therefore represent a larger section of the data set.

Figure 1.6: Heatmap detailing the strength of correlation between attributes of the dataset

Another interesting feature to explore in the data mining process is the correlation between reason for frisking an individual, and the force used by the officer in that incident. This can be achieved by generating another heatmap plotting the physical force used by the officer, and the reason for the frisk. Figure 1.7 is the heatmap that shows the strength of this correlation and brings to light some interesting insights into the data. Firstly, it is clear that the strongest correlation lies between "rf_furt" (REASON FOR FRISK – FURTIVE MOVEMENTS) and "pf_hands"(PHYSICAL FORCE USED BT OFFICER - HANDS) which is expected and logical. Furtive movements is synonymous with fleeing or running, so an appropriate response by an officer would be to give chase and subdue the suspect using their hands.

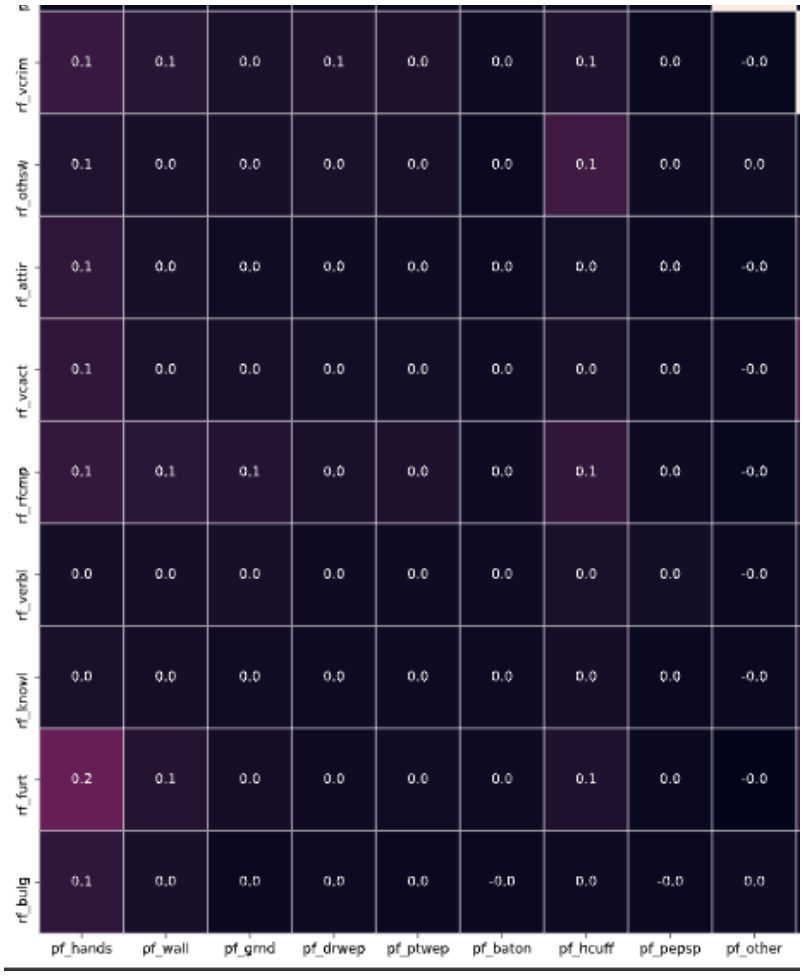

Figure 1.7: Heatmap of Physical Force used, and Reason for Frisk

Another interesting feature to consider is Reason for stop and Physical force used in response, which can be completed in the same fashion:

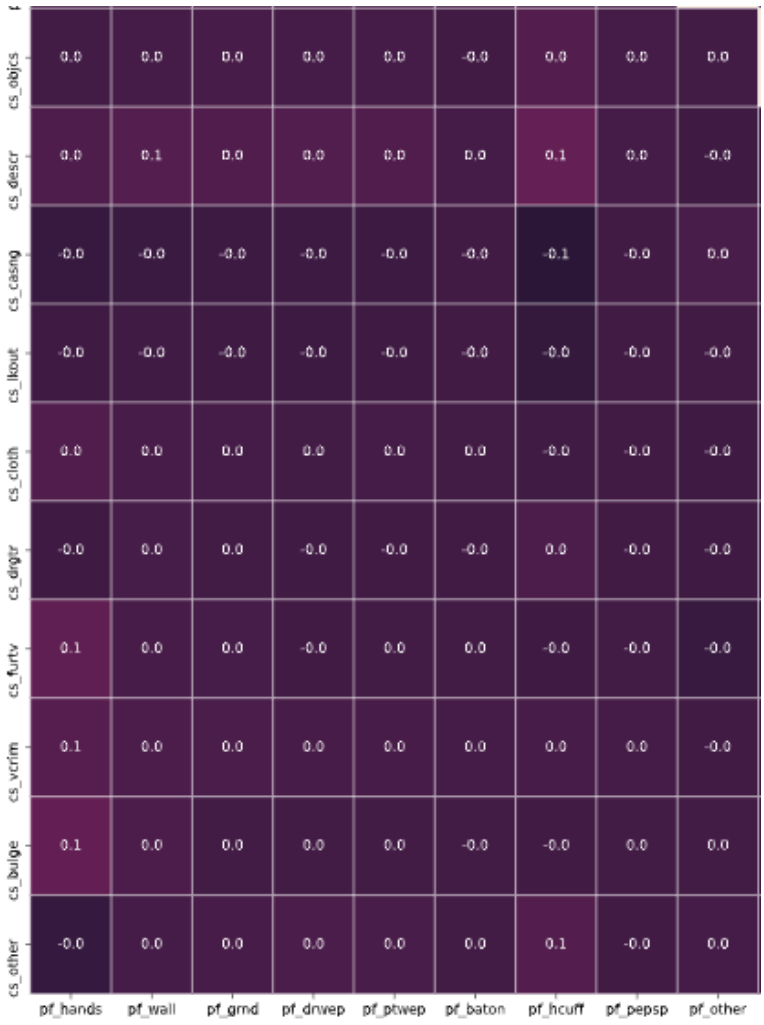

Figure 1.8: Heatmap identifying the relationships between Reason for Stop, and Physical Force used.

This heatmap presents a fairly even distribution of weakly correlated attributes in the dataset. The strength of the relationships are modest at best, however it does show that the "pf_hands" is the most common physical force to be associated with any other column. This also is as expected, seeing as how common the use of hands would be in these SQF scenarios. If we apply the same methodology to all of the

columns, that is every column with the prefix of “pf_”, “cs_”, or “rf_”, or Physical Force, Reason for Stop, and Reason for Frisk, the heatmap tells a much more interesting story.

