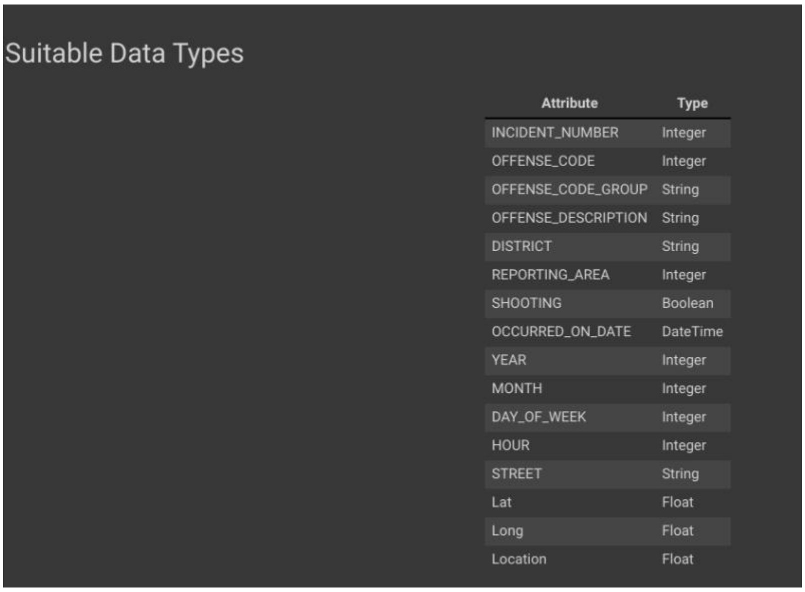
Deliverable 3 (Final Report)

This project revolved around using the concepts and techniques learnt in class to extract meaningful information from a large dataset. We were allotted the Boston crime analysis dataset which contained almost 560,000 entries. What was exciting about this dataset was that analysing it had a real impact because you could potentially reduce crime from happening if you found out significant correlations and trends from the data.

The first stage in this process was the **preprocessing** of the data and exploratory analysis. It is well known that in data science, preprocessing is perhaps the most time consuming task and it was no exception in our case as well. The next few paragraphs will highlight exactly how we achieved this task.

Changing types of columns

The first part was changing the types of columns. The columns in the dataset had types that were not ideal for us to use in our analysis. To fix this, we compiled a list in which suitable data types were determined.



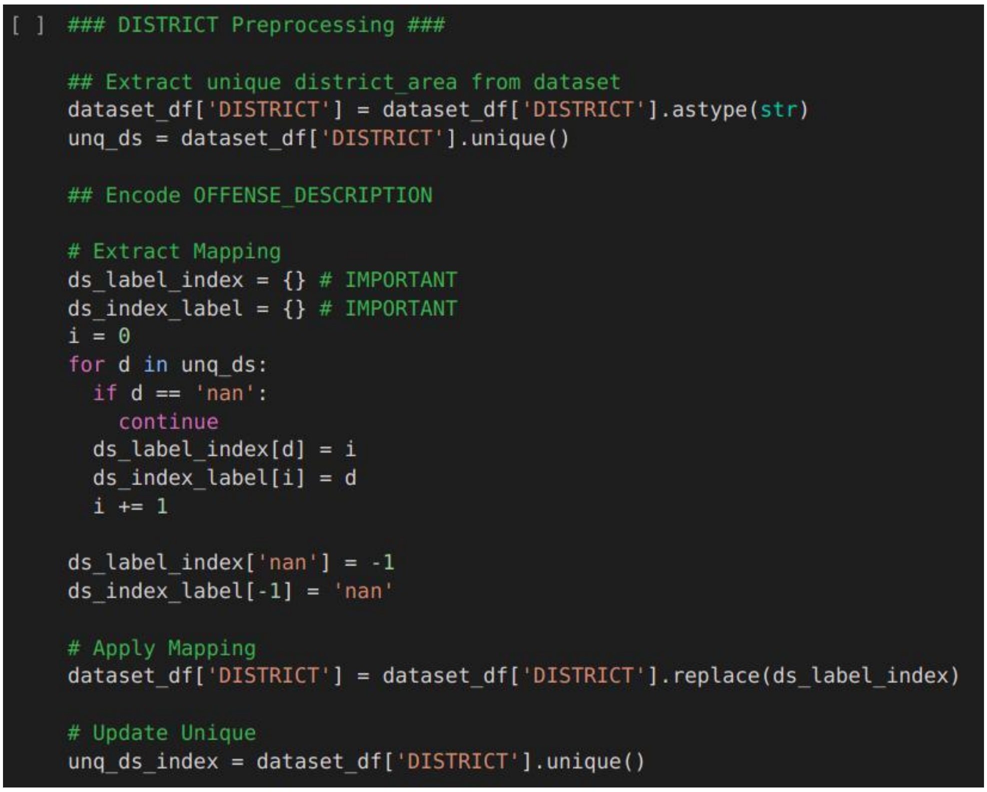
Suitable Data Types

Attribute	Type
INCIDENT_NUMBER	Integer
OFFENSE_CODE	Integer
OFFENSE_CODE_GROUP	String
OFFENSE_DESCRIPTION	String
DISTRICT	String
REPORTING_AREA	Integer
SHOOTING	Boolean
OCCURRED_ON_DATE	DateTime
YEAR	Integer
MONTH	Integer
DAY_OF_WEEK	Integer
HOUR	Integer
STREET	String
Lat	Float
Long	Float
Location	Float

The columns which needed to have their type changed were first converted to a string. Then some manipulation was done which revolved around replacing/removing certain characters which would've made complicated running algorithms in the future. For example, for the column of Incident Numbers, we had to remove the leading "|" if it was present and replace it with an empty string " ". Then it was converted to int. Similarly for reporting area, we labelled the missing values as -1 and then converted them to int. For shooting column, we decided to convert the values into boolean by replacing "Y" with "1" and "Nan" with "0". These are just some of the examples of how we changed the types of columns. Columns of Lat and Long were also dealt with in this step because they required a simple .fillna() and a .replace() method to achieve the desired result (which was converting Nan and -1 to 0).

Encoding

Before we could fill the missing values in the data, we decided to encode certain columns in order to make things easier. The general approach was the same for all columns starting from extracting unique values, creating a mapping between those values, assigning -1 to nan values and finally applying the mapping on the column. The following piece of code highlights the skeleton code we used for this stage with only minor changes for each attribute.



```
### DISTRICT Preprocessing ###

## Extract unique district_area from dataset
dataset_df['DISTRICT'] = dataset_df['DISTRICT'].astype(str)
unq_ds = dataset_df['DISTRICT'].unique()

## Encode OFFENSE_DESCRIPTION

# Extract Mapping
ds_label_index = {} # IMPORTANT
ds_index_label = {} # IMPORTANT
i = 0
for d in unq_ds:
  if d == 'nan':
    continue
  ds_label_index[d] = i
  ds_index_label[i] = d
  i += 1

ds_label_index['nan'] = -1
ds_index_label[-1] = 'nan'

# Apply Mapping
dataset_df['DISTRICT'] = dataset_df['DISTRICT'].replace(ds_label_index)

# Update Unique
unq_ds_index = dataset_df['DISTRICT'].unique()
```

More details can be found in the notebook as each mapping is broken down into sub-headings so that they are easy to follow and understand.

FIlling Missing Values

Incident Number and Offense Code did not have any missing values.

For Offense Code Group missing values, we first created a 1-Many mapping between ‘Offense Code Group’ and ‘Offense Code’. Then we created a mapping between ‘Offense Code’ and ‘Offense Code Group Since every ‘Offense Code’ had only 1 matching entry for ‘Offense Code Group’ -> This showed that ‘Offense Codes’ are a subset of respective ‘Offense Code Group’ that do not intersect. Using this mapping, we were able to fill missing ‘Offense Code Group’ values based on corresponding ‘Offense Code’. 203,806 missing values were filled but 10,565 missing values remained. Another 1 to 1 mapping was created between ‘Offense Description’ and ‘Offense Code’, however using this mapping no missing values were filled.

