BUAN 6340.003: Programming for Data Science

Final Project Report

Crimes in Chicago Python Project

Prepared for:

Dr. Yingjie Zhang

Prepared by:

Adam Butcher | axb122730

Aneesa Noorani| axn180021

Maria Phetteplace | mxp180009

Teenaz Ralhan | txr140230

December 01, 2019

Table of Contents

1. Introduction

It is common knowledge that Chicago has amongst the highest crime rates for major metropolitan areas across the country. Even though overall crime has decreased over the past couple of years, it is still a major concern for city officials (Wire, 2019). Fortunately, the policymakers are taking action towards reducing crime rates by first gathering data on crime types, locations, times they are occuring, etc. Analyzing crime rates can help the City of Chicago in several ways: (i) identify locations where crime, and certain types of crime are more prevalent, (ii) better allocate police department resources, (iii) modify necessary city laws that might deter crime; and (iv) appease Chicago residents by making them feel like the city is indeed aware of the crime problem and that actions are being taken to address it.

2. Problem Statement

Given different features associated with crimes in Chicago including location, type of crime, date, crime description, FBI code, and beat, to name a few, can we accurately predict the number of crimes for a given time period, the location of a crime, the type of crime, and whether the perpetrator of the crime was arrested or not? We aim to accomplish this goal using our knowledge of Python data cleaning techniques, exploratory data analysis techniques, and machine learning.

3. Dataset Description

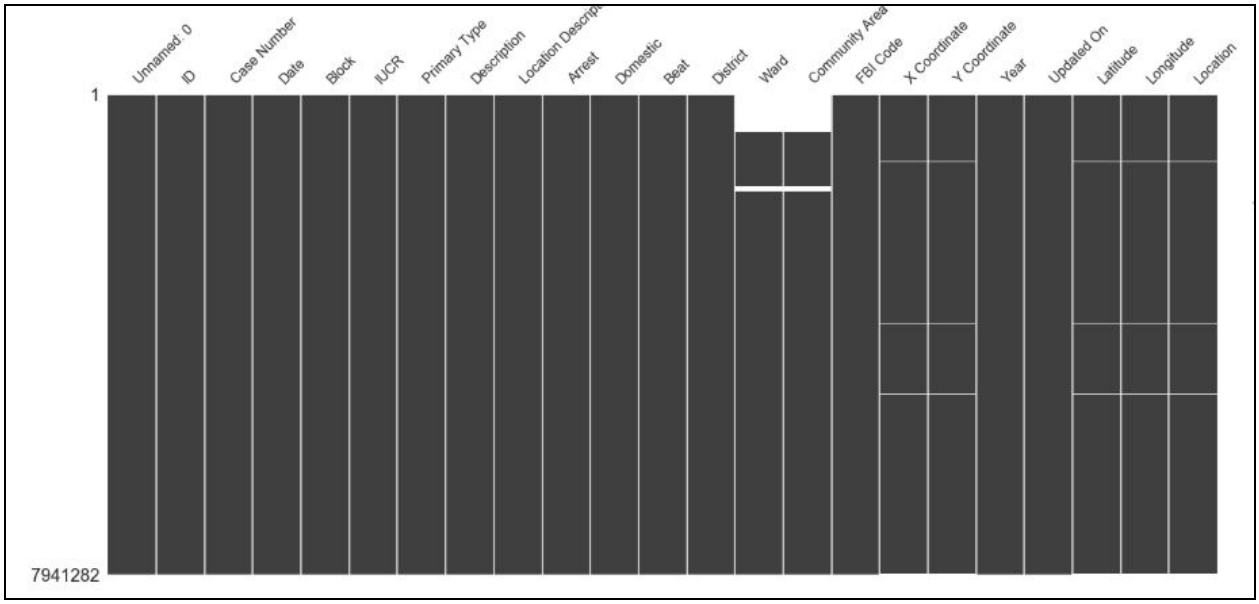
Our dataset came from Kaggle. The original format contained four CSV files which was 2 GB total. Each file had a set of different range of years. We concatenated them together to have a single data frame in python. Before making changes to the dataset, there were around 8 million rows and 23 columns. The original variables (from left to right) were ID, Case Number, Date, Block, IUCR, Primary Type, Description, Location Description, Arrest, Domestic, Beat, District, Ward, Community Area, FBI Code, X Coordinate, Y Coordinate, Year, Updated On, Latitude, Longitude, and Location. Here are the first 5 rows of our concatenated dataset.

Date	ID	Date	Block	Primary Type	Description	Location Description	Arrest	Domestic	Beat	District
2003-03-01 00:00:00	4676906	2003-03-01 00:00:00	004XX W 42ND PL	OTHER OFFENSE	HARASSMENT BY TELEPHONE	RESIDENCE	False	True	935	9.0
2003-05-01 01:00:00	4677901	2003-05-01 01:00:00	111XX S NORMAL AVE	THEFT	FINANCIAL ID THEFT:$300 &UNDER	RESIDENCE	False	False	2233	22.0
2001-01-01 11:00:00	4791194	2001-01-01 11:00:00	114XX S ST LAWRENCE AVE	CRIM SEXUAL ASSAULT	PREDATORY	RESIDENCE	True	True	531	5.0
2003-03-15 00:00:00	4679521	2003-03-15 00:00:00	090XX S RACINE AVE	OTHER OFFENSE	OTHER WEAPONS VIOLATION	RESIDENCE PORCH/HALLWAY	False	False	2222	22.0
2003-01-01 00:00:00	4680124	2003-01-01 00:00:00	009XX S SPAULDING AVE	THEFT	FINANCIAL ID THEFT: OVER $300	RESIDENCE	False	False	1134	11.0

Ward	Community Area	FBI Code	X Coordinate	Y Coordinate	Location
11.0	61.0	26	1173974.0	1.87676e+06	(41.817229156, -87.637328162)
34.0	49.0	06	1174948.0	1.83105e+06	(41.691784636, -87.635115968)
9.0	50.0	02	1182247.0	1.82938e+06	(41.687020002, -87.60844523)
21.0	73.0	26	1169911.0	1.84483e+06	(41.729712374, -87.653158513)
24.0	29.0	06	1154521.0	1.89576e+06	(41.869772159, -87.708180162)

4. Data Preprocessing

We discussed different ways to handle missing data within our dataset. Based on an article posted on TowardsDataScience by Boyan Angelov, here is a visualization that displays our missing data using the package, missingno and a breakdown of missing data per variable.



Based on this knowledge, we felt confident that it would not change our analysis since the proportion of missing values was relatively low.

Unnamed: 0	0
ID	0
Case Number	7
Date	0
Block	0
IUCR	0
Primary Type	0
Description	0
Location Description	1990
Arrest	0
Domestic	0
Beat	0
District	91
Ward	700224
Community Area	702091
FBI Code	0
X Coordinate	105573
Y Coordinate	105573
Year	0
Updated On	0
Latitude	105573
Longitude	105574
Location	105574

After discussing our goals for this project, we removed columns that were irrelevant for our purposes or repetitive in information. The variables that we dropped were Year 'Case Number, Unnamed: 0, IUCR, Updated On, Latitude, and Longitude.

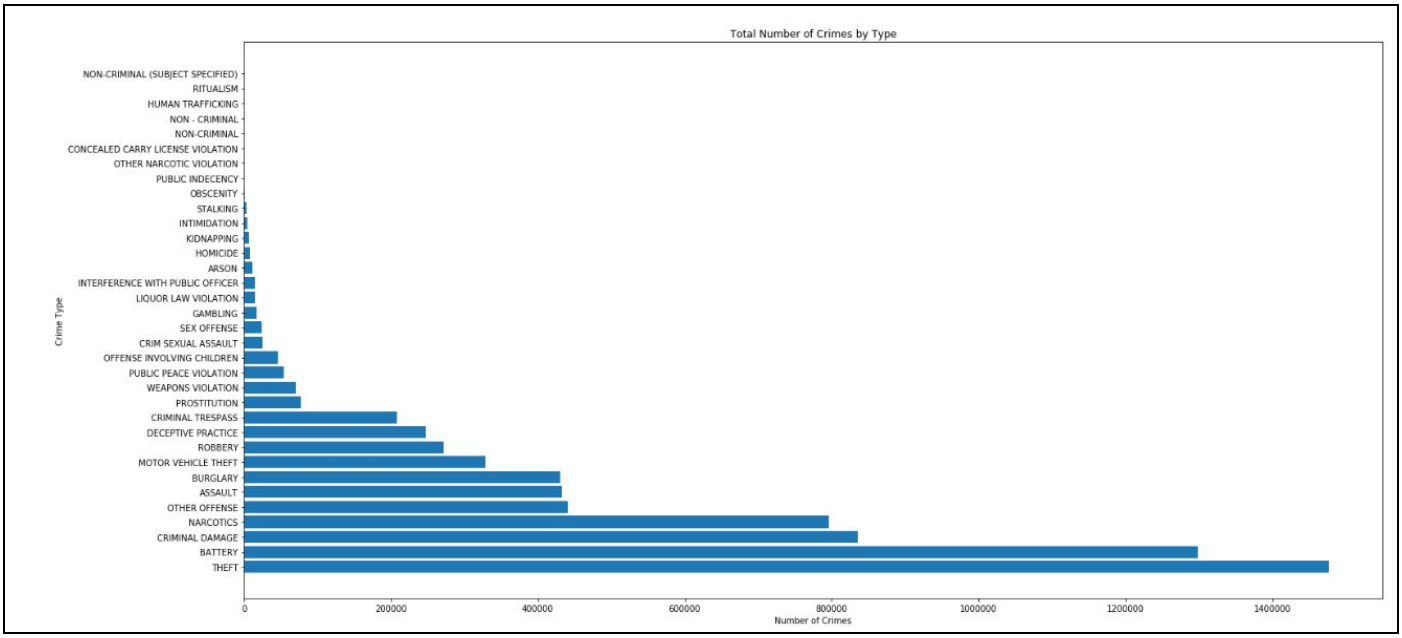
The Date column was in a string format, so we converted it to a datetime format. This made it easy to extract different time periods based on our later needs.