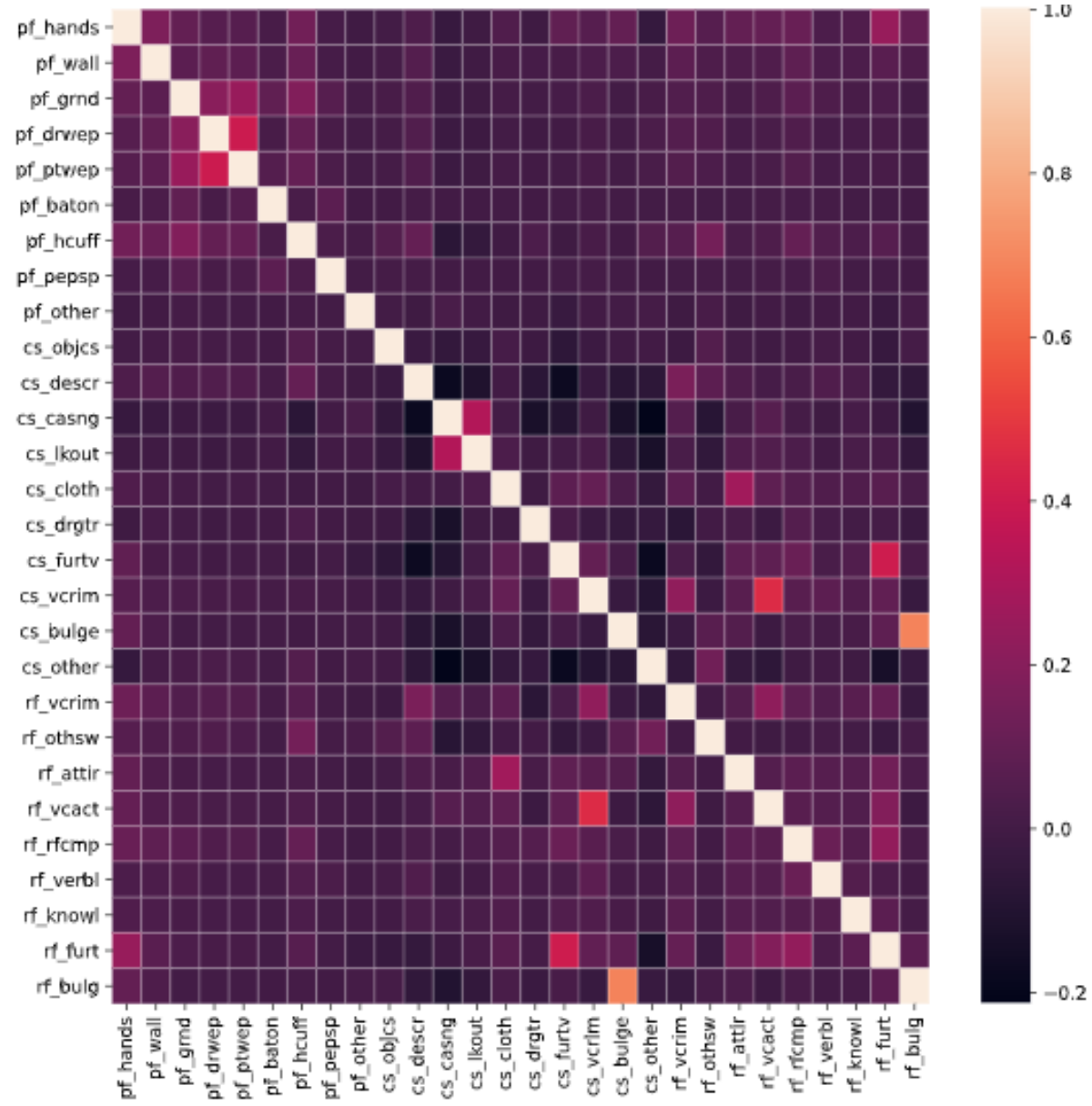

Figure 1.9: Heatmap plotting the correlation strength between Reason for Stop, Reason for Frisk, and Physical Force used.

Some noteworthy relationships are identified within this heatmap, mainly that items that have a pair within he data set – such as “cs_furt” and “rf_furt” or Reason for Stop: Furtive Movements, and Reason for Frisk: Furtive Movements -which means that suspects who fled were likely to be stopped and frisked for that reason – are correlated to one another. There are numerous of these scenarios which occur in the data, such as instances of drawing/pointing a weapon are strongly correlated, because they are such similar actions, or in the case of “rf_vcact” and “cs_vcrim” which both relate to engaging in a violent crime. We also see interesting relationships within the physical forces used that also follow logic explained above, like in instances where handcuffs are used, suspects are commonly put on the ground. With the data presented in this manner, it is not only more visually appealing, but allows the viewer to see the varying strengths of relationships across the data set immediately. This format is very intuitive and useful for understanding the nature of the data, and how the columns and attributes of the individuals are inter-related, and share relationships with one another via the colour gradient feature on the right hand side of Figure 1.9. The solid off-white diagonal ‘line’ through the chart is the meeting point of each column or attribute with itself, having the strongest possible positive relationship value of 1.0.

REPORT 2:

Data Preparation

To begin the Data Preparation step of the CRISP-DM methodology, frequent item sets will be generated from the data set which will determine the strength of correlations between certain attributes within the data set. To handle this step in the CRISP-DM and being to model the data, it needs to be put into a transactional format. To accomplish this, a series of Yes/No values will be converted into a Boolean dataframe, in this case, all instances where a suspect had a weapon or contraband, and where there was a physical force used. The columns for all of the weapons, knives, and contraband will be compiled into a dataframe with all instances of Physical force, called "x". This 'x' dataframe is the combination of columns with a Y value for and "pf_", and a Y for any weapon or contraband. Next, a column named "Armed" is generated, which is populated by the instances in the original dataframe where an individual had a weapon or contraband on their person. To prepare the frequent item sets, the chosen attributes: "race", "age", "city", "build", "sex", "eyecolor" have been passed through one – hot encoder, which will convert their values to binary values in preparation for modelling. It is important to note that all the attributes that are passed through the encoder are originally categorical data, which is unusable by the model.

Modelling

To model the data set the Apriori algorithm will be used which will identify frequent item sets in the database and will extend these to larger and larger sets of items. With the frequent item sets determined, association rules can be used to highlight the trends within the data. First, a few terms need to be covered which are relevant to this technique, 'Support' is the ratio of instances for which a rule is true

out of all instances, 'Confidence' is the ratio of instances for which the rule is true out of the number of instances where the antecedent is true. Minimum support values are provided to the Apriori algorithm, and by specifying this minimum support boundary, the data will be pruned. The minimum support value provided to the algorithm was 0.015, which provided the results of Figure 2.1.

	support	itemsets
0	0.133888	(pf_hands)
1	0.023952	(pf_wall)
2	0.036210	(pf_hcuff)
3	0.017516	(contrabn)
4	0.028233	(armed)
...	...	...
628	0.015902	(sex_MALE, build_THIN, eyecolor_BROWN, city_BR...
629	0.028407	(build_MEDIUM, sex_MALE, eyecolor_BROWN, city_...
630	0.023608	(city_MANHATTAN, build_MEDIUM, sex_MALE, eyeco...
631	0.031189	(build_MEDIUM, sex_MALE, eyecolor_BROWN, city_...
632	0.041531	(build_MEDIUM, sex_MALE, eyecolor_BROWN, city_...

633 rows × 2 columns

Figure 2.1: Description of the frequent itemsets to be used in the Apriori Alglorithm

Inspecting the frequent itemsets generated within the dataset, it is evident that the most common or frequent element within the data set is Physical Force Hands, follow by Wall, and Handcuff. As would be expected, itemsets that include all 6 variables are much less common, as is reflected in the support values for the itemsets. As mentioned, the minimum support value is specified to be 0.015 therefore any of the itemsets which have a support value below this will not be included in the algorithm. This action is the first step of Association Rule Mining, where a minimum support threshold is applied to find all frequent item sets within the data set. Figure 2.1 is the result of completing the first step in this process. To finish

Association Rule Mining, a minimum threshold of confidence will be applied to all frequent itemsets, to form rules on the data set. Figure 2.2 is the resulting chart after applying a minimum confidence threshold of 0.7 to the frequent item sets, sorted by lift. 'Lift' is the measure of the performance of the association rule mining at predicting or classifying cases, and can de defined as target response divided by average response. Lift values that are greater than 1 indicate that the variables involved – the antecedents and consequents – are dependent on one another. Figure 2.2 shows the breakdown of lift values for the frequent itemsets in the data.