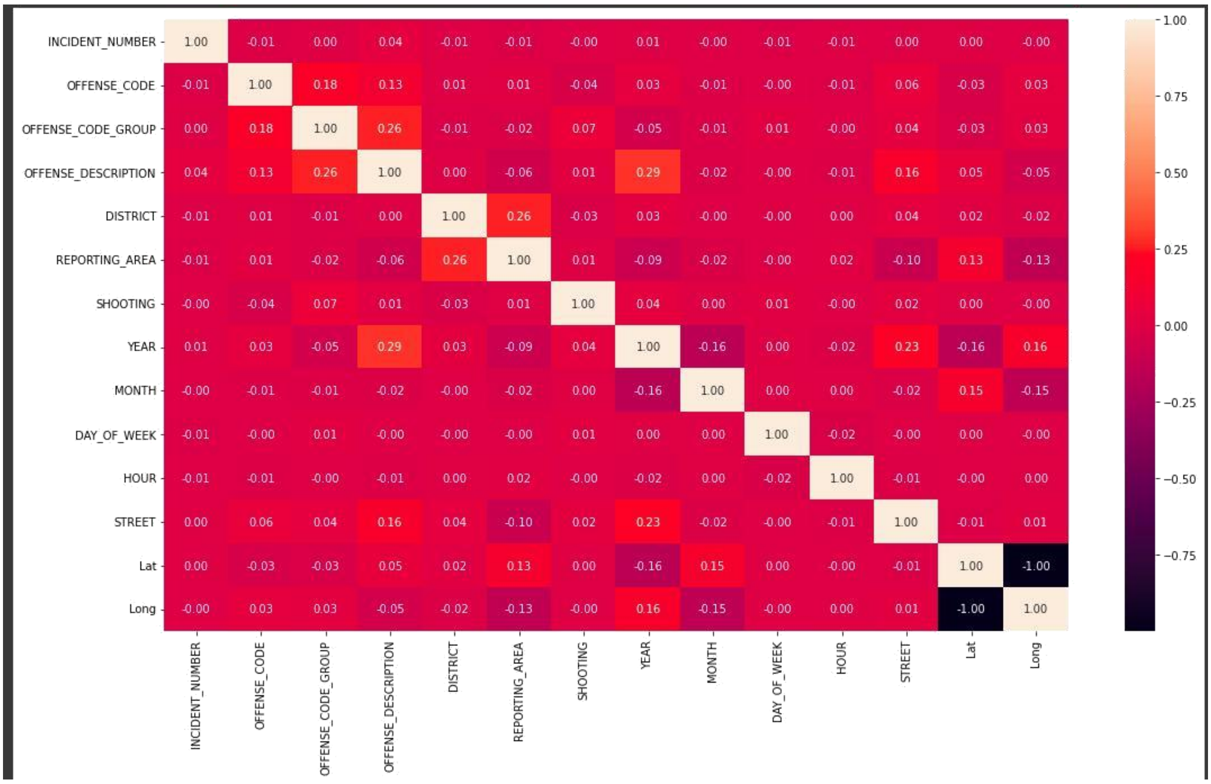
There were no missing values for ‘Offense Description’, therefore only encoding was done for them and no mapping was done.

For District Missing values, we first found the unique areas present in our dataset (by creating ‘District’-’Reporting Area’ pairs). Then we proceeded to create average Lat and Long Maps for said pairs. After this was done, we identified and removed Nan Values from the Lat and Long dictionaries we created in the above part. Then we extracted Area, Lat and Long for Nan Districts and stored these separately. Finally, we replace the Nan Districts with the shortest Area, Lat and Long distance. Thus a lot of district values were filled as accurately as possible, however, not all were filled

We applied the same approach as the above one for filling missing ‘Street’ values (by creating ‘District’-’Street’ pairs and computing average Lat & Long for them). But the computation time was too much, therefore the missing values were not filled.

Correlation Between Attributes

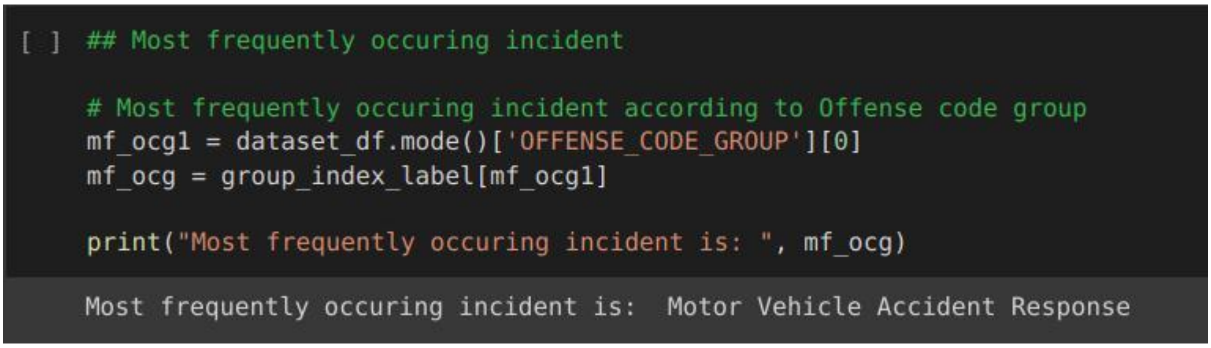
We used the built in library to draw the correlation matrix which is attached below. Most of the values were around 0 which means there is no significant linear correlation. There were however some attributes with correlation values higher than 0.10. Offense Description had a value of 0.29 with Year which means there is some indication towards a positive linear correlation. Similarly Year and Street seem to have a correlation value of 0.23 which at first glance does not make any sense because the increase in the year should not have any effect on the street number.



	INCIDENT_NUMBER	OFFENSE_CODE	OFFENSE_CODE_GROUP	OFFENSE_DESCRIPTION	DISTRICT	REPORTING_AREA	SHOOTING	YEAR	MONTH	DAY_OF_WEEK	HOUR	STREET	Lat	Long
INCIDENT_NUMBER	1.00	-0.01	0.00	0.04	-0.01	-0.01	-0.00	0.01	-0.00	-0.01	-0.01	0.00	0.00	-0.00
OFFENSE_CODE	-0.01	1.00	0.18	0.13	0.01	0.01	-0.04	0.03	-0.01	-0.00	-0.01	0.06	-0.03	0.03
OFFENSE_CODE_GROUP	0.00	0.18	1.00	0.26	-0.01	-0.02	0.07	-0.05	-0.01	0.01	-0.00	0.04	-0.03	0.03
OFFENSE_DESCRIPTION	0.04	0.13	0.26	1.00	0.00	-0.06	0.01	0.29	-0.02	-0.00	-0.01	0.16	0.05	-0.05
DISTRICT	-0.01	0.01	-0.01	0.00	1.00	0.26	-0.03	0.03	-0.00	-0.00	0.00	0.04	0.02	-0.02
REPORTING_AREA	-0.01	0.01	-0.02	-0.06	0.26	1.00	0.01	-0.09	-0.02	-0.00	0.02	-0.10	0.13	-0.13
SHOOTING	-0.00	-0.04	0.07	0.01	-0.03	0.01	1.00	0.04	0.00	0.01	-0.00	0.02	0.00	-0.00
YEAR	0.01	0.03	-0.05	0.29	0.03	-0.09	0.04	1.00	-0.16	0.00	-0.02	0.23	-0.16	0.16
MONTH	-0.00	-0.01	-0.01	-0.02	-0.00	-0.02	0.00	-0.16	1.00	0.00	0.00	-0.02	0.15	-0.15
DAY_OF_WEEK	-0.01	-0.00	0.01	-0.00	-0.00	-0.00	0.01	0.00	0.00	1.00	-0.02	-0.00	0.00	-0.00
HOUR	-0.01	-0.01	-0.00	-0.01	0.00	0.02	-0.00	-0.02	0.00	-0.02	1.00	-0.01	-0.00	0.00
STREET	0.00	0.06	0.04	0.16	0.04	-0.10	0.02	0.23	-0.02	-0.00	-0.01	1.00	-0.01	0.01
Lat	0.00	-0.03	-0.03	0.05	0.02	0.13	0.00	-0.16	0.15	0.00	-0.00	-0.01	1.00	-1.00
Long	-0.00	0.03	0.03	-0.05	-0.02	-0.13	-0.00	0.16	-0.15	-0.00	0.00	0.01	-1.00	1.00

Most Frequently Occuring Incident

The most frequently occuring incident is “Motor Vehicle Accident Response” as seen from the snippet below. This is a vital piece of information to the police department because using this piece of code (with some slight modifications), you can also find a list of the top 5 or top 10 most frequently occurring incidents. Such information could help better organise and concentrate the resources of the police department into preventing these crimes.



