

DEFINITION

Project Overview

Public safety is a major concern for all cities especially the ones with a large university population such as Montreal which has six universities and twelve junior colleges in a five-mile radius, accounting for 248,000 students. Montreal is one of the safest cities in North America however, in 2018 there was a murder rate increase of 33%. Montreal expects roughly 11M+ people to visit each year and those numbers have been declining by 3% since 2017.

The data set was obtained from <http://donnees.ville.montreal.qc.ca> and it includes fields such as crime category, date, part of day, district, and coordinates. The categories are made up of the following: Theft of motor vehicle parts, break-in public/home, misdemeanor, motor vehicle theft, Robbery/violence/armed and murder. The Date column begins from January 2015 and ends in September 26 of 2019 with a total of 149,654 observations. The part of day column is broken down into three sections. "Day" which is between 8:00-16:00, Evening 16:01-00:00 and Night 00:01 – 8:00. The district column is made up of integers which correlate to a district name. During the five-year period there are 44,874 thefts of motor vehicle parts, 41,240 break-ins, 33,900 misdemeanors, 20,765 vehicle thefts, 8,758 robberies and 117 murders. I found another data set on the same website which provides a list of names for those districts.

Problem Statement

This problem will be solved by training a Supervised Learning model to be able to predict the type of crime that will happen next so police units can be active patrol cars near that area. This will lower the number of cases as the police presence will intimidate most criminals and if the criminal act takes place then the police will be able to respond quicker and shorten the investigation time which will also decrease the number of federal spending in ongoing investigations.

The current measures used by Police in Montreal include tightening the video surveillance in certain neighborhoods and assembling special task forces. This will increase city spending substantially and although additional police presence can prevent some crimes there has to be a strategic and more optimized solution.

In order to solve this problem, I will take the following steps:

1. Data Cleaning
 - a. Remove duplicate/Nan values
 - b. Understand the datatype of each feature and transform it if needed
2. Descriptive Analytics
 - a. Produce charts in order to get a deeper understanding of the dataset
 - b. Show reasons as to why I chose the following models

3. Data Pre-processing and Feature Engineering
 - a. Develop new features from existing ones
 - b. Standardize numerical values
 - c. Encode sting datatypes so all features and labels are numeric
 - d. Explain which algorithms will be used
4. Model training and Evaluation
 - a. Train Machine Learning models (XGBOOST, Logistic Regression, Random Forrest)
 - b. Evaluate each model's performance through accuracy and F-score metrics
5. Model Tuning
 - a. Choose the model with the best performance and tune the parameters using Grid Search and or Random Search
 - b. Locate the most important parameters
6. Conclusion
 - a. Reflect on project experience and offer alternative ways to solve the current problem

Metrics

I will use the accuracy as well as the F-score with a lower beta since I will focus on accuracy instead of recall. This way the Police department will be able to utilize resources better instead of dispatching larger number of units than needed.

$$\text{Precision} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}} \quad F_{\beta} = \frac{(1 + \beta^2) \cdot (\text{precision} \cdot \text{recall})}{(\beta^2 \cdot \text{precision} + \text{recall})}$$

ANALYSIS

Below you will see examples of the original Dataset and the updated version:

	CATEGORIE	DATE	QUART	PDQ	X	Y	LONGITUDE	LATITUDE		
0	Vol de véhicule à moteur	2018-09-13	jour	30.0	294904.159001	5.047549e+06	-73.626778	45.567780	CATEGORIE	object
1	Vol de véhicule à moteur	2018-04-30	jour	30.0	294904.159001	5.047549e+06	-73.626778	45.567780	DATE	object
2	Vol de véhicule à moteur	2018-09-01	nuit	7.0	290274.565000	5.042150e+06	-73.685928	45.519122	QUART	object
3	Méfait	2017-07-21	jour	21.0	0.000000	0.000000e+00	1.000000	1.000000	PDQ	float64
4	Méfait	2017-07-29	jour	12.0	0.000000	0.000000e+00	1.000000	1.000000	X	float64
									Y	float64
									LONGITUDE	float64
									LATITUDE	float64

Updated:

The image below shows the appropriate changes made to the dataset in order to make it more readable such as French to English translation, dropping "LONGITUDE", "LATITUDE", "X", and "Y", reason being is that the geocoordinates are repeated per district for multiple instances which means that it's a generic location for the District and not for the instance itself. I added the mean temperatures since there is a positive correlation between temperate and crime according to the scientific research paper found [here](#). The temperature Dataset was found at climate.weather.gc.ca and was preprocessed by day in an excel file. I changed the Data type of the "Date" column to datetimen64 in order to be able to sort and split