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift
116	(contrabn)	(armed, eyecolor_BROWN)	0.017516	0.024938	0.015558	0.888254	35.617842
112	(contrabn, sex_MALE)	(armed)	0.015927	0.028233	0.015927	1.000000	35.420004
6	(contrabn)	(armed)	0.017516	0.028233	0.017516	1.000000	35.420004
115	(eyecolor_BROWN, contrabn)	(armed)	0.015558	0.028233	0.015558	1.000000	35.420004
113	(contrabn)	(armed, sex_MALE)	0.017516	0.026151	0.015927	0.909278	34.769946
...	...	...	...	...	...	...	...
186	(race_WHITE, city_BROOKLYN)	(eyecolor_BROWN)	0.034625	0.902167	0.025965	0.749899	0.831219
190	(race_WHITE, sex_MALE)	(eyecolor_BROWN)	0.084214	0.902167	0.063118	0.749495	0.830772
16	(race_WHITE)	(eyecolor_BROWN)	0.094064	0.902167	0.069950	0.743644	0.824286
532	(race_WHITE, build_THIN, sex_MALE)	(eyecolor_BROWN)	0.025915	0.902167	0.018880	0.728550	0.807556
181	(race_WHITE, build_THIN)	(eyecolor_BROWN)	0.030462	0.902167	0.021902	0.719007	0.796977

Figure 2.2: A table sorted by lift, showing the top 10 records.

A few interesting details are presented from the data mining process here – an Antecedent of Contraband is most strongly linked to the consequents of being Armed, and having Brown eyes. We see similarly high ratios of lift in the top 5 records, and as expected, much smaller ratios in the bottom 5 records. The next step in the modeling process for the purposes of this report is to limit the antecedents and consequents to include only data from the "armed" column, mentioned in the Data Preparation section. Figure 2.3 is the result of performing these operations on the data and is sorted by confidence in descending order.

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift
6	(contrabn)	(armed)	0.017516	0.028233	0.017516	1.000000	35.420004
112	(contrabn, sex_MALE)	(armed)	0.015927	0.028233	0.015927	1.000000	35.420004
115	(eyecolor_BROWN, contrabn)	(armed)	0.015558	0.028233	0.015558	1.000000	35.420004
9	(armed)	(sex_MALE)	0.028233	0.916213	0.026151	0.926278	1.010985
117	(armed, eyecolor_BROWN)	(sex_MALE)	0.024938	0.916213	0.023087	0.925768	1.010430
113	(contrabn)	(armed, sex_MALE)	0.017516	0.026151	0.015927	0.909278	34.769946
111	(armed, contrabn)	(sex_MALE)	0.017516	0.916213	0.015927	0.909278	0.992431
114	(armed, contrabn)	(eyecolor_BROWN)	0.017516	0.902167	0.015558	0.888254	0.984578
116	(contrabn)	(armed, eyecolor_BROWN)	0.017516	0.024938	0.015558	0.888254	35.617842
10	(armed)	(eyecolor_BROWN)	0.028233	0.902167	0.024938	0.883320	0.979109
118	(armed, sex_MALE)	(eyecolor_BROWN)	0.026151	0.902167	0.023087	0.882835	0.978571
119	(armed)	(eyecolor_BROWN, sex_MALE)	0.028233	0.827624	0.023087	0.817750	0.988069

Figure 2.3: Results of Association Rule mining of the data set, limiting antecedents/consequents to "armed". Sorted by Confidence in descending order.

Evaluation

The table in Figure 2.3 uncovers some interesting nuggets of information from the dataset, regarding the correlations between factors and attributes of individual who were stopped in the name of this SQF Program. There are 3 records in this chart with a confidence of 1, the highest confidence can be, This means that through the modeling process, the Apriori algorithm has 100% certainty that these antecedents and consequents are dependent on each other. The algorithm formulates this table based on if-then scenarios, therefore the logic of the table can be explained in the same manner – Line 9 for example: if a person is armed, then they are likely to be a male. For records in the chart with more than one antecedent/consequent, then the argument becomes if-and-if and then, or if and then-and-then. Plotting support vs confidence will provide some deeper insights into the nature of the data and serves as a nice supplemental visual to the tables provided above.

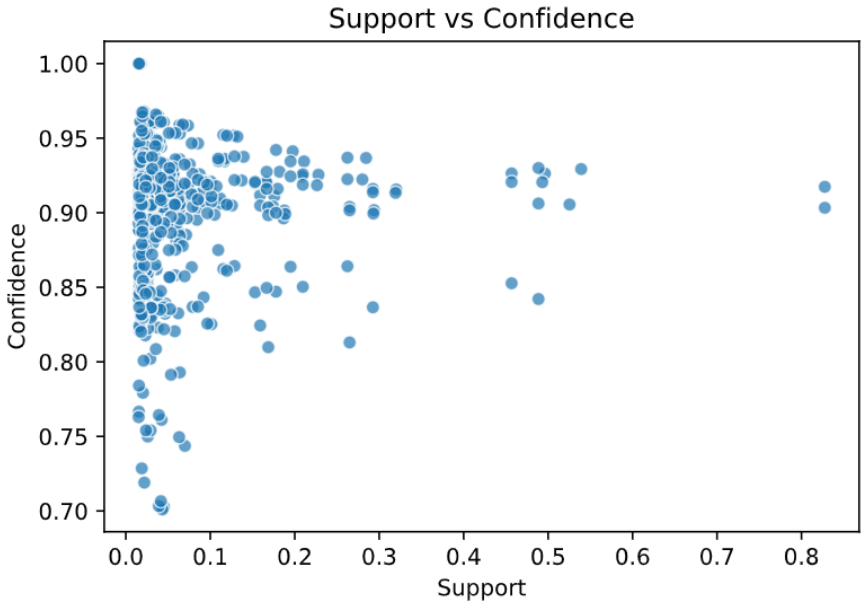

Figure 2.4: Scatterplot of Support vs Confidence

This chart visually represents the tables of data which appear above, however it becomes apparent that through the data mining process, the model has high confidence in relationships which have low support. When this occurs, it essentially means that the model is confident even though there are a relatively small percentage of records in the dataset which meet its criteria. More ideal data points to analyse further would be those with high support and confidence values, such as the two points plotted furthest on the right side of the chart. These 2 points have confidence values > 0.9, and support values > 0.8, making them reliable data points. This chart is generated from the columns of the dataset which were reformatted into a Boolean matrix, or transactional form which only includes 6 different attributes, race, age, city, sex, build, and eye color, not the entirety of the original data set. The scatterplot is denser in the confidence range of 0.85-0.95, and between the support values of 0.0-0.2, and gets drastically less dense as support values increase.

REPORT 3:

Data Preparation

Class variables have been prepared and defined earlier in Report 1, however a brief overview of what operations were done to the dataset, and what the final dataset looks like will be given here in this section.

The data was checked for nulls and duplicate values, all of which were removed from the data set. Next, the 2 original columns for date and time were combined to one datetime column. After this, ranges/filters were imposed on the acceptable age and weight values, eliminating erroneous data. After carefully analysing the original dataset, a series of columns were dropped from the dataset to reduce strain on local machines. Any column which was derivative or was used to generate another column, as well as any column which does not contain relevant data has been dropped from the data set. Figure 3.1 is a list of all columns which were dropped. After these operations the dimension of the dataset are 499,670 records, and 79 columns.

Figure 3.1: A list from VSCode of every dropped column from the original dataframe.

