Springboard--DSC

Capstone Project 1

Investigation of San Francisco Crime Data 2003-2019

By Garrett Yamane

November, 2019

Table of Contents

1. Introduction

San Francisco is a beautiful city that attracts a wide variety of visitors and residents. From Mission District to Chinatown, San Francisco is an amazing city full of different cultures. With a population of 880,000 and an area of 46.87 mi^2, San Francisco's density brings together quite a unique network of people each day. Safety is important for anywhere you travel or live, and being aware of crime and potential risks is always beneficial for one's own safety.

2. Problem Statement

San Francisco crime and safety are concerns for anyone wanting to travel, visit, or live in the city. These concerns highlight the importance of staying educated regarding police reports that have occurred in the past. Are certain areas more dangerous than others? At what times of the day do more crimes tend to occur, and do additional factors such as the time of the year (Christmas, Black Friday, etc.) influence the risk of crime? Answers to these types of questions can help people in San Francisco keep safe and aware of the potential dangers that could pose a direct threat to them. Through observing past crime data and police reports, I can study key features and suggest patterns that contribute to common crime in San Francisco. I will use my findings to model San Francisco crime data in order to decipher important factors that contribute to San Francisco crime patterns and ideally predict future potential crime location/classification. The City of San Francisco could use this to better prepare for certain events when crime rates tend to spike and take appropriate precautionary measures and target areas that are unsafe to keep visitors and residents protected from harm.

3. Target Client

The San Francisco Police Department has to be prepared at all times for any potential report of crime. They must act accordingly and it is essential that they are readily on duty at peak crime times and areas. The more informed they are about the patterns and conditions that trigger spikes in crime rates, the better it will be for both the Police Department and public. My client (SF Police Department) can use the models I build to better understand patterns in past crimes and understand conditions that may lead to more in the future. They can prepare for certain events and be aware of certain conditions that tend to lead to higher rates of crime, making sure that police units are readily available and onsite.

4. Data Wrangling

The San Francisco Government keeps records of police reports filed hosted at https://datasf.org/, dating from 2003-present. For instance, from 2008-2013, "the district attorney's domestic violence team dropped about 72 percent of all domestic violence criminal cases before they reached court" (https://sfpublicpress.org/news/2013-06/domestic-violence-case-that-spurred-san-francisco-reforms-comes-to-a-close). Stacks of hardcopy police report files made record-keeping difficult and cases like these could easily fall through. The San Francisco Government took steps toward improving their access to important records and made moves toward updating to the new Crime Data Warehouse data system, which made accessibility easy for its users.

a) Gathering the Data

Until May 2018, San Francisco crime data was maintained through CABLE, the legacy mainframe used to store their crime reports. However, CABLE was extremely prone to issues related to delays and data accessibility and was discontinued. At the start of 2018, the San Francisco Police Department announced its release of the newly updated Crime Data Warehouse that clearly organizes SF crime data and is quick and responsive.

Because of this update, my dataset contains a union of two datasets: one from CABLE police reports dating from 2003-2018 and the other from the Crime Data Warehouse police reports from 2018-2019. The links to these data sources follow.

- CABLE mainframe 2003-2018 police reports
- Crime Data Warehouse 2018-2019 police reports

b) Consolidating the Data

Prior to merging the two datasets, I wanted to prepare each of the datasets to make merging as easy as possible. I renamed columns from the CABLE mainframe to match their corresponding column names in the Crime Data Warehouse, ordered each crime report by increasing date, and reformatted the joint columns to have the same value format to keep consistency.

There was also some overlap with reports in 2018, so in order to keep all 2018 reports consistent I took 2003-2017 reports from CABLE and 2018-2019 reports from the Crime Data Warehouse to create a merged dataframe. There were some columns that were unique to each of the datasets, leaving many columns in the merged data frame having a high percentage of missing values.

Incident Number	0.00
Incident Category	0.00
Incident Description	0.00
Incident Day of Week	0.00
Incident Date	0.00
Incident Time	0.00
Police District	0.00
Resolution	0.00
Intersection	0.59
Longitude	0.59
Latitude	0.59
point	0.59
Row ID	0.00
SF Find Neighborhoods	1.04
Current Police Districts	0.65
Current Supervisor Districts	0.62
Analysis Neighborhoods	0.64
Incident Year	0.00
Incident Datetime	89.78
Report Datetime	89.78
Incident ID	89.78
CAD Number	92.11
Report Type Code	89.78
Report Type Description	89.78
Filed Online	97.83
Incident Code	89.78
Incident Subcategory	89.78
CNN	90.37
Analysis Neighborhood	90.37
Supervisor District	90.37
HSOC Zones as of 2018-06-05	97.70
OWED Public Spaces	99.48
Central Market/Tenderloin Boundary Polygon - Updated	98.62
Parks Alliance CPSI (27+TL sites)	99.87

Figure 1: Merged data frame's columns and percentage of missing values

The merged data frame has a total of 2,415,298 rows and 34 columns. Imputing values in columns that are missing such a large amount of data would not be realistic. To account for the vast number of missing values, I focused my attention on the shared columns between the two datasets and combined features that contained similar information such as "Analysis Neighborhood" and "Analysis Neighborhoods". My finalized merged data frame has a shape of 2,415,298 rows (each representing a separate crime report) and 15 crime feature columns.

Incident Number	0.00
Incident Category	0.00
Incident Description	0.00
Incident Day of Week	0.00
Incident Date	0.00
Incident Time	0.00
Police District	0.00
Resolution	0.00
Intersection	0.59
Longitude	0.59
Latitude	0.59
point	0.59
Row ID	0.00
Incident Year	0.00
Analysis Neighborhood	0.64

Figure 2: Merged dataframe's columns and percentage of missing values after modifying

5. Exploratory Data Analysis and Initial Findings

With my merged data frame, all data from 2003-2019 that has been cleaned is consolidated into one place. By digging deeper into the data, my goal was to reveal underlying stories behind crimes being reported. Having all crime reports in one place allows me to easily access all available data to answer questions such as "what time of the year are crimes most likely to happen?" and "are certain police districts more at risk than others?"

a) How has crime report rate changed per year since 2003?

I began by examining the number of crimes being reported per year from 2003-2018. In order to not show any biases in the graph, I left out 2019 since data had only been logged up through August.

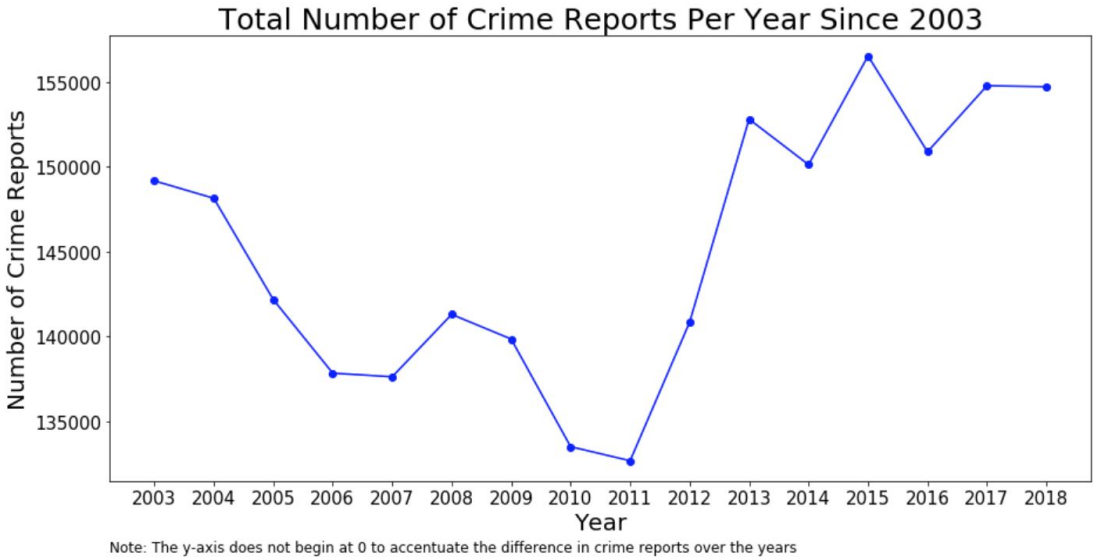


Figure 3: Total crimes reports per year 2003-2018

There are a couple of interesting takeaways from *Figure 3* that are worth noting. Beginning in 2003, the number of crime reports per year was high relative to the total average of 145,192 for 2003-2018. However, the minimum dip in 2011 is interesting. Observing patterns that could have caused this and comparing this year to the peak in 2015 motivate opportunities for further exploration.

b) What classification of crimes is reported the most?

Going back to my merged data frame, there were a couple of things I needed to consider before answering this question. For each crime report, the classification of crime was listed under the "Incident Category" column. However, 2003-2017 crimes from the CABLE mainframe were labeled sightly different compared to the 2018-2019 Crime Data Warehouse crimes. For example, "LARCENY/THEFT" and "Larceny Theft" were two respective classifications for each data set that represented the same type of report. I did not want to treat these two differently, so I split up the reports into 2003-2017 reports and 2018-2019 reports. By doing this, I could examine the original classifications for each report and compared the results to each other.

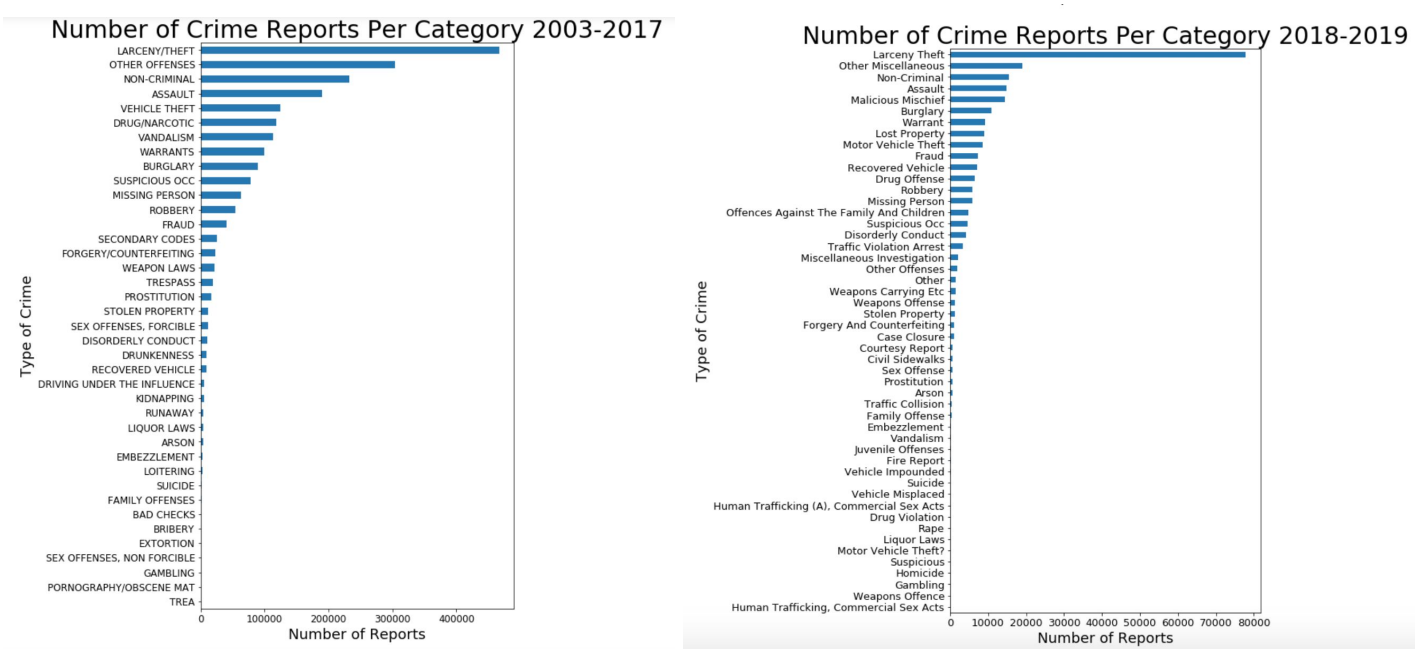


Figure 4: All crime reports by category

c) In what days or months are crimes most likely to occur?

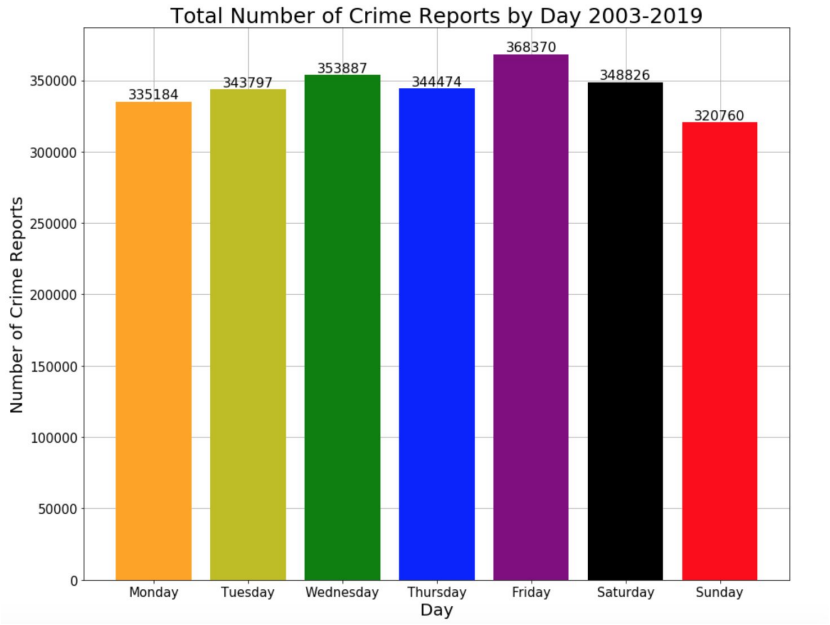


Figure 5: All crime reports by day

From *Figure 5*, it appeared that Fridays have been the most likely for a crime to be reported. This seems reasonable since it marks the beginning of the weekend when more activity during the night outside of work is common. Sunday, the end of the weekend, sees the lowest number of crime reports. The drop from the beginning to the end of the weekend, though small relative to the entire data set (47,610 reports compared to the 2,415,298 total reports), could still affect the likelihood of crimes being reported.