After we completed our initial visualization, we realized that ‘Primary Type” and ‘Location Description’ consisted of similar descriptions that could be grouped together. For example, there were 14 different location descriptions that all dealt with an airport vicinity. We collapsed those different location descriptions to be under one label, 'Airport/Aircraft.’ Doing this change, the accuracy of our models improved which will be addressed in a later section.

After the end of this preprocessing step, our new dataset shape was 7145219 rows and 22 columns.

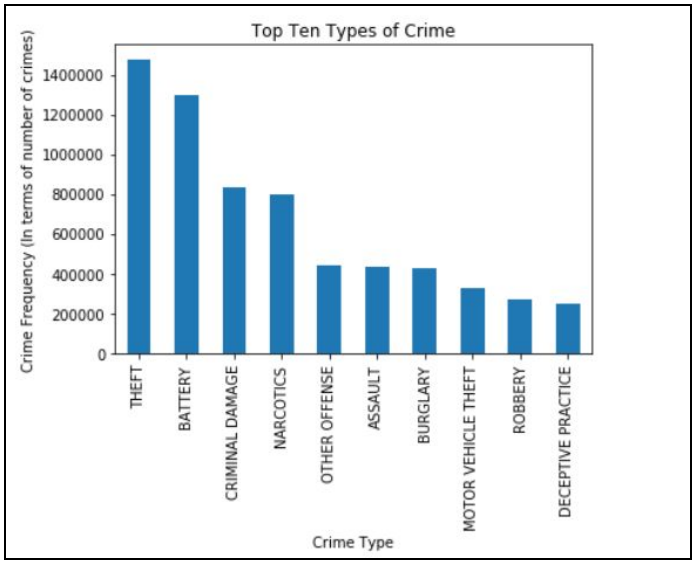
5. Exploratory Data Analysis

a. Total Number of Crimes By Type



In the plot above, we plot the number of crimes for each unique type of crime in the dataset. We see that there are 34 unique types of crime and that Theft and Battery are the most prevalent types of crime while Ritualism and Human Trafficking are the least prevalent types of crime.

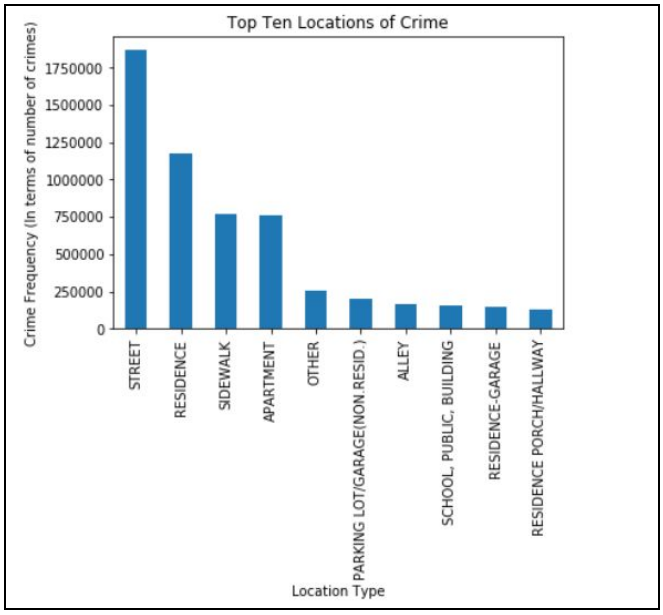
b. Top 10 Types of Crime



The plot above shows the top 10 crimes in the dataset out of a set of 34 unique crimes listed. We observe that Theft and Battery are the most prevalent crime types, which is in line with our

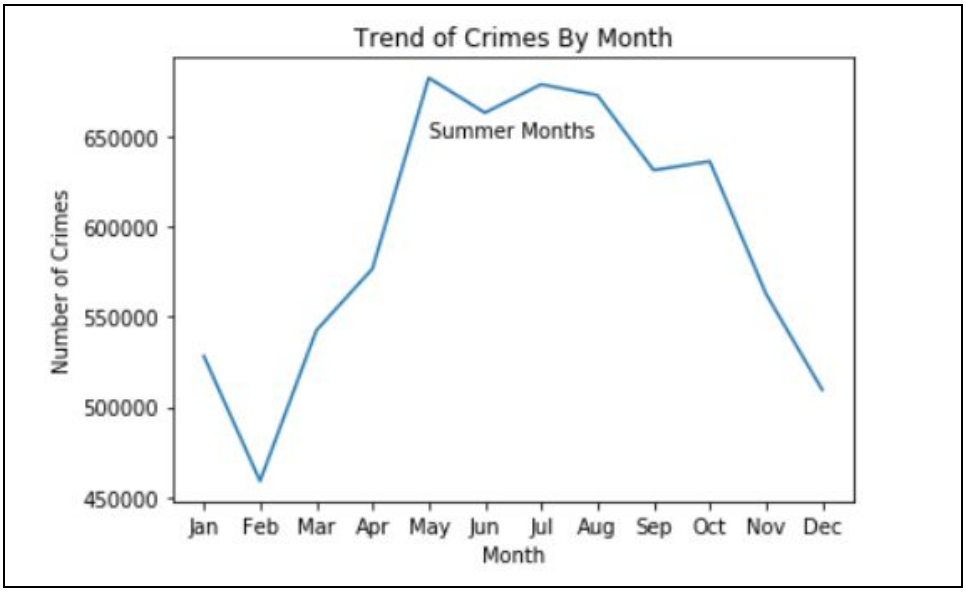
inference from the previous plot, and the Deceptive Practice is the least prevalent crime type in the top 10 types.

c. Top 10 Locations of Crime



In the plot above, we observe the top 10 locations of crime out of a set of 170 unique locations. We observe that the Street is the most prevalent location for a crime to occur in while a Residence's Porch/Hallway is the least prevalent location.

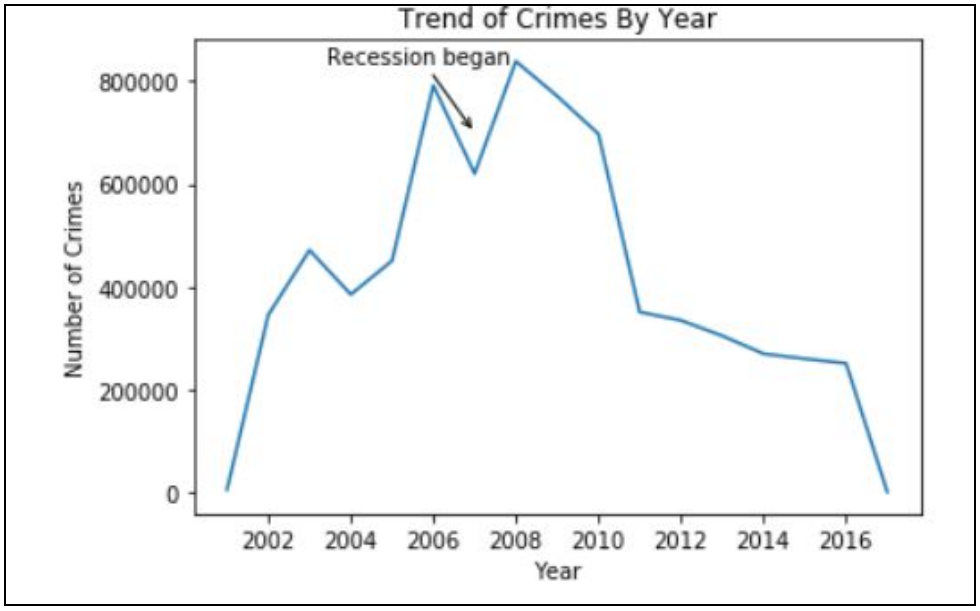
d. Trend of Crimes by Month



After counting the values of crime per month in this plot, it is clear that crimes spike heavily during the summer months. This trend intrigued us, so we tried to research why this is. Based

