Viswadeep Manam
Bronco ID: 013890851

Relationship Between Crime and City Attributes

Data Acquisition

Process to Retrieve Data

To explore the correlation between different types of crimes and the attributes of cities, I needed to find, essentially, two datasets: one that collected crime data, and one that collected demographic information on a city level. However, there was no website that hosted this aggregated information, so I collected both sets programmatically with Selenium and BeautifulSoup. For the city data, I utilized SocialExplorer which hosted the yearly American Community Survey data. To retrieve this data, I utilized Selenium to get city data from 2010 to 2019. This data represents various demographic information a city, and its population can have. Getting the crime data was significantly easier as I was able to just use the FBI's crime data API. In order to combine both sets of data, I scraped a website that hosted the ORI codes- unique identifiers for Law Enforcement Agencies- in counties in a state. I matched each city name that I already retrieved to an ORI code. If that city had an ORI code, I queried the FBI's API and retrieved a summarized list of offenses that happened in a given year in that city.

Data Format

Data Attribute	Meaning	Range of Values
Total Population	Total number of people in a city	Any value greater than 0
Population Density (Per Sq. Mile)	Number of people living in each unit of area	Any value greater than 0
Area	Area of the city split between land, water, and total	Any value greater than 0
Male	Total number of males in city.	Any value greater than 0
Female	Total number of females in city.	Any value greater than 0
Race	Race demographics	Any value greater than 0
Household	Types of households(family	Any value greater than 0

	or nonfamily)	
Household Size in Renter-Occupied Homes	Number of people in each rent occupied housing unit	Any value greater than 0
Educational Attainment	Shows the number of people 25 years and over who received each level of education	Any value greater than 0
Employment Status	Breaks down the number of people in the labor force and the number of people not in the labor force	Any value greater than 0
Industry by Occupation	Shows number of people working for each type of industry	Any value greater than 0
Average Household Income	Shows the average household income	Any value greater than 0
Gini Index	Represents the degree of inequality in a distribution of income	0 <= value <= 1
Means of Transportation	Breaks down the number of people using each type of transportation	Any value greater than 0
Larceny	Actual number of lacernies committed	Any value greater than 0
Robbery	Actual number of robberies committed	Any value greater than 0
Burglary	Actual number of burglaries committed	Any value greater than 0
Rape	Actual number of sexual assualts committed	Any value greater than 0
Motor Vehicle Thefts	Actual number of vehicle thefts reported	Any value greater than 0
Rape Legacy	Actual number of sexual assualts committed	Any value greater than 0

Violent Crime	Actual number of violent crimes reported	Any value greater than 0
Human Trafficking	Actual number of human trafficking reports	Any value greater than 0
Arson	Actual number of arsons committed	Any value greater than 0
Aggravated Assault	Actual number of aggravated assaults reported	Any value greater than 0
Homicide	Actual number of homicide reports	Any value greater than 0
Property Crime	Actual number of property crimes	Any value greater than 0

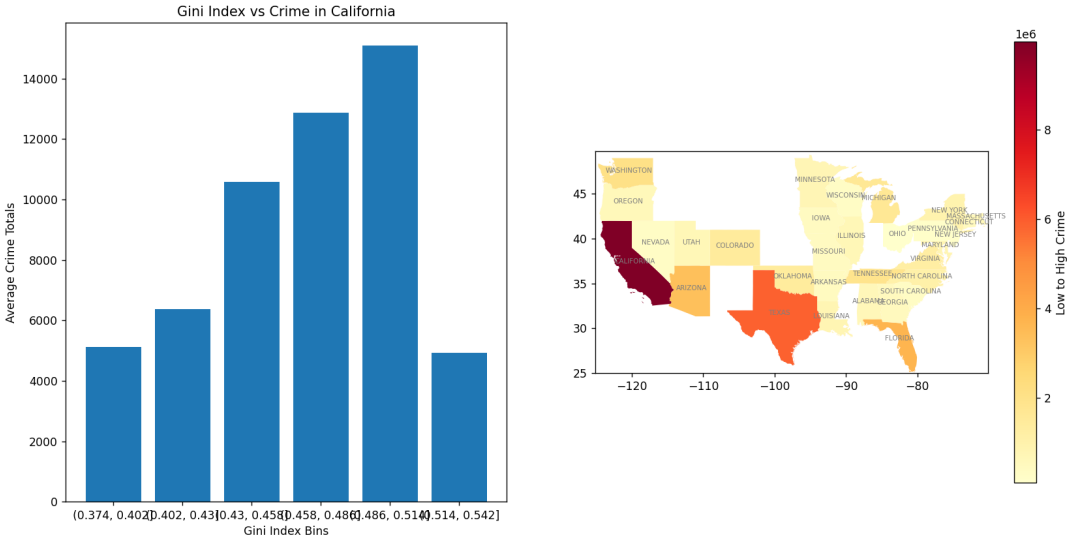
Data Mining

This dataset can potentially be used for both clustering and classification. We can cluster cities with a similar crime history and we can see if there is any overlap between these cities in terms of the city attributes. We can also classify future crime statistics based on these city attributes in a city. Ultimately, these conclusions can be applied to the population of all cities.

Data Analysis

Infographic

This infographic shows the total amount of crime in each city of each state across the time period 2010-2019. It uses a colormap to show how low or high the total crime is in each state. From this map, it was clear that California had the most amount of crime (this is partly due to the fact that in my city dataset California had the most amount of city data), so I wanted to see how this crime in California varied with the Gini Index- a coefficient representing income inequality in a city with 0 representing true equality and 1 representing true inequality. I binned the Gini indices into 6 bins so it would be easier to see visually which range of Gini indices contributed to what total crime counts.