I also did a similar comparison between months to visualize how the number of crime reports change from month to month.

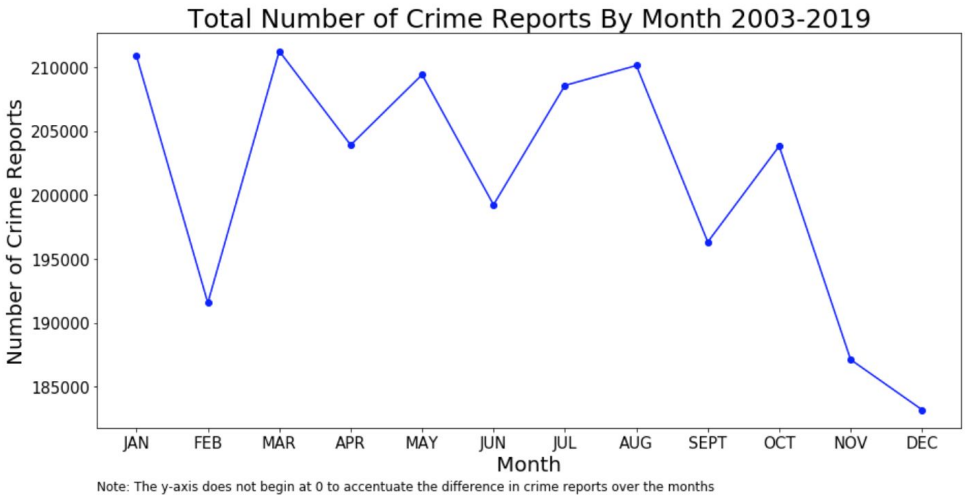


Figure 6: All crime reports by month

Crime reports appear to be more common at the beginning of the year, with the lowest numbers occurring in November and December. The dip in February between January and March (which are the two highest months) could be because there are less days but would be interesting to look further into.

d) Are crimes more likely at certain times of the day or year?

Next, I wanted to isolate each crime by the time it was reported to see if time had any effects on how many crimes were reported. First, I formed buckets representing crimes by the hour. Crimes reported between 12:00am-12:59am were in one bucket, 1:00am-1:59am were in another bucket, and so on. I then took each of these buckets and split them by day reported to get individual graphs representing times crimes were reported for a specific day.

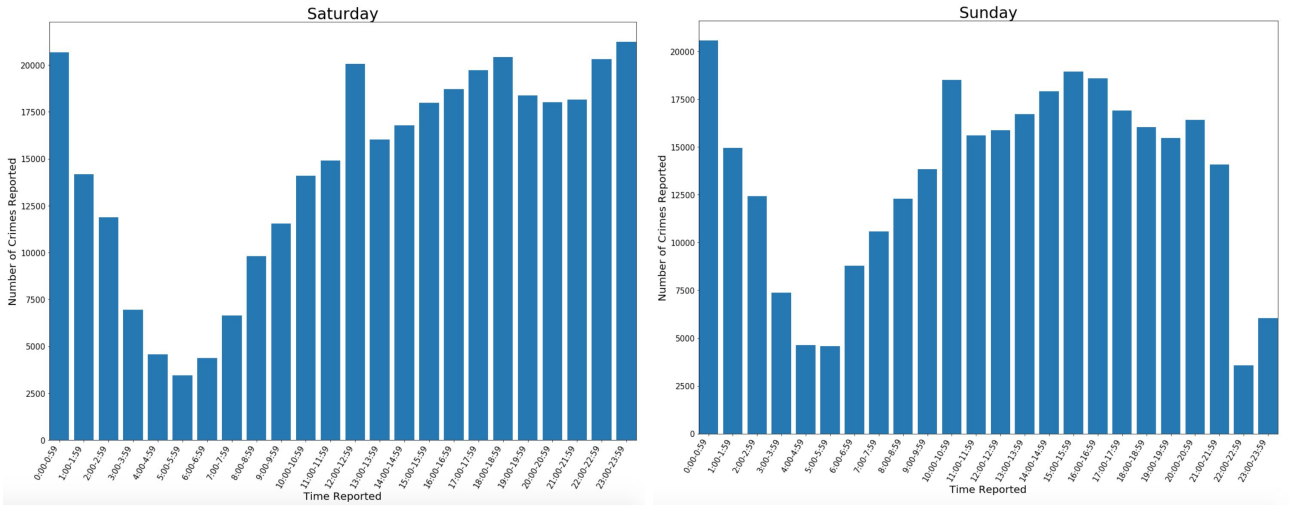


Figure 7: Crime reports by time on Saturday and Sunday

Figure 7 is an example for Saturday and Sunday, where each bin represented the hour in which a crime was reported. Binning the crimes by hour made visualizing patterns relating to time much more clear. Hours 10:00pm-11:59pm highlighted an interesting distinction between the two days. Saturday crimes are more likely compared to Sunday, and there is a large decline in late night crimes toward the end of the weekend.

I decided to take this a step further and look at times of reported crimes per month as well. I was curious to see if the time of the year had any significant effect on the times that crimes were being reported. *Figure 8* shows two graphs for June and December representing the total number of crimes reported for each time bin.

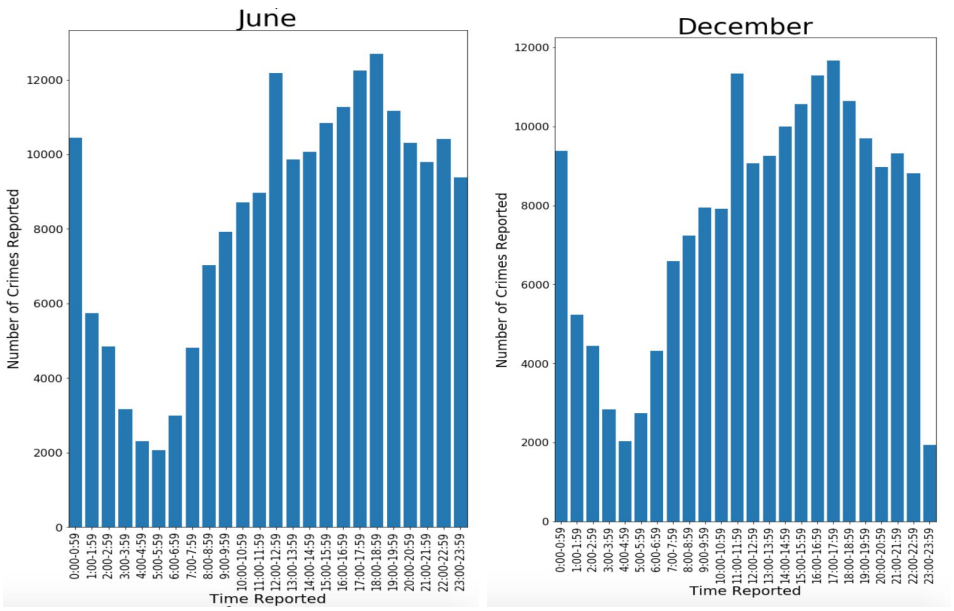
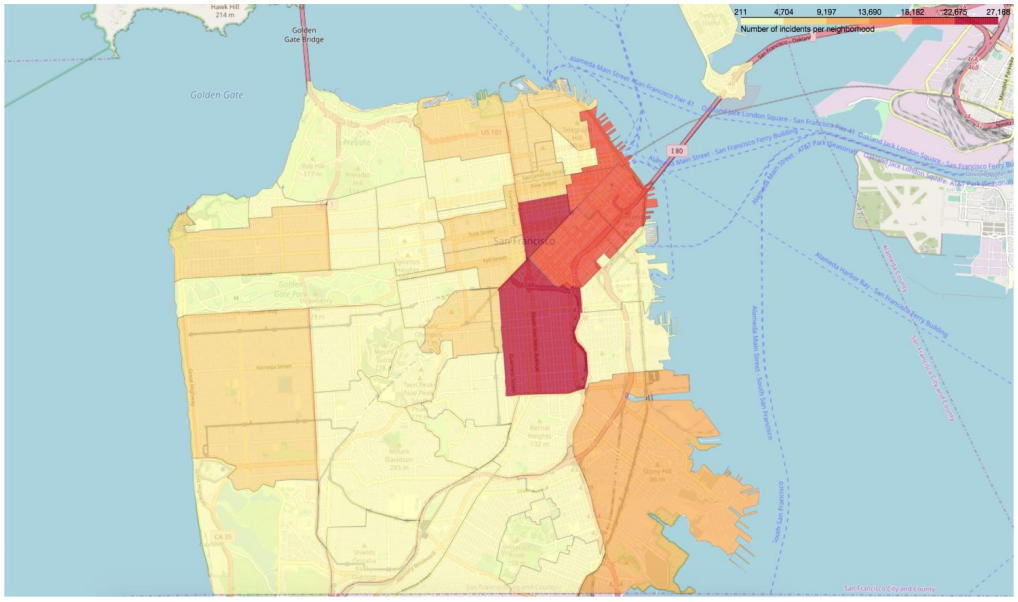


Figure 8: Crime reports by time in June and December

e) In which neighborhoods are most crimes reported?

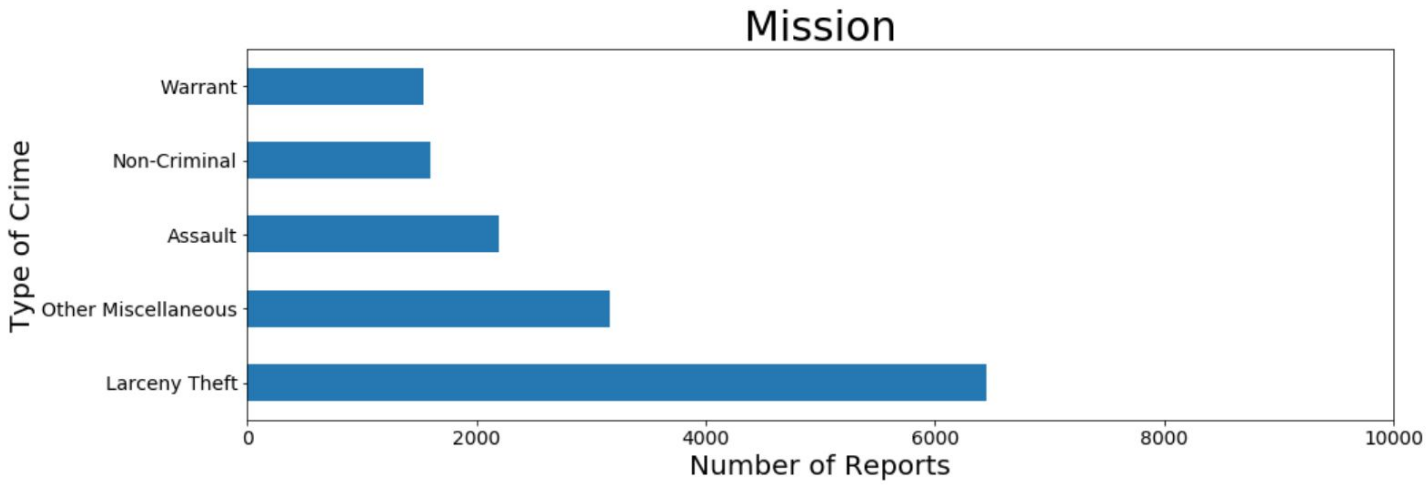
San Francisco consists of 41 different neighborhoods created by The Department of Public Health and the Mayor's Office of Housing and Community Development for the purpose of providing consistency in the analysis and reporting of socio-economic, demographic, and environmental data (https://data.sfgov.org/Public-Safety/Police-Department-Incident-Reports-2018-to-Present/wg3w-h783). Each crime report was associated with one of these 41 neighborhoods, and visually representing the crime reports via a choropleth map allowed for an easily interpretable visualization of areas of San Francisco that were most dangerous. In *Figure 9*, I split each neighborhood into its own segment of the map and colored it based on the number of crimes reported in 2018-2019. Lighter colors represented neighborhoods that didn't have as many crimes reported, whereas darker colors meant that much more crimes were reported in respect to the other neighborhoods.



	Neighborhood	Number of Reports
0	Mission	27168
1	Tenderloin	24788
2	Financial District/South Beach	22052
3	South of Market	20614
4	Bayview Hunters Point	13385
5	North Beach	7746
6	Western Addition	7475
7	Castro/Upper Market	7114
8	Sunset/Parkside	6935
9	Nob Hill	6572

Figure 9: Choropleth Map of Crimes per Neighborhood for 2018-2019 Reports

In *Figure 9*, I decided to look at 2018-2019 reports to examine the current most dangerous neighborhoods. A direction I wanted to take this information was to see if certain classifications of crimes were different among the most dangerous neighborhoods. Was theft more common in Mission than Tenderloin? Or were drug offenses more of a concern in certain neighborhoods? These questions motivated me to hone in on the five most dangerous neighborhoods to see the types of crimes they struggle with.