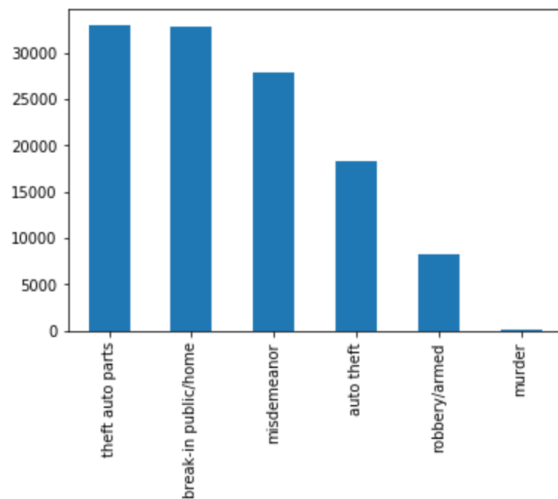
the dataset during the feature engineering step. Lastly, I transformed the " District" column from float to String type as those numbers represent physical locations.

	Category	Date	Part_of_day	District	mean_temp	
0	auto theft	2018-09-13	day	30.0	20.2	Int64Index: 149650 entries, 0 to 149649
1	theft auto parts	2018-09-13	night	33.0	20.2	Data columns (total 5 columns):
2	theft auto parts	2018-09-13	evening	21.0	20.2	Category 149650 non-null object
3	theft auto parts	2018-09-13	day	38.0	20.2	Date 149650 non-null datetime64[ns]
4	robbery/armed	2018-09-13	evening	35.0	20.2	Part_of_day 149650 non-null object
						District 149650 non-null object
						mean_temp 149650 non-null float64

```

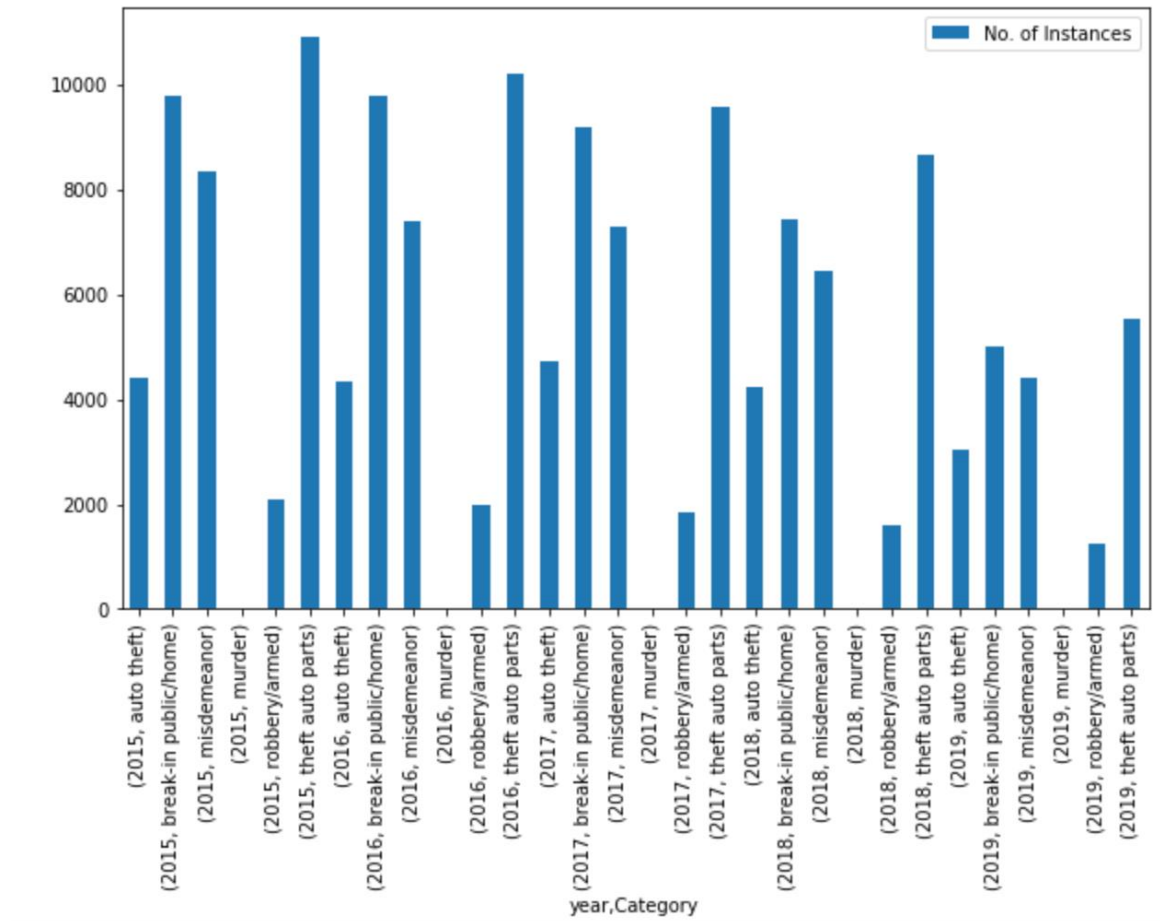
theft auto parts      33001
break-in public/home 32742
misdemeanor          27919
auto theft            18226
robbery/armed         8281
murder                116
Name: Category, dtype: int64