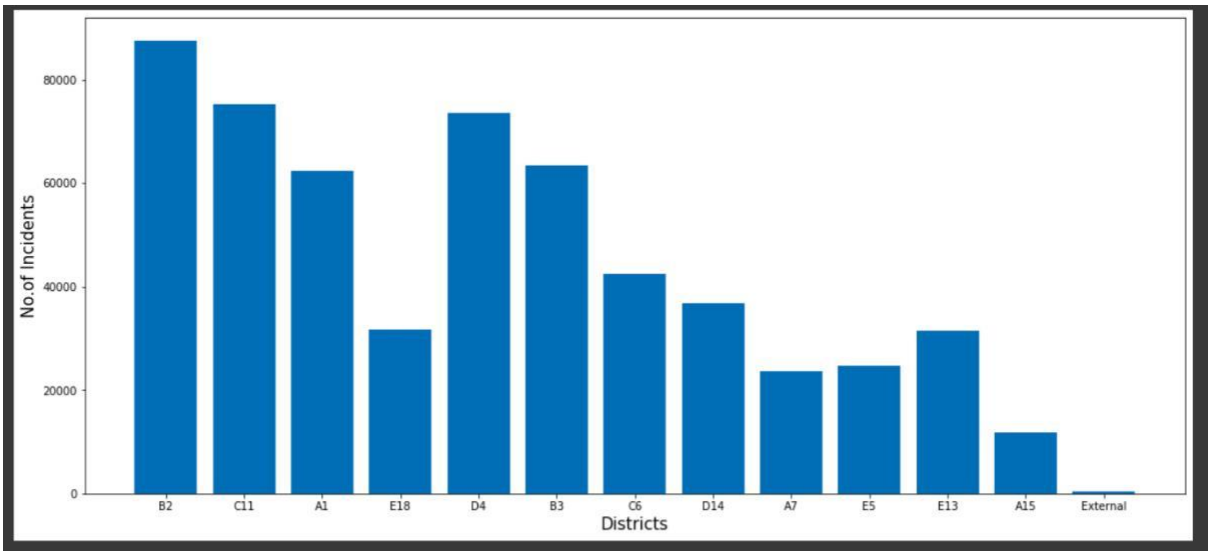
```
[ ]  ## Most frequently occuring incident

     # Most frequently occuring incident according to Offense code group
     mf_ocg1 = dataset_df.mode()['OFFENSE_CODE_GROUP'][0]
     mf_ocg = group_index_label[mf_ocg1]

     print("Most frequently occuring incident is: ", mf_ocg)
```

```
     Most frequently occuring incident is:  Motor Vehicle Accident Response
```

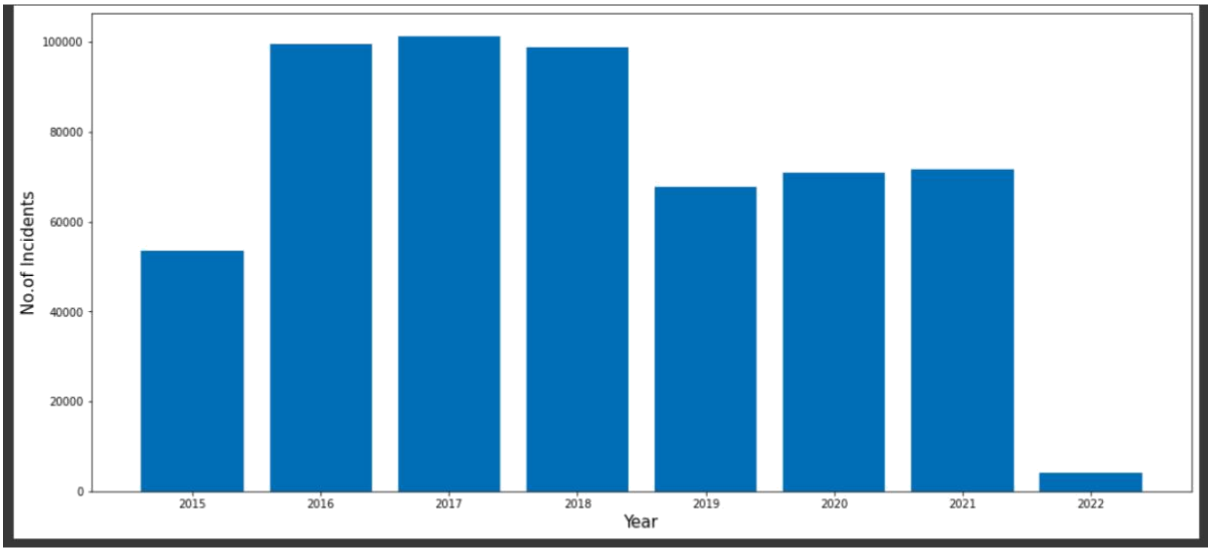
Bar Graphs

The code contains 3 Bar Charts each representing a crucial piece of information which will definitely prove to be helpful in subsequent analysis and tasks,

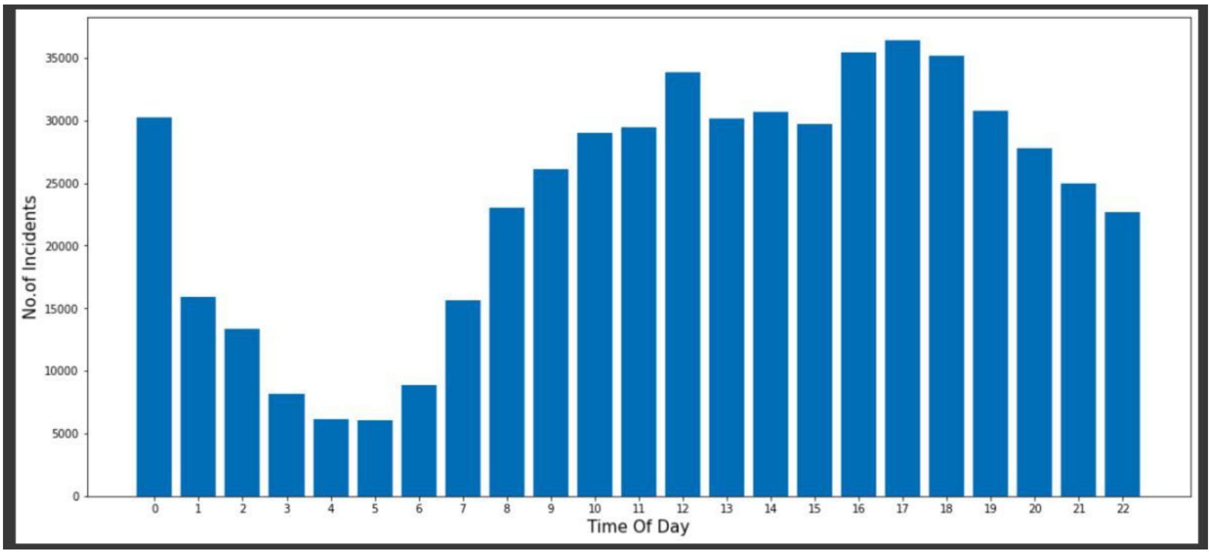
1- District vs Incidents Chart: This bar chart tells the incidents that happen in each district. District B2 has the highest number of incidents whereas A15 has the lowest. This is very helpful for the police department because they now know for a fact which districts have the highest concentration of crime incidents and they would need to take appropriate measures to reduce this number. If the police do not have this information and they deploy resources to sector E18 for example, then that would be a huge tactical error because this district has one of the lowest rate of crimes in the city. Such information is vital for policy and strategy making because it is based on data and therefore facts instead of opinions or any bias.