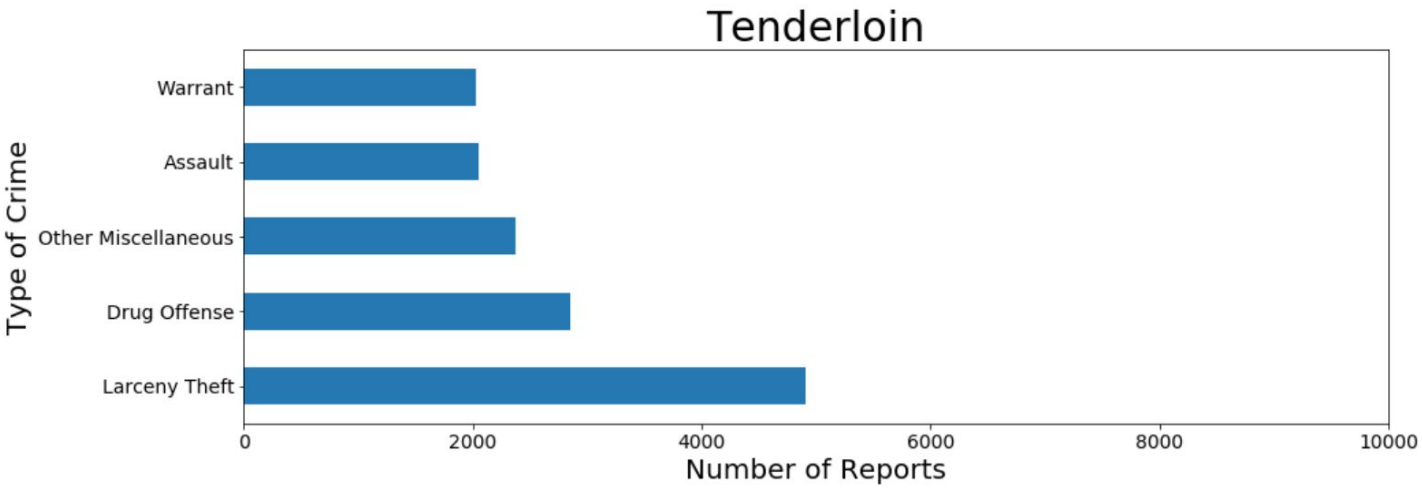


Figure 10: Top-five most reported crime classifications for Mission and Tenderloin

Figure 10 displays the two neighborhoods with the most crime reports from 2018-2019. Drug offense crimes ranked in the top-five most reported crimes in Tenderloin, yet didn't show up for Mission's top-five. This could help the San Francisco Police Department target areas when dealing with specific crimes by knowing which areas are most likely to have those crime classifications reported.

f) Heatmap and Correlation Matrix for Crime Features

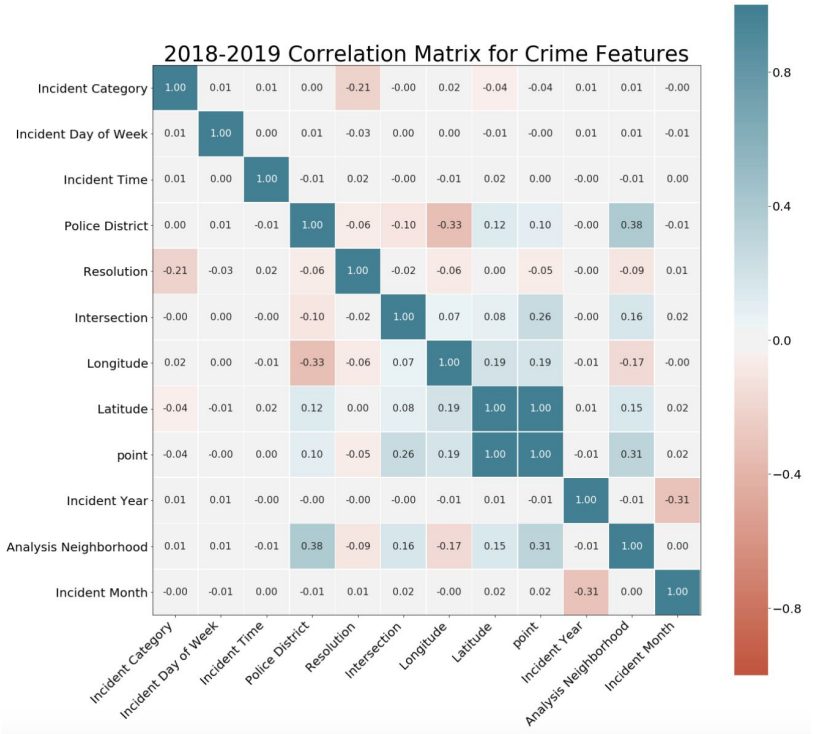


Figure 11: Heatmap and Correlation Matrix for Crime Features

Initially, each crime report contained some categorical information that was stored as a String:

- Incident Category
- Incident Day of Week
- Incident Time
- Police District
- Resolution
- Intersection
- Analysis Neighborhood
- point

String columns initially couldn't be compared for their correlation. Because of this, I converted the above string columns to 'categorical codes' using Pandas' built-in *Series.cat.codes*. This allowed me to create a correlation matrix between the different crime features to visualize how certain features change in relation to others, but it came with some limitations. The features in each category were simply numbered in numerical order, so this didn't accurately capture *how* categories differed (how would one police district be different than another when being compared with just integers?). For example, even though it appears that there was some correlation between "Resolution" and "Incident Category", it is difficult to interpret using simple numerical codes to represent the categories.

6. Applications of Inferential Statistics

a) Crime Report Rate: Is there a statistical significance between the average number of crimes per year between different police districts?

San Francisco is split into 11 different police districts. In 2015, the boundaries for each police district changed. If I were to take all reports from 2003-2018, some reports would be cross-referenced due to the boundaries being different after 2015. Due to the change, I examined the average number of crimes reports per year for each police district starting in 2015 until 2018, leaving me with a sample of 4 data points per police district.

Police District	Mean Per Year	Variance	Total Reports
southern	26848.50	1.949013e+07	107394
mission	20342.50	2.423115e+06	81370
northern	20130.25	5.389582e+05	80521
central	20011.50	1.041427e+07	80046
bayview	13818.50	1.223255e+06	55274
ingleside	11874.50	1.143297e+06	47498
taraval	11501.00	5.272353e+05	46004
tenderloin	11223.75	8.588913e+06	44895
richmond	8993.25	1.796556e+05	35973
park	8479.75	7.493216e+05	33919

Figure 12: Mean reports per year and variance for each police district

i) T-test of difference of means for police districts example: Southern vs. Park

For each comparison between two police districts, I tested the following hypotheses:

- H_0: The true mean crime rate between the two police districts are the same
- H_1: The true mean crime rate between the two police districts are not the same
    - For each, I assumed α-level of 0.05

```
# Compute the test statistic
test_stat = df_districts.loc["southern", "Mean Per Year"] - df_districts.loc["park", "Mean Per Year"]
print("Difference of means for Southern and Park:", test_stat)
```

```
Difference of means for Southern and Park: 18368.75
```

```
sample_size = 4

# Calculate MSE
mse = (df_districts.loc["southern", "Variance"] + df_districts.loc["park", "Variance"]) / 2

# Calculate standard error of test statistic
standard_err = np.sqrt(2*mse/sample_size)

# Compute t-statistic
t_stat = test_stat / standard_err

# Degrees of freedom
dof = 2*sample_size - 2

# Compute p-value
p_val = 1 - stats.t.cdf(t_stat,df=dof)

print("p-value:", p_val)
```

```
p-value: 9.075809058534112e-05
```

Since my p-value was less than my assumed α-level of 0.05, I can reject my null hypothesis that the true means of crimes of "Southern" and "Park" are the same, and therefore the observed difference is statistically significant.

b) Chi-square test of independence to identify important features associated with the type of crime committed

My target feature that I wanted to run Chi-square tests against was *"Incident Category*", which gives the classification of each crime (i.e. theft, assault). Running Chi-square tests against other crime features allows me to test the following hypotheses:

- H_0: There is no association between "*Incident Category*" and the comparison variable
- H_1: There is evidence to suggest that there is an association between "*Incident Category*" and the comparison variable
    - For each, I assumed α-level of 0.05

The five features I tested *"Incident Category"* against were:

- *Incident Hour*
- *Incident Day of Week*
- *Incident Month*
- *Police District*
- *Resolution*

Incident Hour and *Incident Month* contained interval data (1, 2, 3...). Before running the Chi-square tests, I prepped them into more usable categorical data:

- *Incident Hour* = {morning, afternoon, evening, night}
- *Incident Month* = {spring, summer, fall, winter}

```
# Run chi-square independence test against comparison columns
chi_square = ChiSquare(df_2018_2019)
for var in comparison_cols:
    chi_square.TestIndependence(colX=var,colY="Incident Category" )
```

```
Time Category is IMPORTANT for Prediction
Incident Day of Week is IMPORTANT for Prediction
Season is IMPORTANT for Prediction
Police District is IMPORTANT for Prediction
Resolution is IMPORTANT for Prediction
```

Figure 13: Loop running Chi-square test against each feature

Each test that I ran resulted in a p-value less than 0.05, meaning that I could reject my null hypothesis and accept that there is evidence to suggest that there is an association between each of these crime features with "Incident Category" (the type of crime that is committed). Lastly, to test that my Chi-square test worked with irrelevant information, I tested it with"*Row ID*". "*Row ID*" is a unique identifier for each crime report in the dataset, which has no relationship to the type of crime.

```
chi_square.TestIndependence(colX="Row ID",colY="Incident Category")
```

```
Row ID is NOT an important predictor. (Discard Row ID from model)
```

Figure 14: Running Chi-square test against irrelevant crime feature

7. Baseline Model Analysis

From here forward, I approached the business problem from an unsupervised learning perspective. There are a lot of crime features per report, and my goal was to make sense of the data without preexisting labels. Being able to have a collection of crime reports and making sense of any ways to group potentially similar crimes would be beneficial to uncover patterns in the San Francisco crime data. For this reason, I decided to first use the k-means clustering algorithm to build my initial model to group the data. ***I also focused only on data from the Crime Data Warehouse (crime reports starting in 2018) for runtime purposes.***

From the Chi-square tests that I previously ran, I was able to group together a set of crimes that are significant for crime identification and prediction. The following list is the features that I selected to use for my model:

- *Incident Hour*
- *Incident Day of Week*
- *Incident Month*

- *Police District*
- *Incident Category*

a) K-means: Setting up data

All the features to be used in my model are categorical. Due to this, I needed to first convert my data to another usable form: dummy variables. The dummy variables are numerical and allows me to use them in my k-means model by accessing the converted data as a numpy array.

	Incident Hour	Incident Month	Incident Category_Arson	Incident Category_Assault	Incident Category_Burglary	Incident Category_Case Closure	Incident Category_Civil Sidewalks
2168356	1	1	0	1	0	0	0
2168357	1	1	0	0	0	0	0
2168358	1	1	0	0	0	0	0
2168359	1	1	0	0	0	0	0
2168360	1	1	0	0	0	0	0

Figure 15: Data frame containing data as binary dummy variables

b) Choosing the appropriate *k* value: Elbow Method via Sum-of-Squares

One requirement of using k-means clustering is that I need to know the number of clusters to be used prior to running the algorithm. The first method I used to determine my *k* value is the Elbow Method via the Sum-of-Squares error. By calculating the within cluster sum-of-squares error for each data point and plotting the totals, I can choose the "elbow point" in the plot as my target *k* value.

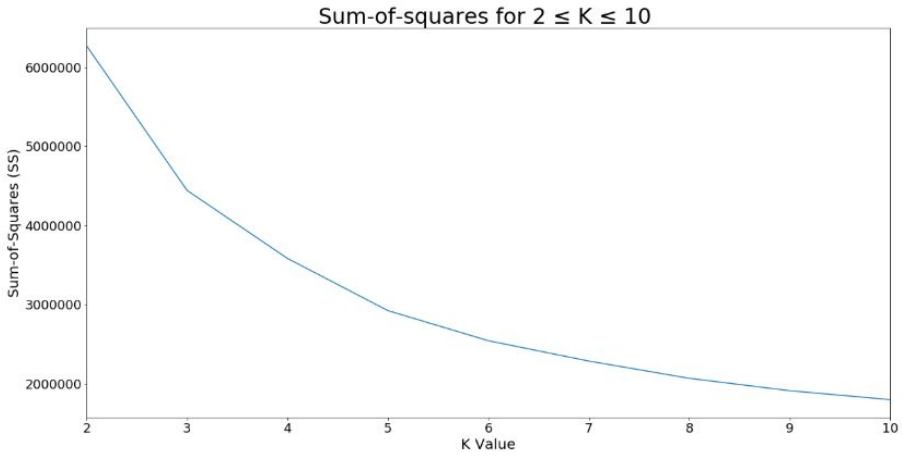
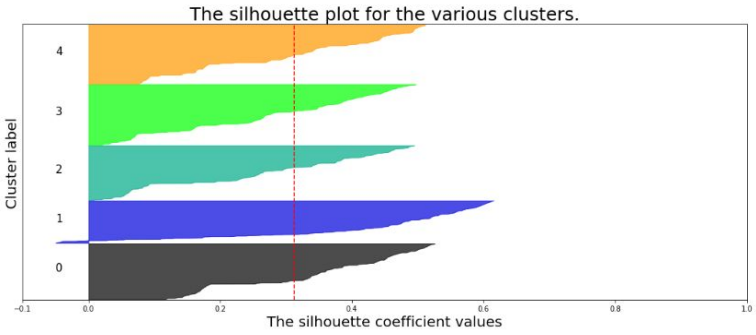


Figure 16: Sum-of-squares plot for different k-values

The "elbow point" is essentially the point in the graph where the slope begins descending at a slower rate. From *Figure 16*, it appears that $k = 5$ may be this point. However, the results are not conclusive. To make my choice of *k* more evident, I chose to use another method of determining my *k* value: The Silhouette Method.

c) Choosing the appropriate *k* value: The Silhouette Method

I initially ran the Silhouette Method on the full Crime Data Warehouse dataset, and the runtime took multiple hours for larger k values. For testing purposes, I decided to focus on a smaller dataset of 5000 randomly sampled crime reports to run the Silhouette method on.