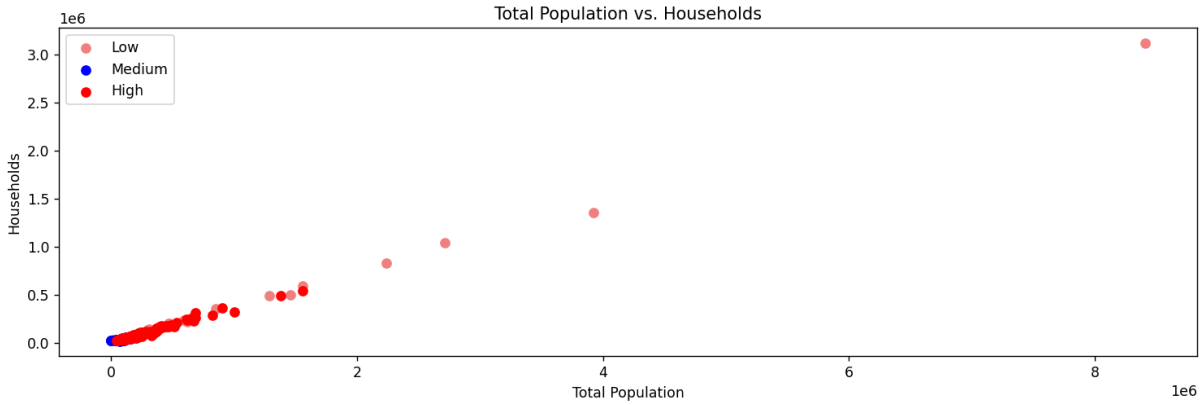


Visualization

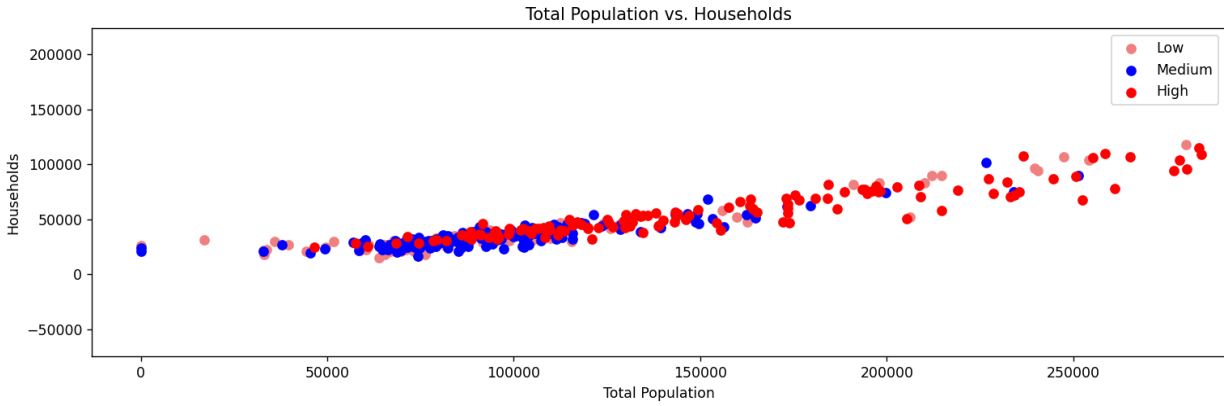
For the visualization, I picked the numerical attributes of total population, and households from the city dataset. The possible values of these attributes are unsigned integers. However in the case that the data did not have this information about these attributes for some city, I used -1 to indicate this null value. I also picked the crime attributes as well. For this visualization, however, I added the crime counts of each city across the years and I took the average.

	Median	Mean	STD
Total Population	102502.65	198861.49405349753	488683.1703283412
Households	37483.85	73548.61850786983	179689.27559219574
Total Crime	5812.875	9561.9960921143	14329.828552749997

To visualize how these two city attributes relate to each other and crime, I plotted a scatter plot. I binned the numerical attribute total crime into 3 separate bins - Low, Medium, High - based on tertiles to identify the points in this plot.



Looking at this plot at face value confirmed two things: my belief that total population and households had relatively positive linear correlation and that there were outliers in my dataset.

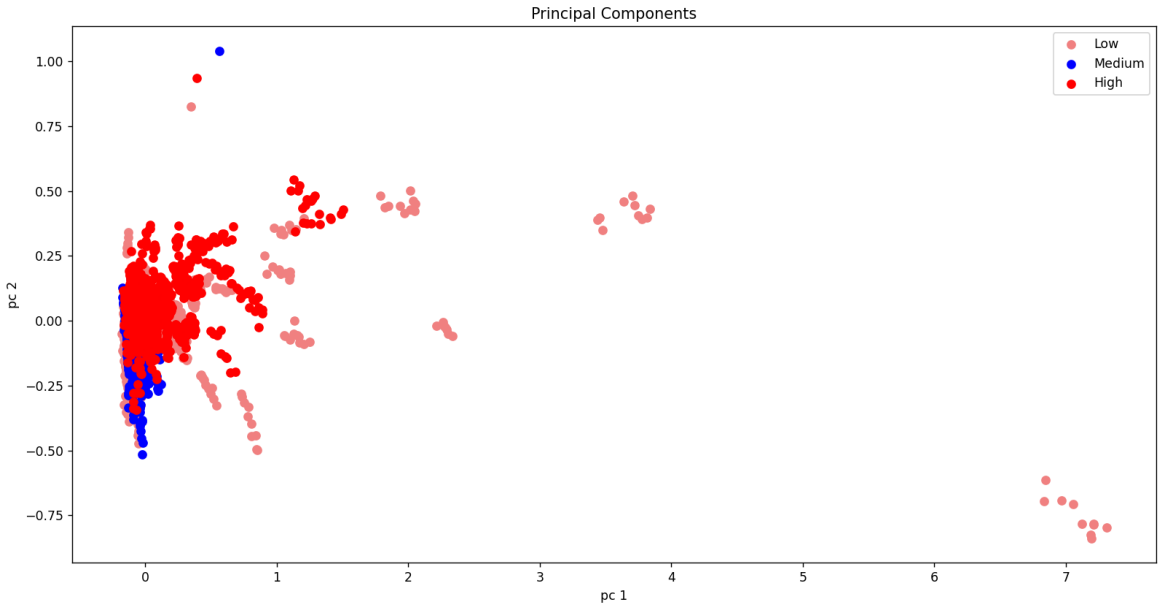


Zooming into this plot, however, I was able to learn that, generally speaking, as both total population and number of households increased, the chance of High Crime appearing also increased.

