Advanced Concepts in Data Analytics

Final Project: New York City Case Study

Southern Alberta Institute of Technology

Matthew Leech

Spring 2020

Table of Contents

Executive Summary:

In 2002, there were 97,296 "stop-and-frisk" stops made by New York police officers; 82.4% resulted in no fines or convictions. The number of stops increased dramatically in 2008 to over half a million, 88% of which did not result in any fine or conviction, peaking in 2011 to 685,724 stops, again with 88% (603,437) resulting in no conviction. Leading to the remaining 82,287 resulting in convictions. On average, from 2002 to 2013, the number of individuals stopped without any convictions was 87.6%. (New York Civil Liberties Union, 2012) This final project aims to utilize the tools and knowledge gained in the Data Science and Analytics program to gain insight into New York's SFQ program using visualization techniques, machine learning techniques with python and its multiple libraries, and statistical analysis.

The bulk of the project was performed in Visual Studio Code and written in the Python programming language. The process saw the use of multiple libraries including Pandas, NumPy, Folim, Seaborn, Matplotlib, SciKit/SKLearn, and others. With these libraries machine learning techniques were utilized and applied to extract information. For Association Rule Mining frequent itemsets and association rule mining was applied to find commonly occurring attributes of SFQ entries. Clustering saw multiple methods applied such as heirarchical clustering, KMeans clustering, and DBSCAN clustering. Finally, predictive modelling was performed using decision trees, logistic regression, multinomial native bayes, and stochastic gradient descent classifiers to understand what traits were likely to indicate weather an individual was carrying a weapon when frisked.

This report found key insights into the data and was able to extract actionable information for future usage. Future research will include applying population statistics to avoid biased results, and deployment into New York in hopes of streamlining the SFQ process by understanding what is driving the crime.

Data and Visualization:

Business Understanding

1) What is the purpose of the SQF program?

The SQF stands for the Stop-Question-and-Frisk program created by The City of New York's Police Department. The goal of the program is to create a policing strategy that officers follow when stop and question hundreds of thousands of pedestrians annually and searched them for weapons and other contraband. In 2016 the program reported 12,404 cases of the program in action, and over 650,000 in its height in 2011.

2) How would you define and measure the effectiveness of such a program?

The program's goal was to reduce overall crime in New York by randomly choosing pedestrians on the street, so taking a look at the crime statistics for the years that SFQ has been in action and comparing it to years where the program was not in place would allow us to define the effectiveness. Due to population fluctuation we should be looking at percentages instead of the raw numbers for measures to derive the effectiveness of the program

3) What data would you need be able to judge its effectiveness?

Looking at a public database for crime in the areas where SFQ is enforced paired with residential population stats would allow us to efficiently judge the programs effectiveness.

Data Understanding

1) Describe the meaning and type of data (e.g., scale, values) for each attribute in the data file.

The dataset comes with over 530,000 rows and 112 columns. It comes in a couple of types, including int64, and object.

```
[7]  df.dtypes

year          int64
pct           int64
ser_num       int64
datestop      int64
timestop      int64
              ...
xcoord       object
ycoord       object
dettypcm     object
linecm       object
detailcm     object
Length: 112, dtype: object
```

Figure 1. Checking the data types in the df dataframe

The values themselves range from 0 into the hundreds of thousands, and even millions. Some values seem to be inaccurate, such as age, for example. The minimum age in this datast is 0, which is impossiblel for a SFQ.

2) Verify data quality. Are there missing values? Duplicate data? Outliers? Are those mistakes? How do you deal with these problems?

The dataset does have outliers. As mentioned above there are values in columns that do not make sense. Some values seen in the Age, Height, Sex, and many other columns do not fit the dataset. There are also many missing values. When using the .dropna operation we can see our dataset drop rows where there are null values. This makes the dataframe go from 532911 to 516968. Some duplicates are expected in the dataframe as they are not unique to the row – for example, multiple people can be 20 years old.

3) Give simple, appropriate statistics (e.g., range, mode, mean, median, variance, counts) for the most important attributes in these files, and then describe what they mean or whether you found something interesting. Note: You can also use data from other sources for comparison.

	year	pct	ser_num	datestop	timestop	perobs	compyear	comppct
count	532911.0	532911.000000	532911.000000	5.329110e+05	532911.000000	532911.000000	532911.0	532911.0
mean	2012.0	66.502354	4979.569876	5.563631e+06	1415.161796	2.445776	0.0	0.0
std	0.0	32.343511	4360.019625	3.413338e+06	737.003276	5.028139	0.0	0.0
min	2012.0	1.000000	1.000000	1.012012e+06	0.000000	0.000000	0.0	0.0
25%	2012.0	41.000000	1790.000000	2.292012e+06	1002.000000	1.000000	0.0	0.0
50%	2012.0	70.000000	3790.000000	5.052012e+06	1615.000000	1.000000	0.0	0.0
75%	2012.0	94.000000	6982.000000	8.252012e+06	2030.000000	2.000000	0.0	0.0
max	2012.0	123.000000	24652.000000	1.231201e+07	2359.000000	955.000000	0.0	0.0

Figure 2. A summary table of simple statistics for the most important attributes in the file.

Some of the data reveals something interesting. For example the column “year” only has values of 2012 for the statistics which means this dataset is only for the year 2012. Things like “age” have outliers with a minimum of 0 and a maximum of 999. The “weight’ column also has odd values, with a minimum of 0 and a maximum of 999.

4) Visualize the most important attributes appropriately (at least 5 attributes). Important: Provide an interpretation for each chart, explaining each attribute and why you chose the visualization you did.

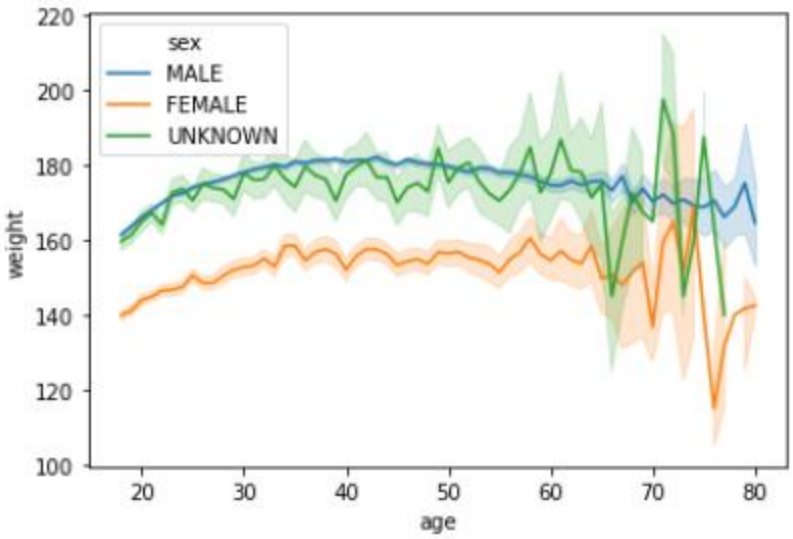

Figure 3. Graph displaying the weight based off of age for all sexes available

This graph displays the weight and age of all people who were chosen for the SFQ protocol. From this information we cannot see any gaps in age or weight until the later end of the graph. This could be due to a lack of people chosen for the SFQ at ages above 60.