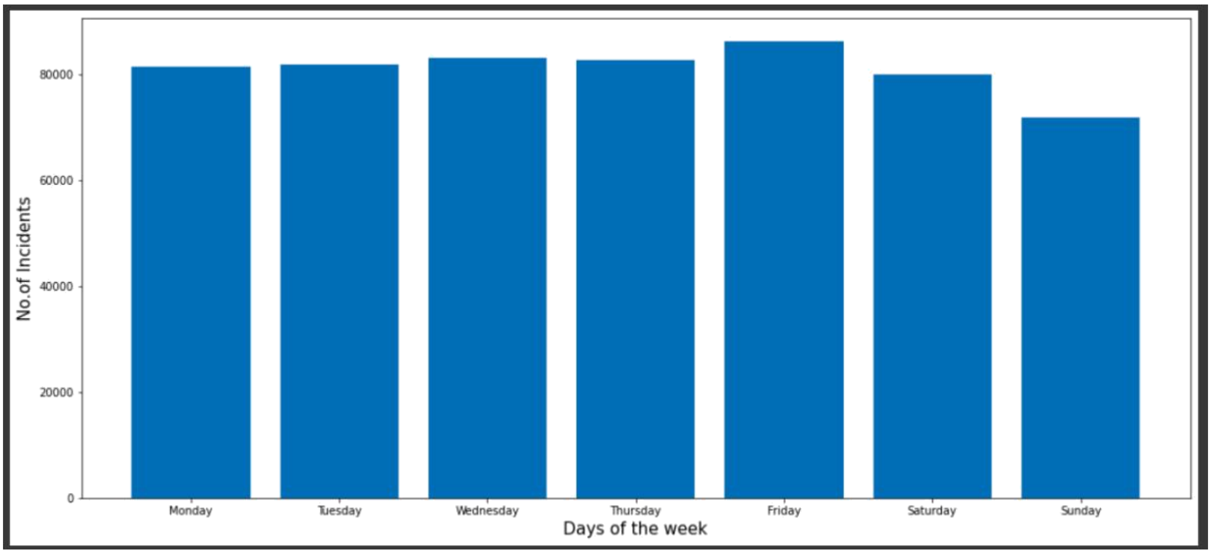
2- Year vs Incidents Chart: Years 2016-2018 had the highest incident rate in the dataset. There was a decrease in 2019 but the number of incidents was still more than 2015 which shows an overall increase in the trend which is proof that the policies and strategies of the police department have had a positive effect.



3- Time Of Day vs Incidents: The incidents are at a lowest between the hours 03-05 and then start to increase. The highest number of incidents occur between hour 16 and hour 18. This graph can help the police shuffle their forces according to the time of day in which crime is most prevalent. The dataset has a large number of entries which have been collected over multiple years. So the chances of these trends being affected by outliers or a bias is fairly low. Therefore, it can be used to predict on which hours of the day, crime is most likely to happen.

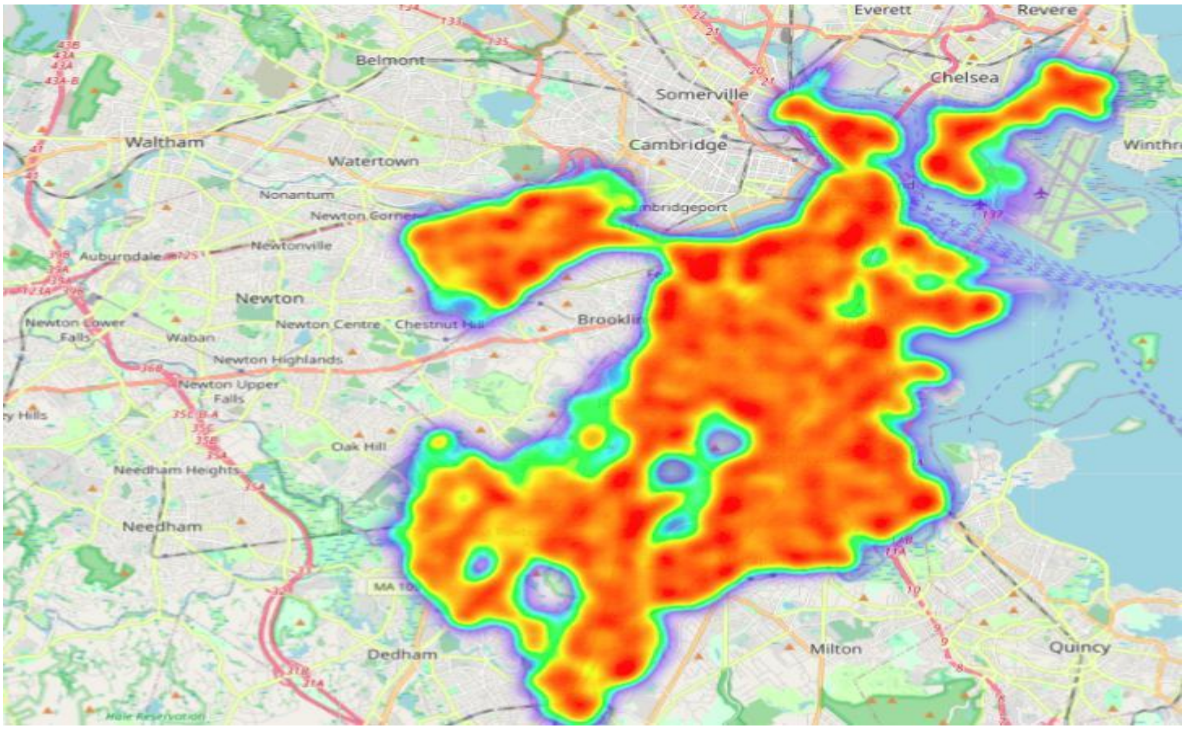


4- Day of the week vs Incidents: Almost all days have the same number of incidents with Friday having the highest (very slight lead). Sunday has the lowest incident rate. This piece of information unfortunately does not help the police department in any significant way because almost all the days apart from sunday have the same number of incidents reported.

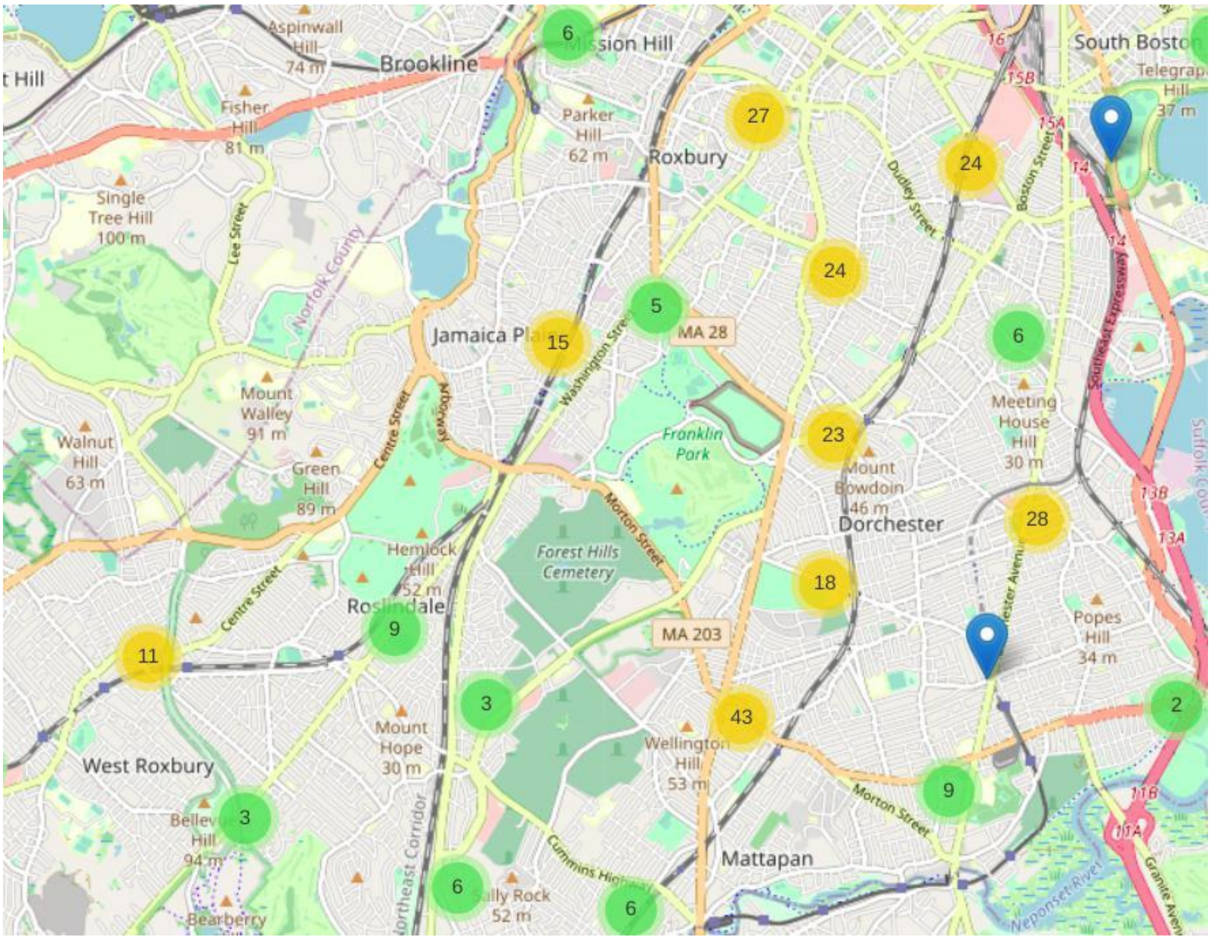


Heat Maps To Visualise Dataset Findings

These graphs are perhaps the most important and valuable for the department because they present the data on a map of the city. The following graph (when zoomed in) would tell you the areas in which most of the crime takes place. As you can see, the majority of the area is in red which might be an alarming sign because crime seems to be concentrated to all parts of the city. There are, however a few green spots in it as well which might prove to be useful while deploying resources or anticipating crime.



