**Crime Type Prediction**

Yuko Kurokawa(301172140)

Data Analytics and Insights -Canadian Context, Centennial College

BA715: Business Analytics Capstone (SEC.001)

Professor David Parent

Professor Mustafa Ahmed

August 14, 2022

Table of Contents

[Executive Summary 7](#_Toc111833346)

[0.1. Executive Introduction 7](#_Toc111833347)

[0.2. Executive Objective 7](#_Toc111833348)

[0.3. Executive Model Description 7](#_Toc111833349)

[0.4. Executive Recommendations 7](#_Toc111833350)

[Introduction 8](#_Toc111833351)

[1.0. Background 8](#_Toc111833352)

[2.0. Problem Statement 8](#_Toc111833353)

[3.0. Objectives & Measurement 8](#_Toc111833354)

[4.0. Assumptions and Limitations 9](#_Toc111833355)

[Data Sources 9](#_Toc111833356)

[5.0. Data Set Introduction 9](#_Toc111833357)

[6.0. Exclusions 9](#_Toc111833358)

[7.0. Data Dictionary 9](#_Toc111833359)

[Data Exploration 11](#_Toc111833360)

[8.0. Data Exploration Techniques 11](#_Toc111833361)

[8.1. Step 1: Data dictionary information 11](#_Toc111833362)

[ID-related columns 11](#_Toc111833363)

[Similar columns 12](#_Toc111833364)

[Unaccounted columns 12](#_Toc111833365)

[8.2. Step 2: Dataset information 12](#_Toc111833366)

[8.3. Step 3: Missing values 14](#_Toc111833367)

[8.4. Step 4: Similar columns 14](#_Toc111833368)

['offence' and 'MCI' 14](#_Toc111833369)

['location\_type' and 'premises\_type' 16](#_Toc111833370)

[Date and time information 16](#_Toc111833371)

[8.5. Step 5: Discriptive statistics and Distribution plot 17](#_Toc111833372)

[‘Long’ 17](#_Toc111833373)

[‘Lat’ 18](#_Toc111833374)

[8.6. Step 6: Bar graph 18](#_Toc111833375)

[‘MCI’ (Target variable) 19](#_Toc111833376)

[‘Division’ 21](#_Toc111833377)

[‘reportedmonth’ 22](#_Toc111833378)

[‘reportedday’ 23](#_Toc111833379)

[‘reporteddayofweek’ 24](#_Toc111833380)

[‘reportedhour’ 25](#_Toc111833381)

[‘Neighbourhood’ 26](#_Toc111833382)

[8.7. Step 7: Heat map 27](#_Toc111833383)

[Data Preparation and Feature Engineering 28](#_Toc111833384)

[9.0. Data Preparation Needs 28](#_Toc111833385)

[9.1. Removal of unnecessary columns 29](#_Toc111833386)

[9.2. Cap and Floor on ‘Long’ 29](#_Toc111833387)

[‘Long’ 29](#_Toc111833388)

[‘Lat’ 31](#_Toc111833389)

[9.3. Data types conversion 33](#_Toc111833390)

[10.0. Feature selection 34](#_Toc111833391)

[11.0. Train data and Validation data 38](#_Toc111833392)

[12.0. Upsampling (SMOTE) 39](#_Toc111833393)

[Model Exploration 40](#_Toc111833394)

[13.0. Modeling Approach/Introduction 40](#_Toc111833395)

[14.0. Model Technique #1: Random Forest 41](#_Toc111833396)

[14.1. Execution of Random Forest 41](#_Toc111833397)

[Important hyper parameters 42](#_Toc111833398)

[Result 42](#_Toc111833399)

[Grid Search 42](#_Toc111833400)

[14.2. Execution of Random Forest with Oversampling 43](#_Toc111833401)

[Result 44](#_Toc111833402)

[15.0. Model Technique #2: XGboost 44](#_Toc111833403)

[15.1. Execution of XGboost 45](#_Toc111833404)

[Important hyper parameters 45](#_Toc111833405)

[Result 45](#_Toc111833406)

[15.2. Execution of XGboost with Oversampling 46](#_Toc111833407)

[16.0. Model Technique #3: Multilayer Perceptron 47](#_Toc111833408)

[16.1. Execution of Multilayer Perceptron 48](#_Toc111833409)

[Important hyper parameters 48](#_Toc111833410)