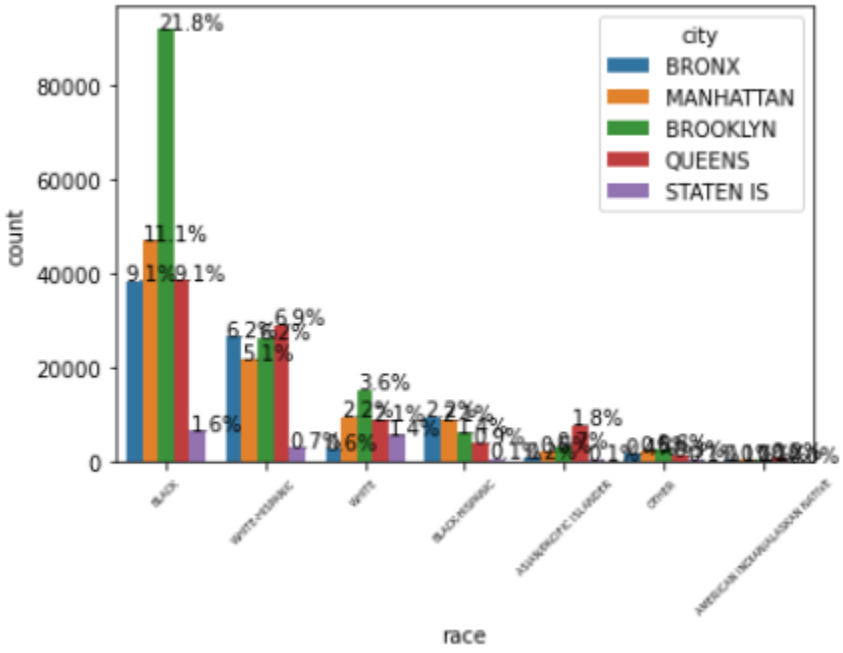

Figure 4. A bar graph showing the count of SFQ searches in certain areas based off of race

This information has some facts that we are able to pull from the SFQ data. Here we can see that the African American race has a significant amount of SFQ searches, and especially in Brooklyn.

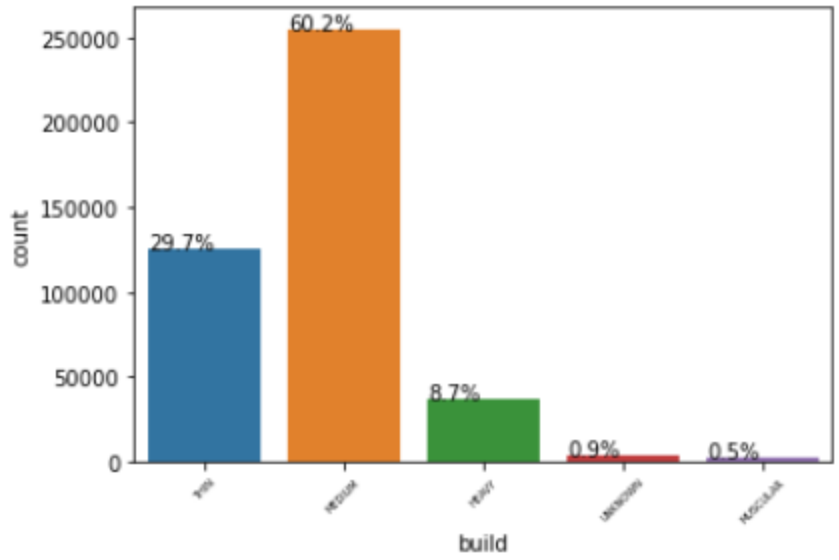

Figure 5. Bar graph displaying the build type of people chosen for SFQ

Although this information isn't too telling, it does show that most of the SFQ searches were performed on medium build types at over 60%

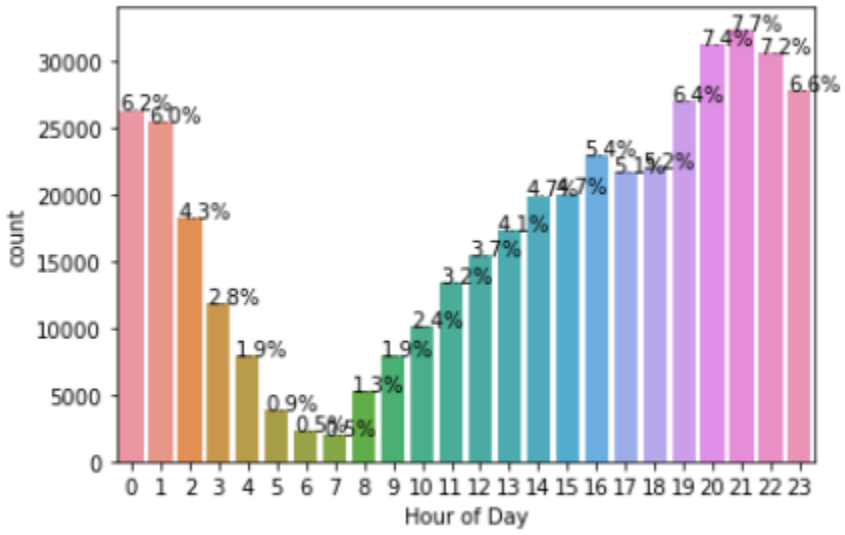

Figure 6. This graph shows the hour of the day when SFQ searchers were performed

This information is quite telling as it states that most SFQ searcher were performed during the night.

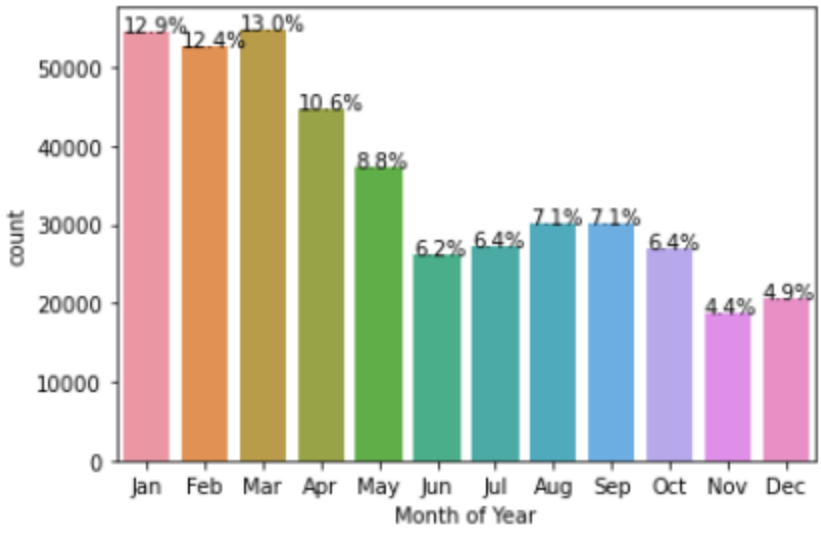

Figure 7. Graph displaying the monthly occurrences

This graph displays what month the SFQ searches occur most. From this we can see that the first few months of the year have the most occurrences, and it slowly declines throughout the year.

5) Explore relationships between attributes. Look at the attributes and then scatter plots, correlation, cross-tabulation, group-wise averages, etc., as appropriate.