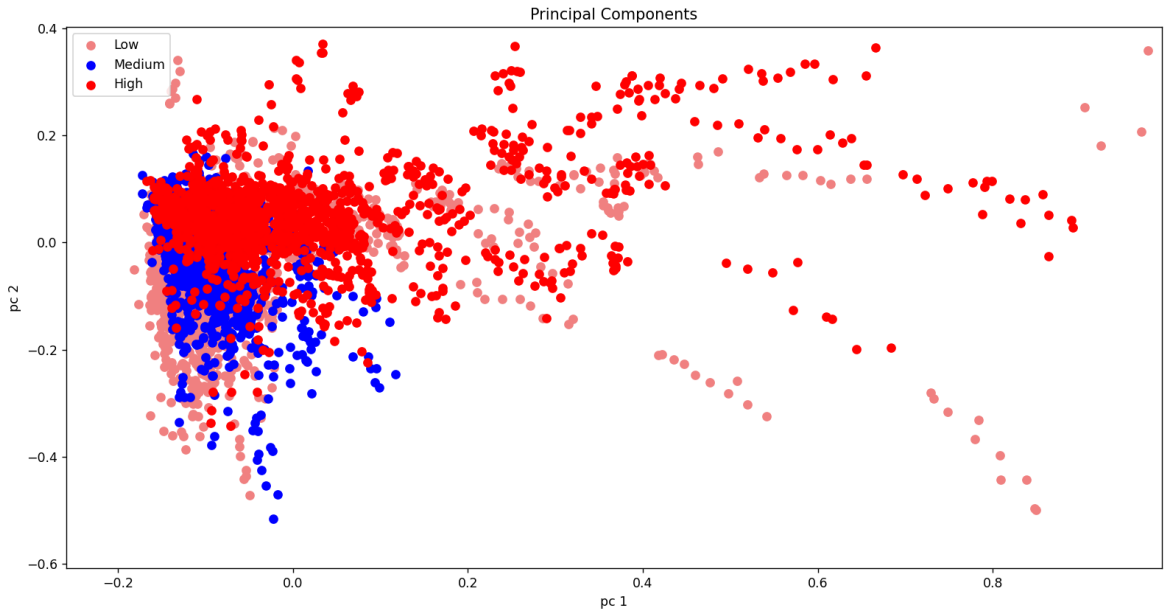
Data Mining Suitability

My dataset has high-dimensional data; it has around 73 city attributes and 11 crime attributes. As a result, to see if it was suitable for data mining I first reduced the dataset. Namely,

I used Principal Component Analysis to reduce my city attributes to 2 and I binned the crime attributes by totaling all the crime and putting them into Low, Medium, or High based on tertiles. I, then, plotted a scatter plot that incorporated this information.



This plot, again, shows that some outliers exist in my dataset. Zooming into this plot, reveals the following:

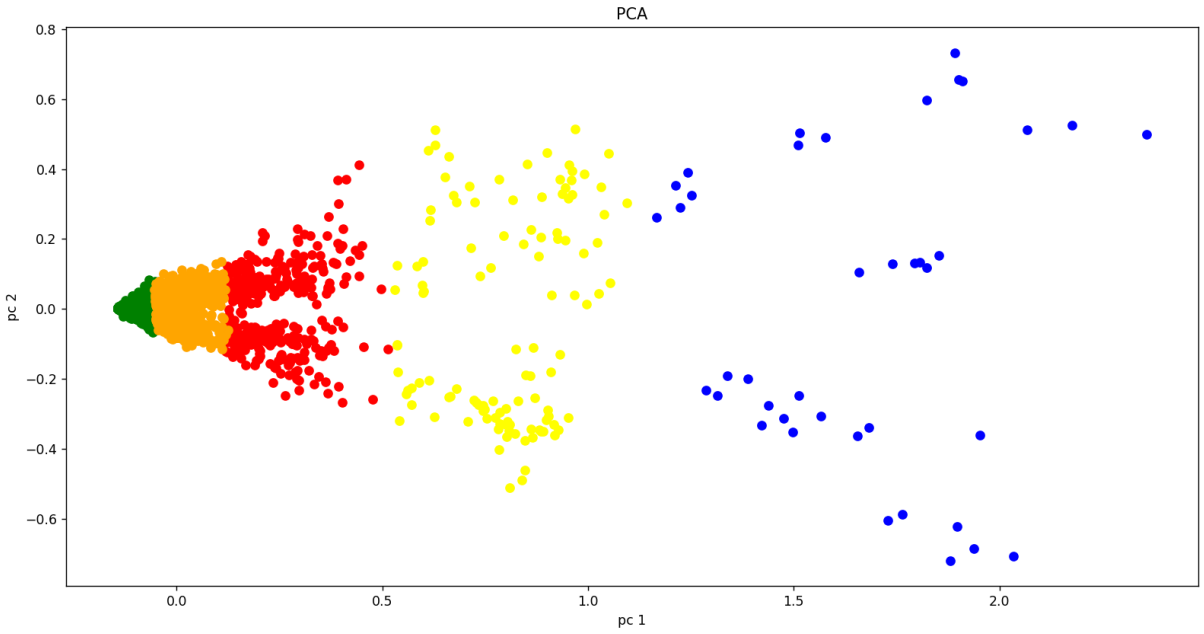


Although it is not very “clean”, I can somewhat see clusters of High and Medium Crime. As a result, I believe that my dataset can still be used for clustering as well as classification.

