CS 480 Project

Crime Classification and Prediction

Andrew Parker

April 23, 2020

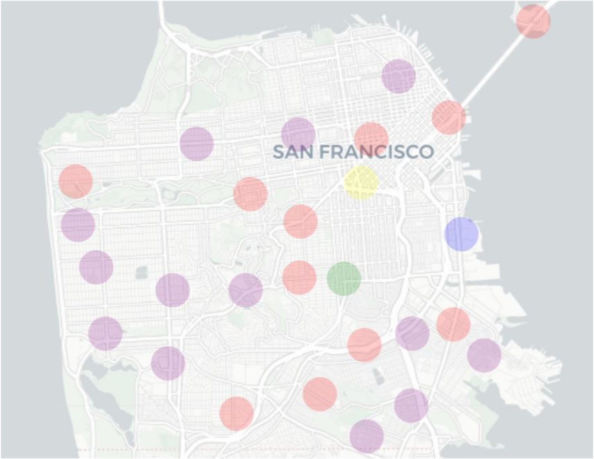


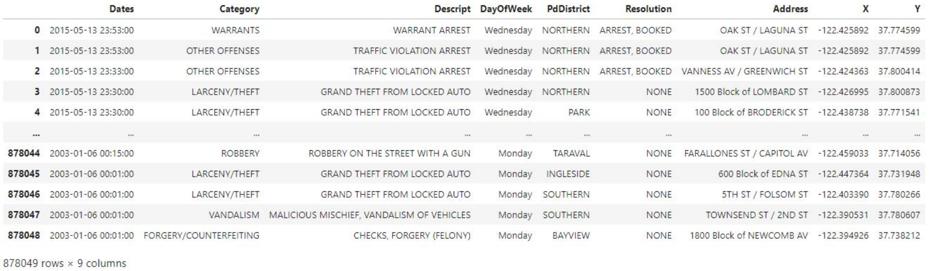
Figure 1: Predicting and classifying crime in San Francisco

1 Project Objective

For this project, I was interested in looking at how well I could classify and predict crime in a city. This task interested me because I wanted to see if patterns could be detected in something as seemingly random and unexpected as crime. The main focus on this project is on the classification part of the program. While doing research, I found programs targeting the prediction of crime geographically, but not much was found on classification. I decided to focus on classification since it seemed more novel. I ended up adding some very basic geographical prediction to add a visual interactive aspect, as well as to add to the general application of my project. The most prominent application of my project would be to use it as a tool to decide what areas of town police officers should be positioned to be able to quickly respond to crime. The classification also would give them an idea of what crimes specifically they should be looking out for at that particular time.

2 Getting Data

After looking at several different cites, I decided to use the San Francisco crime data set found on kaggle.com. This data set was chosen because it had almost all of the features that I was interested in for this project. Figure 2 shows the original crime data set. A notable attribute of the data set is that I am working with data from 2003 to 2015.



	Dates	Category	Descript	DayOfWeek	PdDistrict	Resolution	Address	X	Y
0	2015-05-13 23:53:00	WARRANTS	WARRANT ARREST	Wednesday	NORTHERN	ARREST, BOOKED	OAK ST / LAGUNA ST	-122.425892	37.774599
1	2015-05-13 23:53:00	OTHER OFFENSES	TRAFFIC VIOLATION ARREST	Wednesday	NORTHERN	ARREST, BOOKED	OAK ST / LAGUNA ST	-122.425892	37.774599
2	2015-05-13 23:33:00	OTHER OFFENSES	TRAFFIC VIOLATION ARREST	Wednesday	NORTHERN	ARREST, BOOKED	VANNESS AV / GREENWICH ST	-122.424363	37.800414
3	2015-05-13 23:30:00	LARCENY/THEFT	GRAND THEFT FROM LOCKED AUTO	Wednesday	NORTHERN	NONE	1500 Block of LOMBARD ST	-122.426995	37.800873
4	2015-05-13 23:30:00	LARCENY/THEFT	GRAND THEFT FROM LOCKED AUTO	Wednesday	PARK	NONE	100 Block of BRODERICK ST	-122.438738	37.771541
...	...	...	...	...	...	...	...	...	...
878044	2003-01-06 00:15:00	ROBBERY	ROBBERY ON THE STREET WITH A GUN	Monday	TARAVAL	NONE	FARALLONES ST / CAPITOL AV	-122.459033	37.714056
878045	2003-01-06 00:01:00	LARCENY/THEFT	GRAND THEFT FROM LOCKED AUTO	Monday	INGLESIDE	NONE	600 Block of EDNA ST	-122.447364	37.731948
878046	2003-01-06 00:01:00	LARCENY/THEFT	GRAND THEFT FROM LOCKED AUTO	Monday	SOUTHERN	NONE	5TH ST / FOLSOM ST	-122.403390	37.780266
878047	2003-01-06 00:01:00	VANDALISM	MALICIOUS MISCHIEF, VANDALISM OF VEHICLES	Monday	SOUTHERN	NONE	TOWNSEND ST / 2ND ST	-122.390531	37.780607
878048	2003-01-06 00:01:00	FORGERY/COUNTERFEITING	CHECKS, FORGERY (FELONY)	Monday	BAYVIEW	NONE	1800 Block of NEWCOMB AV	-122.394926	37.738212