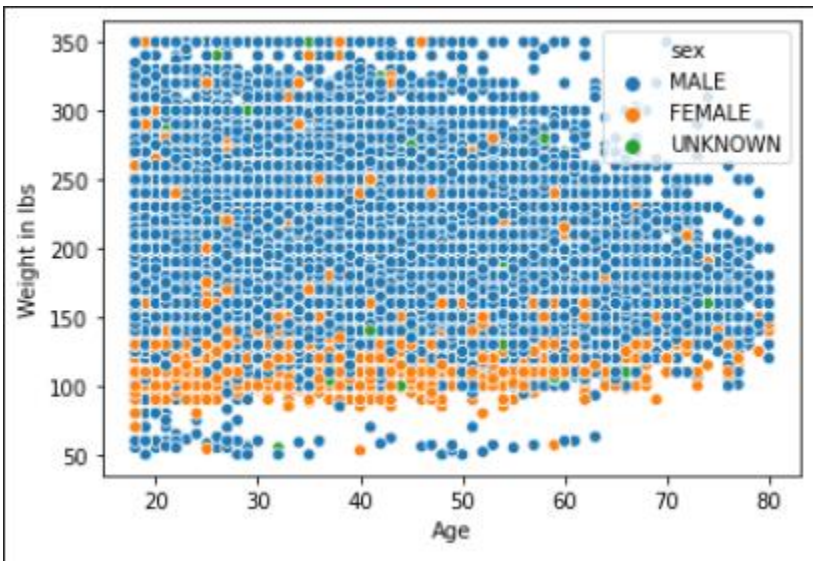

Figure 8. Scatterplot showing age, weight divided by sex

This section looks at the wide range of values for the SFQ searches. With so much data it will be difficult to see any relationships or distinct groups using a scatter plot. Association would likely be a better method for finding similarities. This scatterplot shows us some gaps with much heavier people and much lighter people as well as gaps at ages 60+.

6) Compare the reasons for an SQF and what type of force was used by the officer.

Looking at the dataframe we can see that the category pf_hands was the column that was used the most in the physical force groups. This meant that most of the SFQ stops used hands for force, and not the others. When looking for the reasons they ranged across all types of crime, but one of the most interesting things seen is the fact that most SFQ actions were performed on African Americans. This brings to light the possibility of racial profiling as previous questions highlighted the same information.

Association Rule Mining:

Data Preparation

1) Construct the required transaction data set for frequent itemset and association rule mining.

Constructing the transaction data requires us to create all data into a format that apriori can use – Boolean. To do so we first create a new list called “pfs” which has the columns from the pickle file that start with “pf_”. Then we create a new list called “armed” which includes all of the values in the pickle file that match what we predefine as “armed”. From there we create a new dataframe called x, which includes the both lists of “pfs” and “armed” from our old dataframe: df. Finally we convert the row values to Boolean with the statement x = x == “YES”. Next we create a column within dataframe x to represent if a person is armed. In order to run some association rule mining, we need to select categorical columns from the dataframe df and bring them into the new dataframe x. In this section I have chosen hair color, eye color, build, race, city, sex, and age.

Modelling

1) Create frequent itemsets and association rules.

	support	itemsets
0	0.754921	(haircolr_BLACK)
1	0.899803	(eyecolor_BROWN)
2	0.602412	(build_MEDIUM)
3	0.526263	(race_BLACK)
4	0.918754	(sex_MALE)
5	0.688776	(haircolr_BLACK, eyecolor_BROWN)
6	0.698485	(haircolr_BLACK, sex_MALE)
7	0.544436	(build_MEDIUM, eyecolor_BROWN)
8	0.827877	(eyecolor_BROWN, sex_MALE)
9	0.561285	(build_MEDIUM, sex_MALE)
10	0.637523	(haircolr_BLACK, eyecolor_BROWN, sex_MALE)
11	0.507649	(build_MEDIUM, eyecolor_BROWN, sex_MALE)

Figure 9. Table showing the results of a frequent itemsets function.

The above picture is the results of a frequent itemsets function, using a minimum support of 0.5.

antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage
(build_MEDIUM, eyecolor_BROWN)	(sex_MALE)	0.544436	0.918754	0.507649	0.932432	1.014887	0.007447
(build_MEDIUM)	(sex_MALE)	0.602412	0.918754	0.561285	0.931728	1.014121	0.007816
(haircolr_BLACK, eyecolor_BROWN)	(sex_MALE)	0.688776	0.918754	0.637523	0.925589	1.007439	0.004708
(haircolr_BLACK)	(sex_MALE)	0.754921	0.918754	0.698485	0.925243	1.007062	0.004898
(eyecolor_BROWN)	(sex_MALE)	0.899803	0.918754	0.827877	0.920065	1.001427	0.001179
(haircolr_BLACK, sex_MALE)	(eyecolor_BROWN)	0.698485	0.899803	0.637523	0.912723	1.014359	0.009025
(build_MEDIUM, sex_MALE)	(eyecolor_BROWN)	0.561285	0.899803	0.507649	0.904442	1.005155	0.002604
(sex_MALE)	(eyecolor_BROWN)	0.918754	0.899803	0.827877	0.901087	1.001427	0.001179
(haircolr_BLACK)	(eyecolor_BROWN, sex_MALE)	0.754921	0.827877	0.637523	0.844491	1.020067	0.012542
(build_MEDIUM)	(eyecolor_BROWN, sex_MALE)	0.602412	0.827877	0.507649	0.842694	1.017897	0.008926
(eyecolor_BROWN, sex_MALE)	(haircolr_BLACK)	0.827877	0.754921	0.637523	0.770070	1.020067	0.012542
(sex_MALE)	(haircolr_BLACK)	0.918754	0.754921	0.698485	0.760252	1.007062	0.004898

Figure 10. Table showing the results of an association rules function.

This picture displays the results from the association rules mining function with a minimum threshold of 0.7. Sorted with the highest confidence first, we can see that descriptions such as medium build, brown eye colour, black hair is all high on this list.

2) Use tables and visualizations to help explain your results.

See above for tables summarizing results for both data mining techniques used.

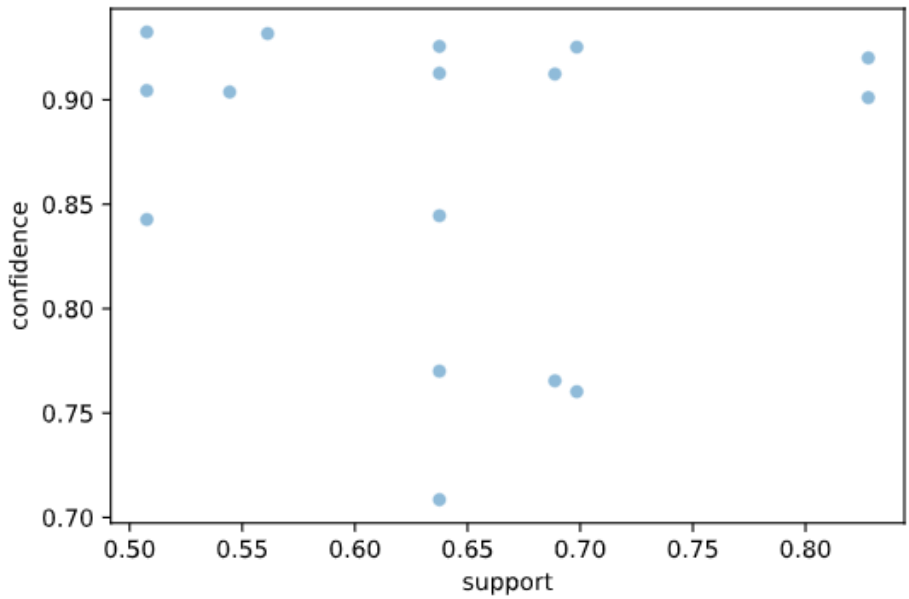

Figure 11. Table showing the confidence and support for the highest values of the association rules function.