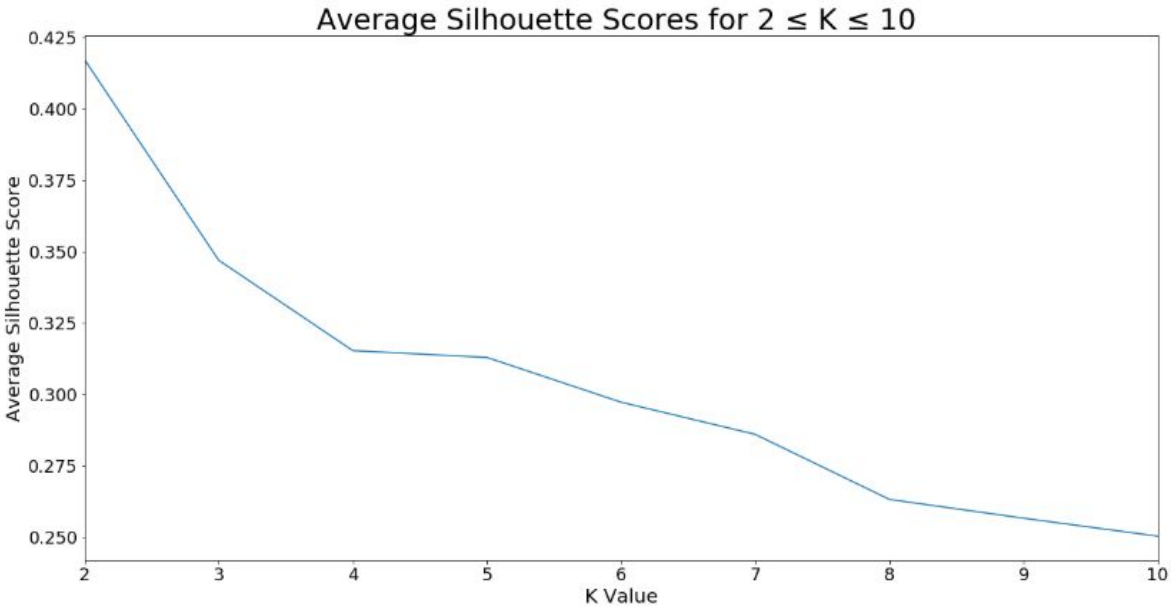


Figure 17 Above: Example silhouette plot with k = 5

Figure 17 Below: Plot of all silhouette scores for k values 2 to 10

Although higher silhouette scores represent stronger clustering structures being found, it appears that the average silhouette score for *k* as *k* increases follows a similar shape to the sum-of-squares method. When $k = 5$, the graph appears to level out a bit, similarly to the sum-of-squares plot. I decided to use this value to build my baseline model and evaluate the resulting clusters.

d) Building the k-means model with k = 5

To begin my baseline model, I ran the k-means algorithm on the Crime Data Warehouse data. The input data to my model was in the form of a numpy array with shape (***# features, # reports***). After running k-means through sklearn's Kmeans toolkit, I had my 5 clusters formed. Now that the crime reports were grouped, I looked deeper into each cluster while asking similar questions in my exploratory data analysis to try and gain some valuable insights for each group.

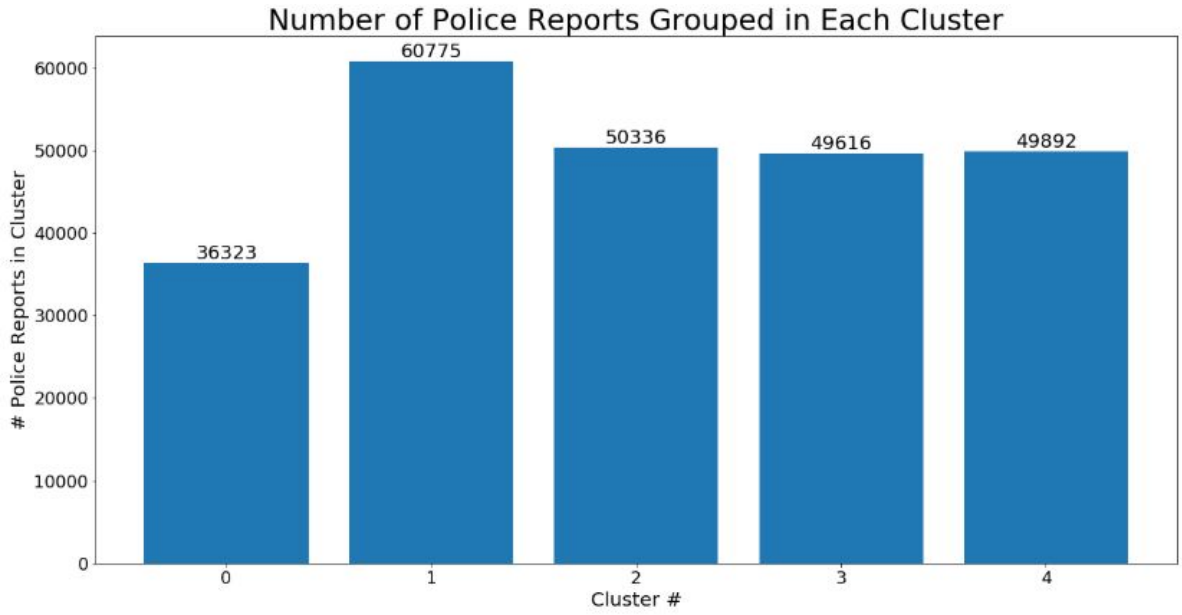


Figure 18: Number of police reports per cluster from k-means model

e) Looking deeper into each cluster

i) What types of crimes are in each cluster?

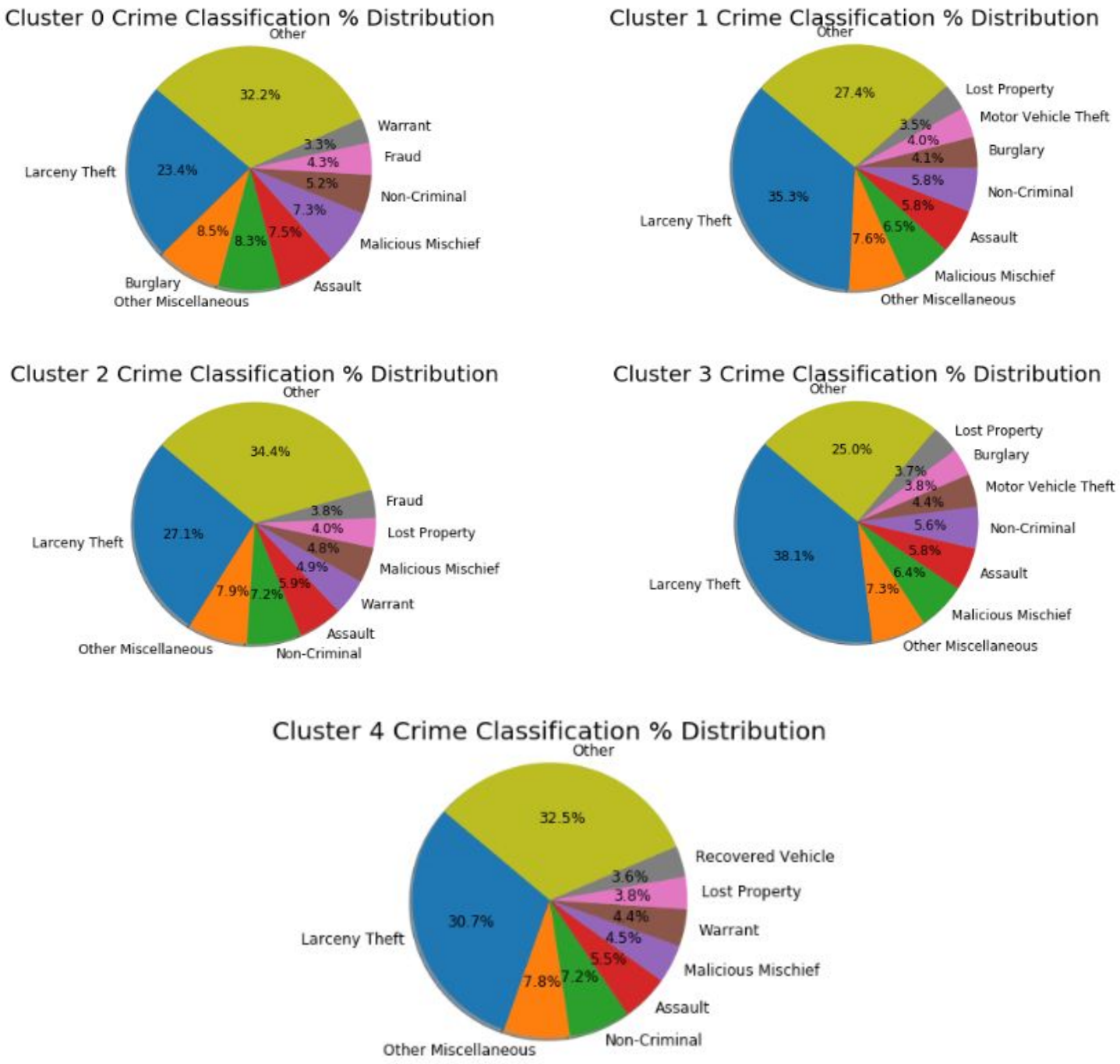


Figure 19: Crime classification % distribution by cluster

In Cluster 0, burglary the second most reported crime type at 8.5% of all crimes in the cluster. No other cluster had more than 4.1% of crimes be burglary. Larceny theft was a common theme throughout the five clusters.

ii) What days of crime reports are most frequent in each cluster?

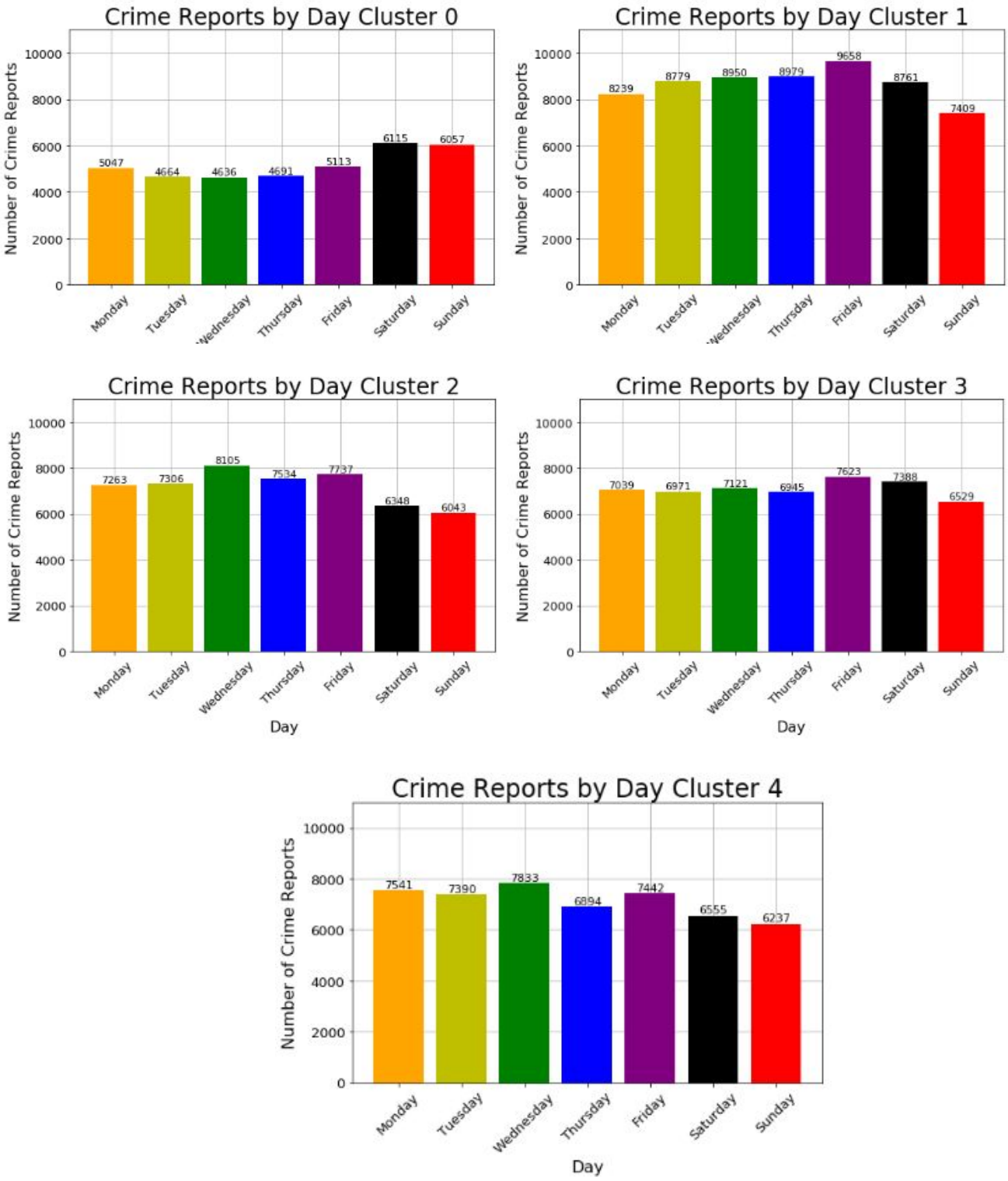


Figure 20: Crime reports grouped by day per cluster

iii) What *hours* of the day are most common for crimes in each cluster?