```



EXPLORATION

Considering that there are not any numerical values in this data set I had to explore it by counting the number of rows based on specific conditions such as which part of the day has the most amount of records, what are the numbers of all instances through the years etc. On the left you will see a value count of all the categories followed by a bar chart and a value count table which shows the discrepancy between each category by year below. All the crime categories are decreasing as the years progress from 2015-2019 with the only increase taking place from 2017-18 in the murder rate. Surprisingly most occurrences occur during the day period between the hours of 8AM-16:00 aside from the murder count with the top part between 16:01-00:00(midnight).



2015	auto theft	4418	2016	auto theft	4352	2017	auto theft	4732
	break-in public/home	9796		break-in public/home	9796		break-in public/home	9197
	misdemeanor	8356		misdemeanor	7405		misdemeanor	7302
	murder	27		murder	23		murder	26
	robbery/armed	2097		robbery/armed	1974		robbery/armed	1851
	theft auto parts	10915		theft auto parts	10207		theft auto parts	9556
2018	auto theft	4236	2019	auto theft	3026			
	break-in public/home	7429		break-in public/home	5021			
	misdemeanor	6432		misdemeanor	4405			
	murder	31		murder	10			
	robbery/armed	1592		robbery/armed	1244			
	theft auto parts	8666		theft auto parts	5528			

Looking at figures above you notice the slight increase in events for auto theft and murder in 2017 as well as the murder count increase for 2018. I decided to investigate this farther by comparing the average temperature of the dataset by incident and then focusing specifically on the categories and year listed above. By glancing over these tables, you notice the clear increase in temperature where the number of instances is higher.