This scatterplot displays the top results from the data mining technique association rules as well as the support. The best values will of course have high confidence and high support.

Evaluation

1) What findings are the most interesting? Why?

The high confidence in the results themselves are very interesting. With some at 95% or above that assumes that 95% of the SFQ have the traits defined in the research. This is not true, what this is displaying is that these descriptors are likely to be together with high accuracy. Meaning that someone with who is male is highly likely to have a medium build and brown eyes, which are dominant traits and likely in a lot of males.

Cluster Analysis:

Data Preparation

1) Define and prepare your class variables. Note: You may have to combine different columns.

Most of the data preparation was done in part one of this report, under the *Data Understanding* and *Data Preparation* sections above. The earlier preparation allowed us to format the data to be used properly in cluster analysis, however there were some alterations that needed to be done with the “coord” columns.

```
# creating coordinates readable for mapping purposes by assigning boolean values
df_prostitution["lat"] = df["coord"].apply(lambda val: val[1])
df_prostitution["lon"] = df["coord"].apply(lambda val: val[0])
```

Figure 12. Visual showing the creation of latitude and longitude columns

The above picture displays how I went about creating different columns for latitude and longitude from a singular “coord” column from a previous dataframe. It also gets the data prepared for machine learning purposes.

```
num_city = df["city"].nunique()
num_pct = df["pct"].nunique()
```

Figure 13. Creation of variables used in cluster analysis

The above figure displays the creation of two variables. These variables are taking the unique values from the original dataframe for the number of precincts and cities within New York. This was important as it helped define the silhouette scores for the different number of clusters. Other variables in this section were created to make for easier computing down the road. These include best_k, m, colors, rf, and many others.

2) Remove variables that are not needed/useful for the analysis.

Because we are creating a new dataframe from the original one, we are not removing variables as it would ruin what we already have. The creation of a new dataframe (df_prostitution) allows us to create a clean dataframe with the variables we require and keep the computing time as low as possible.

3) Describe the final dataset that is used for classification and include the scale/range for the new combined variables.

	perobs	perstop	age	weight	height	lat	lon	label
count	982.000000	982.000000	982.000000	982.000000	982.000000	982.000000	982.000000	982.000000
mean	3.421589	5.725051	34.399185	160.066191	169.264358	40.744990	-73.911695	0.731161
std	5.193647	4.677332	11.262665	34.854331	8.762262	0.069755	0.068623	2.144387
min	0.000000	1.000000	18.000000	53.000000	134.620000	40.562661	-74.192847	-1.000000
25%	1.000000	3.000000	24.000000	130.000000	162.560000	40.685501	-73.952165	0.000000
50%	2.000000	5.000000	33.000000	160.000000	168.910000	40.747228	-73.892562	0.000000
75%	5.000000	5.000000	43.000000	180.000000	175.260000	40.815000	-73.884646	0.000000
max	60.000000	55.000000	70.000000	340.000000	203.200000	40.905736	-73.718028	11.000000

Figure 14. A picture showing the dataframe df_prostitution

The above picture uses the .describe() function for the dataframe df_prostitution with some simple statistics for the integer columns. Limitations were put in place in the earlier stages of the data analysis to get rid of outliers such as ages and weights outside of an acceptable range.

```
pct          object
inout        object
trhsloc      object
perobs      float64
perstop     float64
              ...
coord        object
height      float64
lat         float64
lon         float64
label         int64
Length: 82, dtype: object
```

Figure 15. A visual of the datatypes and length of the dataframe

The dataframe has multiple different data types within it as seen above. The length of the dataframe if 82 and ranges from column to column.

Modelling

1) Perform cluster analysis.

a) Cluster the location for a crime of your choice. Note: Found clusters might be different depending on the time of day.

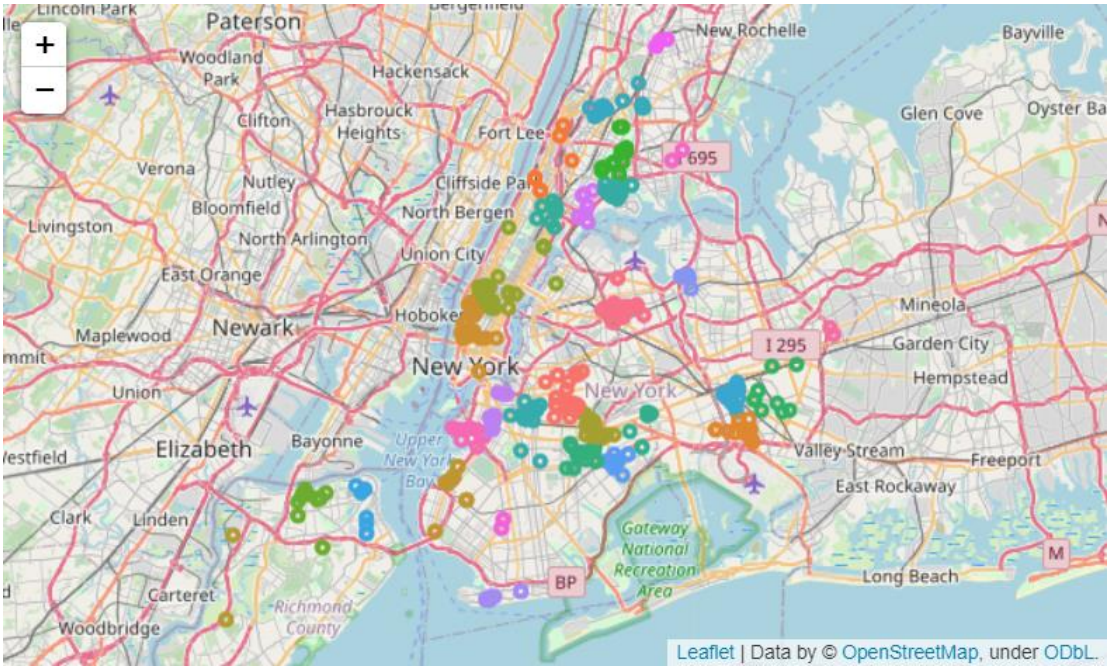

Figure 16. Clusters of instances of prostitution shown on a map of New York using Heirarchical clustering