[Result 48](#_Toc111833411)

[16.2. Execution of Multilayer Perceptron with Oversampling 50](#_Toc111833412)

[17.0. Model Comparison 50](#_Toc111833413)

[Model Recommendation 51](#_Toc111833414)

[18.0 Model Selection 51](#_Toc111833415)

[Accuracy 51](#_Toc111833416)

[F1 score 51](#_Toc111833417)

[19.0 Model Theory 51](#_Toc111833418)

[19.1 Model Assumptions and Limitations 52](#_Toc111833419)

[20.0 Model Sensitivity to Key Drivers 53](#_Toc111833420)

[Validation and Governance 53](#_Toc111833421)

[21.0. Categorical Variable Level Monitoring 53](#_Toc111833422)

[Acceptable ranges for Categorical Variables 53](#_Toc111833423)

[Acceptable ranges for Numerical Variables 56](#_Toc111833424)

[21.1. Caps & Floors 56](#_Toc111833425)

[Before Cap and Floor 57](#_Toc111833426)

[After Cap and Floor 57](#_Toc111833427)

[21.2. Missing Values 57](#_Toc111833428)

[21.3. Variable Drift Monitoring 58](#_Toc111833429)

[22.0. Model Health & Stability 59](#_Toc111833430)

[Initial Model Fit Statistics: 1st model ‘Random Forest’ 60](#_Toc111833431)

[Acceptable ranges: 60](#_Toc111833432)

[23.0. Risk Tiering (e.g., no action, report, refit, rebuild) 60](#_Toc111833433)

[Conclusion and Recommendations 61](#_Toc111833434)

[24.0. Impacts on Business Problem (Scope of the recommended model) 61](#_Toc111833435)

[25.0. Recommended Next Steps 61](#_Toc111833436)

[References 62](#_Toc111833437)

# Executive Summary

## Executive Introduction

There is a tremendous growth in population in Toronto, resulting in demand rising for policing service , especially for emergency responses. Crime type prediction model provides the most probable crime type to 911 dispatchers and police officers, which enables better preparation and more immediate responses.

## Executive Objective

The objective is to provide a crime prediction model for a 911 dispatcher and officer to improve emergency response. One of the five crime categories will be specified, such as assault, break and enter, auto theft, robbery, theft over. In order to identify the crime type, specific information are required, such as premises, hour, day, day of week, month, longitude, and latitude,

## Executive Model Description

The model was created using Random Forest which is one of the machine learning methods for classification. The model’s accuracy was approximately 64%.

## Executive Recommendations

Predicting a crime type will offer better preparation and immidiate response to 911 dispatchers and police officers, optimizing emergency responses. More robust responses could bring the safer city. Therefore, the use of this prediction model is recommended.

# Introduction

## 1.0. Background

Crime has adversely affected people and their society all over the world, and Toronto is no exception (Toronto Police Service – 2022 Operating Budget Request, n.d.). Toronto is the fastest-growing city in North America, and its population has been tremendously grown, resulting in demand rising for policing service. Therefore, Toronto Police Service requires more robust responses to serve community safety. Especially, in the case of emergency, life and death depend on even a seconds-delay in response (McKenzie, 2019).

## 2.0. Problem Statement

There is a situation that a 911 dispatcher can be unsure of the crime type when the caller unable to describe what is happening, and a police officer cannot clarify the crime type when they hear a call for help. To optimize the police response, they need to know the crime tendency in the city and most probable crime type corresponding to corelated features, such as location, time, and premise information.

## 3.0. Objectives & Measurement

I am going to help make a 911 response and a police officer response more efficient, which could contribute to the safety in the city by conducting this analysis. Using crime type prediction, I am going to improve both quantitative and qualitative factors, which are emergency response time and the survey results regarding response preparedness by police officers and 911 dispatchers. If this project is not implemented, it can be possible that more victims may not be rescued properly, and police officers may not be well-prepared and put in danger.

## 4.0. Assumptions and Limitations

Assumptions and limitations are listed in the following table:

*Table 1. Assumptions and Limitations*

|  |
| --- |
| **Assumptions and Limitations** |
| The model was built in the data between 2014-2021 |
| The data set is based on current status quo for crimes, people's lifestyle |
| Data collection was correctly done by police officers |

# Data Sources

## 5.0. Data Set Introduction

The data set is about crime data called ‘Major Crime Indicators (MCI)’ downloaded from Toronto Police Service’s portal website (Major Crime Indicator, 2022). Data was collected between 2014/01/01 – 2021/12/31. Rows are of 281,692, Columns are of 30.