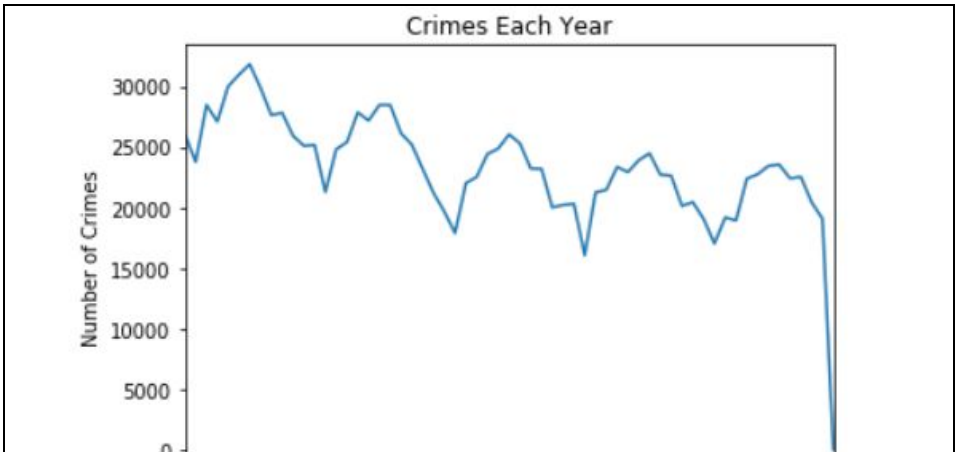
on our intuition, we thought that it was simply because more people were outdoors due to the good weather which led to more crimes. Although this seemed to be a valid reason, there may be more to it. According to a study conducted by the Statistical Analysis Center, "community centers are less active during the summer months therefore teens have less direction. This can lead to higher crime. Another reason that was explored that pertained to less serious assaults only is that those crimes tend not to come to the attention of the police unless they are public. They are more likely to be public in the summertime. In the warm months, an assault is more likely to occur outside, and if it occurs inside, the windows are more likely to be open." Thus, this is a reason for seasonality.

e. Trend of Crimes by Year



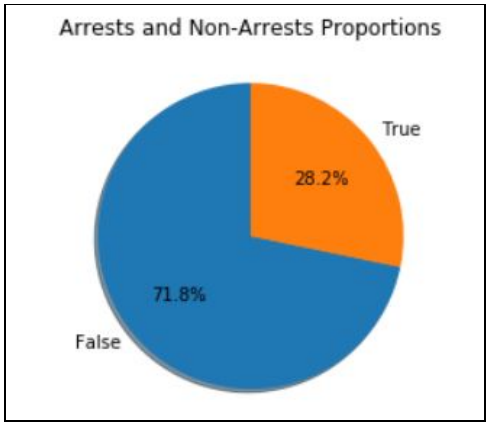
The plot above shows the number of crimes each year with a peak in 2006. The amount of crimes did decrease during 2007 and went at its highest in 2008 due to the recession. Many crimes were committed due to the rise in unemployment as a result of the recession. The amount of crimes dropped down a lot during 2011 with the number decreasing each year after 2011.

f. Crimes for Each Year from 2012 to 2017



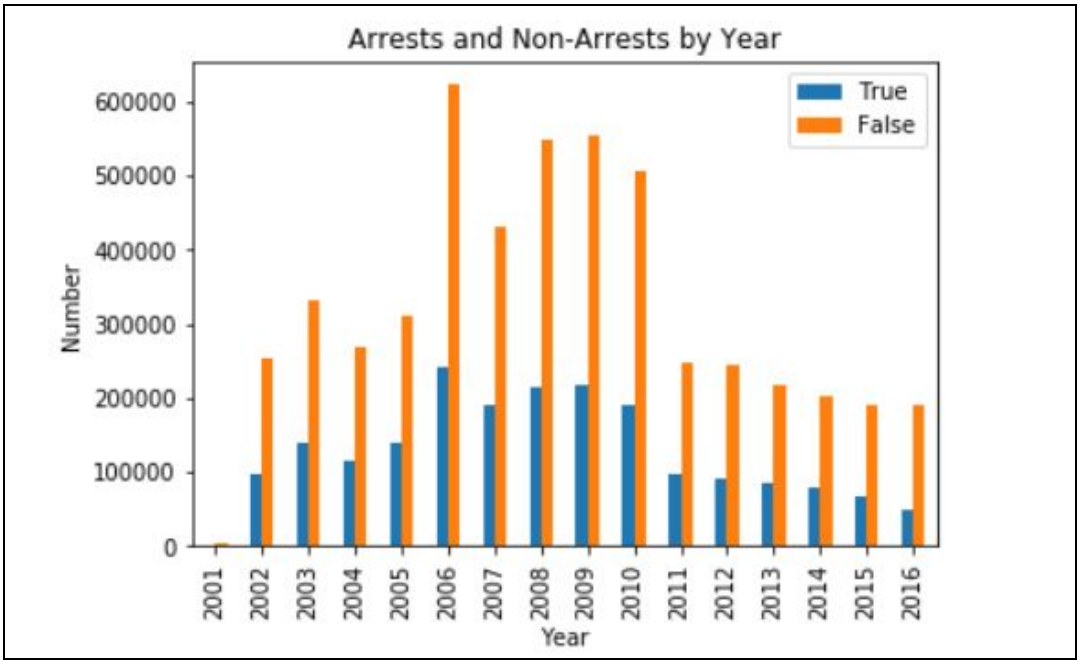
Looking at this plot, we see a downward trend in crime from years 2012 to 2017. There are only a few observations in 2017 which is why there is a dramatic plunge. Because of this, we dropped that year when completing a time series model.

g. Arrests and Non-Arrests Proportions



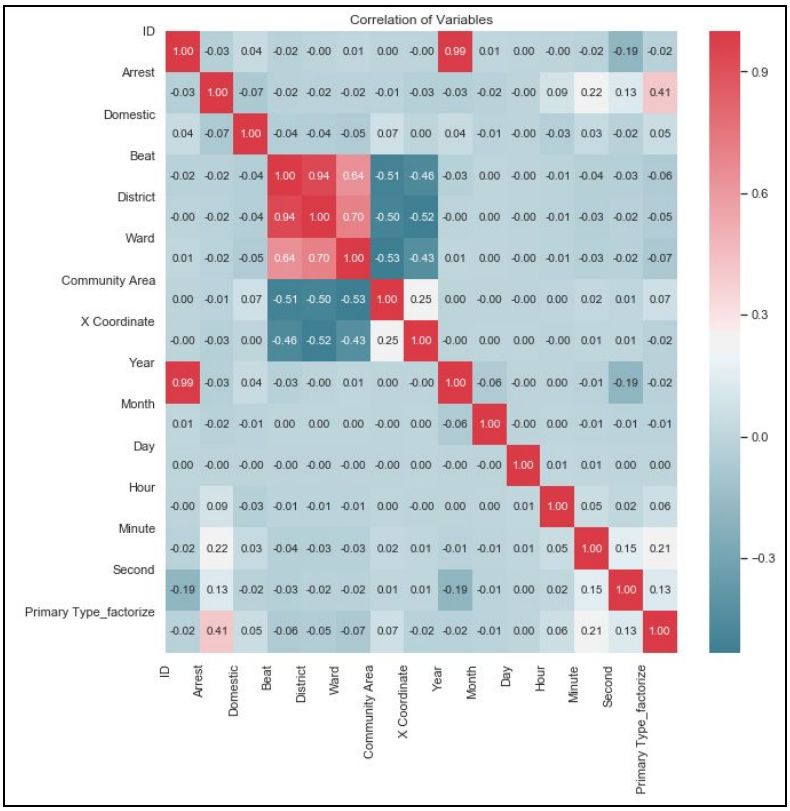
The data shows that most of the time, people don't get arrested for each crime committed. It has been said by experts that there has been a lack of communication between the people and the police. This is due to the fact that people tend to be afraid to speak out as they distrust the police (Bradley, 2019).

h. Arrests and Non-Arrest by Year



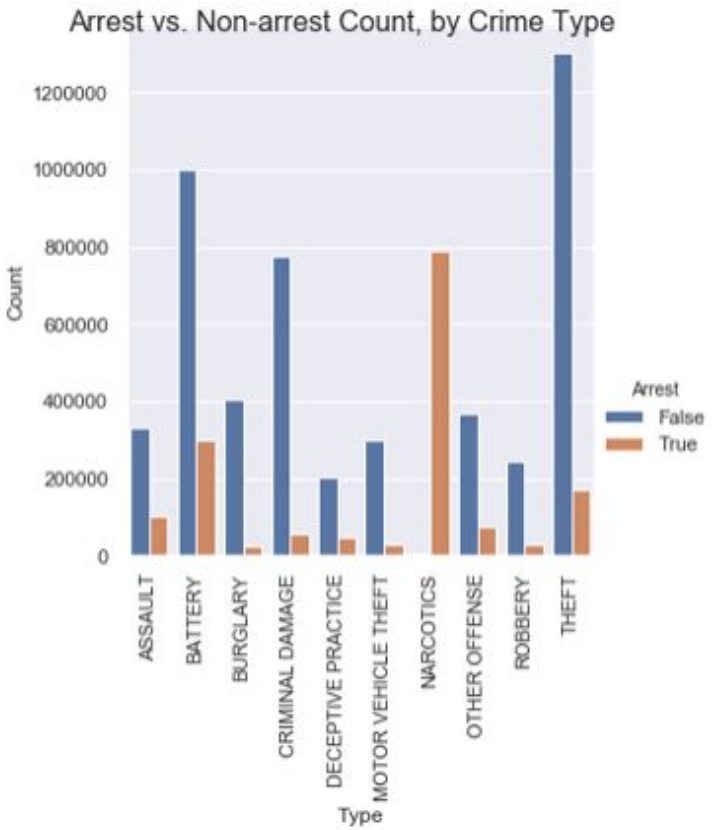
The data shows that the number of arrests made was at its peak in 2006. After that, the recession took place, and the number of arrests still remained very high. It was until 2011 when the number of arrests started to decline along with the number of crimes in general.