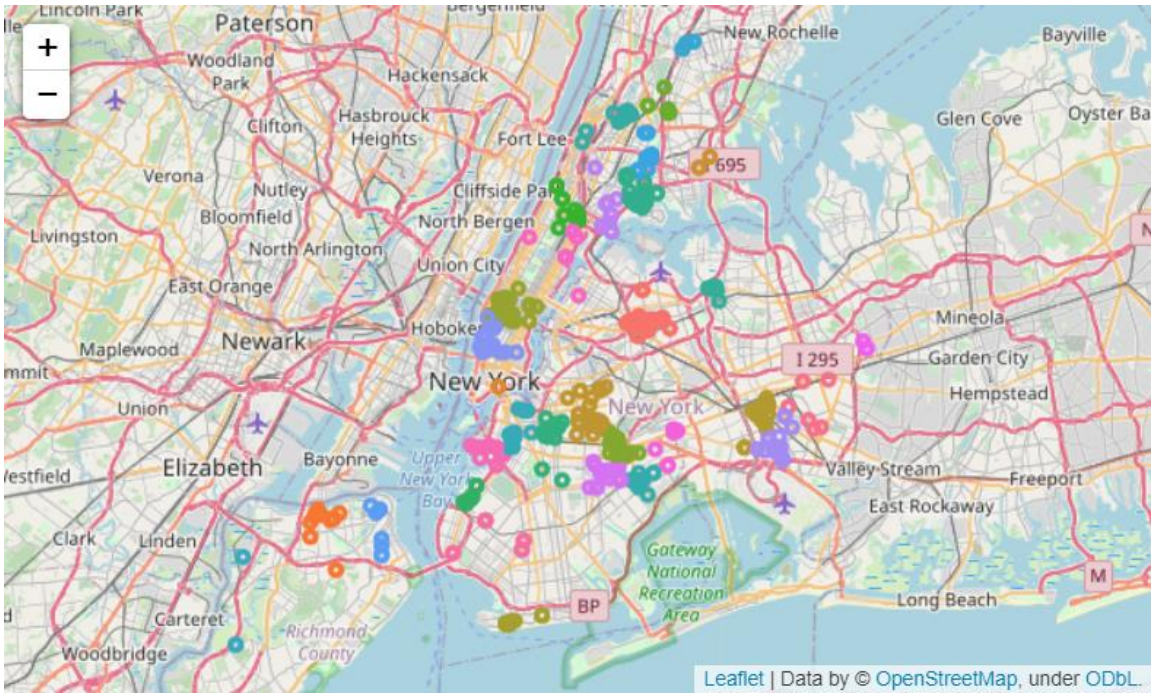

Figure 17. Clusters of instances of prostitution shown on a map of New York using KMeans Clustering

Seen in figures 16 and 17, the maps show different groups of clusters on a map on New York. The method of clustering for the two (Heirarchical and KMeans clustering) is different, however the number of clusters is the same. Using the crime "prostitution" the map shows instances of prostitution and grouping by latitude and longitude. Both Heirarchical and KMeans clustering bases its clusters on proximity to other datapoints and ensuring that the clusters are broadly similar. Using the latitude and longitude of each instance of a prostitute being stopped, each method grouped the clusters slightly differently as seen in the above pictures.

b) Cluster stopped people by reasons for stop.

First, we need to create a new dataframe that takes the reasons people were stopped from the old dataframe.

```
# %% Creating a dataframe for columns that start with cs_ --> meaning reason the person was stopped
css = [col for col in df.columns if col.startswith("cs_")]
```

Figure 18. Code for creating a new dataframe

This snippet of code takes columns in the old dataframe that start with "cs_", which is outlined in the spec files as reasons that people were stopped. This allows us to use the DBSCAN clustering method to define how many clusters to use with the instances defined by the reason they are stopped. DBSCAN is another clustering method that uses density-based clustering – meaning that it defines the number of clusters based on the density of each group.

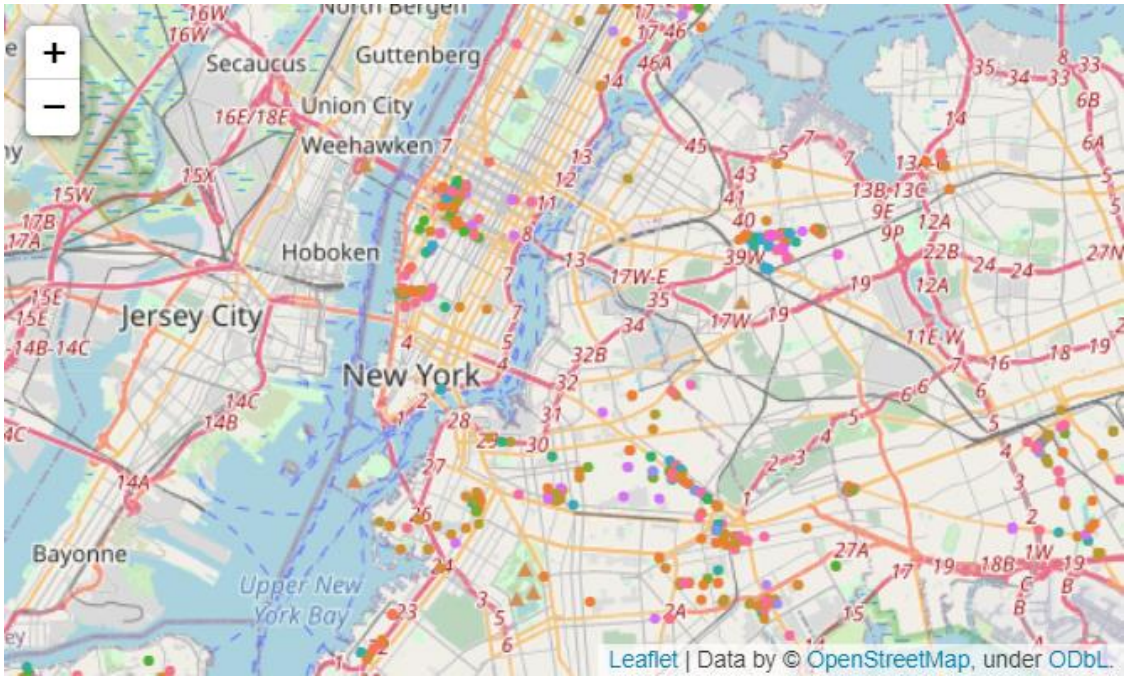

Figure 19. Map of New York displaying reasons for being stopped clustered by density

The DBSCAN method was applied to the reason people were stopped and mapped above, clustered based on the density of each instance with respect to others around it. DBSCAN decides the clusters itself and does not require the user to have any input on the decision-making process. As seen above, it is difficult to get any useful information out of this map. DBSCAN bases the number of clusters based on the proximity of each input to others around it, so like colours don't necessarily have to be together. Similar density clusters could be placed at many points on the map. It seems as if orange and red were the clusters that were most dense, with purple and blue being less dense.

c) What else can you use cluster analysis for in the data set?

Useful applications of cluster analysis are to group like things together. For example, you could decide what percentage an item should be on sale by grouping it to similarly bought products on that price. This would optimize your sales by ensuring the sales of like items are similar and effective. Other applications include banks clustering traits and amounts of money in accounts for credit scoring.

2) How did you determine a suitable number of clusters for each method?

In DBSCAN, the algorithm automatically decided the best number of clusters based on the information given. However, we can use silhouette scores to decide the best number of clusters for Hierarchical and KMeans clustering.