878049 rows × 9 columns

Figure 2: Original San Francisco crime data set

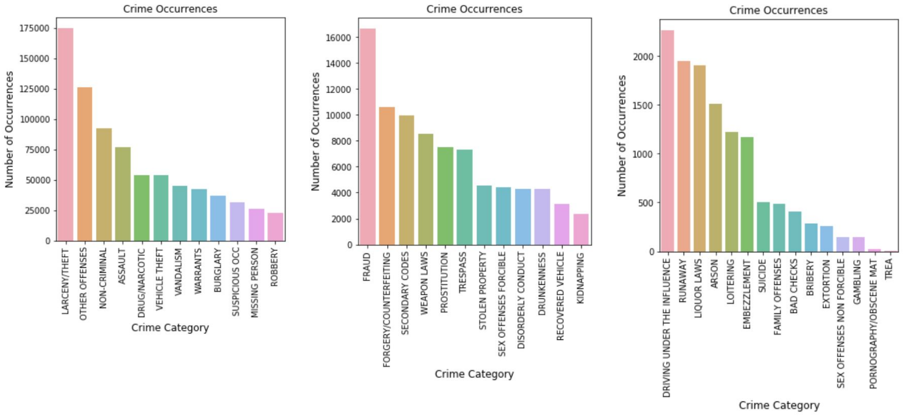
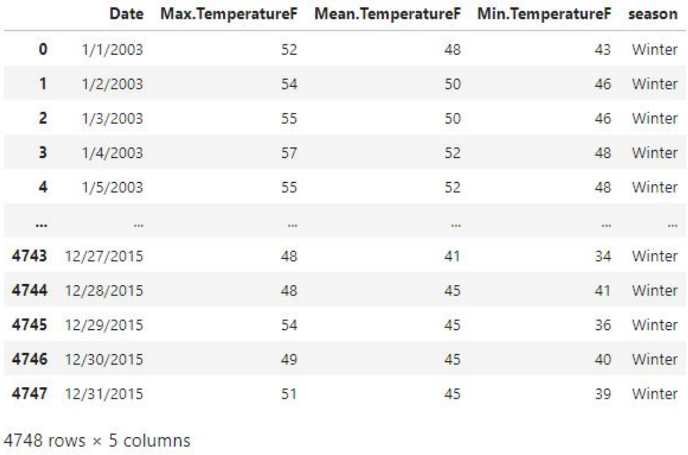


Figure 3: Crime occurrences for original data set

Another notable attribute of the data is the large amount of crime categories provided, as well as the range of occurrences. Figure 3 shows all 39 crime categories and their occurrences. Note the large range. The largest category is ’larceny/theft’ with roughly 175,000 occurrences, and the smallest is ’treason’ with 6 occurrences. A lot of these categories are useless for classifying crime due to this range. Larceny/theft is too large, and covers the whole city map. Categories like treason have so few occurrences that they will almost never be seen. Some categories such as ’suspicious occurrence’ don’t tell us anything about the actual classification, as it could be anything. A large amount of data wrangling needed to be done to even out categories so that the model could be effective.

The second data set I pulled from the internet was the temperature of each day in San Francisco. This was the only feature I wanted that was not present in the SF crime data set. Figure 4 shows the temperature data set. At this point I had cut down the CSV file to only show data in the date range I was using.