i. Correlation of Variables



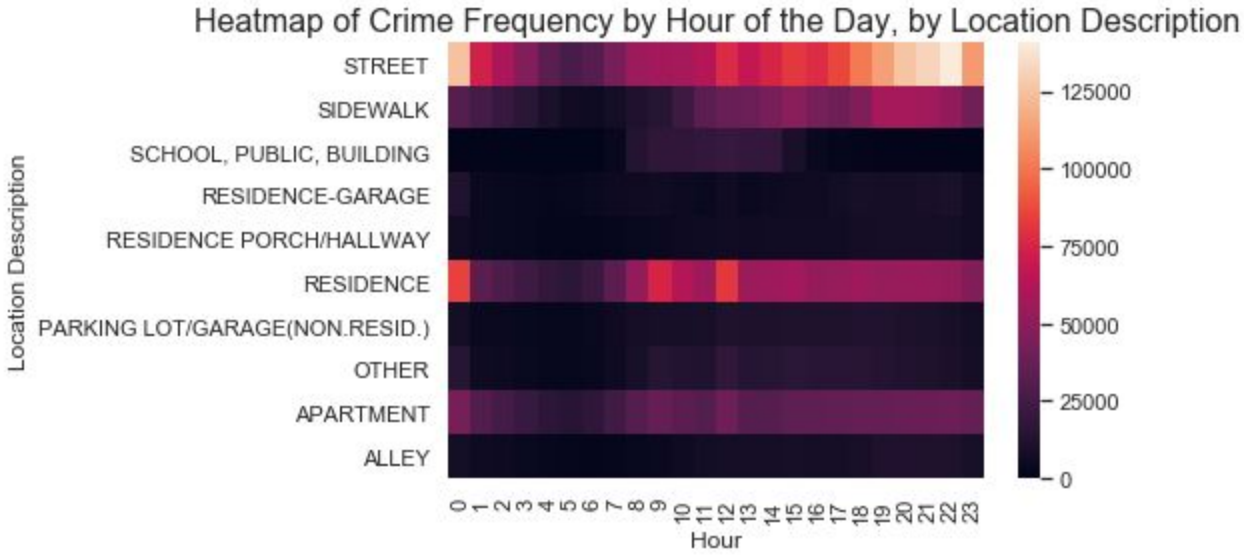
We wanted to see the correlation of variables, so we created a heat map. To include categorical data, we first had to factorize the variables which is what we did for 'Primary Type'. We can see that primary type and arrest are moderately correlated with each other.

j. Arrests vs. Non-Arrests by Crime Type



Of these top 10 crime types, only 'Theft' led to an exponentially higher number of arrests v s. Non-arrests. For type 'Narcotics,' hardly any arrests were made.

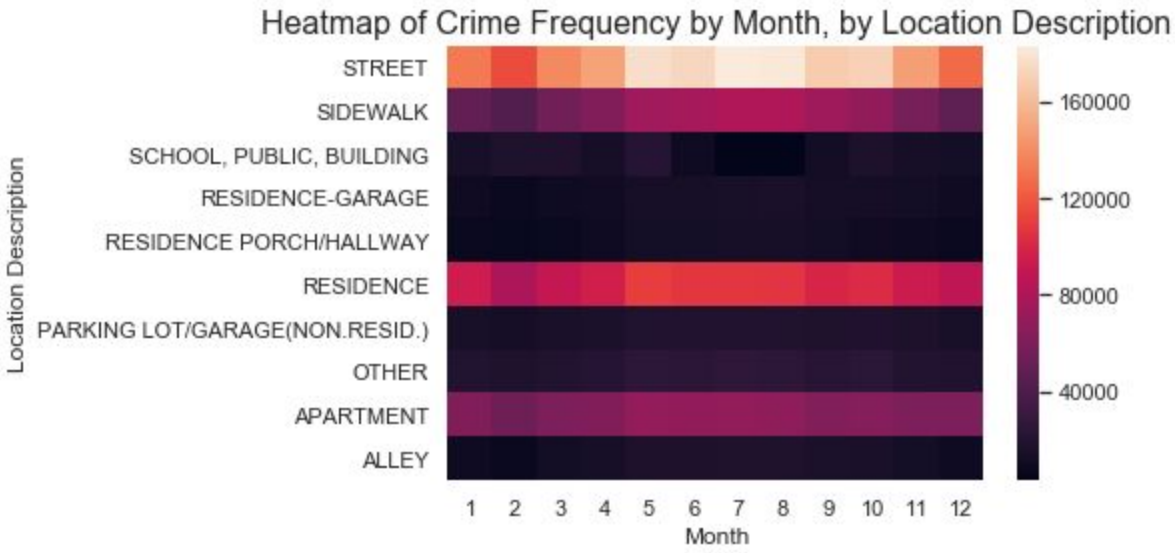
k. Heatmap of Crime Occurrence by Hour of Day



This heatmap makes it a little difficult to discern which crimes are most prevalent, at what hour of the day. Of the top 10 locations, about 5 have a low-enough frequency to where the heatmap does

not produce discernible colors for any of the hours. But for the 'Residence' and 'Street' locations, the colors are lighter for the latter hours of the day, which is expected.

l. Heatmap of Crime Occurrence by Month of Year

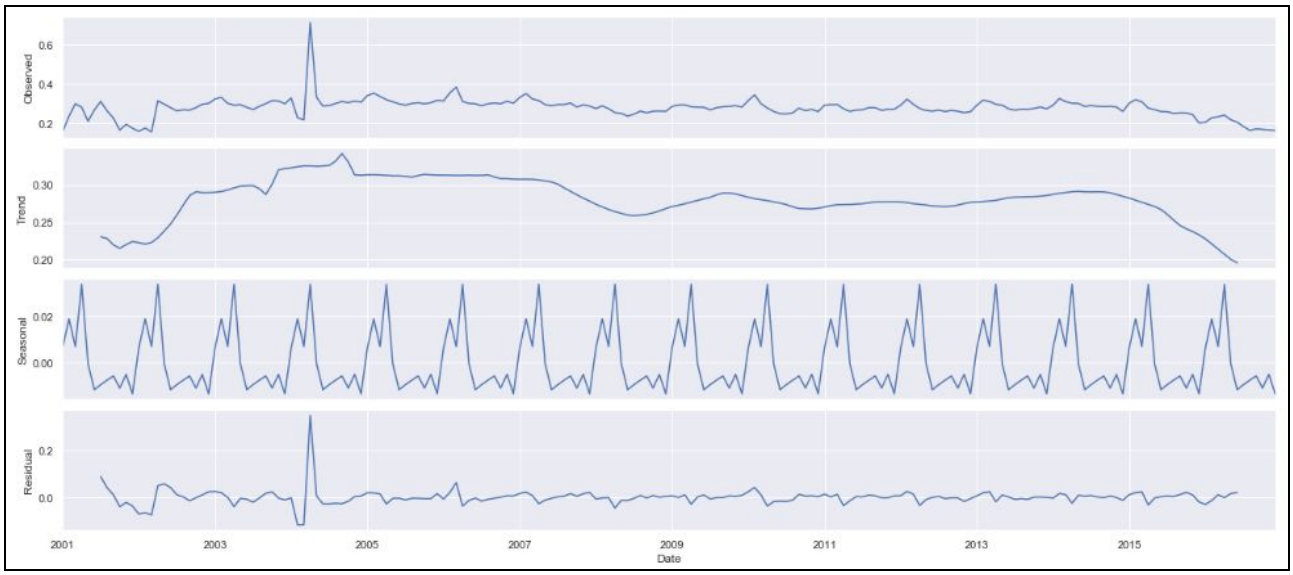


Like above, in this heatmap, there is only enough variation for 4 categories to derive meaningful insights from: street, sidewalk, residence, and apartment. In general, there was more crime during the middle of the year / during the summer months.