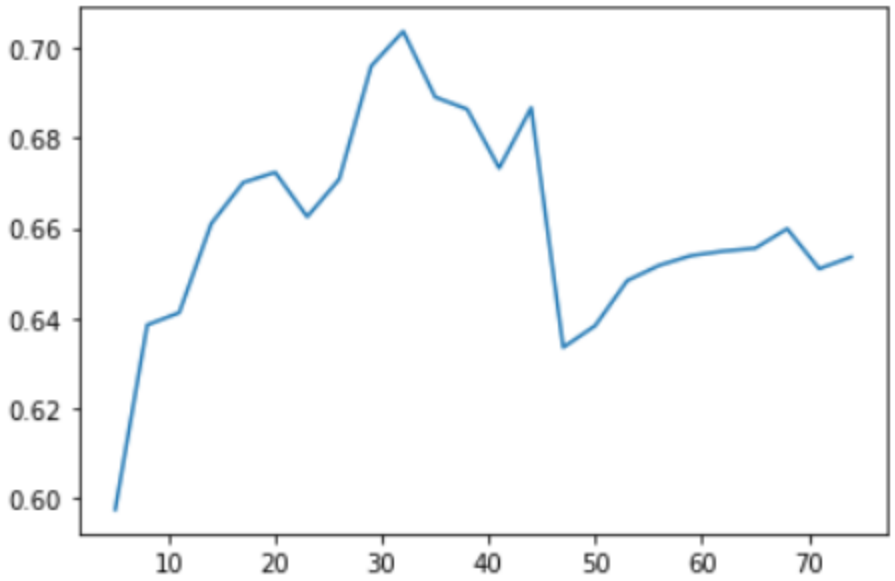

Figure 20. Line graph showing the silhouette score for Heirarchical clustering

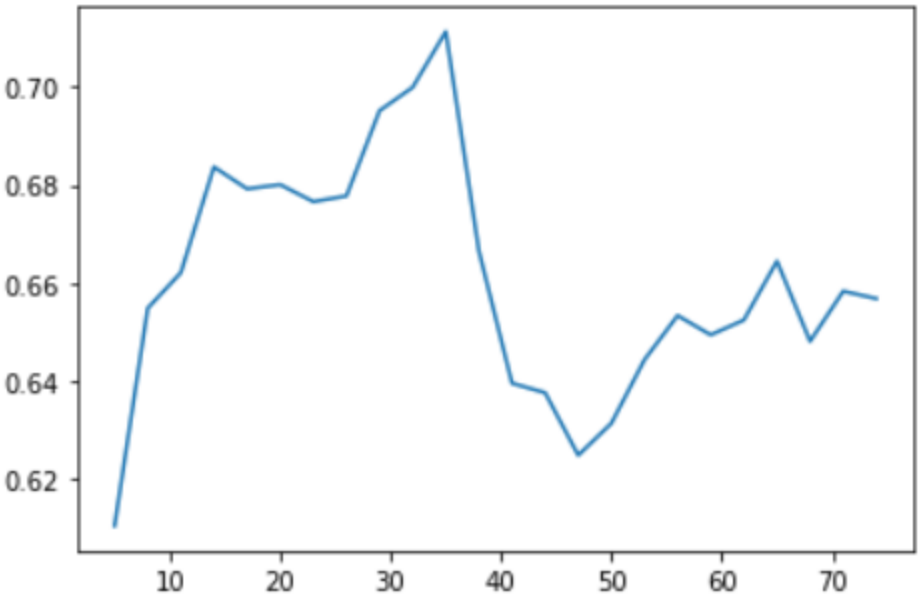

Figure 21. Line graph showing the silhouette score for KMeans clustering

This graph illustrates the silhouette scores based on the number of clusters. The higher silhouette score means that it is more accurate. Both KMeans and Heirarchical clustering had similar clusters being the highest silhouette scores, but varies in its trends.

3) Use internal validation measures to describe and compare the clusters (some visual methods would be good).

Looking at the DBSCAN results for the reason people were stopped, I can take a closer look at individual cluster groups to get a better understanding on how DBSCAN decided to cluster things together.

Figure 22. Map of New York looking at different cluster groups

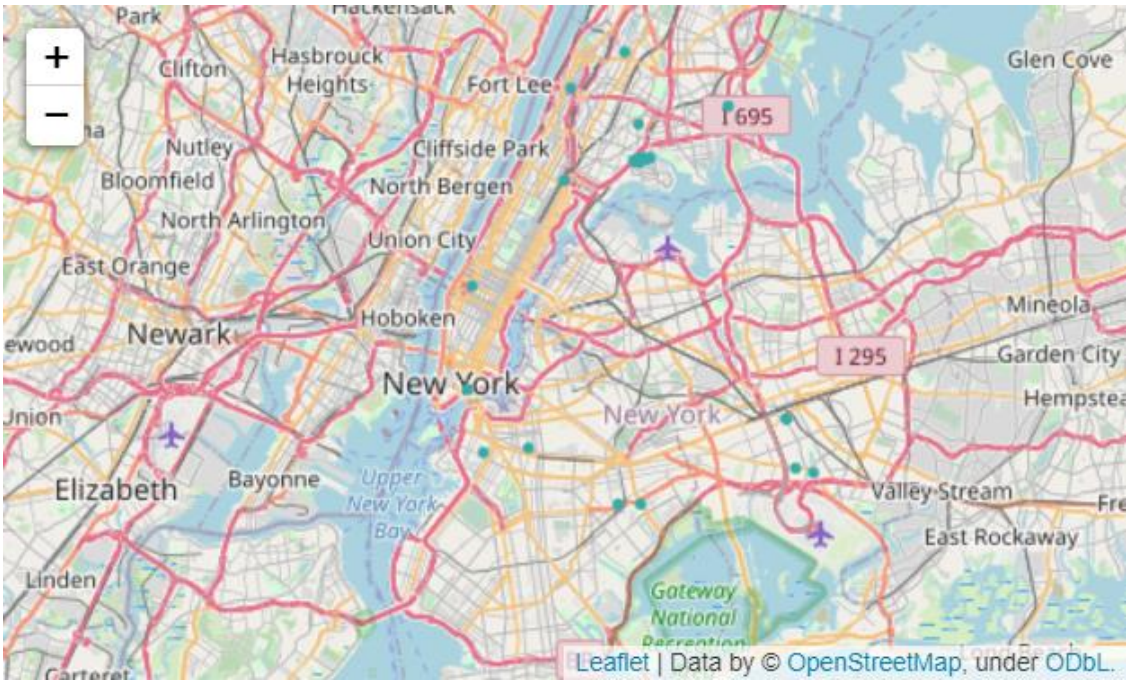

Figure 23. Map of New York looking at different cluster groups

The two maps above look at different cluster groups within the DBSCAN results. It is difficult to understand the method of how they are clustered individually. I have chosen these two cluster groups because they are high- and low-density groups (orange and green) in hopes of comparing the two. My conclusion it that it is difficult to interoperate anything from the DBSCAN, especially if you cannot see the whole picture because the density and grouping is based on other datapoints.

Evaluation

1) Describe your results. What findings are the most interesting? How can these findings be used?

The results of clustering were difficult to interpret as they were density and location-based clustering using crimes that aren't related to location. Giving the whole picture for the DBSCAN, KMeans, and Heirarchical clustering methods can give us an idea of areas where the crime is more likely to occur, which can be important to avoid high density areas. However, the DBSCAN method seemed to be the worst off as it clustered instances based on the density of instances of the area. So you could have a dense area with multiple different clusters, and not visually indicating anything. Further clustering could be used to incorporate similar crimes and show areas of high density to show the most crime ridden areas of New York.