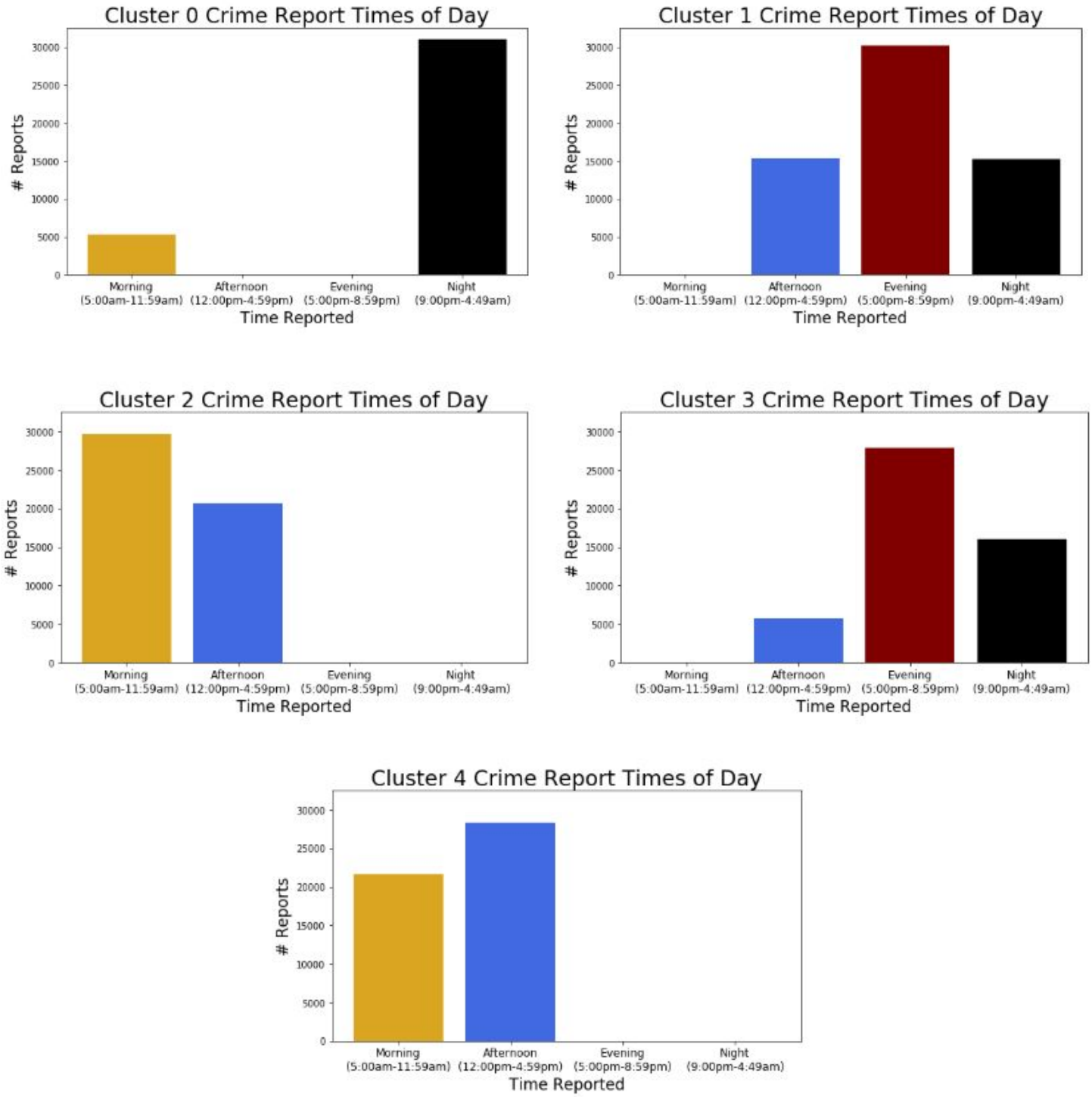


Figure 20: Crime reports grouped by hour per cluster

By splitting the crime reports by the time they were reported, a more distinguishable pattern arose. Cluster 2 and Cluster 4 only had crimes from the morning and afternoon. Cluster 1 and Cluster 3 didn't have any reported

crimes from the morning, and Cluster 0 focused heavily on crimes reported at night. This led me to believe that times that crimes are reported could be a significant factor for my k-means model forming the clusters.

iv) What *months* of the year are most common for crimes in each cluster?

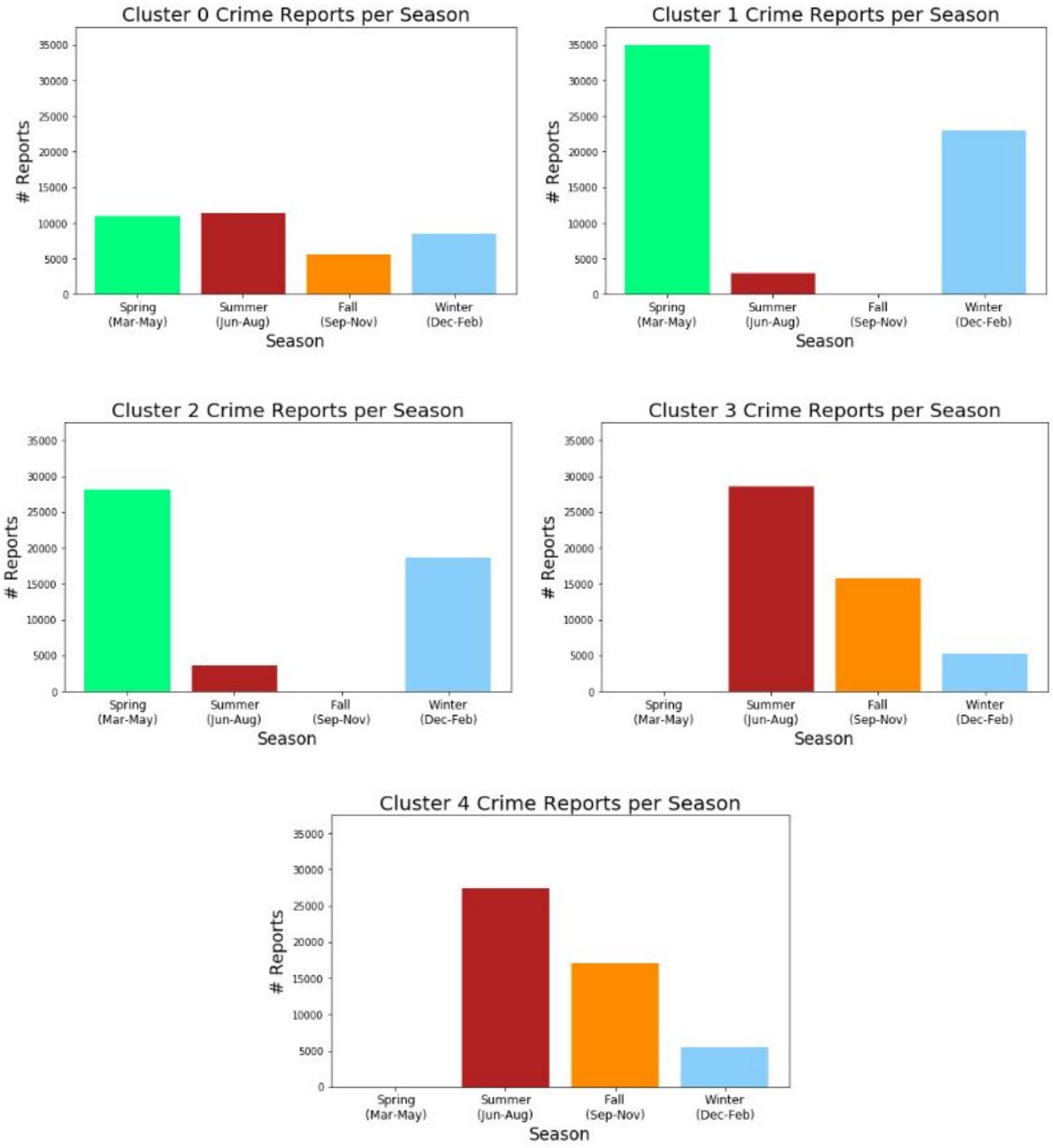
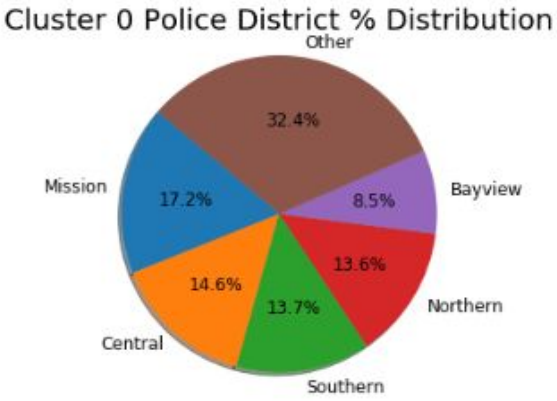
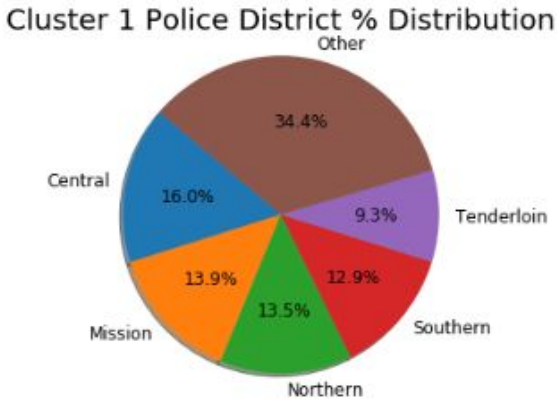


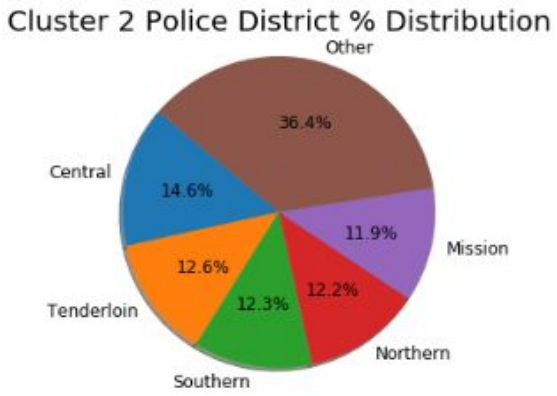
Figure 21: Crime reports grouped by month per cluster

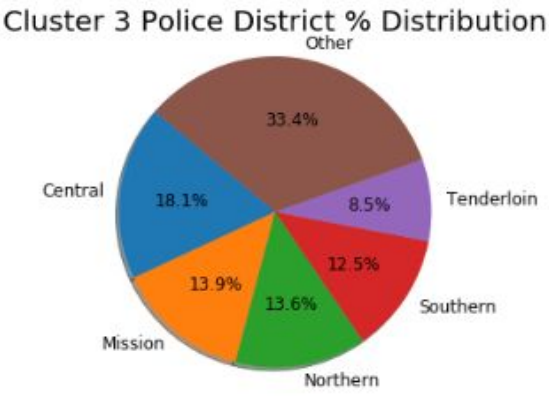
When splitting the crimes by the months (followed by grouping the months into their respective seasons), there was also some relationships between clusters. Cluster 1 and Cluster 2 looked very similar in their distributions, as well as Cluster 3 and Cluster 4. This is interesting because these similarities were different when looking that the reports by time of the day. It appeared that the month that a crime was committed in was a factor in how the different clusters were formed.

v) Which police district were most common in each cluster?









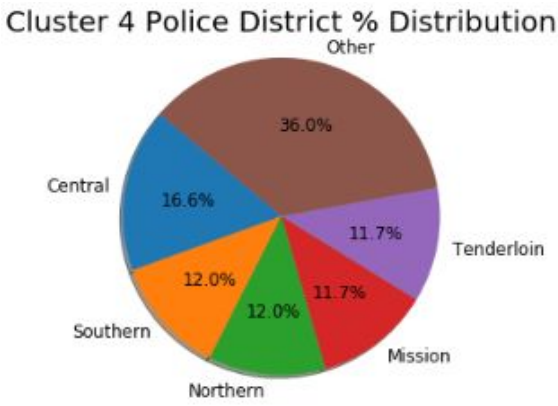


Figure 22: Crime reports grouped by police district per cluster

Lastly, I split the crimes by police district per cluster. Most clusters had a similar percentage distribution for police districts. However, Cluster 0 had a significantly higher percentage of Bayview police reports. Noting back to my previous comment of how burglary was a higher rate in Cluster 0, this could be an interesting relationship that the Police Department could look further into to see if there are internal factors for the Bayview District leading to a higher burglary rate.

f) Visualizing clusters using Principal Component Analysis

Now that the crime reports were clustered, I added each cluster label as a feature in a separate column titled "cluster_id" in the Crime Data Warehouse data frame. Since all crime reports were labeled with a specific cluster, I wanted a way to visualize my clustering. By using Principal Component Analysis, I could reduce the dimensionality of the crime data to lower dimensions and plot them on a graph to see how my clusters looked. For the following visualizations, I randomly sampled 5000 reports to help with the runtime of the plot and principal component computations.

i) Modeling using 2 principal components

By reducing the number of principal components to two, I got two separate numerical values to represent the crime features. By using these two values as the '*x*' and '*y*' points of my plot (*principal_component 1, principal_component_2*), I could get a 2-dimensional point to represent each crime report and plot them a color corresponding to the cluster that it was a part of.