One of the purposes of data visualization is to guide data modeling. The fact that the heatmaps didn't show too much variation for some of the location categories made us realize that there were likely too many similar categories. This led us to decide to collapse the categories, for modeling purposes.

m. Time Series Analysis

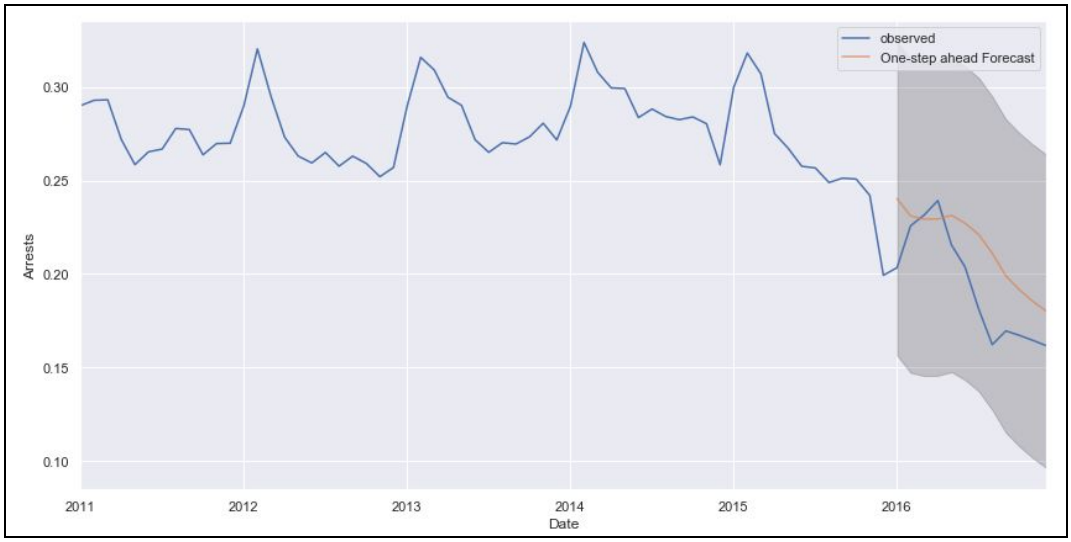
Since our data is heavily dependent on time, we used a time series model to forecast arrest. We referenced Susan Li's article posted on TowardsDataScience. We used average arrests for each month until the end of 2016. We visualized our data using, 'sm.tsa.seasonal_decompose', that decomposed our time series by trend, seasonality, and noise.



We see from these plots that there is clearly a downward trend and seasonality within our data. Using an ARIMA model accounts for this. A grid search allowed up to to select a model with the ideal parameters that let to the best performance. The model that yielded the lowest AIC value of -651.6679 was ARIMA(1, 0, 1)x(0, 0, 0, 12)12.

	coef	std err	z	P>|z|	[0.025	0.975]
ar.L1	0.9982	0.005	220.243	0.000	0.989	1.007
ma.L1	-0.7574	0.052	-14.453	0.000	-0.860	-0.655
sigma2	0.0018	7.35e-05	24.888	0.000	0.002	0.002

To observe the accuracy of our forecast, we compared the actual arrests to the predicted outcome starting from 01-31-2016.



We see the downward trend that was expected of our data.

6. Data Modeling

I. Pre-modeling Data Preparation

As mentioned above, we realized the location description variable had too many categories. So, in order to be able to produce more accurate models, we decide to collapse both the 'Location Description' and 'Primary Type' variables, since we knew we wanted to use these variables as independent variables in our models.

We collapsed the number of possible location categories from 170 to 56 by grouping similar categories together. For instance, we grouped all locations related to airports and aircrafts into one category of 'Airport/Aircraft.' However, in the model, we only used the top 10 location categories, since a model with more than a handful of dummy variables is cumbersome.

We also collapsed the 'Primary Type' column from 34 down to 24 by grouping similar types of crimes.

```
groups2 = {
    'THEFT': ['MOTOR VEHICLE THEFT', 'THEFT', 'ROBBERY', 'BURGLARY'],
    'WEAPONS VIOLATION': ['WEAPONS VIOLATION', 'CONCEALED CARRY LICENSE VIOLATION'],
    'NON-CRIMINAL': ['NON-CRIMINAL', 'NON - CRIMINAL', 'NON-CRIMINAL (SUBJECT SPECIFIED)'],
    'NARCOTICS': ['NARCOTICS', 'OTHER NARCOTIC VIOLATION'],
    'SEX OFFENSE': ['CRIM SEXUAL ASSAULT', 'SEX OFFENSE', 'OBSCENITY', 'PUBLIC INDECENCY']
}
```

```
    #collapsing
dataset['Type_factor'] = dataset['Primary Type']
dataset['Type_factor'] = pycats.cat_collapse(dataset['Type_factor'], groups2)
dataset['Type_factor'] = pd.factorize(dataset['Type_factor'])[0]
dataset['Type_factor'].nunique()
```

```
24
```

II. Modeling

a. Using K-Nearest Neighbors Model to Predict Crime Location

The K-Nearest Neighbors Model is a supervised learning algorithm. This algorithm is very versatile in that it can be used for both regression (prediction of numerical outcomes) and classification (prediction of binary outcome) (Harrison, 2019). In this specific scenario, our aim is to predict the location of a crime. The 'Location Description' column is of type 'string' and is first converted to a categorical type. There are 170 unique location types in the dataset. Since the performance of the KNN algorithm is worse when the number of possible outcomes in the target variable is higher, we decided to 'collapse' the number of possible categories in the location column down from 170 to 10, as mentioned above.

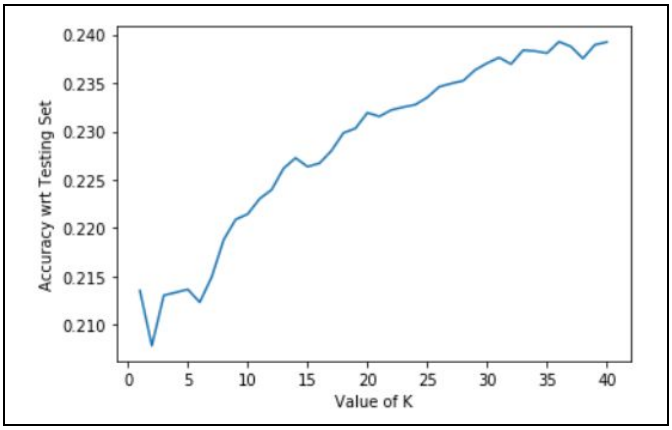
The next step was to take a subset of the entire dataset because running the KNN model on the entire dataset caused a MemoryError in the Jupyter Notebook. We took a random subset of 200,000 records from the entire 7 million rows dataset. The images below show the description for the two

datasets. The first image below shows the description for the entire dataset while the second image shows the description for the subset of the dataset. We can see that the numbers associated with the mean, standard deviation, minimum, 25th percentile, 50th percentile, 75th percentile, and the maximum are fairly close in terms of numerical value. This proves that the subset we took is a good representation of the entire dataset and thus the prediction we get from this subset would be a good representation of what we would get from deploying the KNN model on the entire dataset.

	ID	Block	Primary Type	Description	Location Description	Arrest	Domestic	Beat	District	Ward
count	7.145219e+06	7.145219e+06	7.145219e+06	7.145219e+06	7.145219e+06	7.145219e+06	7.145219e+06	7.145219e+06	7.145219e+06	7.145219e+06
mean	6.317712e+06	9.643948e+03	8.121148e+00	5.602973e+01	2.023515e+01	2.824256e-01	8.708613e-01	1.194928e+03	1.130767e+01	2.261324e+01
std	2.290136e+06	7.368512e+03	5.597462e+00	5.412791e+01	2.895948e+01	4.501793e-01	3.353534e-01	7.034579e+02	6.937785e+00	1.378542e+01
min	6.340000e+02	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	1.110000e+02	1.000000e+00	1.000000e+00
25%	4.695696e+06	3.663000e+03	1.000000e+00	2.600000e+01	2.000000e+00	0.000000e+00	1.000000e+00	6.230000e+02	6.000000e+00	1.000000e+01
50%	6.425439e+06	7.913000e+03	8.000000e+00	4.100000e+01	6.000000e+00	0.000000e+00	1.000000e+00	1.034000e+03	1.000000e+01	2.200000e+01
75%	7.848320e+06	1.439600e+04	1.300000e+01	6.200000e+01	3.500000e+01	1.000000e+00	1.000000e+00	1.731000e+03	1.700000e+01	3.400000e+01
max	1.082334e+07	3.422000e+04	3.300000e+01	3.730000e+02	1.690000e+02	1.000000e+00	1.000000e+00	2.535000e+03	3.100000e+01	5.000000e+01

	ID	Block	Primary Type	Description	Location Description	Arrest	Domestic	Beat	District	Ward
count	2.000000e+05	200000.000000	200000.000000	200000.000000	200000.000000	200000.000000	200000.000000	200000.000000	200000.000000	200000.000000
mean	6.313724e+06	9625.704560	8.135120	56.249395	20.378150	0.282315	0.870450	1194.823000	11.301365	22.586910
std	2.289124e+06	7360.467496	5.586083	54.310275	29.090819	0.450127	0.335809	702.747811	6.930398	13.799386
min	6.340000e+02	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	111.000000	1.000000	1.000000
25%	4.695452e+06	3670.000000	1.000000	26.000000	2.000000	0.000000	1.000000	623.000000	6.000000	10.000000
50%	6.421048e+06	7883.000000	8.000000	41.000000	6.000000	0.000000	1.000000	1111.000000	10.000000	22.000000
75%	7.834634e+06	14378.000000	13.000000	62.000000	35.000000	1.000000	1.000000	1731.000000	17.000000	34.000000
max	1.082032e+07	34219.000000	32.000000	366.000000	164.000000	1.000000	1.000000	2535.000000	31.000000	50.000000

In the first iteration of the KNN Model, we used 12 columns as the independent variables in this model. These included the 'ID', 'Primary Type', 'Description', 'Arrest', 'Domestic', 'FBI Code', 'Year', 'Month', 'Day', 'Hour', 'Minute', and 'Second' columns. We used 70% of the dataset as our training dataset and 30% of the dataset as our testing dataset. We then plotted the k-values in the range of 1 and 40 against the accuracies associated with each k-value.