## 6.0. Exclusions

In this case, there are no exclusions.

## 7.0. Data Dictionary

Data dictionary was cited from OPEN DATA DOCUMENTATION (Major Crime Indicators (MCI), 2022).

*Table 2. Data Dictionary*

|  |  |  |  |
| --- | --- | --- | --- |
| **Variable** | **Role** | **Data Type** | **Description** |
| X | Rejected | Float | N/A |
| Y | Rejected | Float | N/A |
| Index | Rejected | ID | Unique Identifier |
| event\_unique\_id | Rejected | ID | Offence Number |
| Division | Rejected | Category | Police Division where Offence Occurred |
| occurrencedate | Rejected | Category | Date of Offence |
| reporteddate | Rejected | Category | Date Offence was Reported |
| location\_type | Rejected | Category | Location Type of Offence |
| premises\_type | Rejected | Category | Premises Type of Offence |
| ucr\_code | Rejected | ID | UCR Code for Offence |
| ucr\_ext | Rejected | ID | UCR Extension for Offence |
| offence | Rejected | Category | Title of Offence |
| reportedyear | Rejected | Category | Year Offence was Reported |
| reportedmonth | Input | Category | Month Offence was Reported |
| reportedday | Input | Category | Day of the Month Offence was Reported |
| reporteddayofyear | Rejected | Category | Day of the Year Offence was Reported |
| reporteddayofweek | Input | Category | Day of the Week Offence was Reported |
| reportedhour | Input | Category | Hour Offence was Reported |
| occurrenceyear | Rejected | Category | Year Offence Occurred |
| occurrencemonth | Rejected | Category | Month Offence Occurred |
| occurrenceday | Rejected | Category | Day of the Month Offence Occurred |
| occurrencedayofyear | Rejected | Category | Day of the Year Offence Occurred |
| occurrencedayofweek | Rejected | Category | Day of the Week Offence Occurred |
| occurrencehour | Rejected | Category | Hour Offence Occurred |
| MCI | Target | Category | MCI Category of Occurrence |
| Hood\_ID | Rejected | ID | Identifier of Neighbourhood |
| Neighbourhood | Rejected | Category | Name of Neighbourhood |
| Long | Input | Float | Longitude Coordinates  (Offset to nearest intersection) |
| Lat | Input | Float | Latitude Coordinates  (Offset to nearest intersection) |
| ObjectId | Rejected | ID | N/A |

# Data Exploration

## 8.0. Data Exploration Techniques

Several techniques were used in the analysis along the data exploration.

*Table 3. Data Exploration Techniques*

|  |  |
| --- | --- |
| **Techniques** | **Purposes** |
| Value counts | Check the difference in the similar columns |
| Null value ditection | Handle missing values |
| Descriptive statistics | Check distributions and detect outliers in numerical variables |
| Distribution plot | Display distributions and outliers in numerical variables |
| Bar graph | Check distributions and detect outliers in categorical variables |
| Heat map | Look into correlations in categorical variables |

## 8.1. Step 1: Data dictionary information

First of all, I checked the data dictionary to know what they represent. After check all the columns, it was found that there are some columns that seem to have similar information. I also found that there are many ID-related columns and some columns unaccunted in the data dictionary.

### ID-related columns

The columns below will be dropped because they are ID-related.

* ‘Index’
* 'event\_unique\_id'
* 'ucr\_code'
* 'ucr\_ext'
* ‘Hood\_ID'

### Similar columns

Between the columns below, one will be used, and the other will be removed in the model.

* 'offence' and 'MCI'
* 'location\_type' and 'premises\_type'
* The columns with ‘occurrence’ and the ones with ‘reported’

### Unaccounted columns

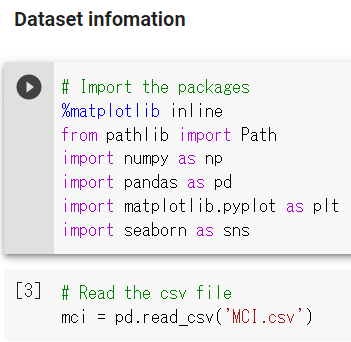
Since the collumns below were unaccounted in the data dictionary, so they will be dropped in the model.