This second graph is even more useful because it highlights the number and type of incidents that happened in the city on their exact coordinates. Police can benefit a great degree from this because they essentially have a solid idea about where crime is most likely to happen (by where I mean they can narrow it down to the very street itself). These heatmaps provide a good reference to police as well as citizens who can be more aware of some areas or streets in the city which have more crime than the others. There are definitely hotspots that can be identified and seen in this graph. If you zoom in further (in the notebook), it breaks the data down into even more detail thus ensuring you know exactly where each of the crime that has happened in the last few years took place and what was the crime exactly.



After preprocessing the data and performing district wise analysis of the incidents, we performed cluster analysis and frequent pattern mining on the data to find a conclusive pattern in occurrence of various crimes/offenses, their frequency of occurrence, their correlation with other factors such as day of the week, hour etc to conclusively help us tackle crime in a better way.

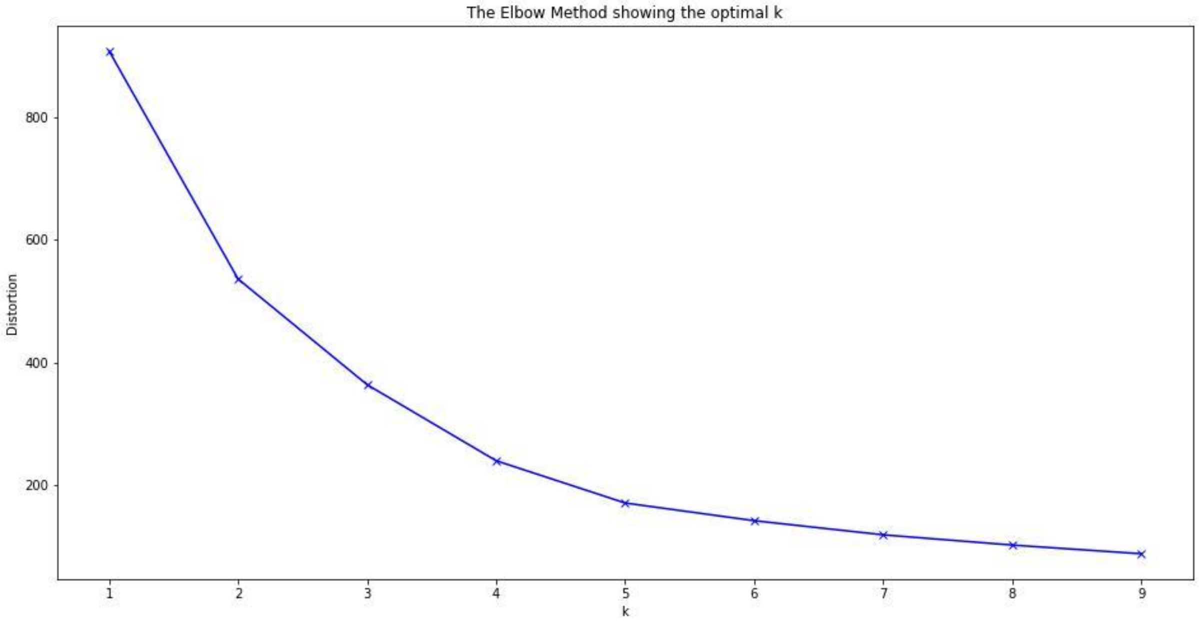
Clustering:

The clustering approach we deployed was k means cluster analysis. The k means clustering algorithm partitions the data into k clusters. This provides an observational point of view to analyze the clusters. The reason for using “K means” over let's suppose Hierarchical clustering is that K means can handle big data quite well. Our dataset in this project contains almost 560k entries so we would ideally want an algorithm that runs in linear time (such as K means) instead of one that is quadratic (Hierarchical clustering).

The Elbow Method for Latitude and longitude:

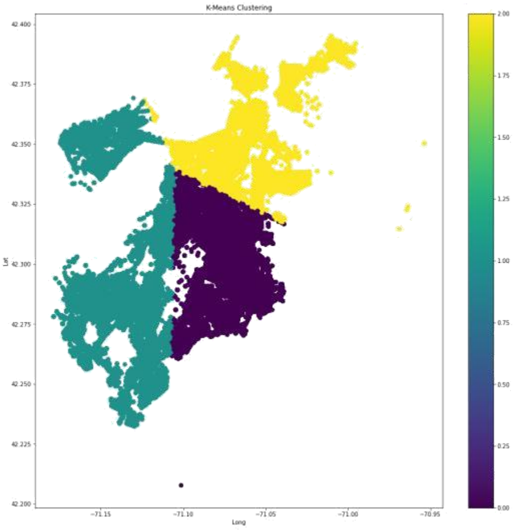
The elbow method is used to define the number of clusters that need to be formed for optimal division of n observations.

The elbow method has shown the optimal value of k is 3, that is the number of optimal clusters is 3 because if you see the graph below, the pattern of the graph becomes linearly decreasing at the elbow point of ‘3’.



Clustering based on latitude and longitude:

We ran clustering analysis on the two columns latitude and longitude. The cluster map that was obtained from it had 3 cluster groups. However, the clustering did not signify any importance in relation to offense data.

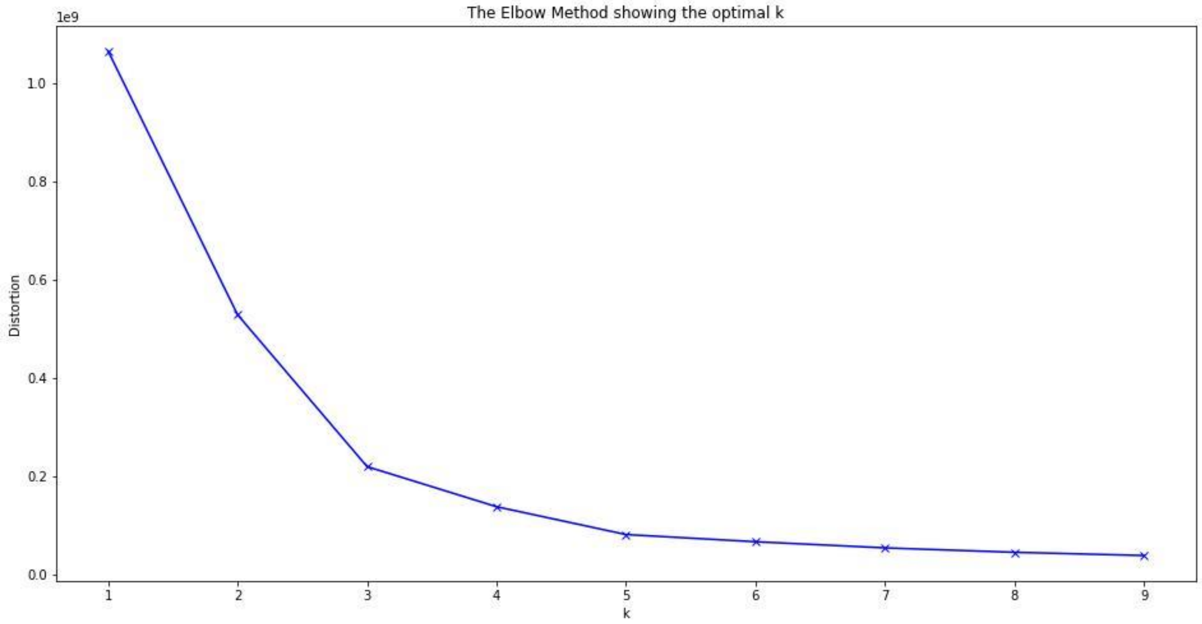


The clusters of color green, purple, and yellow signify the closeness of coordinates in relation to each other which does not lead to any meaningful inferences.