Average Temperature per Incident

Temperature for Increased Categories

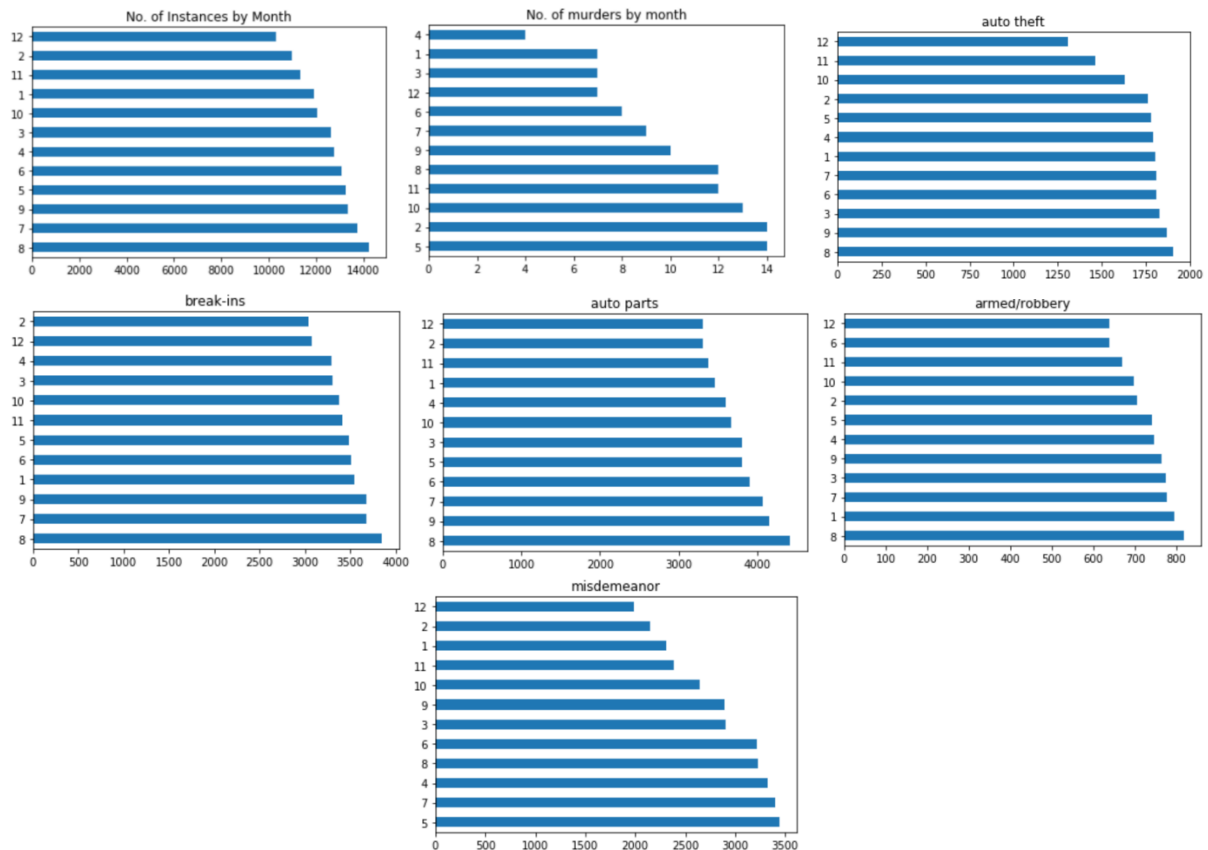
Murder Temp 2017: 9.52
 auto theft temp 2017: 8.59
 Murder Temp 2018: 9.10

Dataset: 8.51
 Murder: 8.63
 Theft Auto Parts: 8.56
 Auto Theft 8.05
 Misdemeanor 9.29
 Armed/Robbery 7.81
 break-in 8.21

For the last part of my exploration I wanted to see which part of the day do most incidents takes place and which months are most active. The “day” time period between 8:00 -16:00 has the highest frequency for the non-violent crimes and the “evening” time period between 16:00-00:00 has the highest frequency for the more aggressive crimes such as murder and armed robbery.

Auto Parts		Auto theft		break-in		Misdemeanor		robbery/armed		murder	
-----		-----		-----		-----		-----		-----	
day	17265	day	8886	day	14035	day	15213	evening	4026	evening	49
evening	11815	evening	4984	evening	12319	evening	9016	day	2678	night	44
night	3921	night	4356	night	6388	night	3690	night	1577	day	23

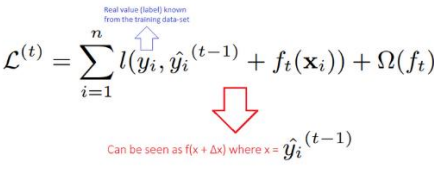
The chart below shows the number of occurrences per month from the dataset followed by a break-down per category.



According to the side bar charts above four out of the six categories have August as the highest occurrence month however “Murder” and “Misdemeanor” occur most often in May.