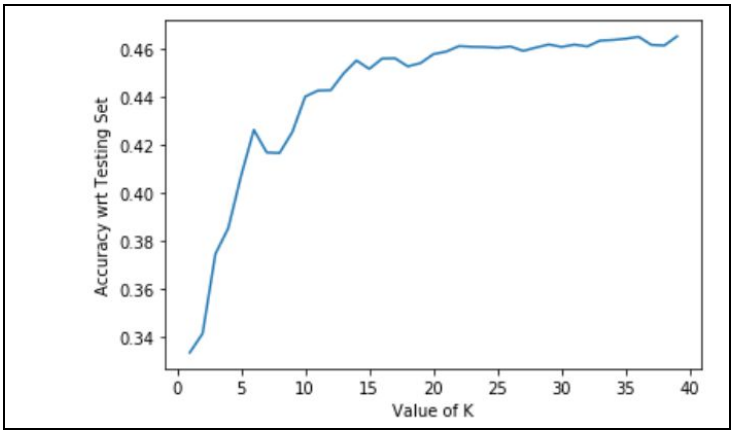


Here, we observe that the accuracy stabilizes at around k=39. To choose the optimal k-value based on this observation on the rules that the k-value must be smaller than the square root of the number of observations in the dataset and that k must also be an odd-number, we chose a k-value of 43. With this value, we got the results below.

```
In [19]: 1 y_pred_model = knn_model.predict(X_test)

In [20]: 1 accuracy_score(y_test, y_pred_model)
Out[20]: 0.24003333333333332

In [21]: 1 precision_score(y_test, y_pred_model, average="weighted")
Out[21]: 0.17429462911491797

In [22]: 1 recall_score(y_test, y_pred_model, average = "weighted")
Out[22]: 0.24003333333333332

In [23]: 1 f1_score(y_test, y_pred_model, average = 'micro')
Out[23]: 0.24003333333333332
```

The accuracy of the model is 0.24. This means that the model is not very good. In order to further determine this, we also analyzed the precision and recall associated with this model. The recall value is higher than the precision value at 0.24 versus 0.1743 which indicates that the model has done better at identifying data points of the positive class (Koehrsen, 2018).

In the second iteration of the KNN model, we used those variables as features which had a correlation of greater than 0.1 with the target variable. There are seven such variables. these are ‘Primary Type’, ‘Description’, ‘Arrest’, ‘Domestic’, ‘FBI Code’, ‘Minute’, and ‘Type Factor’. Once again, we ran the model on a random subset including 200,000 records with the same proportions for the training dataset and testing dataset. Upon plotting the k-value versus accuracy score, we got the plot below.



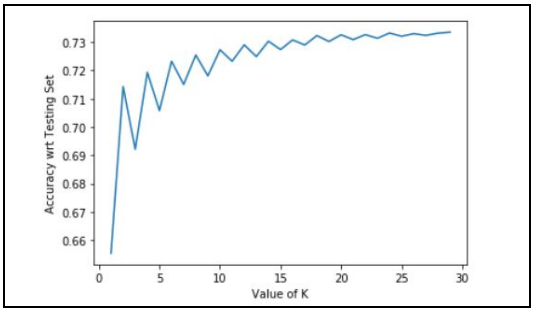
We observe that the accuracy of the model stabilizes at around k=20. So, we ran the model with a k-value of 23 where it is completely stabilized and got the results below.

```
In [35]: 1 accuracy_score(y_test_1, y_pred_model_2)
Out[35]: 0.4610166666666667

In [36]: 1 precision_score(y_test_1, y_pred_model_2, average="weighted")
Out[36]: 0.4246165093935587

In [37]: 1 recall_score(y_test_1, y_pred_model_2, average="weighted")
Out[37]: 0.4610166666666667

In [38]: 1 f1_score(y_test_1, y_pred_model_2, average='micro')
Out[38]: 0.4610166666666667
```

The accuracy of the model is much higher than it was in the first iteration. The precision and recall scores are different in this iteration too. This means that this model has once again done pretty well at identifying data points of the positive class. Given that the number of possible outcomes in the positive class is more than two, the accuracy achieved here is fair. So, this model can be applied in the eventuality that the Chicago police needed to predict the location of a crime.

b. K-Nearest Neighbors: Predict Whether the Perpetrator of a Crime was Arrested

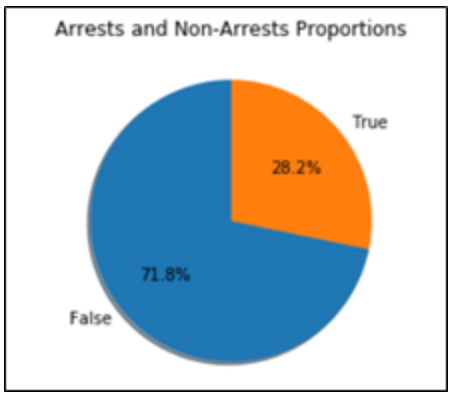
This model was created using Arrest (binary variable with True/False outcome) as the Target Variable and 'Beat', 'Ward', 'District', 'Community Area', 'Year', 'Month', and 'Location Description' as the Independent Variables. The model has been deployed on the same random subset used in the previous scenario of predicting the location of a crime. Once again, we plot the k-values between 1 and 30 against the accuracy associated with each value.



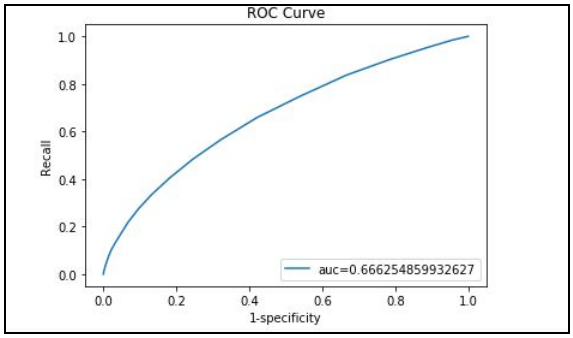
The accuracy stabilizes around the k-value of 25. Using this as the optimal k-value (meets the requirements of being smaller than the square root of the number of observations and an odd number), we got the results below.

```
In [46]: 1 accuracy_score(y_test_arrest_1, y_pred_arrest_1)
Out[46]: 0.7320833333333333

In [47]: 1 confusion_matrix(y_test_arrest_1, y_pred_arrest_1)
Out[47]: array([[40221,  2957],
                [13118,  3704]], dtype=int64)

In [48]: 1 precision_score(y_test_arrest_1, y_pred_arrest_1, average="weighted")
Out[48]: 0.6985535627371231

In [49]: 1 recall_score(y_test_arrest_1, y_pred_arrest_1, average="weighted")
Out[49]: 0.7320833333333333

In [50]: 1 f1_score(y_test_arrest_1, y_pred_arrest_1, average='micro')
Out[50]: 0.7320833333333333
```

The accuracy of the model is 0.7321. The low accuracy is due to the fact that the data in the target variable is imbalanced. The proportion of arrests being True is 28.2% while the proportion of arrests being False is 71.8%. The confusion matrix for this model is shown below. In the results, we observe that the model has correctly classified most of the data points in the model. There is a significant number of False Positives as compared to False Negatives and True Negatives however. This is important because this indicates that there are 13,118 data points that were incorrectly predicted as being arrested or not arrested and this could be a flaw in the legal system because innocent people might sometimes be wrongfully arrested and sometimes real criminals might not be arrested.



The recall score is 0.7321, which indicates that the model is fairly good at correctly classifying the data points of the positive class. The AUC Score for this model is 0.666. Since this value is between 0.6 and 0.7, it can be classified as a model that is neither weak nor strong.