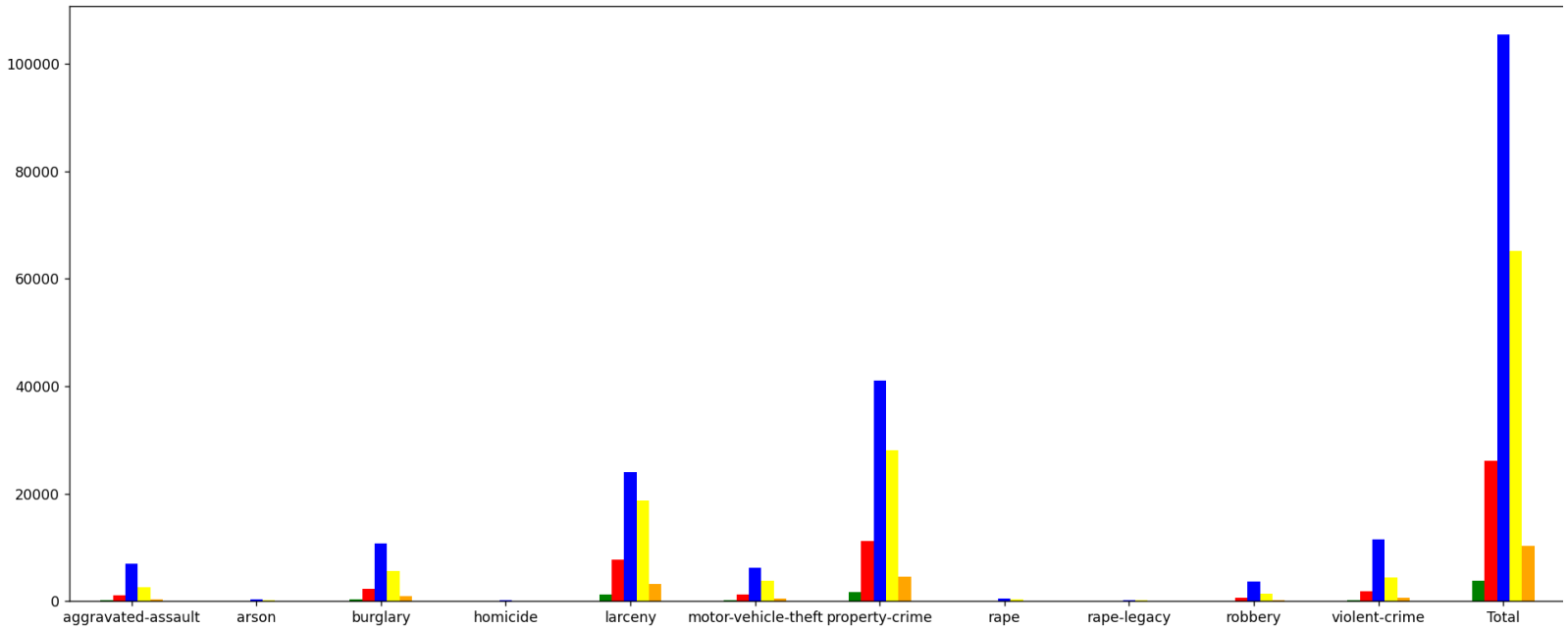
Clustering

I utilized Lloyd's algorithm to cluster my own dataset; for this, I used the crime attributes for clustering with the intent that every cluster will contain data instances that have similar crime statistics such that I could, then, observe the city attributes of each cluster and perhaps determine what city attributes correlate to what "crime pattern".

My implementation of Lloyd's algorithm is standard in the sense that it clusters data instances based on centers after which, based upon the max iteration and epsilon parameters, it will iterate by determining the average data instance in each cluster which will then become the new center for the next iteration. The inputs to this algorithm in my analysis.py are the normalized crime attributes, 5 clusters, a max iteration of 100, and an epsilon of 0. In order to visualize my clusters, I used Principal Component Analysis to reduce the dimensionality of my crime attributes to 2 so that I can plot a scatter plot.

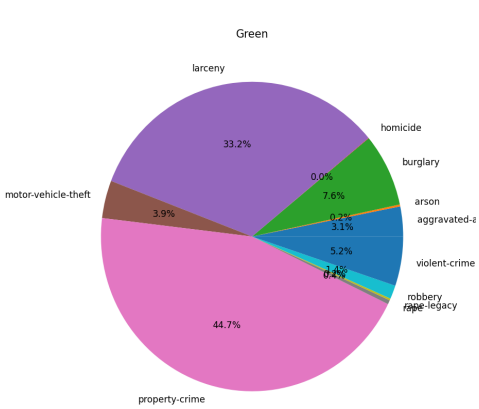


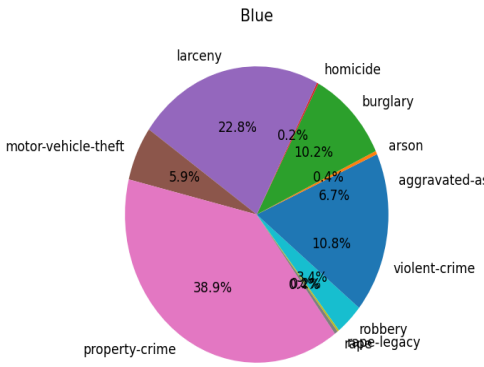
The visualization of my PCA shows that there are identifiable clusters but certain clusters contain high variation perhaps indicating that there are outliers.

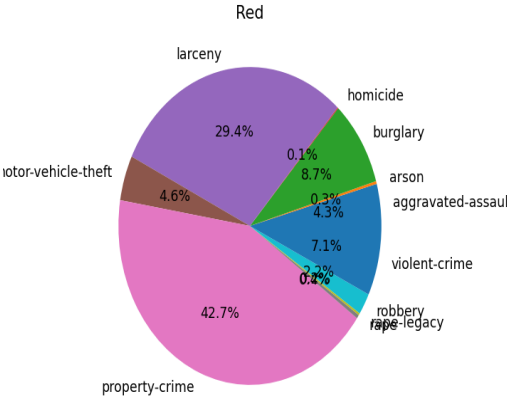


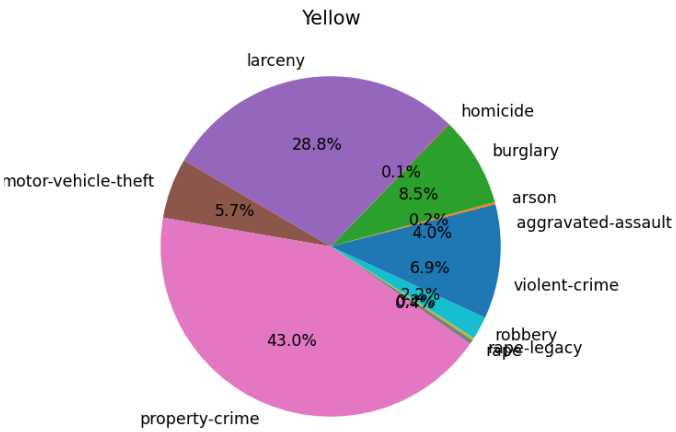
Based upon this bar graph which shows the average crime statistic in each cluster, it seems that the main factor that determines whether a data instance lies in a certain cluster is determined by the scale of the crime. Put simply, the clusters seem to represent the range between high crime and low crime where data instances that are toward the lower end of the spectrum are placed in certain clusters and instances that are toward the higher end are placed in other clusters.