	Date	Max.TemperatureF	Mean.TemperatureF	Min.TemperatureF	season
0	1/1/2003	52	48	43	Winter
1	1/2/2003	54	50	46	Winter
2	1/3/2003	55	50	46	Winter
3	1/4/2003	57	52	48	Winter
4	1/5/2003	55	52	48	Winter
...	...	...	...	...	...
4743	12/27/2015	48	41	34	Winter
4744	12/28/2015	48	45	41	Winter
4745	12/29/2015	54	45	36	Winter
4746	12/30/2015	49	45	40	Winter
4747	12/31/2015	51	45	39	Winter

4748 rows × 5 columns

Figure 4: San Francisco weather data set

3 Data Pre-Processing

I started data pre-processing by looking for outliers and missing data. Luckily my data set did not come with any missing values. I looked at the largest and smallest values for x, y coordinates and only found one coordinate set that was far outside of San Francisco, which I dropped from the data set.

Most of my time on data wrangling was spent on the crime categories. As mentioned and graphed in the previous section, 39 categories were present. I dropped some larger categories such as ’larceny/theft’ since they geographically dominated the entire city, providing no information gain for my model. Similarly, I dropped smaller categories such as ’treason’ and ’pornography’ due to them having very sparse occurrences. I dropped several other crimes that were useless from a patrol perspective. Some examples include ’extortion’, ’bribery’, ’bad checks’, ’gambling’, etc. These also had very few occurrences in the data set. Figure 5 shows the categories left over after clearing out the ones I wouldn’t be using.

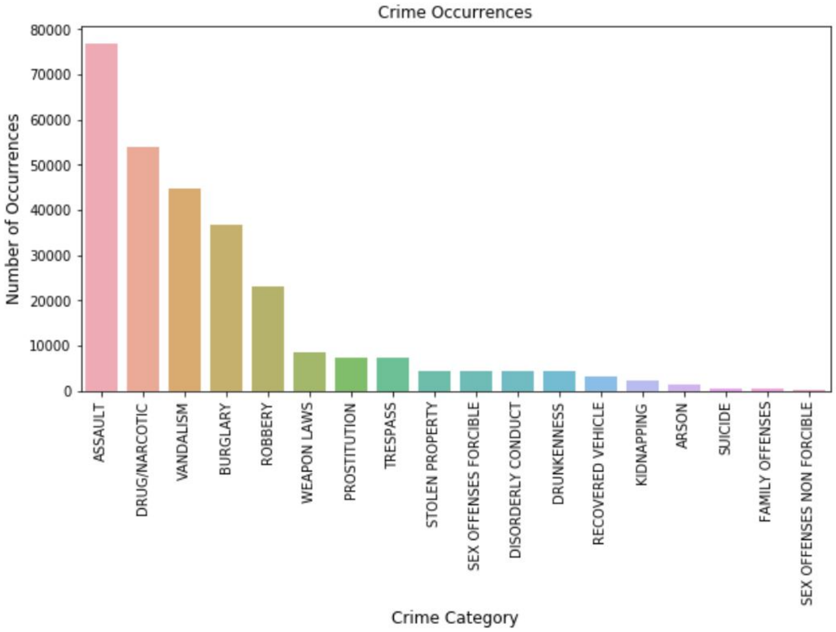


Figure 5: Crime Categories before grouping

The last step to making this data usable was to group together similar crimes to even out the categories I would be trying to classify. I grouped together the types into 5 categories: violent crime, theft, sex offences, drug offences, and mischief (This consists of vandalism, trespassing, disorderly conduct, etc). Figure 6 shows the final groups I attempted to classify.