Modelling

Diving deeper into the dataset, the specific geographic zones where clusters crimes occur can be achieved by performing cluster analysis on the dataframe. Chosen from the "detailcm" list, the crime which will be clustered for the purposes of this report is "Creating a Hazard". This is a misdemeanor crime in the State of New York, which a person is guilty of if they dispose of a

container which a child could lock themselves in accidentally, or if they own property with an abandoned well or cesspool in which a child could harm themselves. Hierarchical clustering will be applied to the dataframe to move forward with the plotting process, where the optimal number of clusters will be determined through defining the silhouette scores for the dataset. In Figure 3.2, the silhouette scores are plotted on the y axis, with k values on the x axis, which will help to determine the optimal number of clusters for use in Hierarchical Clustering. Optimal k value for this dataset is 5, as this is where the highest peak in silhouette score exists.

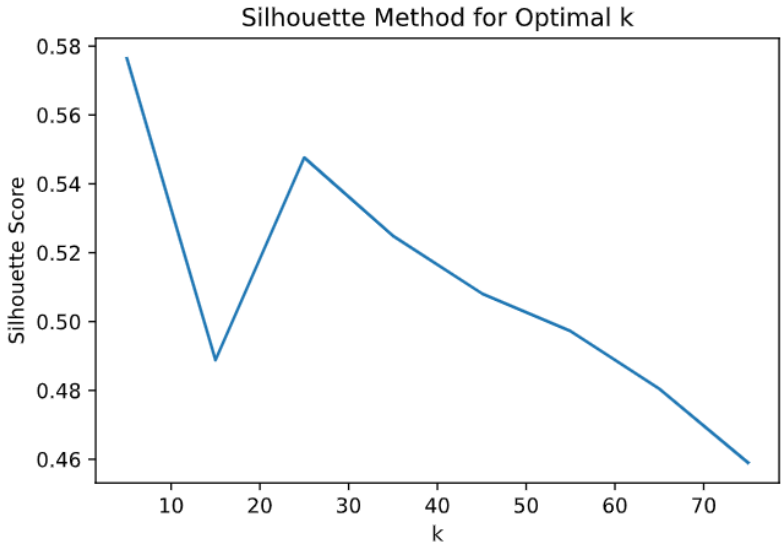

Figure 3.2: Line chart plotted to determine optimal k

Moving forward with this dataframe and applying the hierarchical clustering to a map using a combination of folium and seaborn libraries the clusters were plotted onto a map of New York City in Figure 3.3. This figure provides a geographical display of the where the clustered

groups occur most. To accomplish this, the data from the "coord" column will be split into "lat" and "lon" to accurately cluster on the map. Hierarchical clustering uses the optimal k value for clusters as determined through the line chart plotted in Figure 3.2. Through this process, the optimal cluster to use was determined to be 5 – as outlined in figure 3.3, where only 5 different clusters (colors) are plotted onto the map. Through hierarchical clustering, groups of common items are clustered together by color in this example. In the visual, it is apparent that this "creating a hazard" crime is most common in the Bronx and Brooklyn, and least common in Manhattan. There is almost equal distribution of clusters across Queens and Staten Island.

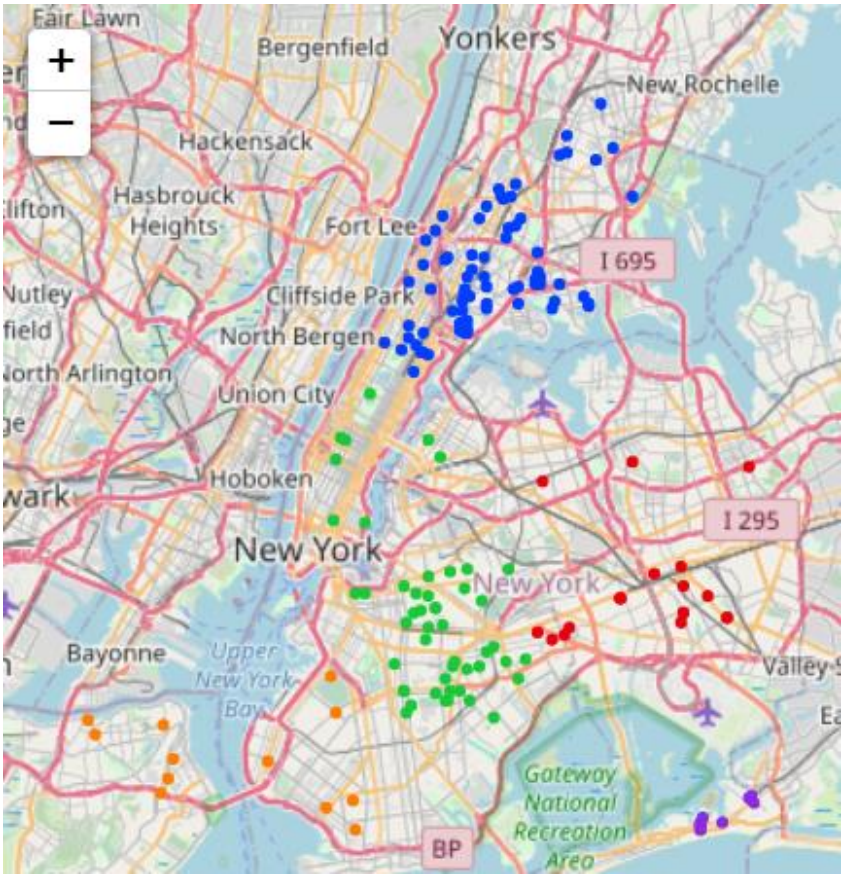

Figure 3.3: Clustered data presented on a map of NYC

There are other methods of clustering which will now be applied to the dataframe, in the hopes that further insights being uncovered about the nature of the dataset. First, DBScan will be applied to a dataframe, which is a density based algorithm, capable of grouping together all data points which are closely packed together, separating low and high density areas. Instead of applying the DB Scan algorithm to a specific crime, it will be applied to a dataframe containing every instance of stopping an individual ("cs_" columns). To do this, the columns with a "cs_" prefix are joined into a single dataframe called CSS. This dataframe will be used for the purposes of DB scan, and when visualized, should present some interesting facts about the density of these stops by reason for stop.