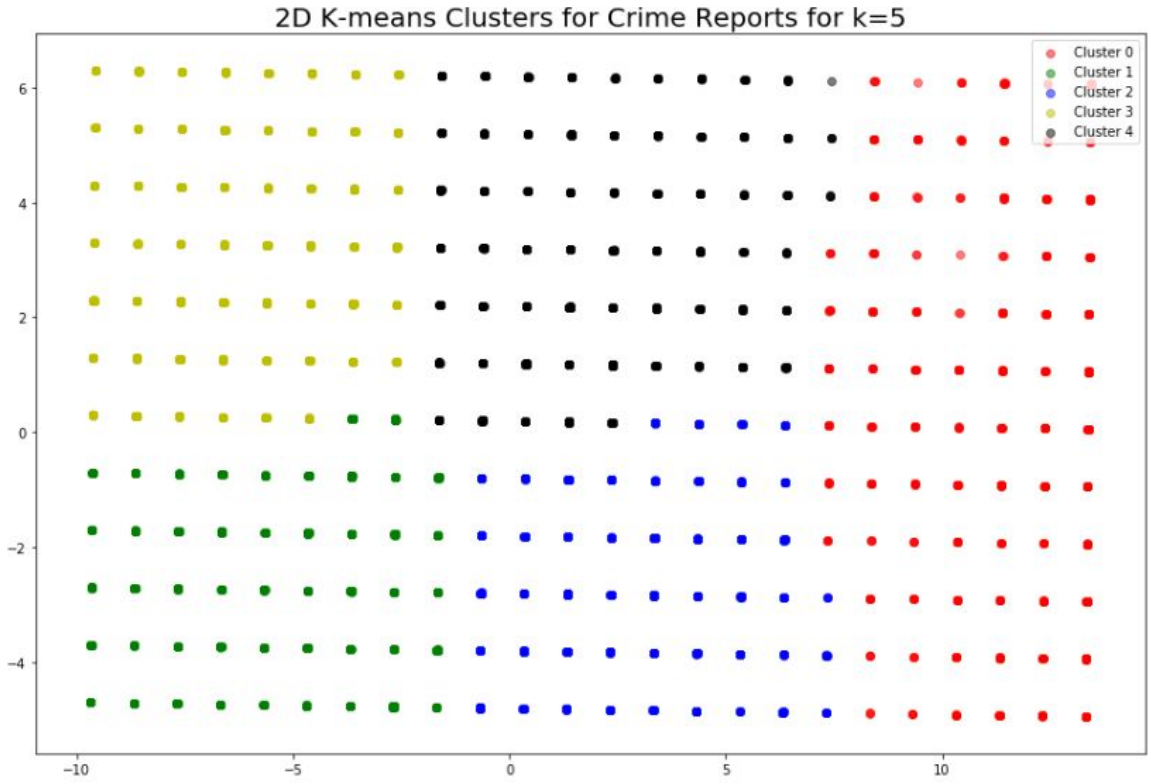


Figure 23: K-means plot using 2 principal components

Initially, it appeared that modeling in 2 dimensions doesn't give me a good intuition about how the crime reports have been clustered from my baseline model. The points were separated by an almost uniform space across x and y values. This made me wonder if adding an additional dimension would give me a clearer idea of the different clusters. To try and better represent the data, I tried introducing a third principal component to represent it in a 3-dimensional space.

ii) Modeling using 3 principal components

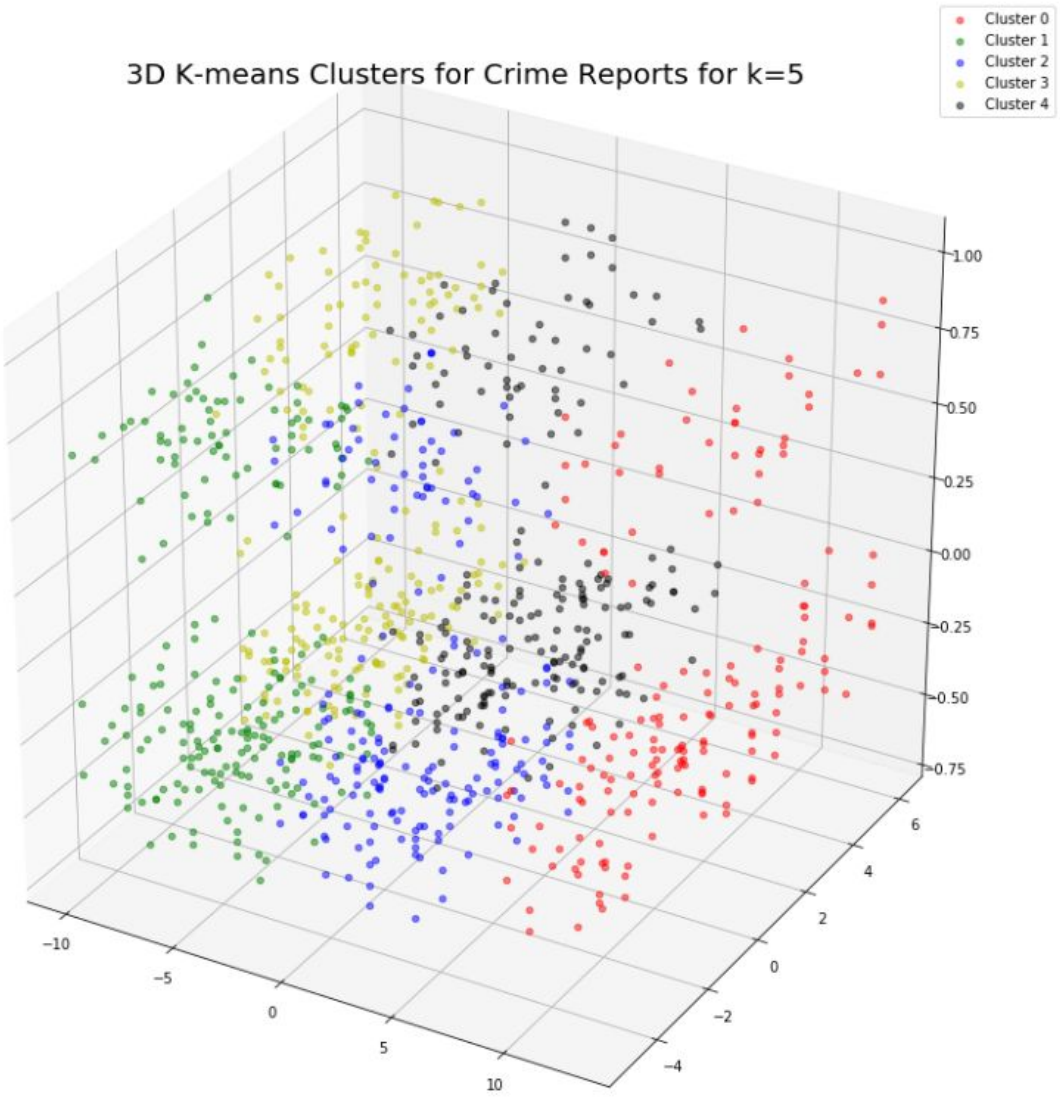


Figure 24: K-means plot using 3 principal components

When modeling the crime reports in a 3-dimensional space, it was a little more clear that where the clusters were forming. It looks like there is some layered elements factoring into the crime reports since it doesn't seem like they are all lined along a single plane. Nonetheless, the segmentation of each cluster doesn't quite provide a clear representation of the crime reports.

Because my clusters are not clearly distinguishable, I continued on by using other clustering algorithms such as k-modes to compare my results. I also use other visualization techniques, such as t-SNE to try and better

represent the clusters in lower dimensions to see if other dimension reduction techniques better represented the data.

8. Extensions from Baseline Model

In my baseline model notebook, I originally used k-means to cluster my crime reports into 5 separate clusters. After clustering, I used PCA to reduce the dimension of my data and visualize the resulting clusters on a 2-dimensional and 3-dimensional plane. I wanted to continue this a step further and see if other dimension reduction techniques and clustering algorithms would provide a clearer representation of the crime data. I proceeded by applying t-distributed stochastic neighbor embedding (t-SNE) to a sample of the data, along with the k-modes clustering algorithm.

a) Application of t-SNE

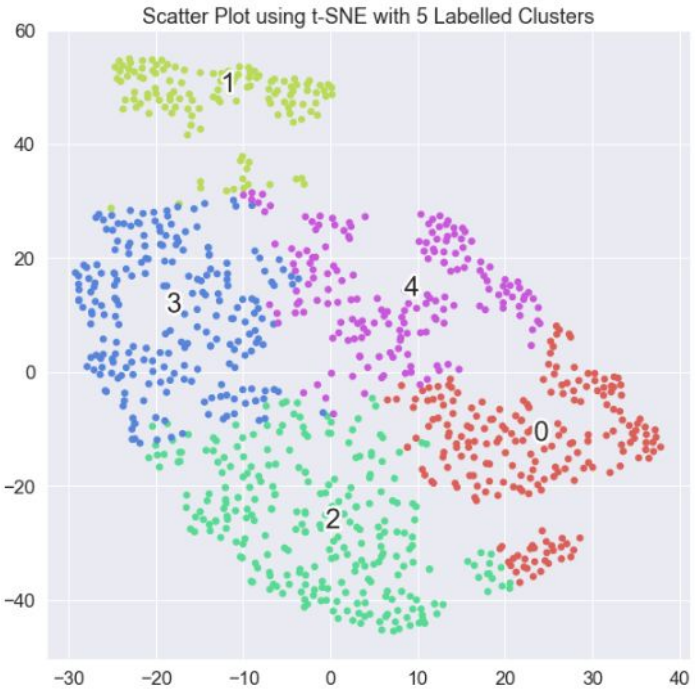


Figure 25: t-SNE application to clusters

I previously ran my baseline k-means algorithm on the data set to construct the potential cluster labels, and from there I created a scatter plot for the crime reports on the output of t-SNE. The output of t-SNE is the resulting

plot (*Figure 25)*, and I colored each data point on what cluster each point would belong to if I used my k-means model.

This scatter plot tells me that there are crime reports that could be similar in nature and can potentially be grouped together. The clusters from t-SNE don't tell me which crime reports should be grouped together, but the visualization does highlight that some reports are more similar to each other than others. It is easier to see some relationships between crime reports compared to my 3-dimensional PCA visualization since my data set has a high number of features and t-SNE is a non-linear dimension reduction algorithm that is good with handling such high-dimensional data.

b) Application of a new clustering algorithm: k-modes

K-modes is similar to k-means due to the fact that they require the number of clusters to be known before running, and k-modes is essentially an extension of k-means. However, one key difference is the way that distances between points are calculated. Rather than computing Euclidean distances in k-means (which requires numerical data for the distances to be calculated), k-modes uses the dissimilarity between objects and finds the mode (cluster centroid) vector that minimizes the dissimilarity between itself and its corresponding data. This makes k-modes good for categorical data since it doesn't require numerical input. All of the features I selected to build my model around are categorical, making k-modes great at handling the crime reports.

c) Choosing the appropriate *k* value: Elbow Method via k-modes' cost function

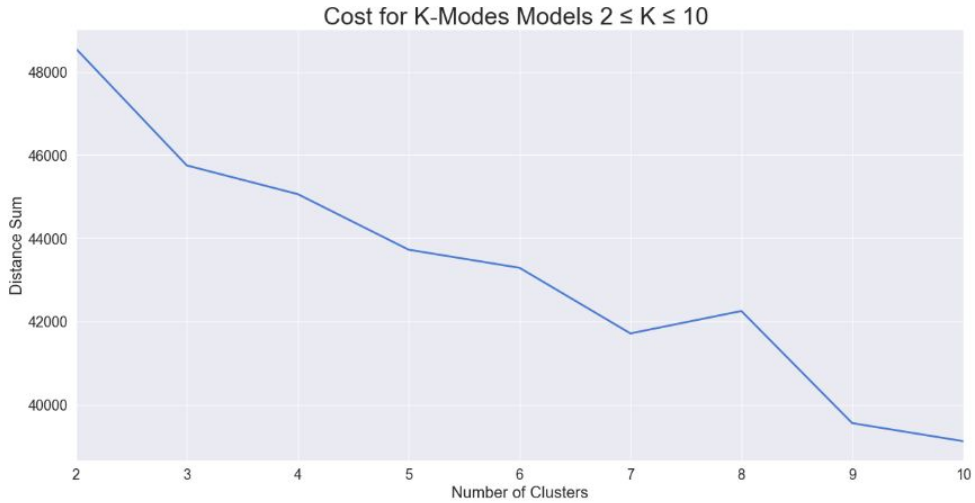


Figure 26: Plot of k-modes cost function

The "distance sum" (cluster cost) for each cluster value is defined as the sum distance of all points to their respective cluster centroids, meaning that lower values represent tighter clusters. Using the Elbow Method to select my k-value, when k=5 the cost function flattened out. I built my k-modes model with 5 clusters and compared my results to my baseline k-means model (which also had 5 clusters).

d) Building k-modes model with 5 clusters

For my k-modes model, I ran the k-modes algorithm on a sample from Crime Data Warehouse data. The code referenced from https://pypi.org/project/kmodes/ runs the algorithm on multiple iterations, so in order to decrease the runtime I randomly sampled 10,000 crime reports from the Crime Data Warehouse data set to cluster my data.