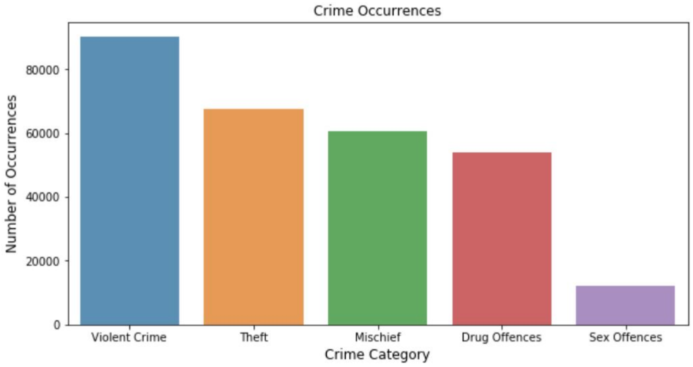
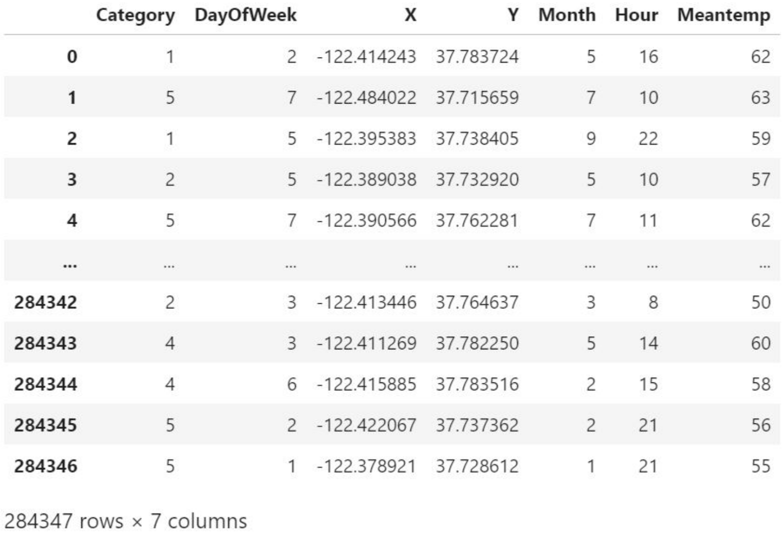


Figure 6: Crime Categories after grouping

The data in the 'Dates' column was split into the columns, 'Year', 'Month', 'Day', and ’Hour’. This was done to make the data usable by the model. The crime category and day of the week data were both mapped to integer values for the model to use. Later in the process I end up finding that the model performs slightly better when ’Day’ is dropped, so I removed this from the data frame. I also removed ’Year’ due to the fact that this feature would be useless when trying to predict crime more recently than 2015. I also dropped the features ’Address’, ’Descript’, ’Resolution’ and ’PdDistrict’, as they would not be useful for my model.

The last main part of the pre-processing step was to pull in the weather data. For each crime entry in the data frame, my code used the San Francisco weather data frame to look up the mean temperature for that day. This data was put into the new column ’Meantemp’ and was added to the main data frame. Figure 7 shows the final data frame that I used for the model.



	Category	DayOfWeek	X	Y	Month	Hour	Meantemp
0	1	2	-122.414243	37.783724	5	16	62
1	5	7	-122.484022	37.715659	7	10	63
2	1	5	-122.395383	37.738405	9	22	59
3	2	5	-122.389038	37.732920	5	10	57
4	5	7	-122.390566	37.762281	7	11	62
...	...	...	...	...	...	...	...
284342	2	3	-122.413446	37.764637	3	8	50
284343	4	3	-122.411269	37.782250	5	14	60
284344	4	6	-122.415885	37.783516	2	15	58
284345	5	2	-122.422067	37.737362	2	21	56
284346	5	1	-122.378921	37.728612	1	21	55

284347 rows × 7 columns

Figure 7: Final data set

I did end up running the data through a scaler to normalize it, as I found that this improved accuracy slightly. For the train/test split, I did a random 80/20 split of the data.

Cross-validation was not performed due to the fact that the model I chose was random forests, which does not have issues with over-fitting.

4 Model

I ended up choosing random forest for classification . Random forest is a very robust algorithm that can deal with a large range of data. The fact that it can handle both discrete categorical data as well as continuous data was also important for my data set.

Another reason I chose this model was due to the fact that it is good with unbalanced data, which was an issue that my data set had.

For the crime prediction part of my program, I decided to use a clustering algorithm to identify groups of crime over time. I then focused on cluster centers to identify where crime was most likely to happen for that given set of inputs. I found that the mean-shift clustering algorithm was the most useful for this problem. This is because cluster centers diverge at points of maximum density and find local maximums that other algorithms would skip over to simply find the global maximum. I want to be able to see those local maximums. Mean-shift is also useful because it determines the number of clusters based off of the given data set rather than the number of clusters being predetermined. My model should be able to adjust the number of crime hot spots depending on the input.

5 Training and Testing

I tried several models out to see how each performed for classification. The ones I tried were multinomial naive bayes, random forest, SVM, and a neural network. The SVM and the neural network were getting an accuracy of roughly 30-35%. The naive bayes had an accuracy of roughly 50%, but I realized that this was because it was always classifying crime under violent crime or theft, so this was useless. After settling on the random forest model, I was getting an accuracy of around 40%.