The Elbow Method for OFFENSE_CODE and OFFENSE_CODE_GROUP:

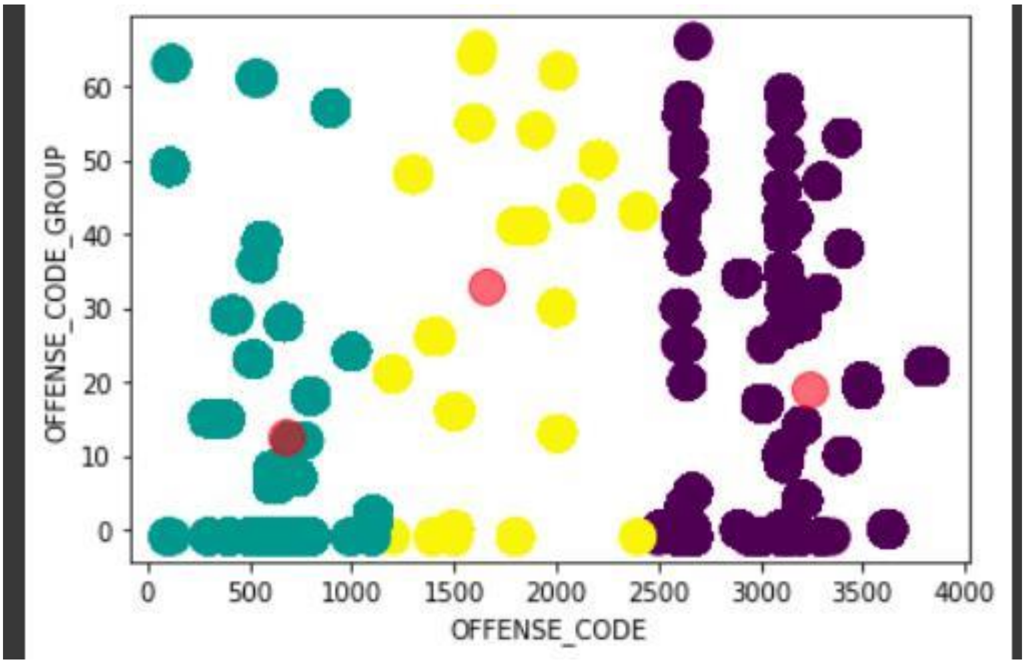
The elbow method is used to define the number of clusters that need to be formed for optimal division of n observations.

The elbow method has shown the optimal value of k is 3, that is the number of optimal clusters is 3 because if you see the graph below, the pattern of the graph becomes linearly decreasing at the elbow point of ‘3’.



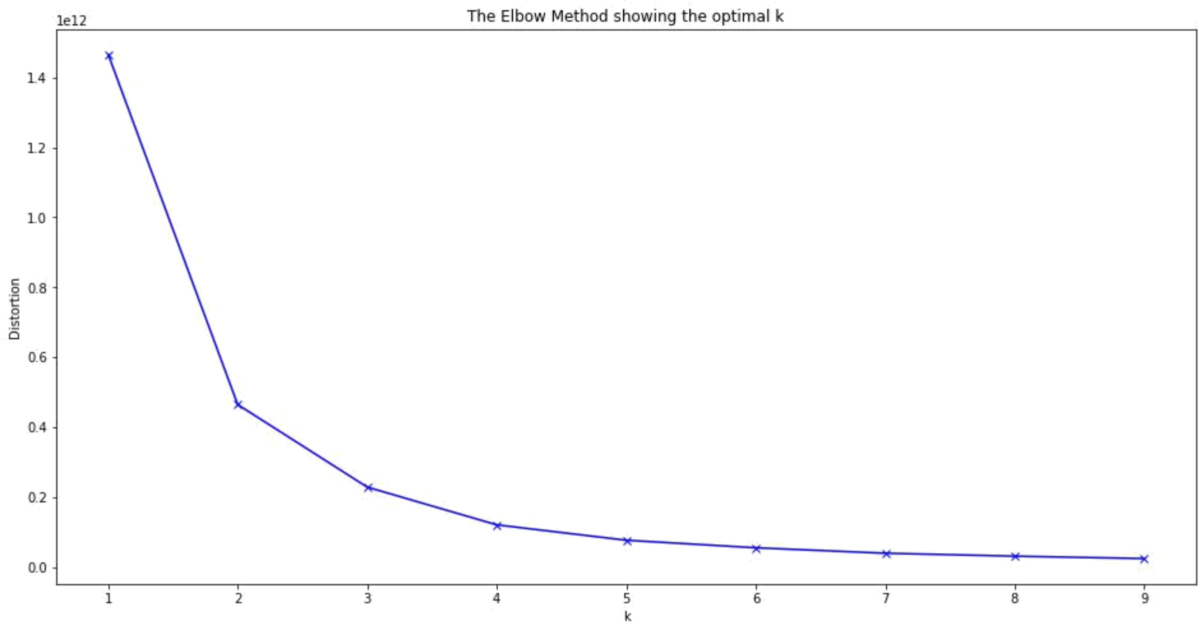
Scatter Plot of OFFENSE_CODE and OFFENSE_CODE_GROUP:

The scatter plot is quite inconclusive. Although the k means clustering has identified the centre point for the 3 clusters, the overall scatter plot does not lead to the inference of any significant result.



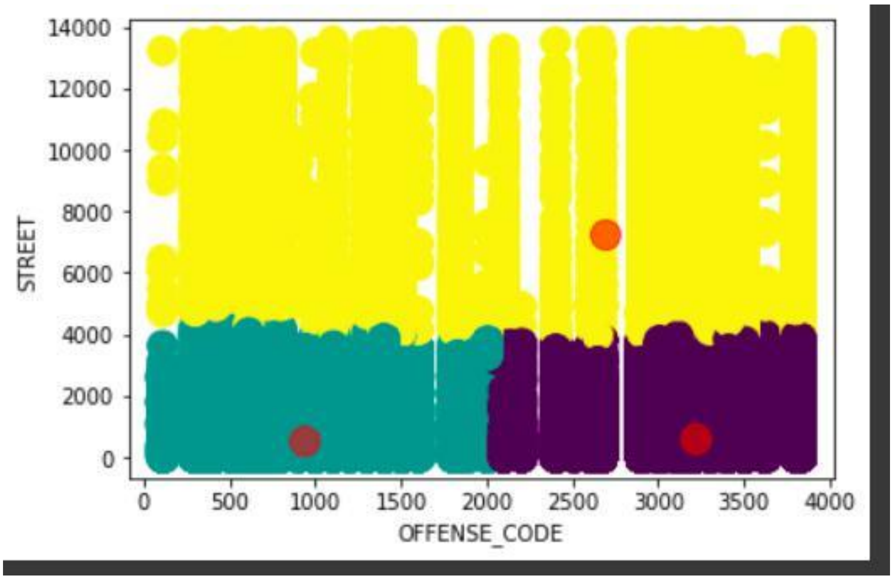
The Elbow Method for OFFENSE_CODE and STREET:

The elbow method has shown the optimal value of k is 3, that is the number of optimal clusters is 3 because if you see the graph below, the pattern of the graph becomes linearly decreasing at the elbow point of ‘3’.



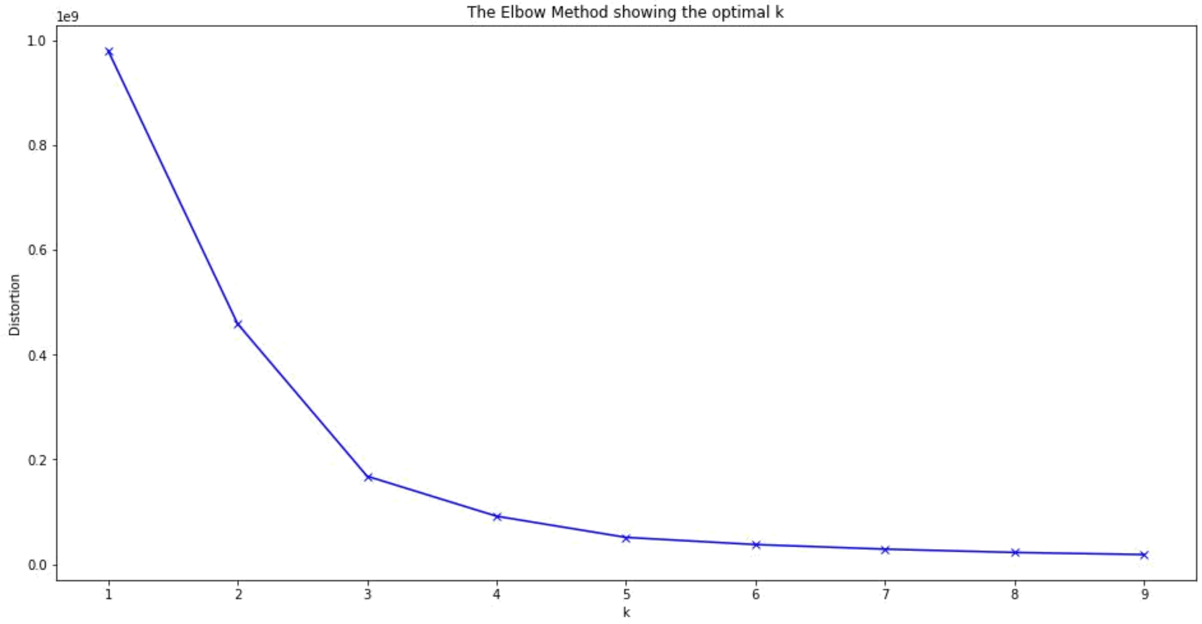
Scatter Plot of OFFENSE_CODE and STREET:

Once again, the scatter plot is quite inconclusive. Although the k means clustering has identified the center point for the 3 clusters, the overall scatter plot does not lead to the inference of any significant result.



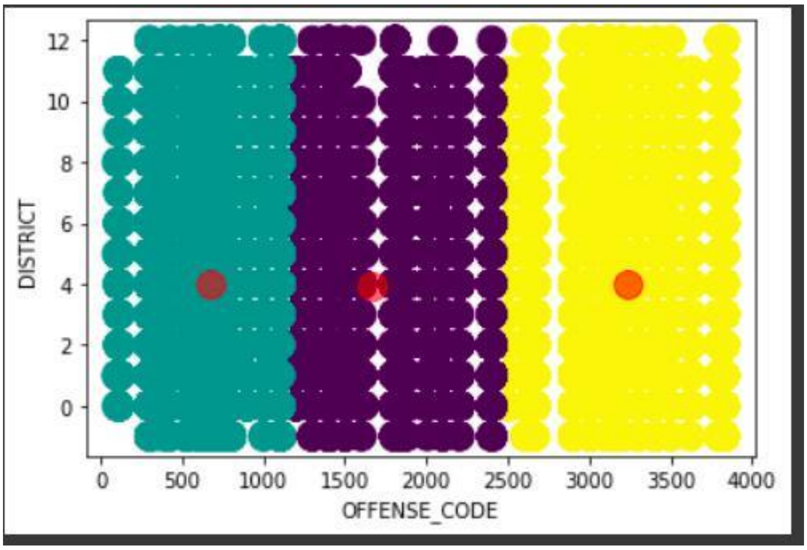
The Elbow Method for OFFENSE_CODE and DISTRICT:

The elbow method has shown the optimal value of k is 3, that is the number of optimal clusters is 3 because if you see the graph below, the pattern of the graph becomes linearly decreasing at the elbow point of ‘3’.



Scatter Plot of OFFENSE_CODE and DISTRICT:

Once again, the scatter plot is quite inconclusive. Although the k means clustering has identified the center point for the 3 clusters, the overall scatter plot does not lead to the inference of any significant result.



Note: We consulted the correlation matrix generated in the previous part and ran those attributes which had a decent enough correlation with each other. This correlation however, is not reflected when we run k-means clustering as evident from the graphs attached above. We will use the same attributes and apply frequent pattern mining to them in order to extract any meaningful and relevant patterns.

After performing k means clustering algorithm on **OFFENSE_CODE and DISTRICT, OFFENSE_CODE and STREET, OFFENSE_CODE and OFFENSE_CODE_GROUP, and Latitude and longitude,** the results were inconclusive. To reiterate our analysis in part 2, we have re attached the scatter plots to clearly show that the scatter plots did not visually represent any significant information for crime analysis.

Frequent Pattern Mining:

Since clustering did not yield any beneficial information for curbing various crimes, we used the fp growth algorithm to find association of crimes with different attributes of the dataset. The correlation matrix was valuable in finding frequent patterns because we used those attributes which had moderately high correlation amongst each other. **DAY_OF_WEEK & SHOOTING, YEAR & OFFENSE_DESCRIPTION** showed which items had high frequency of occurrence.

First of all we had to encode the data. This is a very important prerequisite to apply the fp growth algorithm via the mlxtend library.

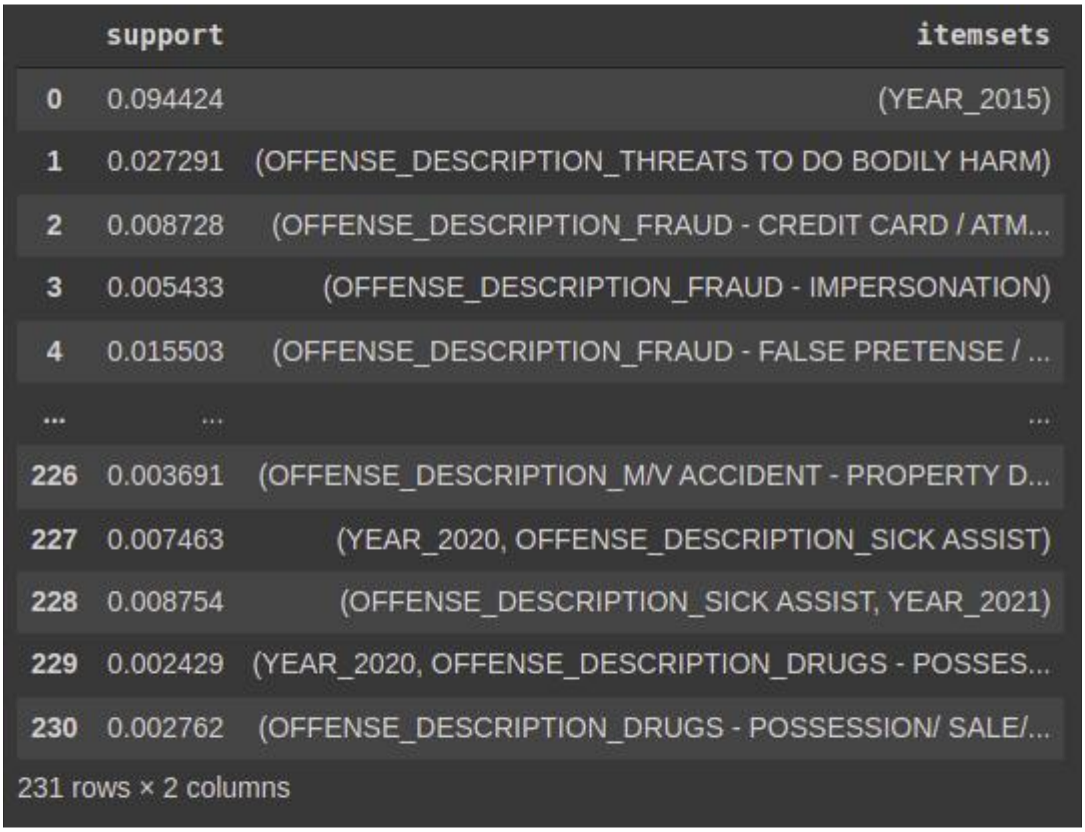
- Previously the columns OFFENSE_CODE_GROUP, OFFENSE_DESCRIPTION, DISTRICT, STREET and DAY_OF_WEEK were encoded by integer values starting from 0 uptill (n-1)th unique value.
- We applied this encoding on the remaining columns OFFENSE_CODE, REPORTING_AREA, SHOOTING, YEAR, MONTH and HOUR.
- Note that each column contains -1 integer value representing nan.
- The combinedEncoder function essentially deep copies the dataset (so that mutating it won't change the original dataset) and then applies combined encoding over the passed columns. The following example explains this:
    - Let's say there are 2 columns A and B, with unique values (0, 1, 2) and (0, 1) respectively.
    - The mappings for these integral values are as follows:
        - A = {0: ‘a0’, 1: ‘a1’, 2: ‘a2’}
        - B = {0: ‘b0’, 1: ‘b1’}
    - The dataset is as follows D = [[0, 0], [1, 0], [2, 1]]
    - After the combineEncoding function runs, it returns D = [[0, 3], [1, 3], [2, 4]], meaning that the mapping is now altered to:
        - A = {0: ‘a0’, 1: ‘a1’, 2: ‘a2’}
        - **B = {3: ‘b0’, 4: ‘b1’}**
    - And also returns the mapping in the format: [‘A_a0’, ‘A_a1’, ‘A_a2’, ‘B_b0’, ‘B_b1’,]
- Hence, we feed the numpy array of the returned dataset and the corresponding mapping array to the transcoder therefore generating a one-hot encoded dataset.