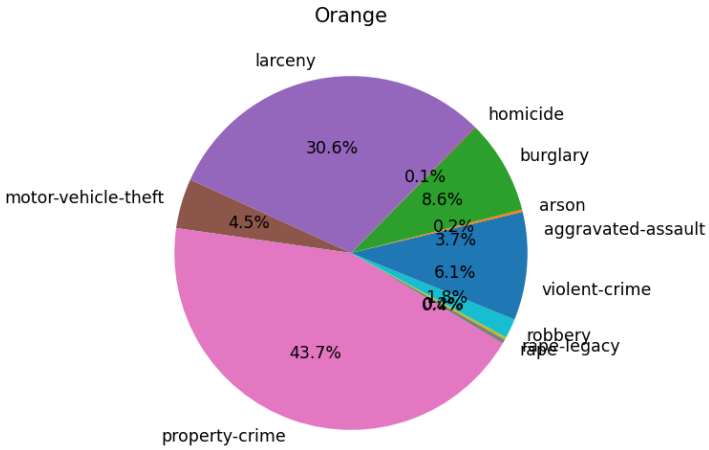
Unlike the expectation I had where each cluster will hold different instances that have different and prominent crime patterns(e.g., the breakdown of the most prominent crimes in cluster 1 is 50% property crime, and 25% burglary with the breakdown in cluster 2 being 75% violent-crime, and 10% arson), the breakdown of all the crime attributes in each cluster is relatively the same as evidenced by the following pie charts.





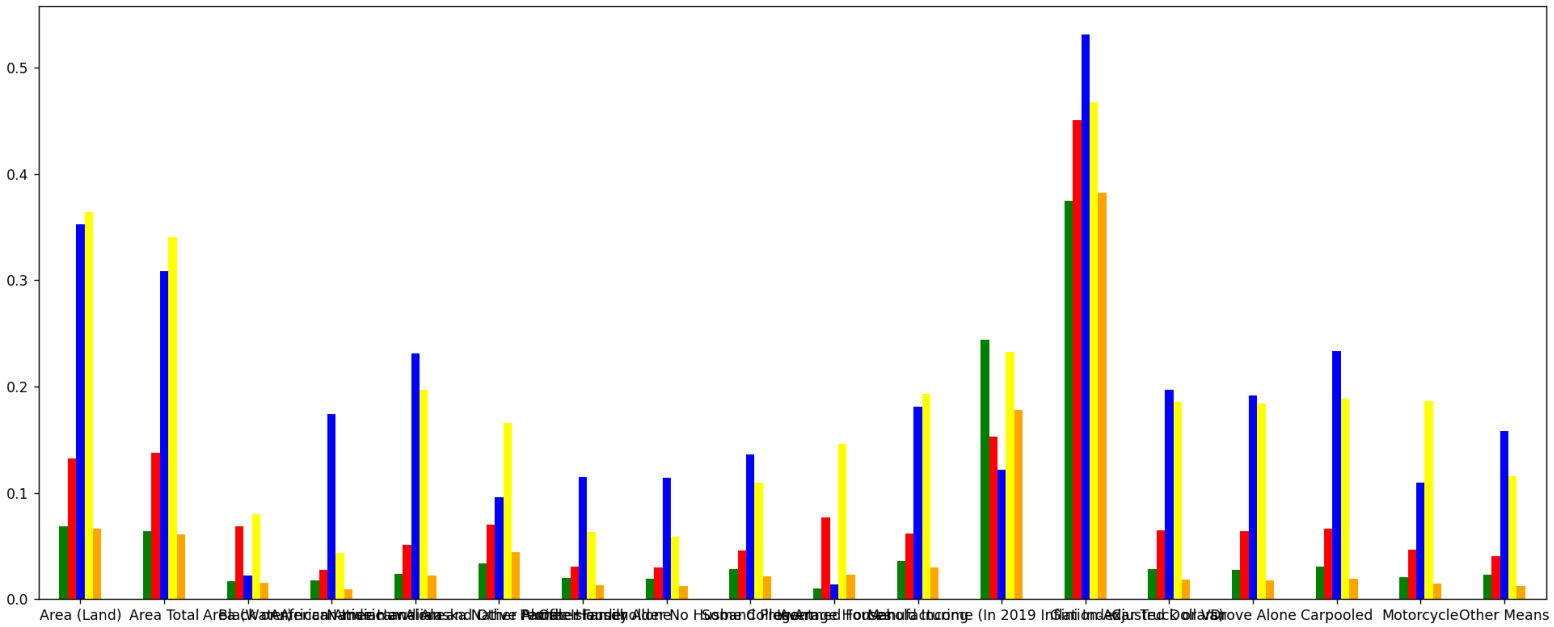




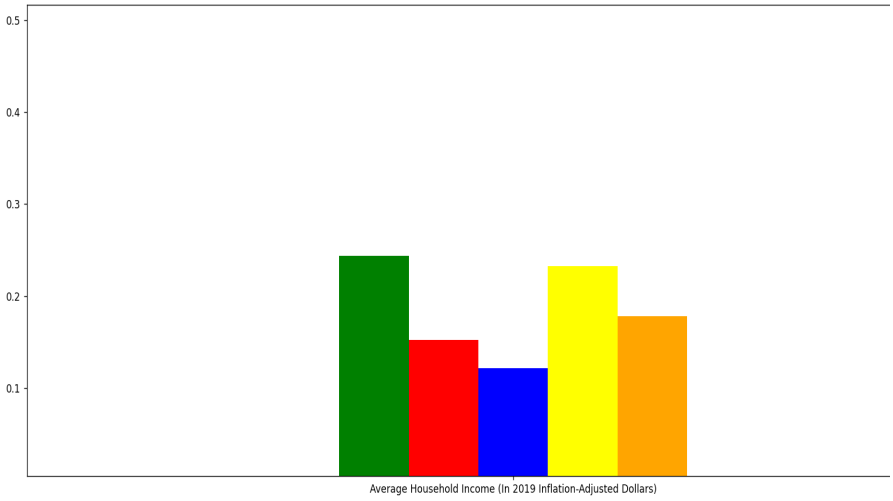


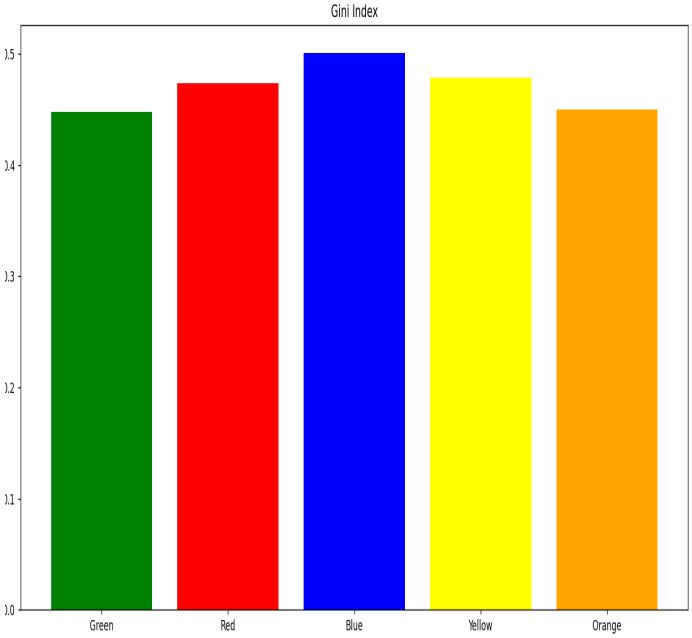
As a result, the main determining factor in regards to what cluster an instance lies in is the scale of the crime statistics of that instance.