Predictive Modelling:

Data reparation

1) Define and prepare your class variables. Note: You may have to combine different columns.

Similar to the *clustering* section, most of the data preparation was done in the earlier sections. Outliers, null values, and unused columns were either coerced or taken out of the dataset. Creation of a new dataframe x, was important as we had to ensure that it was ready for machine learning. To do this, we had to create columns and convert them to a Boolean fashion as seen below.

```
# %% Using columns of reasons for frisked and reason for stopped
rfs = [col for col in df.columns if col.startswith("rf_")]
css = [col for col in df.columns if col.startswith("cs_")]
armed = [
    "contrabn",
    "pistol",
    "riflshot",
    "asltweap",
    "knifcuti",
    "machgun",
    "othrweap",
]

x = df[rfs + css + armed]
x = x == "YES"

# create label for the dataset, then remove the weapons columns
y = (
    x["contrabn"]
    | x["pistol"]
    | x["riflshot"]
    | x["asltweap"]
    | x["knifcuti"]
    | x["machgun"]
    | x["othrweap"]
)
x = x.drop(columns=armed)
```

Figure 24. Image showing the creation of a new dataframe.

This dataframe took the columns of rf_ and cs_ from the original dataframe and converted each unique value into a new column in the new dataframe and making it Boolean. Other information for the dataframe was needed, so creating numeric and categorical columns from the original dataframe was

necessary. I decided to use age, height, wright, race, city, build, haircolr, and eyecolor to predict weather a suspect is armed or no.

```
num_cols = ["age", "height", "weight"]
cat_cols = ["race", "city", "build", "haircolr", "eyecolor",]

x[num_cols] = df[num_cols]
x[cat_cols] = df[cat_cols]
```

Figure 25. Creation of lists to bring into the new dataframe

Next, we needed to create training and testing splits of the data. Using a function of SciKit's model selection library we were able to make x and y training and testing splits.

2) Remove variables that are not needed/useful for the analysis.

Because most of the information was taken out in earlier sections, the only thing I needed to remove was the "armed" column which had the weapons in it as it defines if they are armed which is what we are trying to predict.

```
x = x.drop(columns=armed)
```

Figure 26. Code displaying the dropping of the armed column

3) Describe the final dataset that is used for classification and include the scale/range for the new combined variables.

```
<bound method NDFrame.describe of        rf_vcrim  rf_othsw  rf_attir  rf_vcact  rf_rf
verbl  rf_knowl  \
0           True     False     False      True      True     False     False
1          False     False     False     False     False     False     False
2          False     False     False     False     False     False     False
3          False     False     False     False     False     False     False
4          False     False     False     False     False     False     False
...          ...       ...       ...       ...       ...       ...       ...
532905     False     False     False     False     False     False     False
532906     False     False     False     False     False     False     False
532907     False     False     False     False     False     False     False
532909     False     False     False     False     False     False     False
532910     False     False     False     False     False     False     False

       rf_furt  rf_bulg  cs_objcs  ...  cs_bulge  cs_other  age  height  \
0         True    False     False  ...     False     False   20  175.26
1        False    False     False  ...     False      True   18  170.18
2        False    False     False  ...     False     False   19  175.26
3        False    False     False  ...     False     False   37  165.10
4        False    False     False  ...     False     False   21  170.18
...        ...      ...       ...  ...       ...       ...  ...     ...
532905   False    False     False  ...     False     False   25  165.10
532906   False    False     False  ...     False     False   21  177.80
532907   False    False     False  ...     False     False   22  162.56
532909   False    False     False  ...     False     False   18  172.72
532910   False    False     False  ...     False     False   18  177.80
```

```
0          False
1          False
2          False
3          False
4          False
           ...
532905     False
532906     False
532907     False
532909     False
532910     False
Length: 423826, dtype: bool
```

Figure 27. Using .describe to see the dataframes created

The above picture shows the x and y dataframes used in the machine learning process. Both dataframe contain Boolean values, but x also contains some numeric and categorical values. Both contain ~420,000 records.

Modelling

1) Create at least three different classification models (different techniques) for each of the classification tasks.

ExtraTreeClassifier

```
ExtraTreeClassifier

Training Results:
Accuracy: 0.9992764314859266
Precision Score: 0.999675885911841
Recall: 0.9760548523206751
F1 Score: 0.9877241673783091

Testing Resuls:
Accuracy: 0.9485734779202885
Precision Score: 0.1420345489443378
Recall: 0.13827468078480223
F1 Score: 0.14012939876913363
```

Figure 28. Measures of the ExtraTreeClassifier

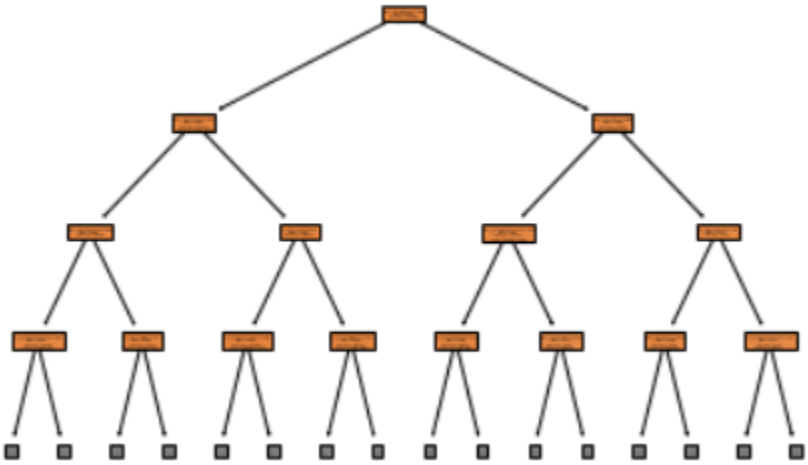

Figure 29. Tree visual for the ExtraTreeClassifier

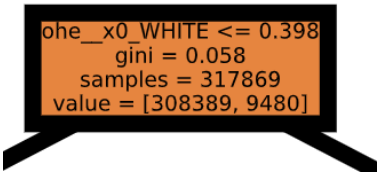

Figure 30. Top result for the ExtraTreeClassifier

MultinomialNB

```
MultinomialNB

Training Results:
Accuracy: 0.9694433870556739
Precision Score: 0.38244197780020184
Recall: 0.0399789029535865
F1 Score: 0.07239041161302645

Testing Resuls:
Accuracy: 0.9686099077927839
Precision Score: 0.31511254019292606
Recall: 0.030520087200249145
F1 Score: 0.055650198750709826
```