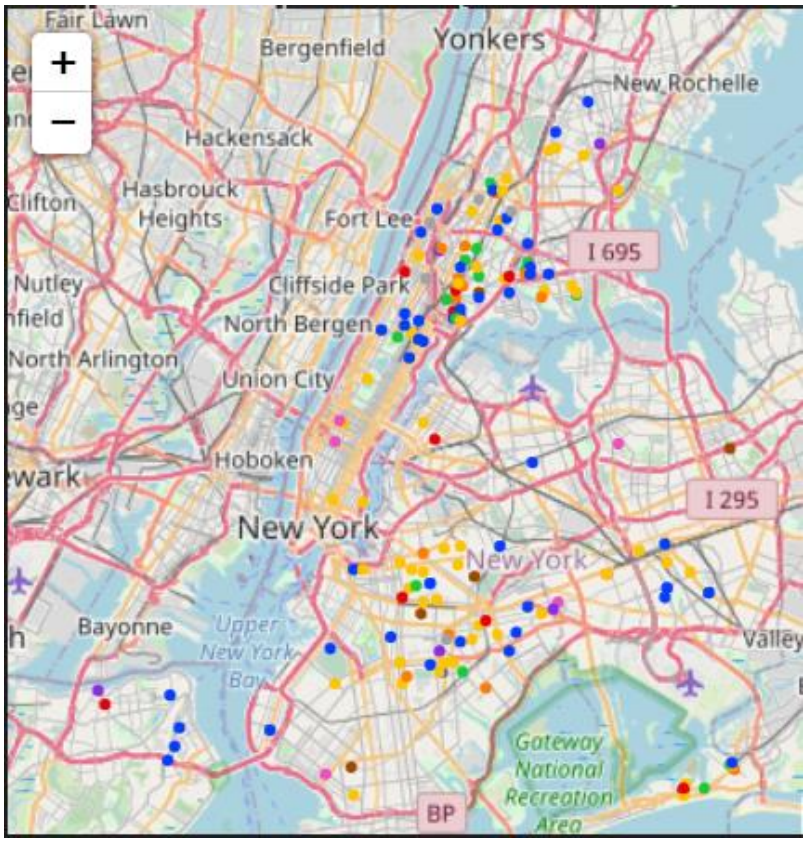

Figure 3.4: Visualized DB Scan on map of NYC

This figure includes 9 clusters or colors representing differing densities of stops across the city made for the SQF program. From this map it is apparent that the highest density of stops occurs in the Bronx and Brooklyn, followed by Queens, and then Manhattan and Staten Island. The colours of the dots represent differing densities of stops being made, so in areas where there are many dots nearby, it simply means that in these areas, there are differing levels of stops being made. In regions with no clusters, there is very low density of stops being made. Some of the clusters appear to be very near to one another in this map view, however if the zoom feature is utilized, we can see that they are separated in most cases by at least 1 City block, Figure 3.5.The Db Scan algorithm decides the optimal number of clusters to use which requires no input from the user. The algorithm provides a silhouette score of 0.537 to use and has grouped together similar data points into clusters of differing densities.

Figure 3.5: Spacing of DB scan clusters mapped on NYC.

The KMeans algorithm can also be applied to this dataframe, which is an iterative algorithm which partitions the data into pre-defined and non overlapping groups or clusters. The dataframe which will be used contains the CSS dataframe used earlier, as well as every instance of "detailcm"=="CREATING A HAZARD" in the dataset. KMeans requires an input parameter for the k value, and as mentioned earlier, the optimal clusters determined for the "Creating a hazard" crime is 5. The KMeans algorithm will be applied to a dataframe which contains every instance from the original data set where a person is stopped and charged with the crime of Creating a hazard. The 2 visuals provided, Figures 3.4 and 3.6 show roughly the same things, which of course is expected. The 2 algorithms used are clustering the same data into groups of differing densities, however the methods used by each algorithm is different, which is why there are some small visual differences between the maps. In Figure 3.6 the locations of the clusters are

the same as presented in Figure 3.4, however the coloration is different, which signifies the differences between the two algorithms used to map the data. Both algorithms essentially made the same conclusion and clustered them on the map, however the methodology of each algorithm is significantly different.

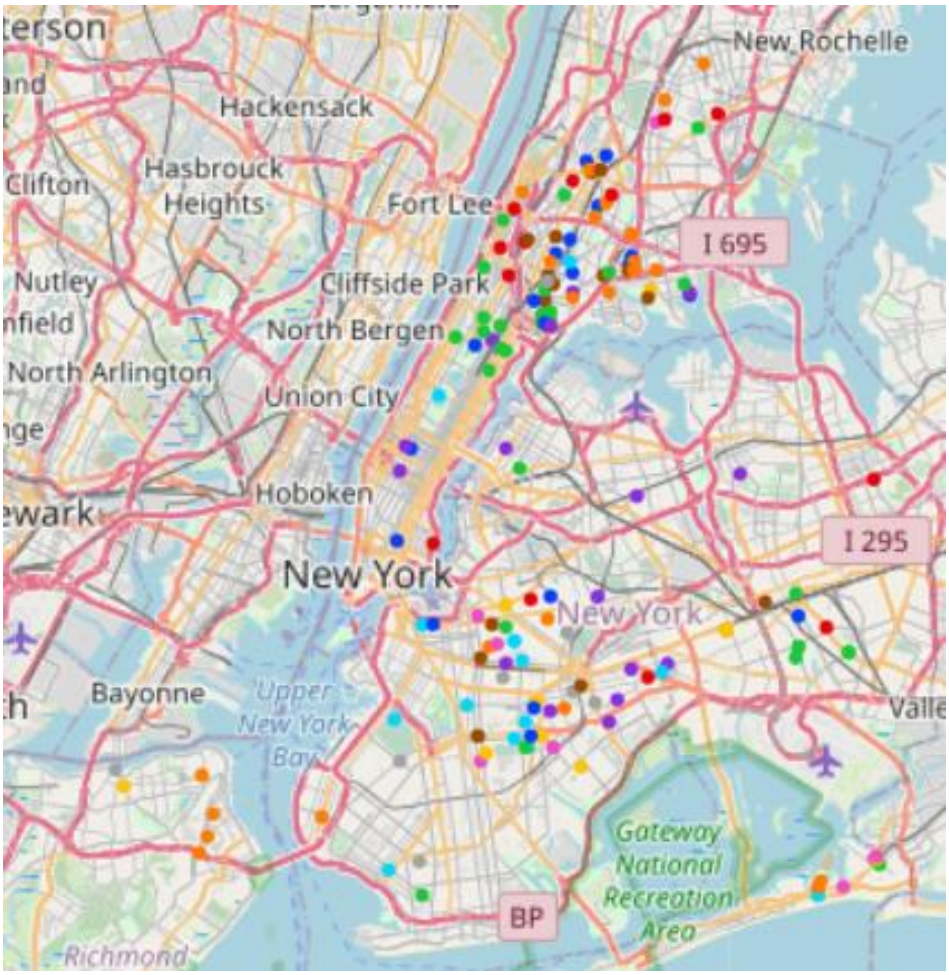

Figure 3.6: KMeans clustering of the x dataframe.

As mentioned, the KMeans algorithm requires an input parameter for k values, or number of clusters. To determine the optimal number of clusters to use for this operation,

another Silhouette Score line chart was created. Figure 3.7 shows the results of this operation, which is produced using the same method as before for hierarchical clustering.

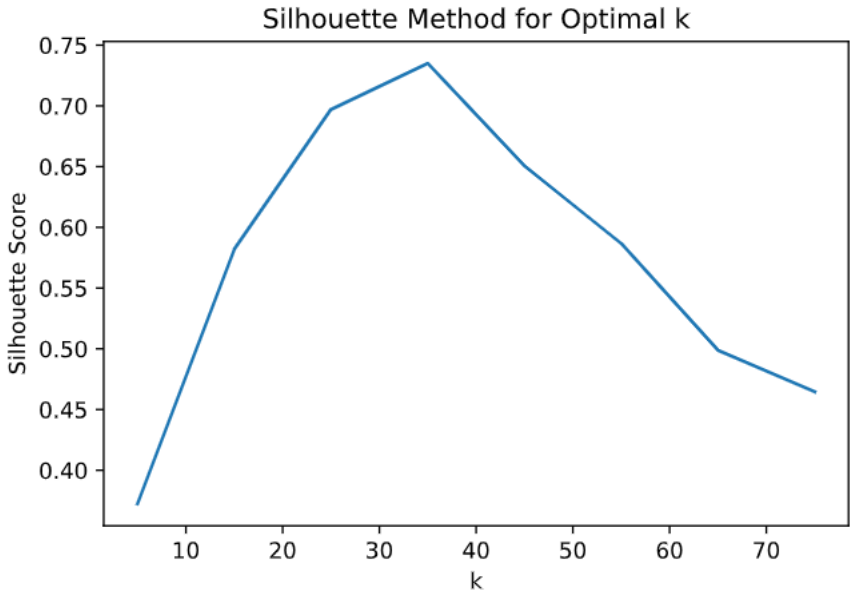

Figure 3.7: Optimal k line chart for KMeans method

The chart in Figure 3.7 does have some noticeable differences compare to the line chart for hierarchical clustering as expected. The optimal k value is determined to be 35, where the silhouette score is at its maximum value for this method, 0.74. The difference in number of clusters, or k, is the reason why the maps appear vary so much in color palette. Figure 3.3 was produced using an optimal k value of 5, represented by the 5 colors plotted on the map. Figure 3.6 was produced using an optimal k value of 35, which is why this map has such variation in color compared to the others. The greater the number of clusters used by the algorithm, the greater the number of colors on the map.