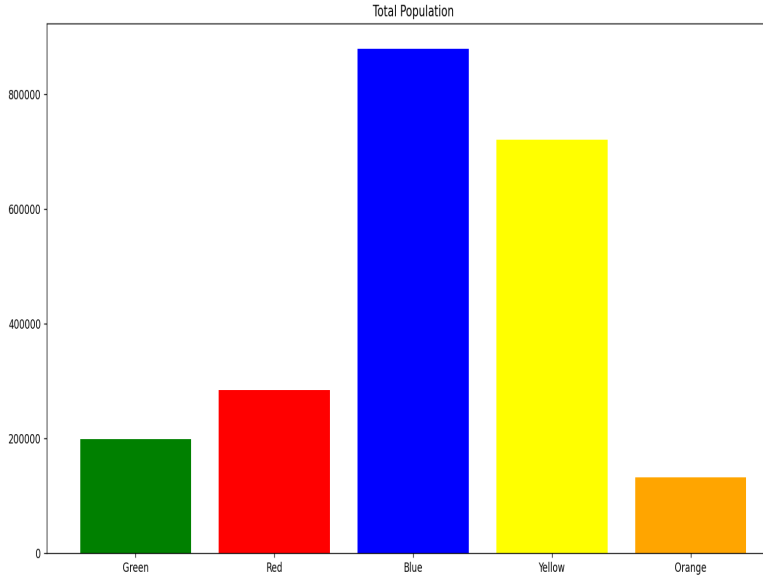
To visualize the average values of the city attributes in each cluster, I plotted a bar graph. Since I have too many city attributes, I selected a city attribute to be included in this bar graph based upon how the average difference between the normalized attribute in each cluster compared to the average of the average differences. If that difference was higher, I selected it to be included in my plot.



The main thing of note is that most of the bars in this plot follow the same pattern as that of the previous bar graph where the scale is what is being shown with the implication being, in general, as the numerical value of each city attribute increases, so does the total crime in that city. This idea seems to be reasonable since logically the more people you have in a city the more crime you have. Most of the city attributes can be related to the total population. In general as the population increases, as do the households, number of people carpooling, number of people working in construction, area of the city, etc. However this is not always the case. The bar of the average household income, along with a few other attributes, do not follow this pattern. The total crime or population seem to not correlate with these attributes. Additionally, an attribute like the Gini Index, which describes the income inequality in a city, cannot also really be correlated with the population. Nevertheless, for the most part most of the city attributes seem to do somewhat correlate with the total population and in turn correlate with the total crime.







Based on these two graphs, it seems that the Gini Index has little to no impact on the crime statistics of each cluster as the Gini Indices of each cluster are similar. This was another unexpected result as I had believed that crime increases in a city as the income inequality also increases. As I mentioned previously, the population of a city seems to have the most impact on the crime statistics.

In conclusion, the clusters seem to be grouped by the total number of crimes with this being correlated with the total population of a city. This was somewhat unexpected as I had hoped that the clusters would reveal certain crime patterns which could be attributed to certain combinations of the city attributes. Based on this conclusion, it would be reasonable to assume that if I had clustered the other way(using the city attributes to determine cluster groupings), I would have gotten the same results as most of my attributes can be correlated with the total population; each cluster will show a portion in the range of the total population and total crime as it does now.

Frequent Pattern Mining

I implemented the Apriori algorithm to mine frequent patterns in my own dataset. My implementation of the algorithm is standard except that I added constraints in order to mine and observe specific patterns. In the join phase of the Apriori algorithm, I check if the union between two sets contains a specific item in the set of constraints. If it does, then I use this item set for the next iteration. For example, a constraint can be such that an element of the set "Total Crime Low", "Total Crime Medium", and "Total Crime High" has to be included in all item sets. As a result, all the frequent item sets will contain an element from this set of all total crime. This reduced my search space as it eliminated any sets that contain only city attributes. Similarly, I added constraints for the antecedents and consequences that can appear in a rule. This allowed me to get rules where the consequences are crimes and the antecedents are city attributes.

For my tests, I used two sets of data. One dataframe was general in the sense that I only used the categorized Total Crime attribute. The other used all the categorized crime attributes to get more specific frequent itemsets and rules.

```
-----Freq Itemsets Total Crime-----
Civilian High,In Labor Force High,Total Population High,Male High,Population 16 Years and Over High,Total Employed Civilian Population 16 Years and Over High,Civilian Population in L
abor Force 16 Years and Over High,Female High,Population 25 Years and Over High,Total Crime High,Employed High,Retail Trade High 0.18
-----RULES Total Crime-----
Civilian High,Civilian Population in Labor Force 16 Years and Over High,Female High,In Labor Force High,Population 25 Years and Over High,Total Population High,Employed High,Retail Trade High,Ma
le High,Population 16 Years and Over High,Total Employed Civilian Population 16 Years and Over High => Total Crime High 0.58
```