Algorithms

For this project I will use the following algorithms

Logistic Regression	Random Forest	XGBOOST
$p = \frac{1}{1 + e^{-(b_0 + b_1 x_1 + b_2 x_2 + \dots + b_p x_p)}}$	$E(S) = \sum_{i=1}^c -p_i \log_2 p_i$	$\mathcal{L}^{(t)} = \sum_{i=1}^n l(y_i, \hat{y}_i^{(t-1)} + f_t(\mathbf{x}_i)) + \Omega(f_t)$ 
<p>Typically, this algorithm is used for binary classification however there is an option to set the multi_class parameter to “multinomial” or “ovr” which is more appropriate for this case since I am trying to classify between six classes. The way logistic regression executes this is by using the one-vs-rest scheme or the cross-entropy loss depending on the choice above. The logistic curve is constructed by the natural logarithm from the odds of the target variable. This algorithm makes a great choice for this dataset because the predictors do not have to be normally distributed or have equal variance in each group.</p>	<p>The equation above is for the “Entropy” which is the backbone of information Gain as part of the Decision Trees algorithm. Random Forest is just many decision trees working together which also where the Ensemble Models name comes from. Features are split based on the amount of information which can be gained (range 0-1) and the lower the entropy the higher the information that can be gained. In order for the decision tree to no longer be able to make split the entropy has to be 0 which means information gain is 1. The way Random Forrest gets to a result is by taking all the answers from each tree and returning the answer which occurred most often.</p>	<p>The equation corresponds to the Loss function and regularization of the Gradient Boosted Trees algorithm. This algorithm is trying to approximate a function by tuning the loss function as well as other regularization techniques. As part of the Ensemble Methods XGBoost is a boosting algorithm where Random Forrest is a Bagging algorithm. What this means is that each point has an initial weight and after each iteration or better yet after each learner the algorithm punishes miss classified points made from the previous learner. This algorithm will create sequential decision trees and assigns weights to each record. These weights are the Probabilities of each record getting selected by by the first tree. After the first tree finishes training and it attempts to make classifications it will take the weights of the misclassified records and increase their weights. The first tree is called the 1st weak classifier and after its training is complete the 2nd tree/weak classifier will use the weights created from the first tree in order to select the more important records.</p>

METHODOLOGY

Data-preprocessing & Feature Engineering

As the first step of pre-processing I split the dataset into features and labels prior to dealing with each column and it according data type. Since datetimen64 datatypes will be difficult to work with I decided to split the Date column into three parts, Day of the week: 1-7, Month:1-12 and year: 2015-2019. Datetimes cannot be standardized the way most integers values are because they are Cyclical continuous features which means that they use difference ranges than just -1 to 1 or 0 -1. For example, if you think of time the difference between five minutes before noon and five minutes after noon is only 10 minutes and in order to help the machine understand this difference vs assuming that the difference is larger than it truly is I split the day, week and month column into 6 additional columns. Then, I compute the appropriate sin and cos in order to map each variable into a circle so the smallest value appear next to the largest value in the column range.

The code example of the day of week and month feature is the following:

```
features['day_sin'] = np.sin((2*np.pi)/30*features.DayofWk)
features['day_cos'] = np.cos((2*np.pi)/30*features.DayofWk)
features['mnth_sin'] = np.sin((features.Month-1)*(2*np.pi/12))
features['mnth_cos'] = np.cos((features.Month-1)*(2*np.pi/12))
```

Below I placed the dataset prior to the day and month processing followed by an example of the properly processed version

	Category	Date	Part_of_day	District	DayofWk	Month	year	mean_temp
0	motor vehicle theft	2018-09-13	day	30.0	4	9	2018	20.2
1	theft of motor vehicle parts	2018-09-13	night	33.0	4	9	2018	20.2
2	theft of motor vehicle parts	2018-09-13	evening	21.0	4	9	2018	20.2
3	theft of motor vehicle parts	2018-09-13	day	38.0	4	9	2018	20.2
4	Robbery/violence/armed/armored vehicle	2018-09-13	evening	35.0	4	9	2018	20.2

	Date	Part_of_day	District	mean_temp	DayofWk	Month	year	day_sin	day_cos	mnth_sin	mnth_cos
0	2018-09-13	day	30.0	0.818692	4	9	0.75	0.743145	0.669131	-0.866025	-0.5
1	2018-09-13	night	33.0	0.818692	4	9	0.75	0.743145	0.669131	-0.866025	-0.5
2	2018-09-13	evening	21.0	0.818692	4	9	0.75	0.743145	0.669131	-0.866025	-0.5
3	2018-09-13	day	38.0	0.818692	4	9	0.75	0.743145	0.669131	-0.866025	-0.5
4	2018-09-13	evening	35.0	0.818692	4	9	0.75	0.743145	0.669131	-0.866025	-0.5

I drop the Data column as this is not a forecasting problem(if I attempted to predict the number of crimes for a specific date I would have to convert the data column as the index and so on).The integer datatypes in features have to be scaled in order to fit the model properly and string types have to

encoded using the `get_dummies()` method to turn them into numbers of either 0 or 1 depending on which one below to each instance.

The labels are converted into integers by using the label encoder which assigns each category to a number depending on which one occurred. Since in this case I have 6 different categories the numbers found in the `target_final` variable will be a list of ranged from 0 to 5.

