# Metropolis crime

## Dataset:

The Metropolis Police Department(MPD) has published a record of the crimes over a period. Our objective is to predict the type of crime based on the data provided.

The input has 5 features incident date time, incident coordinates(x,y), number of victims, and location type.

The response variable ‘Crime type’ is a categorical variable which can take 11 distinct values.

- “AGG ASSAULT"

- “AUTO THEFT"

- “BURGLARY-NONRES"

- “BURGLARY-RESIDENCE"

- “HOMICIDE"

- “LARCENY-FROM VEHICLE"

- “LARCENY-NON VEHICLE"

- “RAPE"

- “ROBBERY-COMMERCIAL"

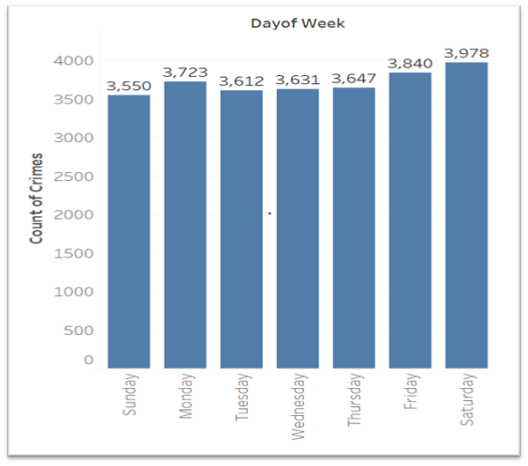
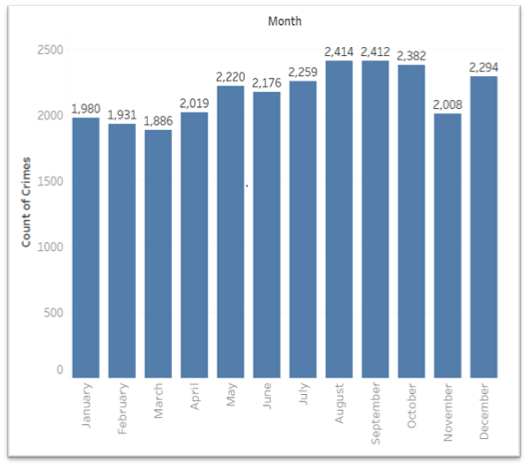
- “ROBBERY-PEDESTRIAN"

- “ROBBERY-RESIDENCE"

## Approach:

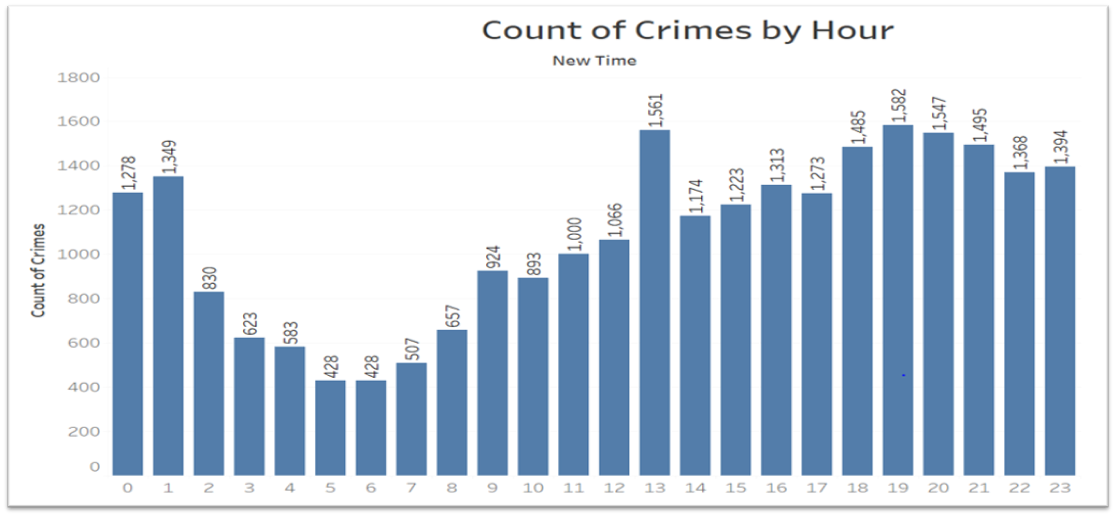
### Exploratory data analysis:

The below are graphs obtained by performing basic Exploratory Data Analysis

The above graphs represent the count of the crimes from Metropolis city and they are segregated based on the month and day. We can intuitively say that the crimes do not appear to have a particular trend.