For the general total crime data frame, I used a support of .18. There was already an upper bound of .33 because I used tertiles to categorize my numerical attributes. As a result, the chances of my attributes appearing in a 1-itemset was already around 33%. Through experimentation, however, support thresholds in the range of .18-.22 yielded in item sets with more than one dimension. For my association rules, I used the Kulczynski metric with a threshold of .5. This resulted in what is shown above. The output shows what I already learned from my clustering analysis. A significant number of my city attributes are correlated with the total population attribute. When the total population is high, every other city attribute is also high with the ultimate consequence being high total crime. Interestingly enough, both my frequent itemsets and rules contain information only about high total crime. This is because as the number of dimensions of my item sets increase, the item sets that include low total crime and medium total crime become more infrequent. This is perhaps indicating that as the dimensions increase there are fewer city attributes that correlate with each other in regards to low and medium total crime. Or, as another way to put it, the city attributes that correspond to low and medium total crime become more varied which results in overall low support counts.

```
-----Freq Itemsets Total Crime Low-----
Female Low,Population 16 Years and Over Low,Total Crime Low,In Labor Force Low,Civilian Low,Civilian Population in Labor Force 16 Years and Over Low,Male Low 0.14
-----RULES Total Crime Low-----
Female Low,Population 16 Years and Over Low,In Labor Force Low,Civilian Low,Civilian Population in Labor Force 16 Years and Over Low,Male Low => Total Crime Low 0.48
```

```
-----Freq Itemsets Total Crime Medium-----
Other Family Medium,Female Householder No Husband Present Medium,Total Crime Medium 0.14
Civilian Medium,Civilian Population in Labor Force 16 Years and Over Medium,Total Crime Medium 0.14
Male Medium,Total Population Medium,Total Crime Medium 0.14
Female Medium,Total Population Medium,Total Crime Medium 0.14
-----RULES Total Crime Medium-----
Other Family Medium,Female Householder No Husband Present Medium => Total Crime Medium 0.46
Civilian Medium,Civilian Population in Labor Force 16 Years and Over Medium => Total Crime Medium 0.43
Male Medium,Total Population Medium => Total Crime Medium 0.44
Female Medium,Total Population Medium => Total Crime Medium 0.44
```

I ran two additional tests where I only focused on sets that included either low or medium total crime which resulted in the above outputs. The dimensions of these item sets are low again indicating that, for these two cases of total crime, there are only a core group of city attributes that are frequent.

Another interesting thing that I noticed from the itemsets in all three of the above outputs is the appearance of employment and “in labor force” city attributes. I expected that if there was low employment or fewer people in the labor force, there would be high crime. However the opposite is true for all three outputs (i.e when the number of people in the labor force is high, there is high crime, when it is medium there is medium crime, etc.).

Finally, I ran a test that utilized the data frame that included all the crime attributes. I made one minor modification in my Apripori algorithm to yield good results, however. Namely, I added a count variable in the join phase that counts how many elements from the set of constraints appear in the union between two item sets. If the count is only one, I added this union to be used for the next iteration. I did this in an effort to eliminate item sets that contain only crime attributes because as the cardinality of the item sets increase only the crime attributes appear together.

For example, when the range of the count is between the number of all possible crime attribute categories and 1, the output is the following where all the item sets are crime attributes:

```
-----Freq Itemsets ALL Crime-----
violent-crime High,burglary High,robbery High,property-crime High,larceny High 0.21
violent-crime High,burglary High,aggravated-assault High,property-crime High,larceny High 0.21
violent-crime High,robbery High,aggravated-assault High,property-crime High,larceny High 0.21
violent-crime High,burglary High,robbery High,aggravated-assault High,property-crime High 0.21
-----RULES ALL Crime-----
violent-crime High => property-crime High,burglary High,larceny High,robbery High 0.76
burglary High => violent-crime High,larceny High,robbery High,property-crime High 0.76
robbery High => violent-crime High,burglary High,larceny High,property-crime High 0.76
property-crime High => violent-crime High,burglary High,larceny High,robbery High 0.78
larceny High => violent-crime High,burglary High,robbery High,property-crime High 0.76
violent-crime High,burglary High => property-crime High,larceny High,robbery High 0.83
violent-crime High,robbery High => property-crime High,burglary High,larceny High 0.75
violent-crime High,property-crime High => burglary High,larceny High,robbery High 0.87
violent-crime High,larceny High => property-crime High,burglary High,robbery High 0.86
burglary High,robbery High => violent-crime High,larceny High,property-crime High 0.83
property-crime High,burglary High => violent-crime High,larceny High,robbery High 0.82
```

With a more strict range, I was able to get rules that included the city attributes. However, the overall output was still dominated by the item sets that only included crime attributes.

```
In Labor Force High,Civilian High,Civilian Population in Labor Force 16 Years and Over High => larceny High 0.59
```

As a result, I opted to make it even more strict such that the count can only be equal to 1 which result in the following output:

```
-----Freq Itemsets ALL Crime-----
Male High,Population 16 Years and Over High,property-crime High,Total Population High 0.20
Population 16 Years and Over High,property-crime High,Total Population High,Female High 0.20
Civilian High,In Labor Force High,Civilian Population in Labor Force 16 Years and Over High,larceny High 0.20
Male High,property-crime High,Total Population High,Female High 0.20
Civilian High,In Labor Force High,property-crime High,Civilian Population in Labor Force 16 Years and Over High 0.20
-----RULES ALL Crime-----
Total Population High,Male High,Population 16 Years and Over High => property-crime High 0.60
Total Population High,Female High,Population 16 Years and Over High => property-crime High 0.60
Civilian High,In Labor Force High,Civilian Population in Labor Force 16 Years and Over High => larceny High 0.59
Total Population High,Male High,Female High => property-crime High 0.61
Civilian High,In Labor Force High,Civilian Population in Labor Force 16 Years and Over High => property-crime High 0.60
```

The results from this test were also similar to what I got from the other tests I ran.

Classification

For this lab, I implemented the decision tree algorithm and used my existing dataset. My implementation is standard in the sense that it recursively builds interior nodes by splitting my dataset based on some column, calculating the entropy of each resulting piece, and choosing the best column that results in the most information gain. In addition to this basic algorithm, I also implemented two post-pruning algorithms in order to combat overfitting. The first one is basic in that it prunes the tree if the current depth is greater than some max depth value. The other prunes the tree based on the number of misclassifications that result from using the validation data set. For this algorithm to function, I added majority and node error attributes to each node. The majority attribute represents the majority class if this node was a leaf and the node error represents the number of misclassifications that result when this majority attribute is different from the actual class of some validation data instance. I, then, collected all the "twigs", nodes with only leaves as children, in the tree. For each twig, I get the leaf error, the number of misclassifications this twig will make if it is a leaf, and the node error, the number of misclassifications this decision node is currently making. I subtract the leaf error from the node error and this value represents the priority at which the twig should be deleted. Only twigs with node errors greater than leaf errors will be deleted. As a result, twigs with low leaf errors and high node errors will be converted to leaves first. This algorithm will delete n number of twigs until the total number of leaves deleted is greater than or equal to the user provided num_leaves_to_prune value or if there are no more viable twigs to be deleted.