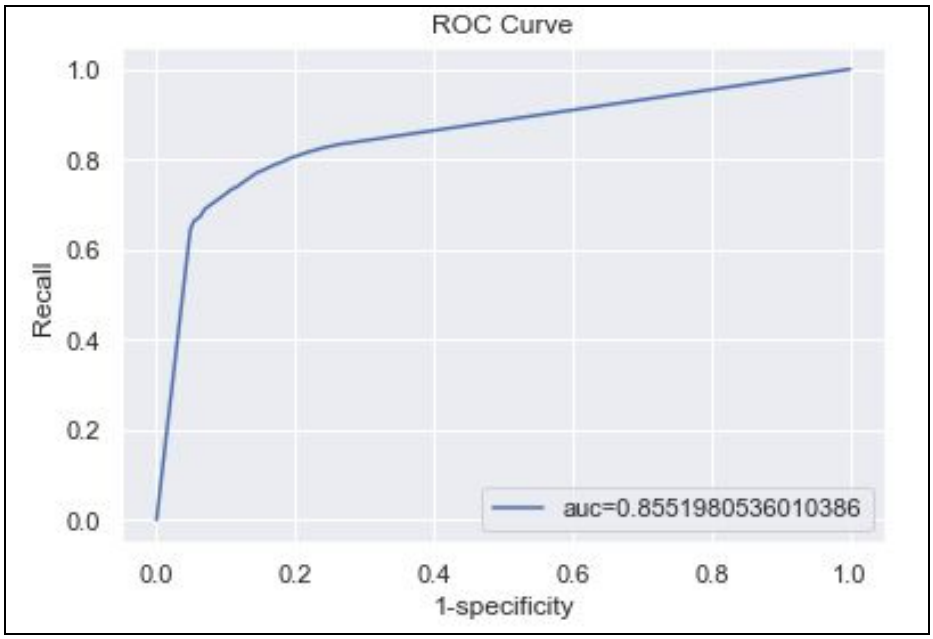


c. Decision Tree: Predict Whether the Perpetrator of Crime was Arrested

Before starting this decision tree, we knew that this had the potential to the best model to predict arrest. We know that decision trees are great when the data consists of categorical and numerical data. To get a true concept on which model was test, we used the same features to predict arrest. Before checking for accuracy, we administered a features test. The testing accuracy was 86.20% which we were very happy with.

```
Feature Set Used    :  ['Beat', 'District', 'Ward', 'Community Area', 'Year', 'Month', 'Location Description', 'Type_
factor']
Target Class        :  Arrest
Training Set Size   :  (5001653, 26)
Test Set Size       :  (2143566, 26)

DecisionTreeClassifier(class_weight=None, criterion='gini', max_depth=None,
                       max_features=None, max_leaf_nodes=None,
                       min_impurity_decrease=0.0, min_impurity_split=None,
                       min_samples_leaf=1, min_samples_split=2,
                       min_weight_fraction_leaf=0.0, presort=False,
                       random_state=None, splitter='best')

array([0.08651648, 0.00561239, 0.03166485, 0.03141121, 0.08651651,
       0.1061353 , 0.04870979, 0.60343347])

DecisionTreeClassifier(class_weight=None, criterion='gini', max_depth=None,
                       max_features=None, max_leaf_nodes=None,
                       min_impurity_decrease=0.0, min_impurity_split=None,
                       min_samples_leaf=1, min_samples_split=2,
                       min_weight_fraction_leaf=0.0, presort=False,
                       random_state=None, splitter='best')

Accuracy on training set: 0.933
Accuracy on test set: 0.862
```

Looking at the ROC Curve, the AUC is .86 which still indicates a good model.



d. Decision Tree to Predict Crime Type

We decided to also predict what type of crime was committed. This could potentially help the Chicago Police Department decide what type of crime will be committed based on the location. The model's accuracy raised considerably when adding 'hour' and 'month' which makes since if you consider the seasonality of crime and when crime usually happens. We could have gotten a significant higher accuracy if we include the variable, 'FBI code,' but due to collinearity between variables, we decided it was best to leave it out.

After completing a features test to decide which variables were important to the model, we removed 'District'. After changing the parameters to receive the highest accuracy, max_depth seemed to have the most impact.

```
Feature Set Used    :  ['X Coordinate', 'Y Coordinate', 'Year', 'Hour', 'Month', 'Arrest', 'Ward', 'Beat', 'Domesti
c', 'Community Area', 'Location Description']
Target Class        :  Type_factor
Training Set Size   :  (5001653, 26)
Test Set Size       :  (2143566, 26)

DecisionTreeClassifier(class_weight=None, criterion='gini', max_depth=100,
                       max_features=None, max_leaf_nodes=None,
                       min_impurity_decrease=0.0, min_impurity_split=None,
                       min_samples_leaf=50, min_samples_split=2,
                       min_weight_fraction_leaf=0.0, presort=False,
                       random_state=0, splitter='best')

Accuracy on training set: 0.563
Accuracy on test set: 0.537
```

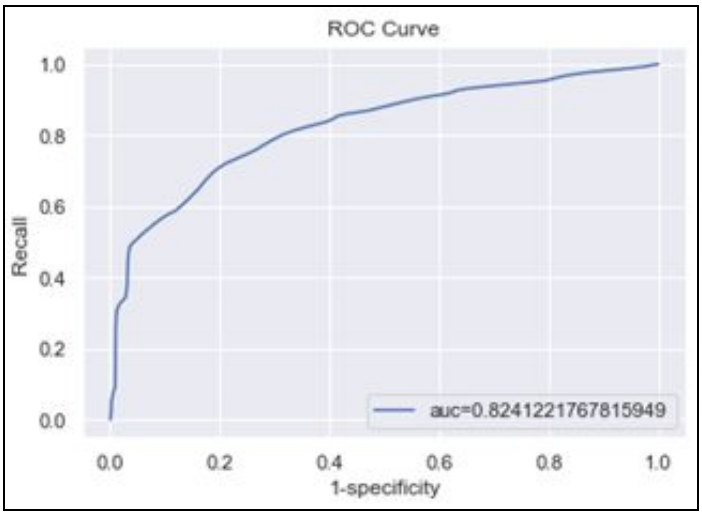
Although the testing accuracy is only 53.70%, it seems like it is the highest possible when predicting primary type.

e. Logistic Regression to Predict Whether the Perpetrator of Crime was Arrested

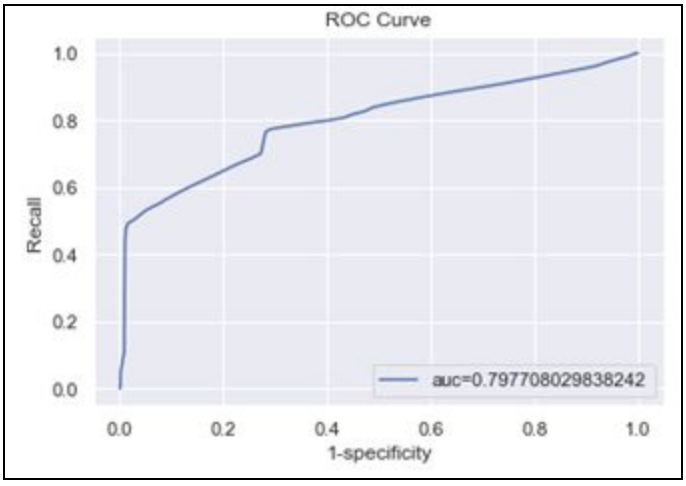
We also ran a logistic regression model to see if it could accurately predict whether the perpetrator of a crime was arrested or not. This model is designed for binary classification, so it is perfect for our purposes. As part of the pre-modeling data preprocessing, we converted the top 10 location categories we obtained earlier into dummy variables. Then, we concatenated the dummy variables with the rest of the data frame, for ease of modeling.

```
dummy = pd.get_dummies(df['Location Description'])
dummy
```

For the first iteration of logistic regression, we used 10 independent variables, plus 10 location dummy variables, for a total of 20 predictor variables. We also converted 'Arrest' to a Boolean type before calling the train_test_split function. In the LogisticRegression function, we used the parameter *class_weight* = 'balanced' to indicate to Python that the class weights might be imbalanced. When we use this argument, we tell Python to automatically weigh classes inversely proportional to their frequency (Albon, 2017). We see that the first iteration resulted in a training set accuracy of 74.6%, a test set accuracy of 74.7%, and an area under the curve of 0.824. Below is an ROC curve of this iteration.



In the second iteration, I only kept 7 variables; I removed the dummy variables associated with 'Location Description.' Surprisingly, this only decreased both the training and test scores by less than 2%, down to 72.8%, and the AUC down to 0.798. Below is the associated ROC curve.



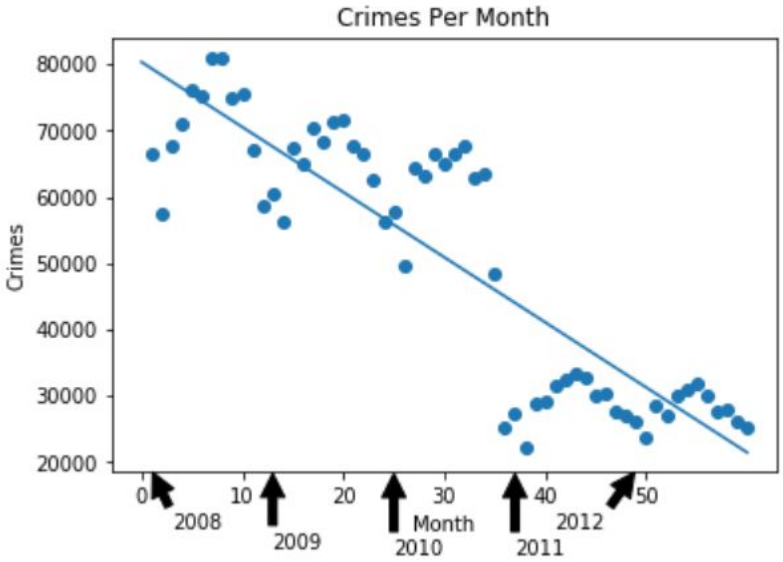
In the second iteration, when we deleted the 10 location description dummy variables, the accuracy only decreased by 1.9%. This means

f. Linear Regressions to predict how many crimes and arrests will happen each month