Through applying these and other methods of clustering, any group of similar entities within the dataset can be clustered and plotted. For instance, a cluster map of any crime can be produced, as was shown in the hierarchical clustering example. An example of another iterative algorithm which could be used to find the density center point for stop and frisks is Mean-shift clustering, which also automatically identifies and uses the optimal number of clusters. Clustering could be applied to this dataset specifically to find areas where certain crimes clusters are most dense, such as violent or sexual crimes, non-violent or drug related crimes, etc. To internally validate the model, individual clusters can be isolated and examined. This technique is useful for the purposes of analyzing the methods used and determining the areas of high/low density on the map visuals.

Evaluation

Most interestingly from the results of the modeling performed in this report is the ways in which each algorithm uses clustering, and determining numbers of clusters to use, be it through user input or automatic defining. The plotted line charts, Figures 3.2 and 3.7 show how the algorithms make these decisions using silhouette scores. What is also interesting, is that regardless of the algorithm used the outputs are providing similar insights into the dataset, specifically regarding density of crimes, or stops, as outlined earlier. Also noteworthy are the plethora of possibilities for clustering using this dataset, where one could cluster for any crime of their choosing, physical force used, or simply the density of arrests made clustered by borough.

REPORT 4:

Data Preparation

Class variables have been prepared and defined earlier in Report 1, and further spoken on in Report 3, however a brief overview of what operations were done to the dataset, and what the final dataset looks like will be given here in this section.

The data was checked for nulls and duplicate values, all of which were removed from the data set. Next, the 2 original columns for date and time were combined to one datetime column. After this, ranges/filters were imposed on the acceptable age and weight values, eliminating erroneous data. After carefully analysing the original dataset, a series of columns were dropped from the dataset to reduce strain on local machines. Any column which was derivative or was used to generate another column, as well as any column which does not contain relevant data has been dropped from the data set. After these operations, the dimension of the dataset are 499,670 records, and 79 columns. For the purposes of Report 4, a dataframe including all columns with prefixes “cs_” and “rf_”, or reason for stop, reason for frisk, as well as “armed” as previously defined, are combined. Labels were created for these columns, and then the “armed” columns were dropped, as they will be the predicted target in the modelling stage of this report. The numerical columns used for predictive modelling are “age”, ”height” and “weight”. The categorical columns used for predictive modelling are “race”, “city”, “build”, “eyecolor”, “haircolr”, and “sex”.

Modelling

After a little data preparation the dataframe is broken down into a 80/20 training/test set. 80% of the data will be used to train the model, and 20% of the data will be used for testing the model. The dimensions of the training dataframe are 399,736 records, and 28 columns. The dimensions of the testing dataframe are 99,934 records, and 28 columns. At this point the categorical columns which are to be used are passed through OneHotEncoder, and a ColumnTransformer for utilizing the fit function which will read the values from the OneHotEncoder output. For the purposes of the predictive modeling in this report, the model will be predicting for whether or not a person is armed based on other variables.

The first predictive modeling technique is the Decision Tree Classifier, which produces a flow chart like structure, where nodes are linked by decision rules, ending in an outcome node. The results of applying this method of Classification to the train/test data sets are depicted in Figure 4.1 below.

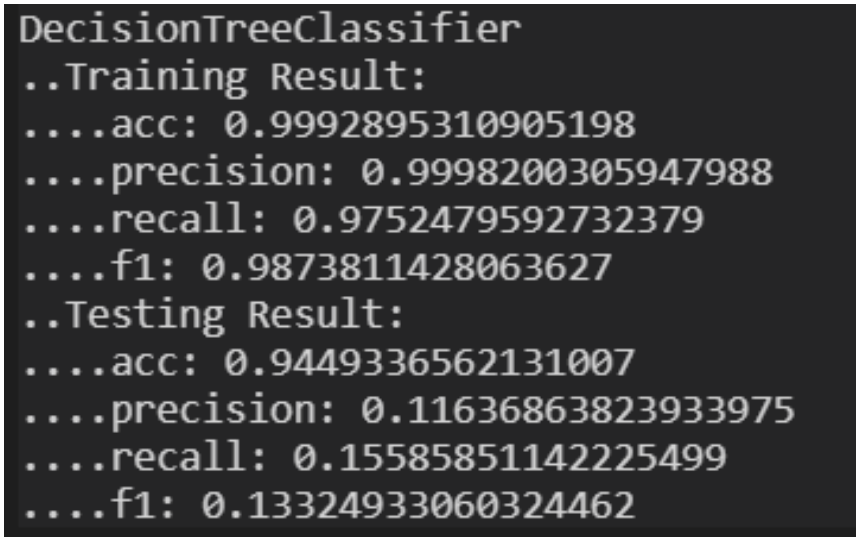

Figure 4.1: Printed results of Decision Tree Classfication

Presented in these results is that the Decision Tree Classifier works very well with the train set, and not as well with the test set. The train set sees high values in every category, however the test data set only has a high accuracy score. The best metric to compare for this dataframe, as it contains many "false" values, is f1 score, as it is the mean of precision and recall, and is a general measure of accuracy for the model. From the low f1 score of 0.13, or 13%, it is clear that this particular classification model is not that accurate for making predictions on this dataframe. That being said, looking at the plotted tree diagram provides more context.