Figure 31. Measures of the MultinomialNB

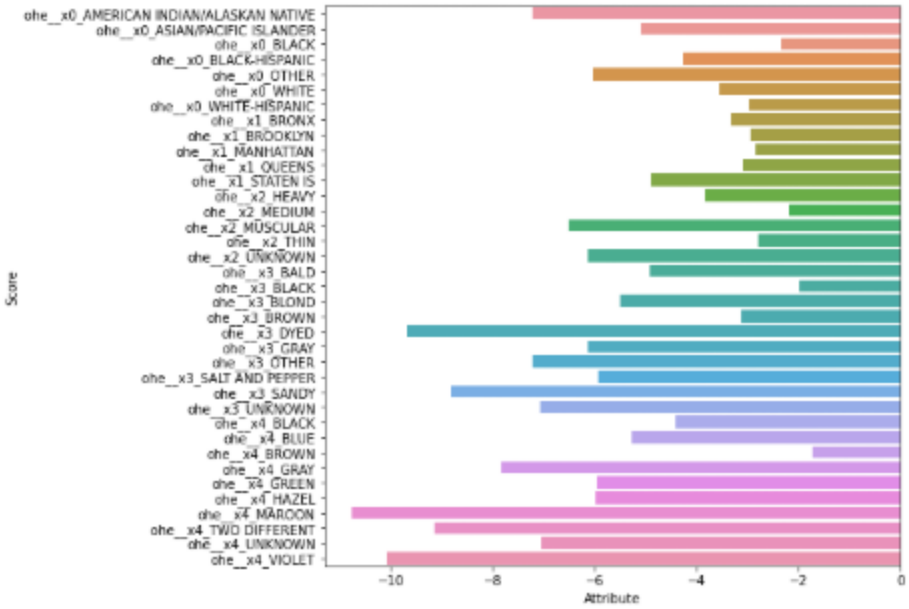

Figure 32. Results of the MlutinomialNB

SGDClassifier

```
SGDClassifier

Training Results:
Accuracy: 0.9701480798693801
Precision Score: 0.4801762114537445
Recall: 0.01149789029535865
F1 Score: 0.022458019985577417

Testing Resuls:
Accuracy: 0.9697046915258076
Precision Score: 0.5060240963855421
Recall: 0.013080037371535347
F1 Score: 0.025500910746812388
```

Figure 33. Measures of the SGDClassifier

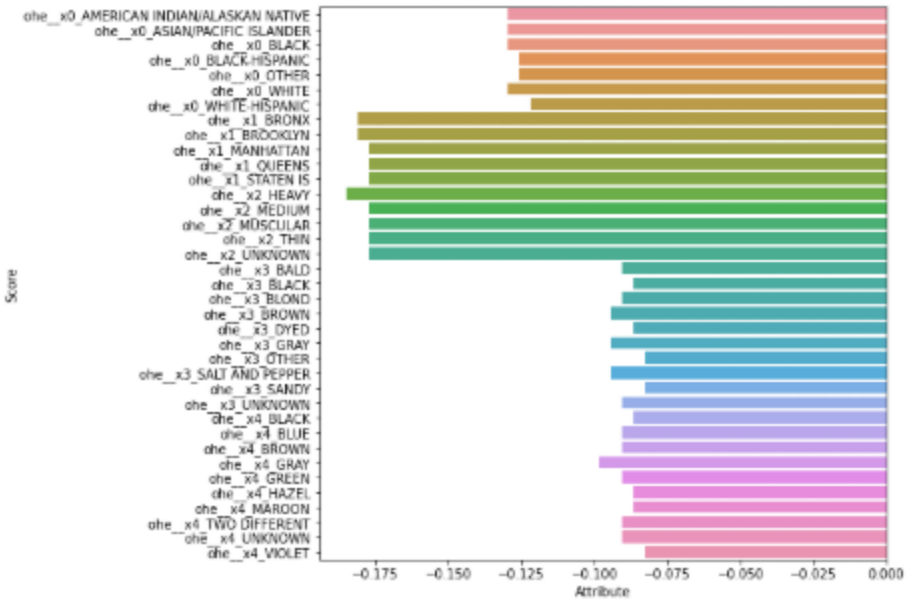

Figure 34. Results of the SGDClassifier

2) Discuss the advantages of each model for this classification task.

ExtraTreeClassifier is useful because it applies a number of randomized decision trees (extra trees) on sub samples of the dataset and uses averaging to improve accuracy and attempt to control over-fitting. This result gave me the highest accuracy at <96% and the highest f1 score. The advantage to this method is that it uses multiple different methods and apples an average over them. It is also useful for finding the most prevalent thing to look for to decide if someone is armed or not.

MultinomialNB strengths compared to regular or bernouli native bayes is that it
is tailored for classification with discrete features. In our case, the discrete features are the descriptions given by the police in the report. Usually this method is used for things like word counts for text classification. This also yielded us with a high accuracy score and strong f1 score as summarized above, with the hair colour of maroon being a high indicator of distinguishing if someone is armed or not.

SGDClassifier is a group of linear classifiers (like SVM, logistic regression, etc) that has stochastic gradient decent learning built in. This means that the gradient o floss is estimated in each sampling at a time and with that information, updating the model continually. It is effectively a better way of matching a line of best fit under convex loss functions. This method found another high accuracy and shows us that one of the most important factors when classifying if someone is armed is if their build type is heavy.

3) What are the most important variables found by each model?

ExtraTreeClassifier found that the individual being White was the most important thing when classifying is someone is carrying a weapon, followed by if the individual was black, or having blue eyes.

MultinomialNB found that a maroon hair colour was the most important, followed by violet and dyed hair colours.

SGDClassifier found that a heavy build was the best indicator, followed by several locations such as the Bronx, Queens, Staten Island, Manhattan, and Brooklyn.

Evaluation

1) How useful is your model for the police? How would you measure the model's value if it were used?

This model is quite useful as it gives an outline for common things found when someone was armed in the past. This means that they can understand what things to look for when completing SFQ's. The model's success would be defined by applying its findings to future SFQ statistics to see if its accuracy held up.

2) How would your implement your model to improve policing? What other data should be collected? How often would your model need to be updated?

Implementation of the models would be to apply its findings to precincts and inform police officers of the predictive highest indicators for seeing if someone is armed or not. This could give police officers an advantage or edge for identifying people who are a threat by carrying a weapon. The same data collected in our SFQ dataset should be collected as well as a simple Y/N column for seeing if the main findings in this report lead to identifying someone who is armed. Updates should be done periodically, likely quarterly as we saw seasonality having a large factor in the number of SFQ searches earlier in the report.

Conclusion:

In conclusion this dataset provided many powerful information about the historical data for New York's Stop Question and Frisk (SFQ) program. Using Visualizations, Association Rule Mining, Cluster Analysis, and Predictive Modeling we were able to distinguish patterns and outliers in our data, common attributes of SFQ's, Common locations and similarities, and predict information. Data and Visualization allowed us to understand simple statistics of our data like the average age, weight, and height. It also gave us information on what race was frisked the most, or what day or month more stops occurred. Association Rule Mining allowed us to look at all attributes in a standard SFQ and see frequently occurring traits like being White or African American, being a medium build, or having brown or black hair. Cluster analysis allowed for looking at specific crimes seen and where they were located on a map of New York. It also

allowed us to cluster similar instances and define where high rates of crime were. Predictive modelling allowed for predictions to be made based on the criteria of past SFQ results. The findings saw that being black or white had a high likelihood of a person carrying a weapon. It also saw that having an obscure hair colour like maroon or violet was a high indicator. All this data was in raw numbers and did not account for population. This means that it might be biased toward certain races as they would count for a high percentage of SFQs, but only account for a small percent of the population. Future findings should include public datasets of population and include weighting of race.