* ‘X’
* ‘Y’
* 'ObjectId'

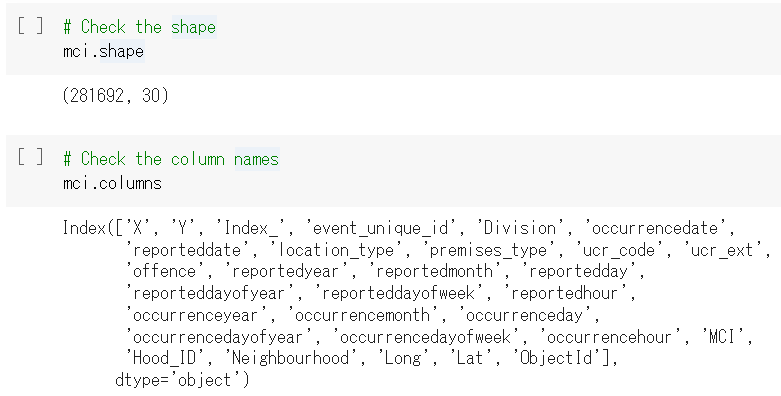
## 8.2. Step 2: Dataset information

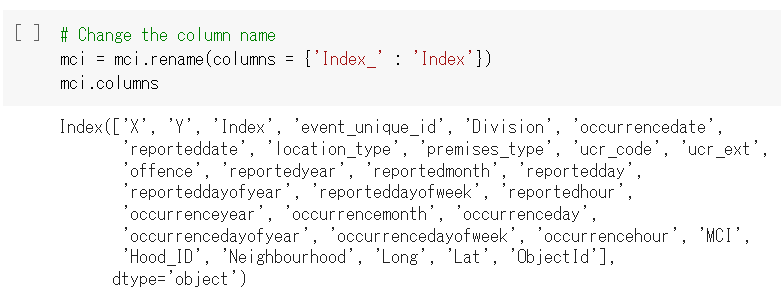
The dataset was downloaded and explored in Python. The data shape and column names were checked.

*Figures 1. Dataset information*



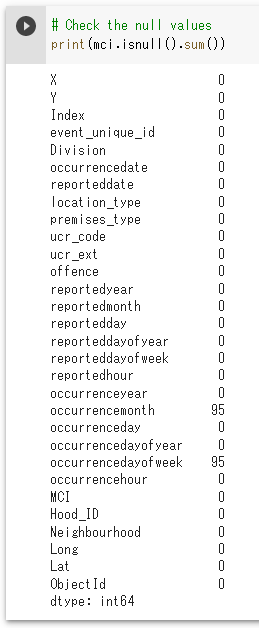
*Figures 1. Dataset information*





## 8.3. Step 3: Missing values

*Figures 2. Missing values*

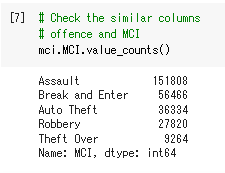
 Only ‘occurrencemonth’ and ‘occurrencedayofweek’ have null values. Therefore, it was determined that the columns with ‘reported’ were used for date and time information in the model.

## 8.4. Step 4: Similar columns

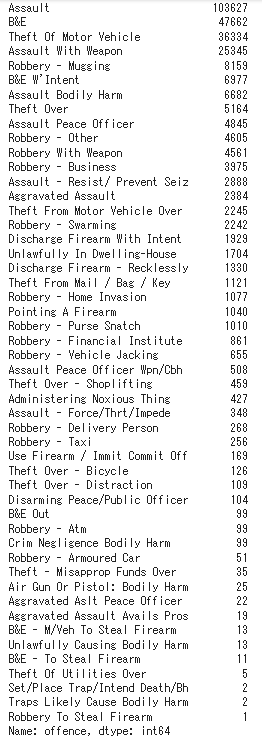
### 'offence' and 'MCI'

The values in 'offence' are aggregated into fewer categories in 'MCI'. Therefore, ‘MCI’ was taken as a target variable.