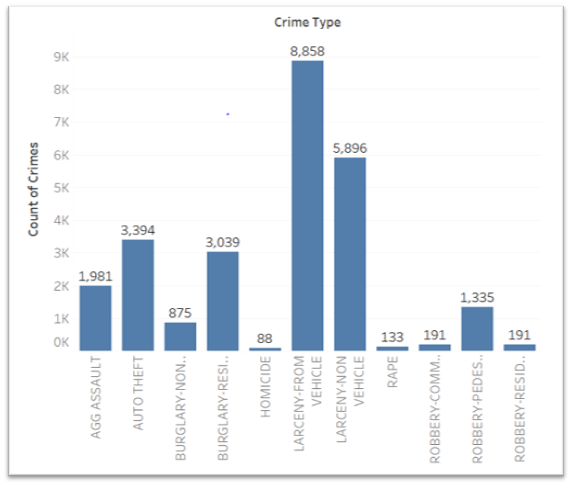
The below graph shows that the number of crimes are more during evenings and very less during the day time.



Data preprocessing:

Upon doing our basic EDA, we found and removed few data points where the field incident\_timestamp was given as ‘NaT’ and also few outliers in our data which were the records from the year 1944 & 1948. Further, for the field num\_victims, NaN values were populated for the entire HOMICIDE class, after further instructions from the instructor we have changed the value to 0. We have split our data set at 80:20 ratio for our train and test sets.

As we can see from the below graph the entire dataset is totally imbalanced and any prediction made will be more biased towards the dominant classes like(LARCENY). So, we have planned to proceed further to resample our training dataset (Up Sampling the smaller classes and Down Sampling the Larger Classes).



### Feature engineering:

Based on the Features available in the dataset we have derived few new features like NewDate, NewTime, Month, DayofWeek and Year from the incident\_datetime.

### Models:

Initially, we have run all our models on our imbalanced dataset to obtain a base line value which ranged from 36 – 40% for all the models.

As explained above we have resampled our data and used the resampled training dataset to check if the accuracy score is improved or not. But, to our surprise there was not much change in the accuracy and other evaluation metrics after training the model with the Resampled training set.

In the above two iterations of runs we have identified that Random Forest gave us reliable results and we then focused more on Random Forest to improve the prediction Accuracy. We have tried different combinations and hyper tuned our parameters for Random Forest where there is no Overfit of the data. In fact, the final test results mentioned by the instructor were inline and similar to the proposed accuracy in the presentation.

The following Classification models were implemented to solve the problem at hand.

* **KNN**

KNN is a non-parametric method used for both Classification and Regression.

In k-NN classification, an object is classified by a majority vote of its neighbors, with the object being assigned to the class most common among its k nearest neighbors. We have identified k as 71 based on the formula

K = sqrt (Number of Records) / 2

We got an accuracy of 43% which was not very satisfactory. So, we drifted more towards other complicated and sophisticated models which are more frequently used for Classification.

* **SVC** – Rbf, Linear kernel

SVC, again is a Classifier which supports both Linear and non-Linear Classifications based on the kernel type used. We have used Linear and rbf kernel for our project and after hyper parameter tuning we have got an accuracy of 39.42, 45.44 for linear and non-linear respectively

* Neural Network

Neural Network was one of the models we thought that would give us the best results. But, in contradiction we have got a low accuracy rate of 39% for this model.

* L**DA**

LDA (Linear Discriminant Analysis) is a technique which is more preferred when the problem at hand is a multiclass classification problem. We have got a 40% accuracy for this model with our test dataset.

* **Random Forest**

RF is an ensemble learning method for Classification, Regression and other tasks, that operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classification. With hyper parameter tuning and balanced weights assignment we have achieved an accuracy of 45.2%. The final test dataset was also predicted with the same accuracy.

* **OneVsRest Classifier**

Regular Classification models would work well for binary classification. But, the problem at hand is an imbalanced multi class classification problem. After careful study we found that OneVsRest classifier is a go to classifier for multi class imbalanced problems. This basically isolates each class from the other available classes and builds a separate tree for each class. This reduces the number of trees and improves the processing speed of the classifier. We used our Random Forest model as an input to this model which gave us an accuracy of 49%. This was a considerable increase in accuracy when compared to the Random Forest model alone.

|  |  |
| --- | --- |
| **Model** | **Accuracy** |
| KNN | 43% |
| SVC – linear | 39.42% |
| SVC -rbf | 45.44% |
| Neural Network | 38.9% |
| LDA | 40% |
| Random Forest | 45.2% |
| OneVsRest Classifier | 49% |

# Final Prediction:

After hyper tuning the parameters for various models and multiple iterations of testing, we decided to go forward with the Random Forest model which gave us an accuracy of 45%.