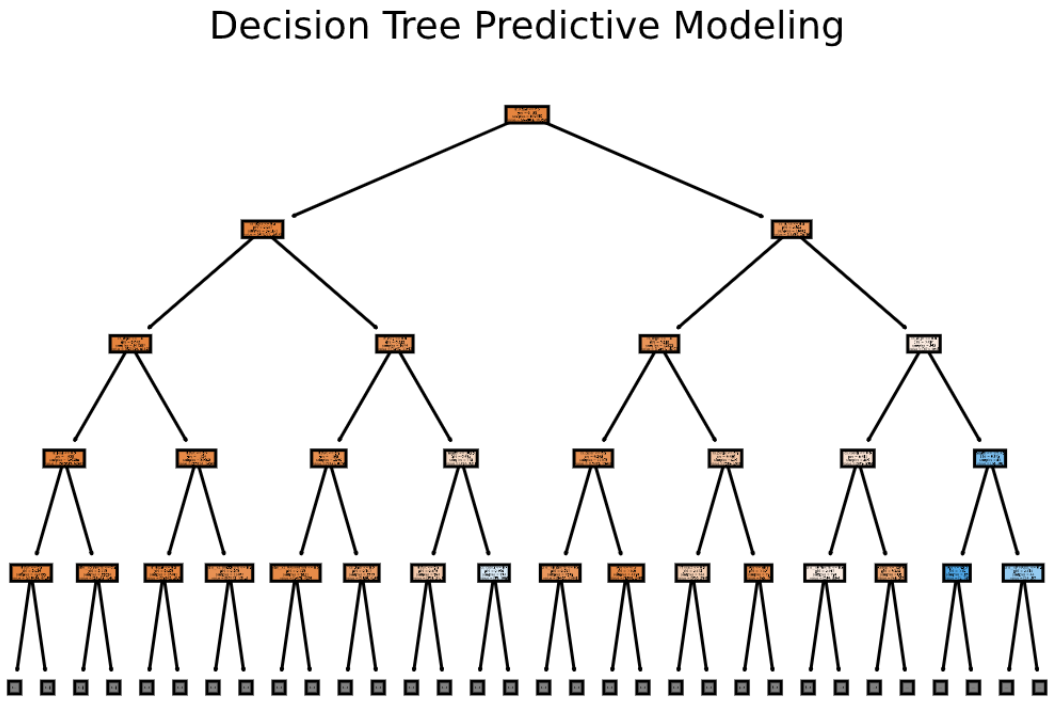

Figure 4.2: DecisionTreeClassifier output

The flowchart in Figure 4.2 begins with the most significant factor for determining whether a person is armed or not, as predicted by the model. This node, known as the root, is for column “rf_othsw” , or reason for frisk – other suspicion of weapons, which means that a person is most likely to be armed if the officer involved in the stop has some other suspicion of the individual carrying a weapon. The sample size of this node is 399,736 records, the entire training data set. The value split for the True/False condition were 388,343 False, 11,393 True. The next most significant factor in determining whether a person is armed is reason for stop – carrying a suspicious object, which of course make logical sense. Thirdly, is reason for stop – actions indicative of a drug trade, which is due to the relation of the sale of contraband and violent crime. At this stage, the sample size is a mere 883 records, and continues to lessen with each tier.

Since there exists a binary dependent variable in the dataframe, Logistic Regression can be an appropriate method of predictive modeling. This method will be useful in describing the relationship between the binary variable () and more than one categorical variable. The data has been sufficiently prepared for this modeling technique, and the results of applying it to the train/test data set are seen in figure 4.3.

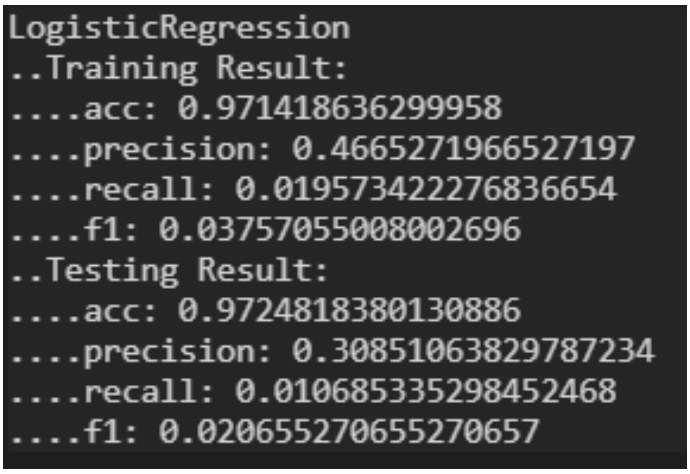

Figure 4.3: Results of the Logistic Regression predictive modeling technique

This particular method resulted in even lower f1 scores and therefore overall accuracy on both the train/test data sets than the Decision Tree did. Perhaps surprisingly, the accuracy statistic of both the train/test results are quite high, around 97%. The f1 score for the test set is a mere 2%, making this method have questionable value in predicting whether or not an individual will be armed. The method does produce a descriptive chart in Figure 4.4 which shows the influence these attributes have on one another.

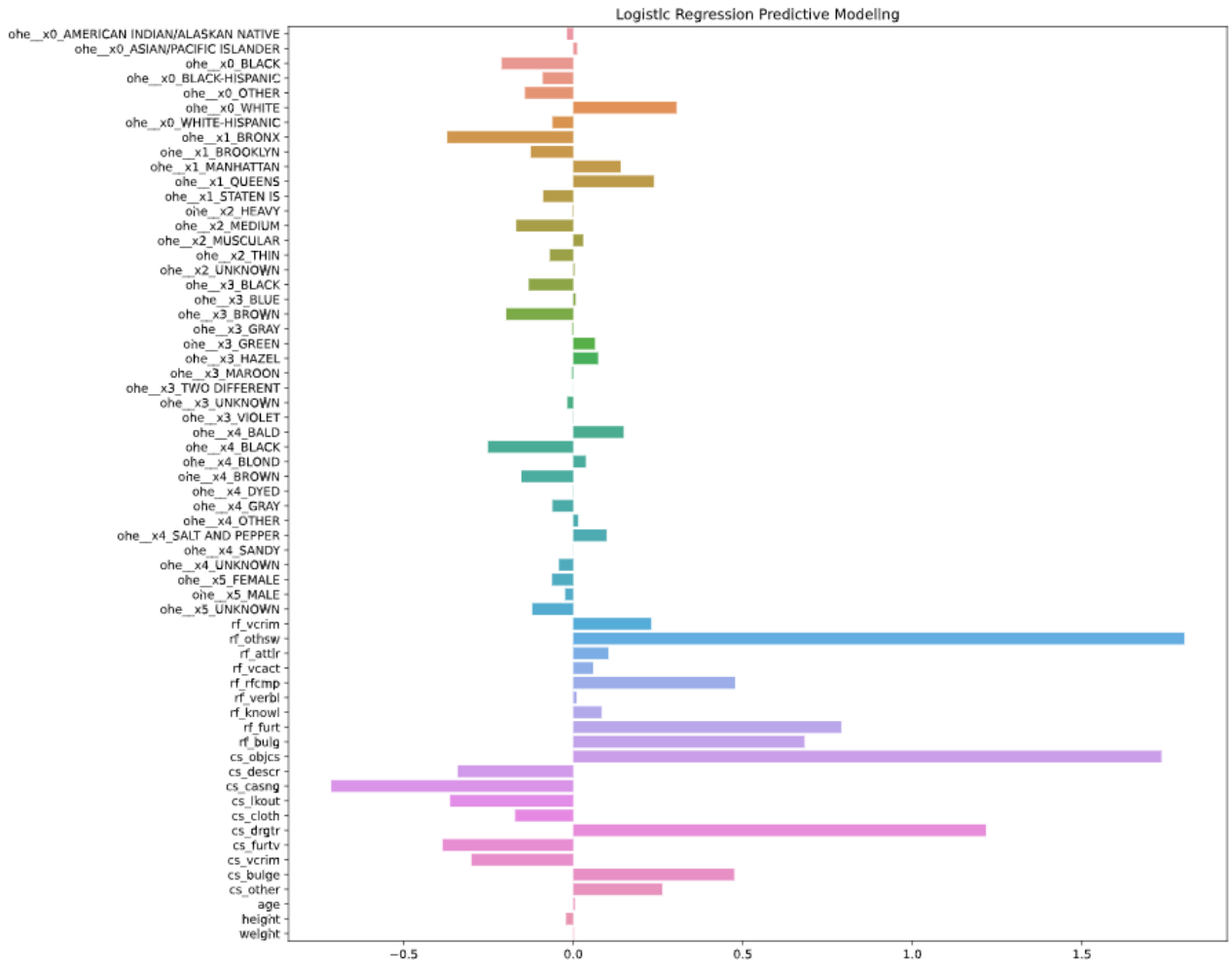

Figure 4.4: Bar chart of Logistic Regression results

Figure 4.4 confirms what the Decision Tree modeling had predicted earlier - that a person is most likely to be armed if they are frisked for other suspicions of weapons, followed by reason for stop – carrying a suspicious object. The negative values exist because this chart is generated using a logarithmic scale, and the values with the largest positive value are the most dependent factors in the dataframe. Some changes could be made to the included columns at this

stage to fine tune the model, and only use variables with positive values, however all of the data used will remain for the next steps of this Report.