- Thus, when the fp growth algorithm runs, it takes the column names that essentially store the column_value strings rather than simple mapping.

Note: In the fp growth algorithm, we are setting the minimum threshold for various columns on which we run the fp growth to find a decent amount of frequent itemsets so that we can draw a reasonable conclusion.

FP GROWTH ON YEAR & OFFENSE_DESCRIPTION:

The fp growth algorithm has given us a list that shows in which year what type of crime which most frequently occurred. For example, in Year 1 the most occurred crime was harassment. Similarly, more patterns related to year and offense type have been mined via fp growth.

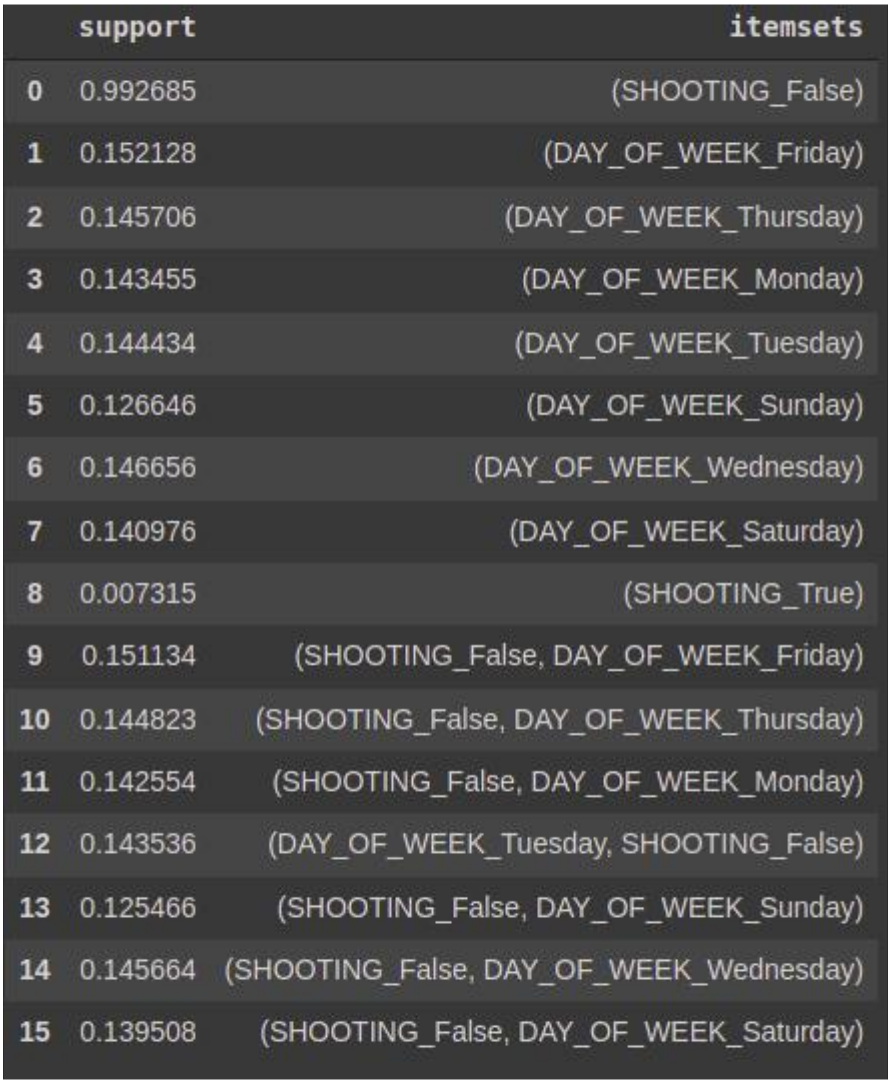


	support	itemsets
0	0.094424	(YEAR_2015)
1	0.027291	(OFFENSE_DESCRIPTION_THREATS TO DO BODILY HARM)
2	0.008728	(OFFENSE_DESCRIPTION_FRAUD - CREDIT CARD / ATM...
3	0.005433	(OFFENSE_DESCRIPTION_FRAUD - IMPERSONATION)
4	0.015503	(OFFENSE_DESCRIPTION_FRAUD - FALSE PRETENSE / ...
...	...	...
226	0.003691	(OFFENSE_DESCRIPTION_M/V ACCIDENT - PROPERTY D...
227	0.007463	(YEAR_2020, OFFENSE_DESCRIPTION_SICK ASSIST)
228	0.008754	(OFFENSE_DESCRIPTION_SICK ASSIST, YEAR_2021)
229	0.002429	(YEAR_2020, OFFENSE_DESCRIPTION_DRUGS - POSSES...
230	0.002762	(OFFENSE_DESCRIPTION_DRUGS - POSSESSION/ SALE/...

231 rows × 2 columns

FP Growth on DAY_OF_WEEK & SHOOTING:

After selecting multiple values of minimum support for pattern mining between these two columns, the results were quite in line with our previous work. This is because the occurrence of shooting on different days of the week were totally negligible and the minimum support had to be really small to show areas where shootings had occurred but since the value of the minimum support had to be quite small the relation would not matter. This, in our opinion, highlights that shooting had a very weak correlation which can also be verified via the correlation matrix.



	support	itemsets
0	0.992685	(SHOOTING_False)
1	0.152128	(DAY_OF_WEEK_Friday)
2	0.145706	(DAY_OF_WEEK_Thursday)
3	0.143455	(DAY_OF_WEEK_Monday)
4	0.144434	(DAY_OF_WEEK_Tuesday)
5	0.126646	(DAY_OF_WEEK_Sunday)
6	0.146656	(DAY_OF_WEEK_Wednesday)
7	0.140976	(DAY_OF_WEEK_Saturday)
8	0.007315	(SHOOTING_True)
9	0.151134	(SHOOTING_False, DAY_OF_WEEK_Friday)
10	0.144823	(SHOOTING_False, DAY_OF_WEEK_Thursday)
11	0.142554	(SHOOTING_False, DAY_OF_WEEK_Monday)
12	0.143536	(DAY_OF_WEEK_Tuesday, SHOOTING_False)
13	0.125466	(SHOOTING_False, DAY_OF_WEEK_Sunday)
14	0.145664	(SHOOTING_False, DAY_OF_WEEK_Wednesday)
15	0.139508	(SHOOTING_False, DAY_OF_WEEK_Saturday)

After selecting multiple values of minimum support for pattern mining between these two columns, the results were quite in line with our previous work. This is because the occurrence of shooting on different days of the week were totally negligible and the minimum support had to be really small to show areas where shootings had occurred but since the value of the minimum support had to be quite small the relation would not matter. This, in our opinion, highlights that shooting had a very weak correlation which can also be verified via the correlation matrix.

Significance to the Police Department:

The Boston Police Department can use this data to find the exact time and place when and where the most crimes occur. By narrowing down the exact time of occurrence and place, the Police Department can divert their time and resources to these neglected areas to curb crime in a more efficient manner. The results are valuable in conjunction to the “Crime Pattern Theory” which states that a crime occurs in a particular area because there is an opportunity for the crime to occur due to the awareness of the law breaker regarding that space. The “Crime Pattern Theory” is solidified when it is backed up by the results of fp growth. Similarly, the Police Department can use the data to divide the respective police departments for the crimes that occur in different areas of Boston for a better control on the crimes.

What more can be done:

For better statistical data analysis and mining, the collection of more **correct** data is valuable. A better data extraction and pattern mining will be possible due to the larger dataset. If more crime patterns are found, the police department will be able to catch the offenders quicker and map out the crimes in different locations with greater accuracy. Additionally, the Isolation Forest algorithm could also have been used on the dataset to remove anomalies(outliers) in each attribute so that the data would have the minimum amount of bias.

Conclusion:

The techniques taught and implemented on the Boston Crime Dataset has shown the undeniable importance of Data Mining. The techniques of clustering and frequent pattern mining provide valuable insight on the occurrence of crimes and enables authorities to fight and resolve these issues promptly. Data mining on crime can be expanded in more cities to provide people with a higher level of security.