After the completion of the Feature Engineering step it's time to split them into the training and testing set. I chose to split by 80/20 with the training set containing 119,720 samples and the test set 29,930.

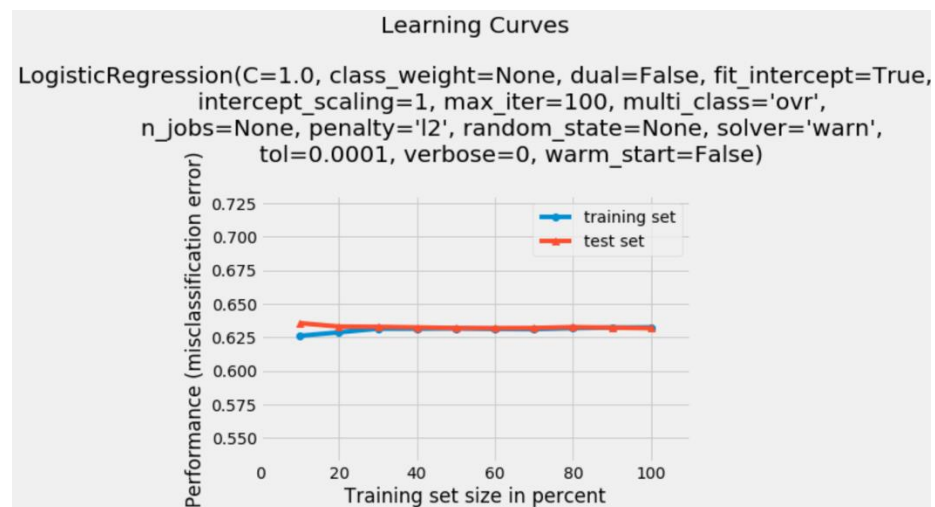
BENCHMARK MODEL

As a bench model that randomly guesses the one of the six incidents which will establish an accuracy score of 16%.

Model training

I trained the XGBoost, Random Forrest and Logistic Regression models as mentioned earlier which resulted into a poor outcome:

```
RF Final F-score on the testing data: 0.3277
LOG Final F-score on the testing data: 0.3682
XGBOOST Final F-score on the testing data: 0.3687
```



As you can see in the image above the model is overfitting and therefore is not able to be robust enough to make accurate classifications from new features as it only memories this specific dataset. I will attempt to use Cross validation in order to utilize a greater amount of my dataset

I choose to optimize the Logistic regression model due it's faster performance. The way I turned my Logistic Regression is displayed in the code below:


```

logistic = LogisticRegression()

# Create regularization penalty space
penalty = ['l1', 'l2']

# Create regularization hyperparameter space
C = np.logspace(0, 4, 10)

# Create hyperparameter options
hyperparameters = dict(C=C, penalty=penalty)

```

Best Penalty: l1
Best C: 1291.5496650148827

After my Grid Search results there was no difference and I received the same accuracy score which was extremely disappointing. Below the metrics I placed a confusion matrix which shows

Logistic Regression after GridSearch Accuracy score on testing data: 0.3679
Logistic Regression after GridSearch F-score on the testing data: 0.3679

```

array([[ 506,  444,  253,    0,   62,  288],
       [1538, 4482, 2659,   13,  893, 2828],
       [ 400,  739, 1001,    1,  175,  709],
       [   0,    0,    0,    0,    0,    0],
       [   0,    0,    0,    0,    0,    0],
       [1740, 2690, 2823,    4,  660, 5022]], dtype=int64)

```

CONCLUSION

Alternative solutions

- 1) Due to the lack of features in the dataset I would recommend attempting to gather additional information such as number of units dispatched per instance and the exact location of each incident. If this information is unavailable then I would use Independent component Analysis in order to generate more features however, there is no better replacement than real data.
- 2) If this dataset was all I had to work with then I would attempt to train a Neural Network or combine some of the target variables in order to turn this problem into a binary outcome of “violent” and “non-violent crimes” which will have it’s own challenges as almost 90% of the data is constructed from non-violent crimes.

Project Recap

Although the results are disappointing the slight upside is that they are better than random chance. Aside from the results, I enjoyed working on this project especially figuring out how to properly engineer the datetime features. It would be great if we were provided a dataset based on the topic of our choosing from Udacity and then allow us to complete the machine learning pipeline from Data cleaning to final prediction.