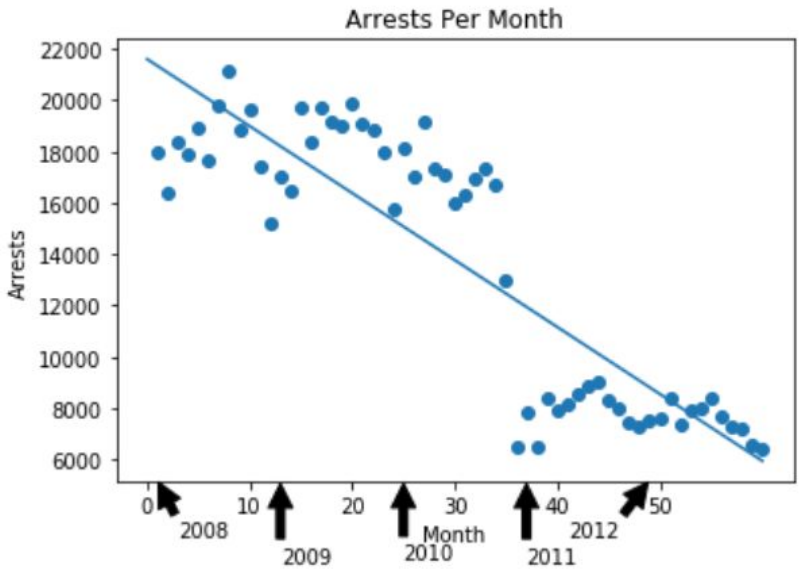
The data from which the crimes and arrests per month were calculated were from 2008 to 2012. This time period was chosen because there was a spike in crimes during the Great Recession, and then the amount of crimes fell after the recession. This leads to an averaging out of the crimes and arrests from higher values to lower values. Since arrests are caused by crimes, both datasets have a similar pattern although there was not an arrest for every crime.

The test size was chosen to be 0.3. From here, the linear regression is very accurate with 0.78 for both training values for crimes and arrests per month. The test values for both datasets were also close to the training values by being both 0.74 and 0.72 for the crimes and the arrests respectively, making them fairly accurate. Furthermore, the models were evaluated by the MAE, MSE, RMSE, and

R^2. The MAE, MSE, and RMSE were both lower for the amount of arrests per month as the values were smaller than for the amount of crimes per month. The MAE is the best evaluation of the 3 error methods as the errors for both datasets were smaller. The R^2 was also higher (but similar) for the crimes per month, which means that there was slightly more correlation between the variables.




```
lr.coef_: [[-981.67172571]]
lr.intercept_: [80036.20973578]
Training set score: 0.78
Test set score: 0.74

Model slope:  -981.4671853292582
Model intercept:  80341.08248587571

Mean Absolute Error:  8426.94137662342
Mean Squared Error:  99699583.61894862
Root Mean Squared Error:  9984.967882719935

R^2:  0.7374626836020501
```




```
lr.coef_: [[-263.73853492]]
lr.intercept_: [21714.69820716]
Training set score: 0.78
Test set score: 0.72

Model slope:  -260.77413170325104
Model intercept:  21595.49435028249

Mean Absolute Error:  2273.821620287991
Mean Squared Error:  7236164.244460577
Root Mean Squared Error:  2690.011941323045

R^2:  0.7214028948305319
```

7. Model Comparison and Conclusion

Overall, the models where numerical data are predicted instead of categorical data tend to be more accurate. The decision tree was very strong in predicting whether the perpetrator of the crime was arrested or not, and had a very high AUC of 0.86 along with a test value of 0.862. It did, however, have a 0.537 test value when predicting the type of crime committed. This was due to the fact that the data was categorical. It is still a good model for predicting categorical and numerical data.

The KNN Model performed weaker than the decision tree as it had lower accuracy values. It had an AUC score of 0.664 when predicting whether the perpetrator of the crime was arrested or not, which was not weak nor strong. Also, when predicting the location of the crime, the model did not perform well. It had an initial accuracy of 0.241 with k-value of 43, but then it almost doubled to 0.461 when we decided to use variables with a correlation of greater than 0.1 with the target as model features. This model performed poorly because the number of possible outcomes in the target is greater than two.

The linear regressions for both the crimes per month and arrests per month were very similar with a test score in the lower 0.7 range. It is a good model when dealing with data over a time period which is strictly numerical. The amount of crimes and arrests were showing a decreasing pattern, so both linear regressions were able to successfully plot a decreasing trend line. All of the errors were lower for the amount of arrests per month. The R^2 value, was also lower for the amount of arrests per month, but there is less correlation compared to the amount of crimes per month. The main problem is that the linear regression models may not work long term as the values of crimes and arrests per month will be negative, which is impossible.

An alternative regression model would be a reciprocal regression model as the values won't reach 0 as the function 1/x won't be 0 or negative for all positive values. The logistic regression to predict whether the perpetrator of a crime was arrested or not had a very similar test score to the linear regression models of around the low 0.7 range. Therefore, the performances of the linear and logistic regressions are very similar.

The logistic regression seemed to have very low overfitting as it is best for predicting the binary outcomes. The main problem with the logistic regression is that it is harder to predict numerical variables or variables with several categories as it mainly good for binary classification.

Below is a table that compares all of the models we ran for predicting whether or not a perpetrator of a crime was arrested. If multiple iterations of the same model were run, we only included the best one.

Model	Accuracy	AUC	R^2	Independent Variables
KNN	0.7321	0.666	N/A	Year, Month, Ward, Beat, Community Area, and Location Description, and District
Linear Regre-ssion	0.740	N/A	0.737	Month
Decision Tree	0.862	0.855	N/A	Year, Month, Ward, Beat, Domestic, Community Area, Location Description, District, and Type Factor
Logistic Regre-ssion	0.747	0.824	N/A	'X Coordinate', 'Y Coordinate', 'Year', 'Month', 'Ward', 'Beat', 'Domestic', 'Community Area', 'District', 'Type_factor', 'Alley', 'Apartment', 'Other', 'Parking Lot', 'Residence', 'Retail', 'Road', 'School', 'Sidewalk', 'Transport Site'

Looking at the models predicting 'Arrest,' we can conclude that decision trees are the best at predicting if the perpetrator of the crime was arrested. The model had an accuracy of .862 and an AUC score of 0.855. The AUC value indicates that this is a fairly good model.

References

- Albon, Chris. “Handling Imbalanced Classes In Logistic Regression.” *Chris Albon*, 20 Dec. 2017, chrisalbon.com/machine_learning/logistic_regression/handling_imbalanced_classes_in_logistic_regression/.
- Angelov, B. (2017, December 13). Working with Missing Data in Machine Learning. Retrieved November 5, 2019, from https://towardsdatascience.com/working-with-missing-data-in-machine-learning-9c0a430df4ce.
- Block , C. R. (1984). IS CRIME SEASONAL? . *IS CRIME SEASONAL?* , 8–9. doi: https://bjs.gov/content/pub/pdf/ics.pdf
- Diaz, A. (2018, September 17). More than half of violent crimes in the U.S. never reported to police. Retrieved October 15, 2019, from https://www.foxnews.com/us/more-than-half-of-violent-crimes-in-the-u-s-never-reported-to-police.
- Harrison, O. 14 July 2019. “Machine Learning Basics with the K-Nearest Neighbors Algorithm.” Medium, Towards Data Science. https://towardsdatascience.com/machine-learning-basics-with-the-k-nearest-neighbors-algorithm-6a6e71d01761. Retrieved November 26, 2019.
- heng8835. 4 May 2019. “Classification with ML: Predict Crime Type.” Kaggle, Kaggle. https://www.kaggle.com/heng8835/classification-with-ml-predict-crime-type. Retrieved November 26, 2019.
- Koehrsen, W. 10 Mar 2018. “Beyond Accuracy: Precision and Recall.” Medium, Towards Data Science. https://towardsdatascience.com/beyond-accuracy-precision-and-recall-3da06bea9f6c. Retrieved November 26, 2019
- Li, S. (2018, September 5). An End-to-End Project on Time Series Analysis and Forecasting with Python. Retrieved November 18, 2019, from https://towardsdatascience.com/an-end-to-end-project-on-time-series-analysis-and-forecasting-with-python-4835e6bf050b.
- Russell. (2017, August 30). Creating and Visualizing Decision Trees with Python. Retrieved November 5, 2019, from https://medium.com/@rnbrown/creating-and-visualizing-decision-trees-with-python-f8e8fa394176.
- Wire, Sun-Times. “Chicago Shootings at Four-Year Low through First Half of 2019: Police.” *Chicago Sun-Times*, Chicago Sun-Times, 1 July 2019, chicago.suntimes.com/news/2019/7/1/20676793/chicago-shootings-four-year-low-through-first-half-of-2019-police-crime-stats.
- Odds of getting away with murder in Chicago are better than 50 percent. (2019, November 7). Retrieved from https://wgntv.com/2019/11/06/odds-of-getting-away-with-murder-in-chicago-are-better-than-50-percent/.