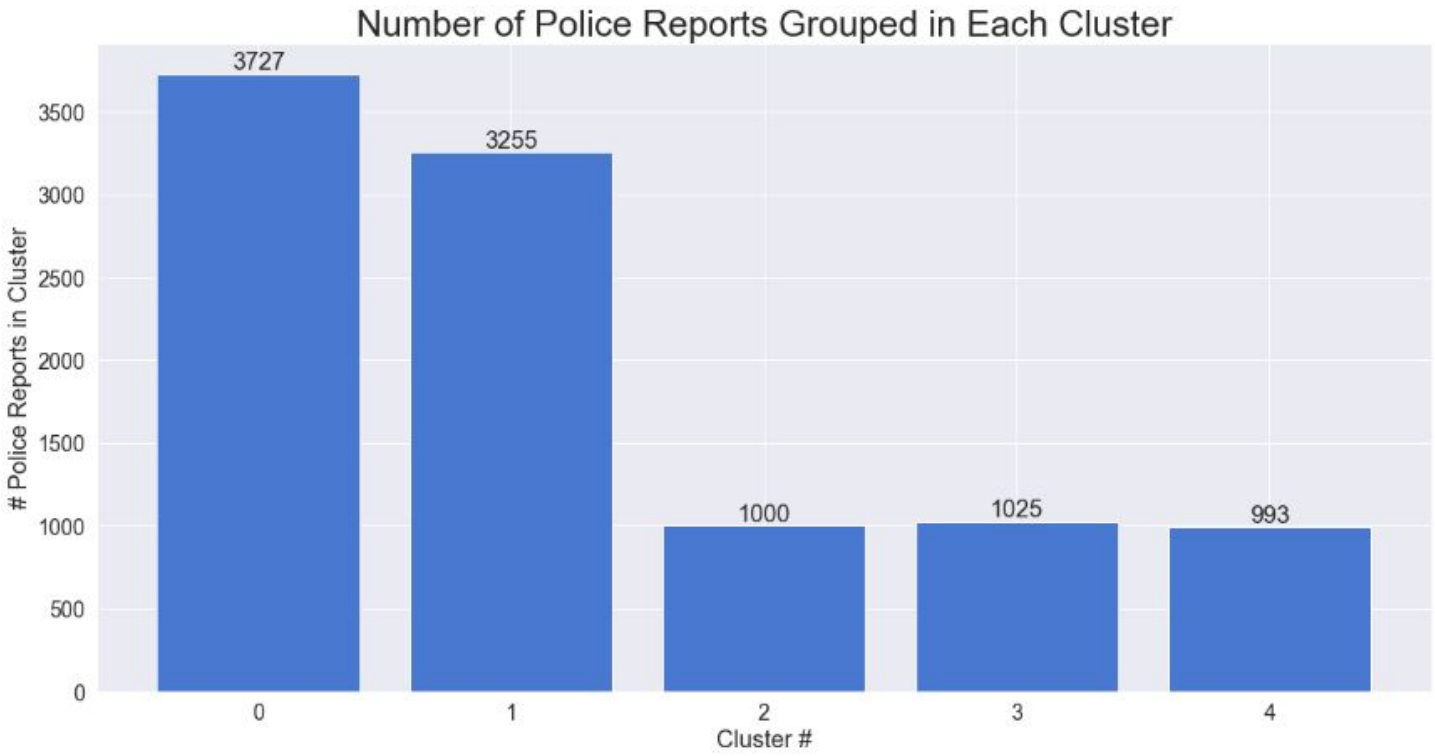
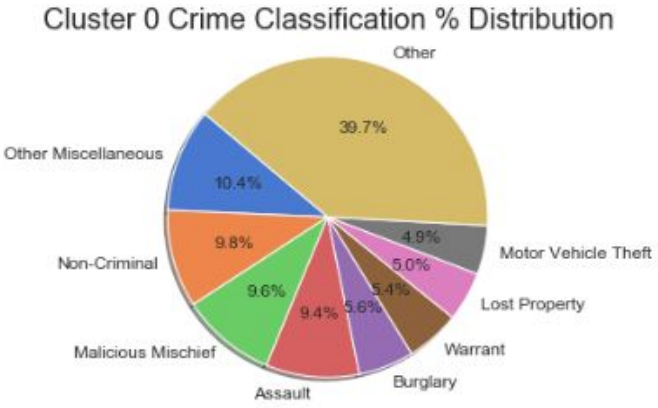


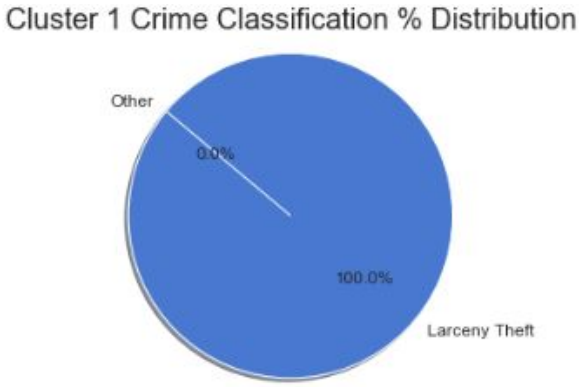
Figure 27: Number of police reports per cluster using k-modes

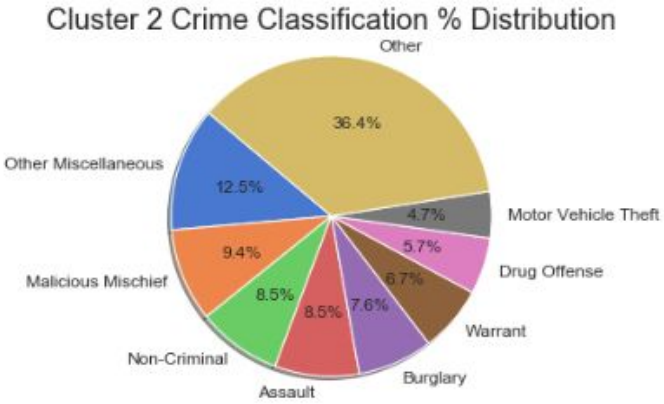
Now that my crime reports were labeled, I wanted to compare my results to those of the k-means model. Through asking similar questions and utilizing visualizations, I hoped to be able to highlight any key differences or similarities between the two models. Due to the randomness of my sample for k-modes, there could be some bias in terms of the reports that were selected. This is why I am going to look deeper into what each cluster contains and see if there are underlying patterns that these clusters tell me.

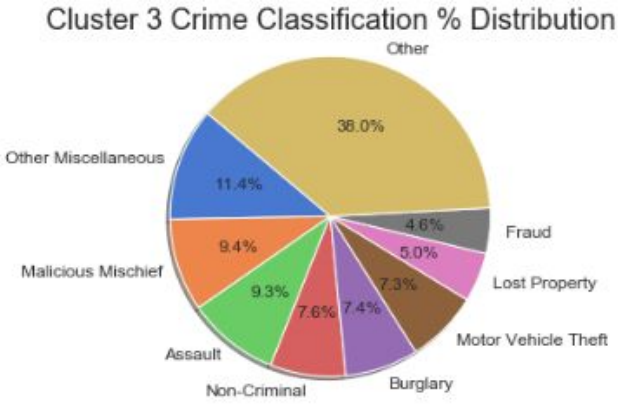
e) Looking deeper into each cluster

i) What types of crimes are in each cluster?









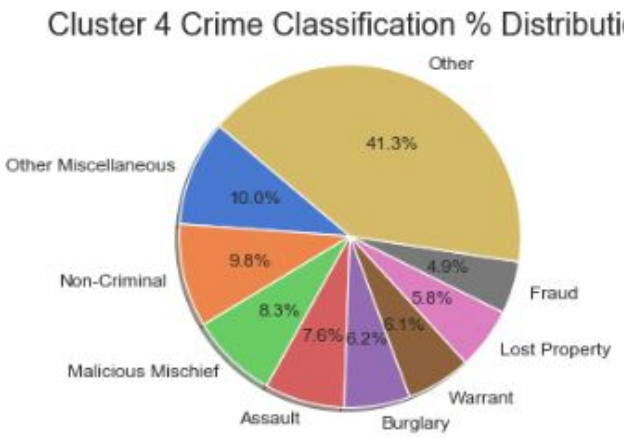


Figure 28: Crime classification % distribution per k-modes cluster

A big difference that I noticed for my k-modes model is that all "Larceny Theft" crimes were separated into their own cluster. "Larceny Theft" is the most reported crime type, and this appears to have a large impact on how these reports were separated from the rest especially since Cluster 1 is the second largest cluster.

ii) What days of crime reports are most frequent in each cluster?

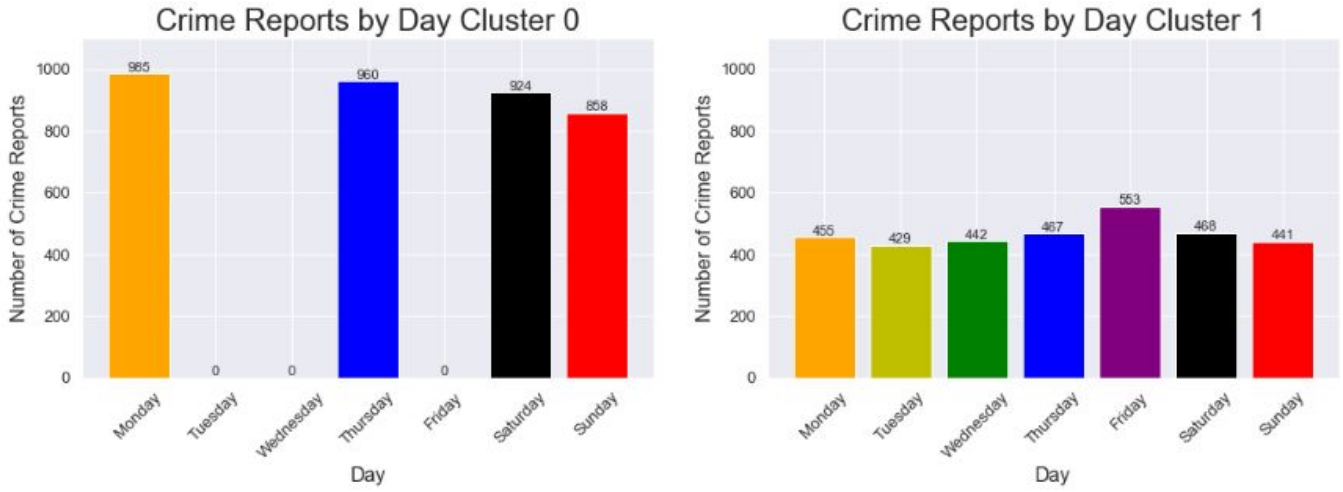


Figure 29:Crime reports by day per k-modes cluster

When splitting the crimes by day in each cluster, another major difference from my k-means model was highlighted. Unlike the k-means model, when I ran k-modes there were certain clusters that only contain reports from a single day. Cluster 2, Cluster 3, and Cluster 4 contained reports only from Wednesday, Friday, and Tuesday respectively. These three days were completely missing from Cluster 0. However, Cluster 1 had reports from every day, so this makes me wonder if there is some relationship between how crimes were separated from the other four clusters.

iii) What *hours* of the day are most common for crimes in each cluster?