The last method of predictive modeling which will be used for this case study is the Naïve-Bayes method, specifically the Multinomial variation, which will calculate the probability of a data point belonging to a certain class. This is useful for this report as it will predict the most likely attribute an armed individual will have. As always, the model is applied to the train/test data sets, and the resulting output is represented in Figure 4.5

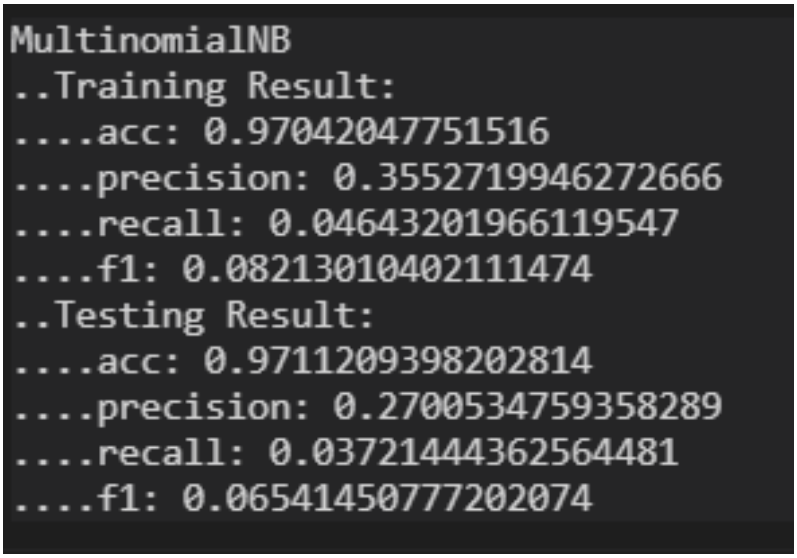

Figure 4.5: The printed results of Naïve-Bayes on the dataframe

This particular method had a f1 scores slightly higher than seen in Logistic Regression, however they are still less than desirable at only 8.2% and 6.5% for the train/test sets. As seen in the other methods results, the model is very accurate on both data sets, however, is significantly

lacking in the other measures. The Naive-Bayes method finds the attributes of an armed individual that are most probable. Figure 4.6 is the visual representation of the output of the Naïve-Bayes method, as a bar chart. The sections which extend furthest to the left are the most probable features an armed individual will have. The columns which have the "ohe_" prefix are the categorical columns passed through OneHotEncoder earlier, present on the left side of Figure 4.6.

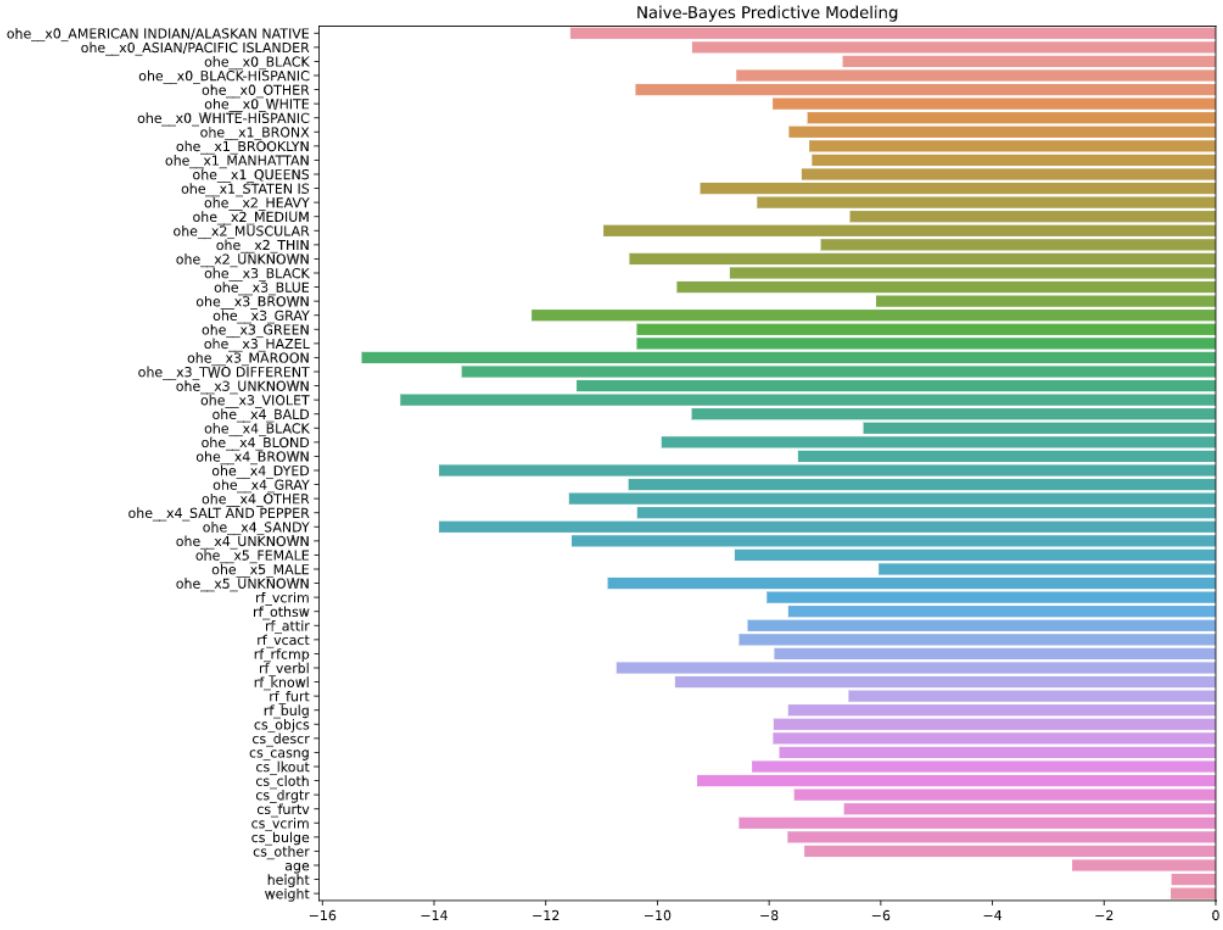

Figure 4.6: Naïve-Bayes predictive modelling bar chart output

Clearly from the bar chart above, the individual characteristic which is most strongly tied to whether an individual is armed, is having a “Maroon” eye color. The next factor which is most influential on whether an individual is armed is having a “Violet” eye color. The third most influential characteristic is tied between having “Dyed” or “Sandy” hair. Both the categorical and numerical columns are listed on the left side, and interestingly the numeric columns have very little influence on determining whether a person is armed or not. The transformed columns have the prefix “ohe_” are the categorical variables included and passed through OneHotEncoder, and in this case the x0 class = race, x1 = city, x2 = build, x3 = eye color, x4 = hair color, and x5 = sex.

Evaluation

This predictive model was used to aid in determining whether an individual is armed based off of other variables in the data set, more explicitly, the individuals race, age, sex, height, weight, eye color, hair color, and which city they lived in. This model is not a very effective tool for determining whether a person is armed, as is demonstrated by its continuously low f1 scores in the test modeling. For these reasons, the model is not advised to be used in policing. If it were to be implemented, the police force could understand the model as saying any person with sandy colored hair is more likely to be armed, so they could essentially be targeted. The effectiveness of the model could be measured through a series of year over year findings, each contributing to a finer tuning of the model, increasing the amount of data it can learn from. This could unlock further relations and correlations within the data, and potentially uncover other factors influencing whether an individual is armed or not. With frequent updates to the model, further fine tuning can be done to increase confidence in the model, and if the variety of data included

was wider a more effective model could be designed, however the model overall is inaccurate and should not be used in the real world.