The two features I dropped were 'Day' (not day of the week) and 'Year'. The day was dropped because the day of the month generally shouldn't have any sort of correlation with crime type. My model's accuracy increased slightly after dropping 'Day'. The year was dropped because while it helped slightly with classification accuracy, The scope of my problem was centered around predicting crime more recently than the data set. The year should always be higher than 2015.

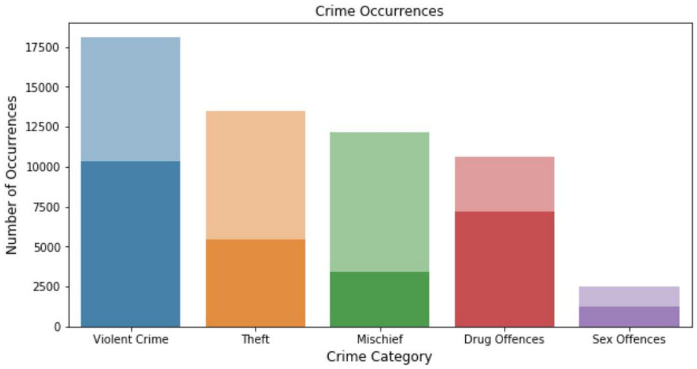
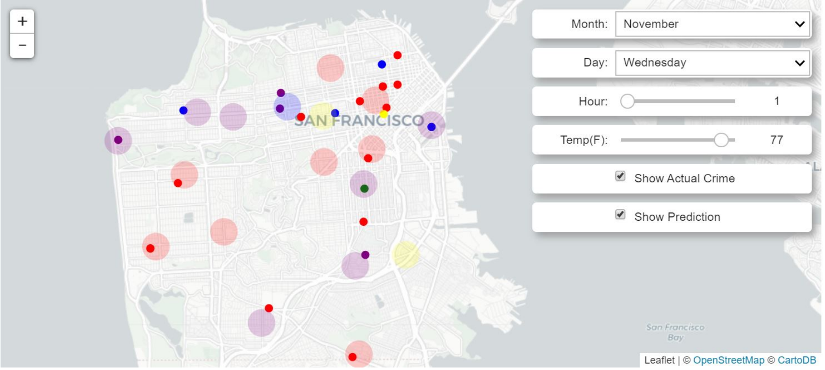


Figure 8: Total test data and correct classifications

Some other steps such as normalizing data, adjusting how crime types were distributed into the categories, and adding temperature data all improved my model's accuracy. I spent some time hypertuning the model which had a very slight improvement as well. I increased ’n_estimators’ from 100 to 120 and increased ’min_samples_leaf’ from 1 to 2 to improve the model. My final accuracy for classification averages at 49%. This is graphically shown in figure 8. It shows the total occurrences in the test set compared with the total correct classifications. I believe the main factor impacting accuracy is simply the relationship between the features and crime. Crime is much more complicated than simply looking at external features such as date, time, location, and temperature. Something else impacting accuracy is how I organized crime types into the 5 categories. An example is the mischief category. This category consists of several different crime types put together, making it hard to classify. There are better ways to organize the crime groupings that would improve the model's accuracy.

As for crime prediction, I built a basic tool in jupyter lab to have an interactive way of looking at the data. After entering inputs for a time period, the program will run the mean-shift clustering algorithm on the data matching those inputs. After the location of clusters are identified, the inputs and cluster coordinates are run through the crime classification model to attempt to classify the crime in that area. Figure 9 shows a screenshot of the tool. The large colored circles show the predicted crime type and location, and the colored dots show the actual crime from the test data. (Violent Crime - Red, Theft - Purple, Sex offence - Green, Drug offence - Yellow, Mischief - Blue)



Month: November

Day: Wednesday

Hour: 1

Temp(F): 77

☑ Show Actual Crime

☑ Show Prediction

Figure 9: Crime classification and prediction tool

6 Conclusion

Although people may look at an accuracy of 49% and be disappointed with the results, I think that it is still a success. Due to the random nature of crime and having features that are unable to capture all of the complicated variables involved with crime, expecting a high classification accuracy like 80% or 90% is unrealistic with the data used in my model. I have still demonstrated that my model picked up on patterns with crime type, and I believe the accuracy is appropriate.

As for the secondary goal of predicting crime hot spots geographically, I found it difficult to calculate a meaningful accuracy. I wanted to produce a map with the most probable locations for crime across the city. Just because no crime was reported in a hot spot doesn't necessarily make the prediction wrong. This section of the project is also a success, mainly because the purpose was to have the user be able to look at the data geographically, as well as to interact with it.