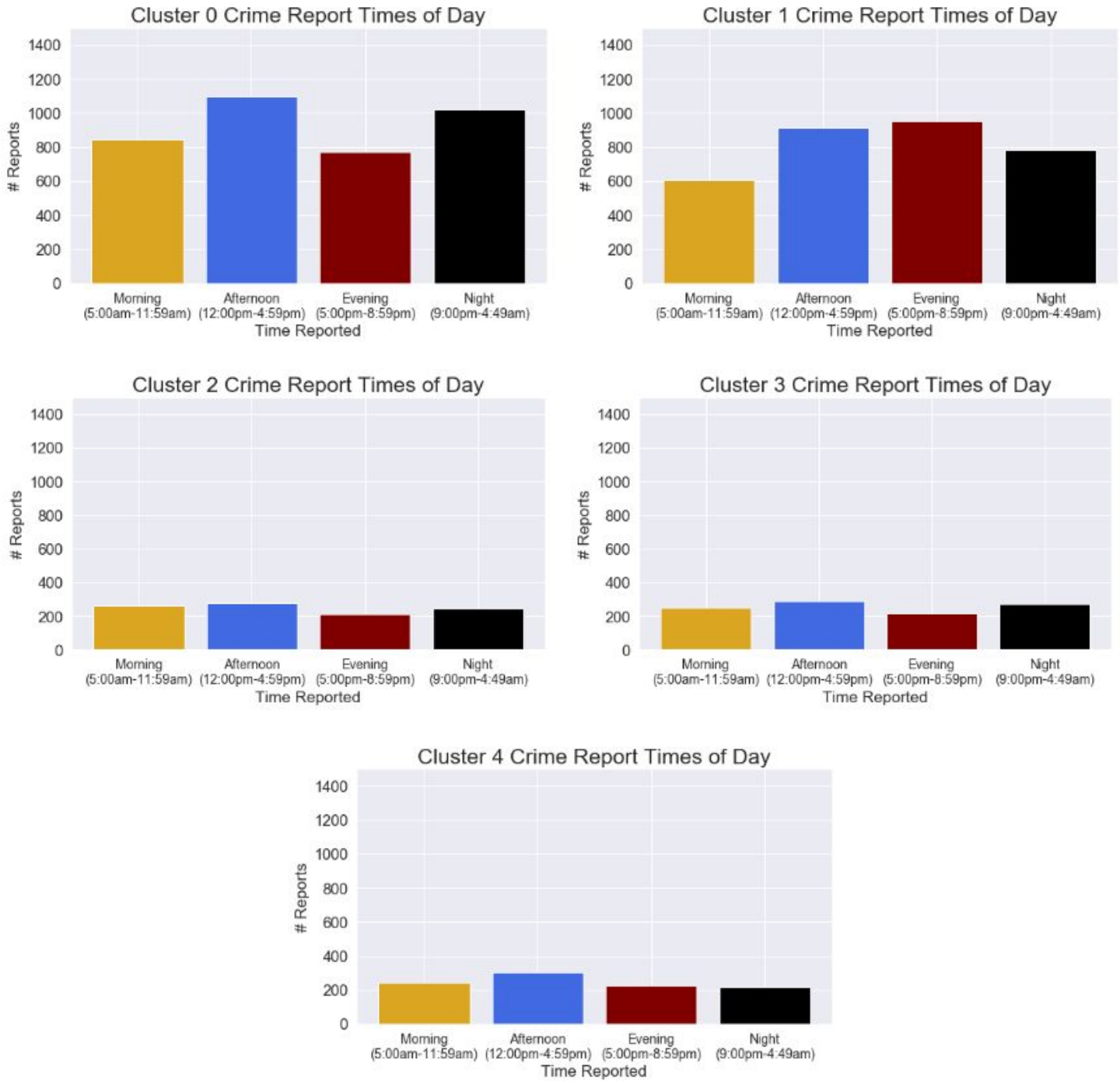


Figure 30:Crime reports by hour per k-modes cluster

Unlike my k-means model, the k-modes model's clusters didn't appear to be affected as much by the hour of the day a crime was reported. Within each cluster, the number of crime reports by hour was relatively evenly distributed.

iv) What *months* of the year are most common for crimes in each cluster?

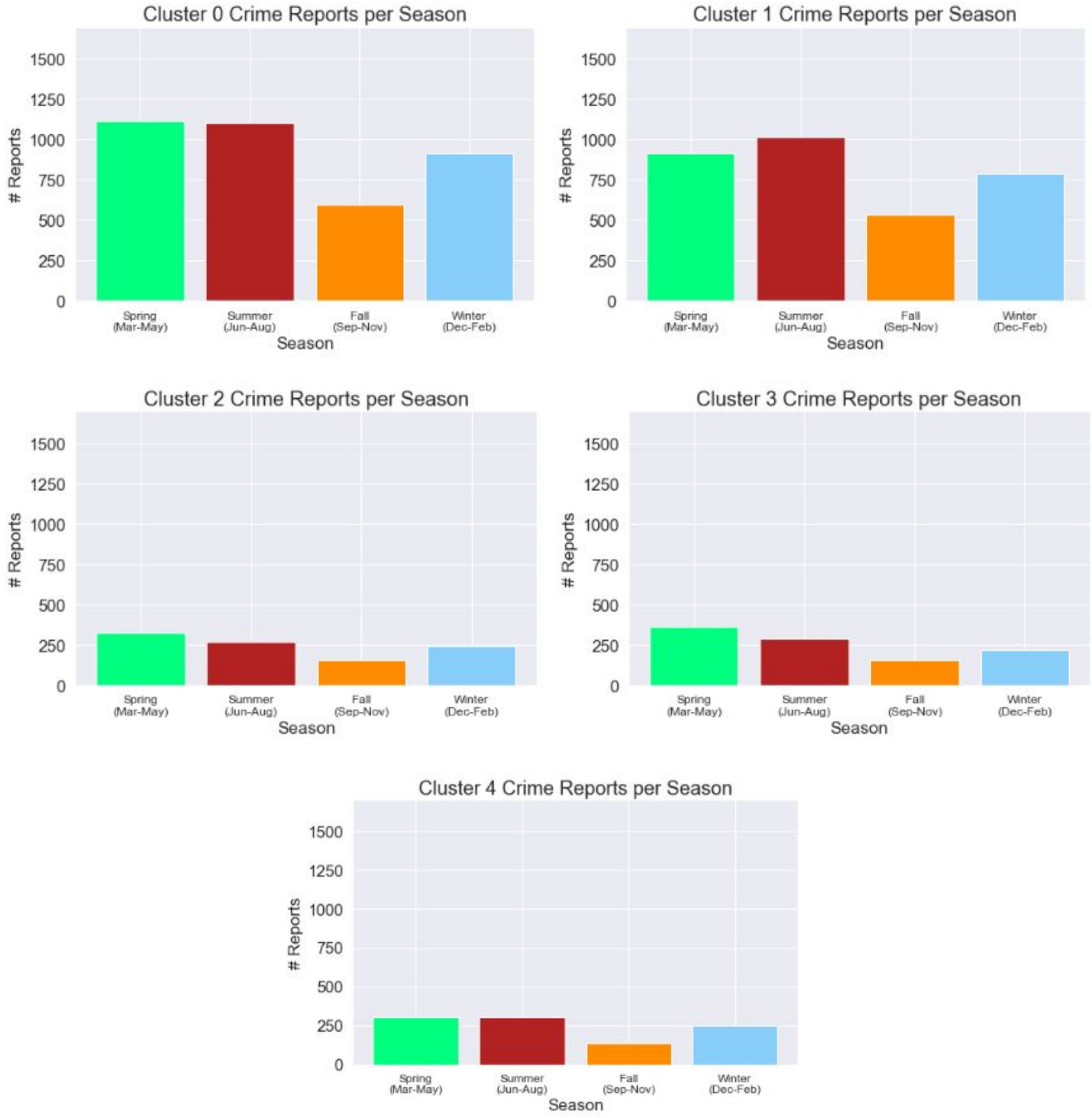
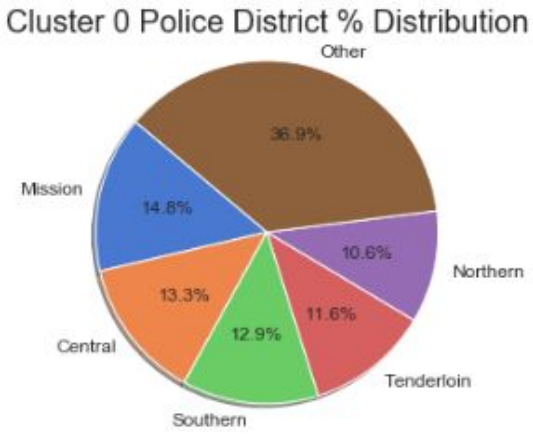
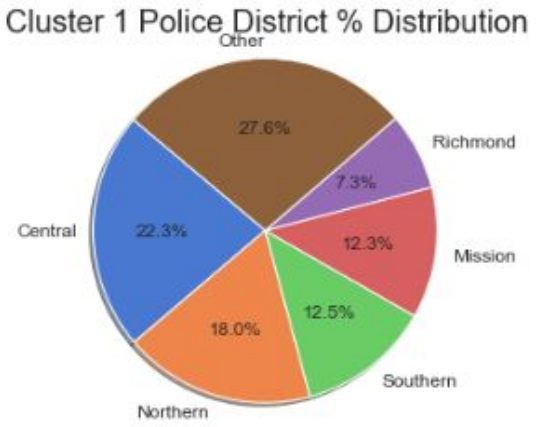


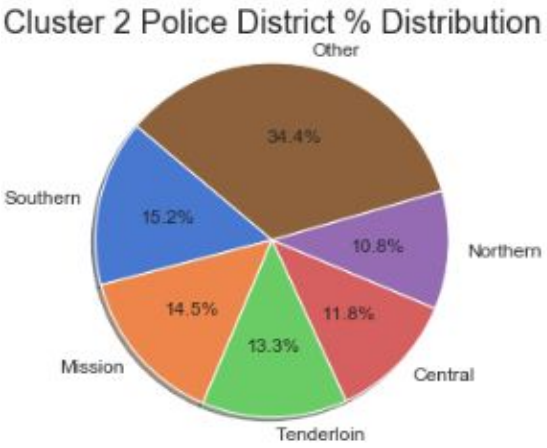
Figure 31:Crime reports by month per k-modes cluster

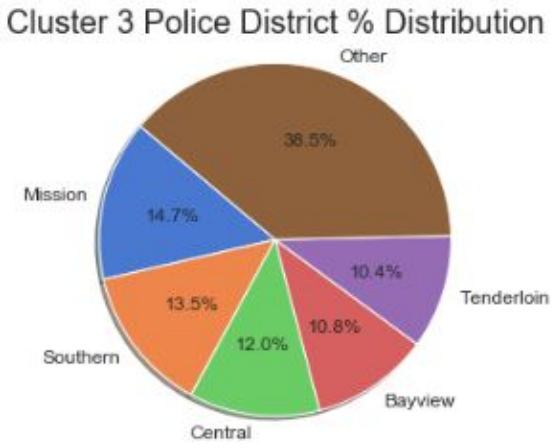
Similar to the charts for hours that crimes were reported, it also looked like the month that crimes were reported didn't affect the clustering as much as the k-means model. The distributions within each cluster were similar to each other.

v) Which police district were most common in each cluster?









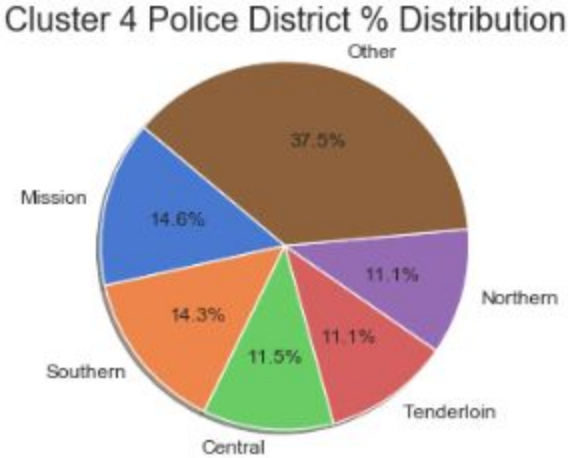


Figure 32:Police district % distribution per k-modes cluster

Clusters 0, 2, and 4 looked similar in their split of police district percentages. However, Clusters 1 and 3 stood out a bit because they each contained a unique police district that was not found in the top 5 of the others. Cluster 1 contained Richmond and Cluster 3 contained Bayview. Since Cluster 1 also only has crimes classified as "Larceny Theft", it could be interesting for the San Francisco Police Department to target these police districts as higher priority when dealing with theft cases. Since Cluster 3 only has crime reports from Friday, these police districts could also be more at risk of crimes happening on this day of the week.

f) Final Comments

These individual unique characteristics are quite different compared to my k-means algorithm. A big takeaway that I got from running different clustering algorithms on the Crime Data Warehouse data is that no single algorithm can reveal all the underlying patterns of my crime data. Some may be better for handling certain data or looking into specific features. However, because k-modes specifically targets categorical data, I feel like this model would be a better characterization of my data set. If I were to continue working with this algorithm, I would dig deeper into clustering on different features other than the ones I chose and on different cluster numbers to see how each model differs. I believe further exploring on a different sample would also be beneficial to account for the randomness in my current clustered sample.

9. Conclusions and Future Work

The San Francisco Police Department has taken major steps forward in terms of their crime preparation and data accessibility, as seen through their development of the Crime Data Warehouse. They are aware of the importance of keeping data accessibility reliant and quick. My goal was to make sense of the data and potentially uncover patterns in the crime reports that could help the San Francisco Police Department be aware of. Through the hypothesis testing and machine learning models I built, I hope that the San Francisco Police Department can understand more about the data that is being submitted for each crime report. Even though there are hundreds of thousands or crime reports in the past two years alone, I hope I showed some ways to interpret, visualize, and apply models to reveal underlying crime patterns.

Some recommendations I could make for future crime analysis is the way in which crime classifications are dealt with when building the machine learning models. I handled every unique case that the Crime Data Warehouse labeled a crime as ("Incident Category" crime report feature). I believe that configuring my own classifications so that way there weren't 50 unique crime classifications could create more ways to further explore the data in both a supervised and unsupervised learning context. The k-means/k-modes clustering algorithms I used could be affected by more general crime classifications. Additionally, the business problem could be taken from a supervised learning perspective by making crime classifications the target feature and predicting the types of crimes committed based on certain conditions (time, day of the week, location of crime, etc.). Focusing on creating more general crime classification buckets would help the data not be as spread out and would help narrow the focus in terms of the business problem of understanding specific conditions that lead to crimes.

10. Recommendations to Client

After running k-modes clustering on the data, "Larceny Theft" is by far the most reported crime and has a significant impact on crimes that could be connected or related to each other. These classifications of crimes were put into their own separate cluster, and I think it is important to target police districts such as Central and Northern with high priority when dealing with theft cases due to the inflated probability of theft cases. Allocating more resources to districts where crimes are most likely going to be reported may help in reducing crime rate and prevent similar crimes from happening.