I, first, ran a test with a decision tree without any pruning. This resulted in what is pictured below:

```
DECISION TREE WITHOUT PRUNE
Height of Tree:  74
Decision Tree Training Accuracy: Accuracy: 0.99
Decision Tree Training  Precision:  0.9902554715414289
Decision Tree Training  Recall:  0.9902552204176334

Initial Leaves:  691
Decision Tree Validation Accuracy: Accuracy: 0.67
Decision Tree Validation  Precision:  0.671176374526559
Decision Tree Validation  Recall:  0.6710893854748603

Decision Tree Testing Accuracy: Accuracy: 0.68
Decision Tree Testing  Precision:  0.6848090611652861
Decision Tree Testing  Recall:  0.6841463414634147
```

The results for the training set were extremely high unlike the results from the validation and testing sets. This was expected since the tree was most likely overfitting. As a result, I ran

another test this time using the validation set to prune the tree by deleting all the twigs with node errors greater than leaf errors. I opted to not prune the tree again by a max depth since the depth of the pruned tree was already reduced as a result of the deleted twigs.

```
DECISION TREE WITH PRUNE
Height of Tree:  74
Decision Tree Training Accuracy: Accuracy: 0.99
Decision Tree Training  Precision:  0.9902554715414289
Decision Tree Training  Recall:  0.9902552204176334

Initial Leaves:  691
Leaves After Prune:  397
Height After Prune:  47
Decision Tree Validation Accuracy: Accuracy: 0.72
Decision Tree Validation  Precision:  0.721718847629049
Decision Tree Validation  Recall:  0.7213687150837989

Decision Tree Testing Accuracy: Accuracy: 0.69
Decision Tree Testing  Precision:  0.6879725575561942
Decision Tree Testing  Recall:  0.6865853658536586
```

Although my validation results have improved, the testing results have remained relatively the same. As a result of this and because of the fact that my decision trees were still heavily reliant on my actual training and validation set, as I was getting different trees each time I ran my tests, I wanted to use a random forest classifier to see if I can improve my testing results. I ran two tests with the random forest classifier. The first test uses the base random forest classifier with all the columns whereas the second only uses the top thirty most important columns identified from the previous test along with a n_estimators value of 2000. The first test resulted in the following results for the training, validation, and testing sets along with top thirty most important features:

```
RANDOM TREE BASE
Random Forests Tree Training Accuracy: Accuracy: 0.99
Random Forests Tree Training  Precision:  0.9902725937020167
Random Forests Tree Training  Recall:  0.9902552204176334

Random Forests Tree Validation Accuracy: Accuracy: 0.74
Random Forests Tree Validation  Precision:  0.7559370895695647
Random Forests Tree Validation  Recall:  0.7381284916201117

Random Forests Tree Testing Accuracy: Accuracy: 0.74
Random Forests Tree Testing  Precision:  0.7428221860674307
Random Forests Tree Testing  Recall:  0.7390243902439024
```

```
Area Total                                                          0.022195
Average Household Income (In 2019 Inflation-Adjusted Dollars)       0.020366
Population Density (Per Sq. Mile)                                   0.019268
Area (Water)                                                        0.017927
Black or African American Alone                                     0.017195
Area (Land)                                                         0.014878
Gini Index                                                          0.010000
Bachelor's Degree                                                   0.009756
Renter-Occupied Housing Units                                       0.009634
Nonfamily Households                                                0.007439
Some Other Race Alone                                               0.007439
Asian Alone                                                         0.005732
Total Population                                                    0.005366
White Alone                                                         0.005000
Other Family                                                        0.004756
Less than High School                                               0.004756
Households                                                          0.004390
Public Transportation (Includes Taxicab)                            0.004390
In Armed Forces                                                     0.004024
Female Householder No Husband Present                               0.004024
Two or More Races                                                   0.003902
Female Householder                                                  0.003293
Manufacturing                                                       0.003293
Walked                                                              0.003171
Male                                                                0.002927
Male Householder                                                    0.002805
3-Person Household                                                  0.002683
Arts Entertainment and Recreation and  Accommodation and Food Services  0.002561
2-Person Household                                                  0.002439
Public Administration                                               0.002195
dtype: float64
```

I was expecting to see total population listed as one of the most important attributes. However, I was surprised to find that it was ranked somewhat low. The way the random forest

classifier finds the most important features is based on the information gain. The reason why total population would be ranked low is because the entropy of this attribute will be high or, when the dataset is split based on this attribute, the labels in each piece would be varied. This seems almost contradictory to my clustering results as in those results each cluster was distinct in relation to total population. More specifically, the crime range in each piece or cluster corresponded to a certain total population threshold. Here, it seems to be the opposite as an attribute like the Gini Index, which did not vary across my clusters, is listed as important with the implication being that the entropy for this attribute is low or when the data set is split with this feature the labels in each piece are less varied; unlike my clustering results, where again the Gini Index of each cluster was relatively the same or each piece was varied in the sense that a Gini Index could have corresponded with multiple crime ranges. This could be a consequence that results from binning my numerical data. The bins are, essentially, too "discrete" and I lose some of the interesting relationships. Nevertheless, the subsequent test that only used the top thirty most important features resulted in a better result for my test set as pictured below:

```
RANDOM TREE WITH BEST FEATURES
Random Forests Tree Training Accuracy: Accuracy: 0.98
Random Forests Tree Training  Precision:  0.9820289965521637
Random Forests Tree Training  Recall:  0.9819025522041763

Random Forests Tree Validation Accuracy: Accuracy: 0.74
Random Forests Tree Validation  Precision:  0.7566238619208987
Random Forests Tree Validation  Recall:  0.7353351955307262

Random Forests Tree Testing Accuracy: Accuracy: 0.77
Random Forests Tree Testing  Precision:  0.7751398062404778
Random Forests Tree Testing  Recall:  0.774390243902439
```

In conclusion, although my decision tree and random forest classifiers were able to have reasonably high results in terms of the accuracy, precision and recall, I believe that if I had used a regression model such that I did not bin my city or crime attributes, I would have gotten significantly better results as I would have been able to retain some of the interesting relationships between the city and crime attributes.