*Figures 3. 'offence' and 'MCI'*



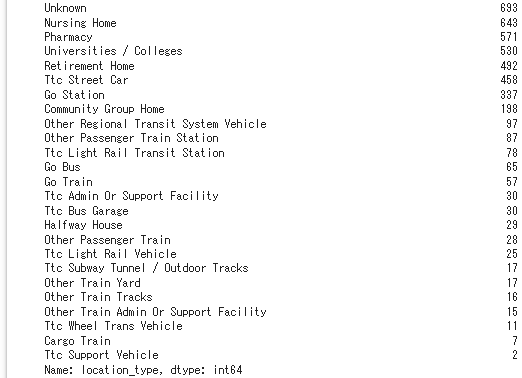
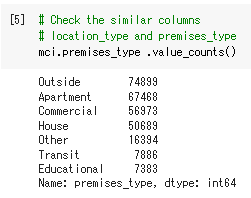




### 'location\_type' and 'premises\_type'

The values in 'location\_type' are aggregated into fewer categories in 'premises\_type'. Therefore, 'premises\_type' was taken as an input.

*Figures 4. 'location\_type' and 'premises\_type'*



### Date and time information

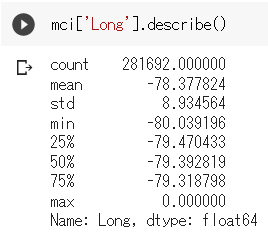
After deciding that ‘reported’ columns would be used as inputs, I looked deeper into ‘reported’ columns. Consequently, it was found that the values of 'reporteddate' are separated into‘reportedyear’, 'reportedmonth', 'reportedday', 'reportedhour'. Also, I decided to exclude ‘reportedyear' and 'reporteddayofyear' to avoid too much features for date information.

## 8.5. Step 5: Discriptive statistics and Distribution plot

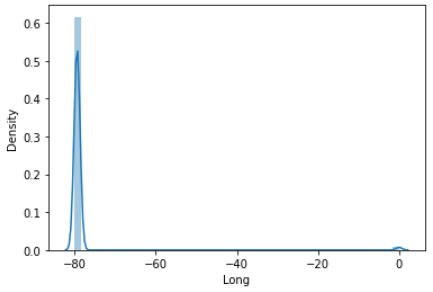
Discriptive statistics for numerical features were showed to detect outliers.

### ‘Long’

*Figures 5. ‘Long’*



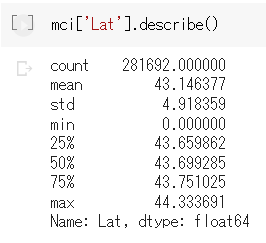




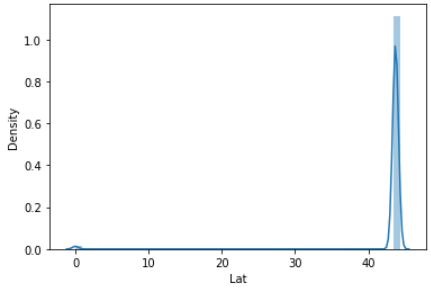
There are some outliers with the value of 0.0. Cap and Floor will be performed.

### ‘Lat’

*Figures 6. ‘Lat’*







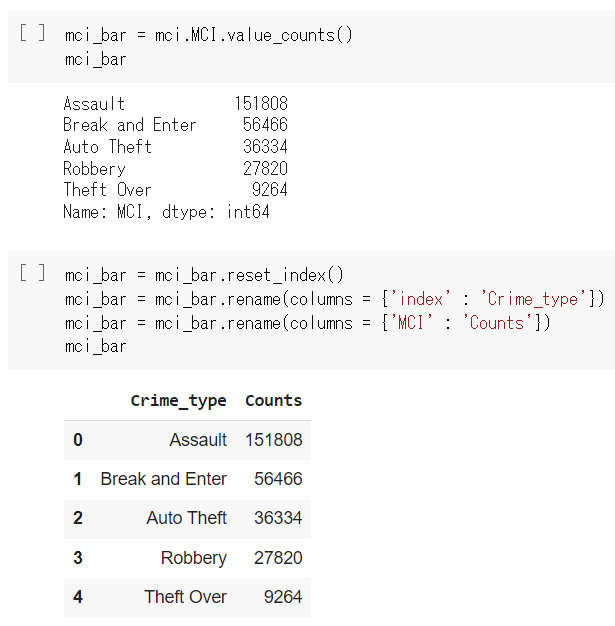
There are some outliers with the value of 0.0. Cap and Floor will be performed.

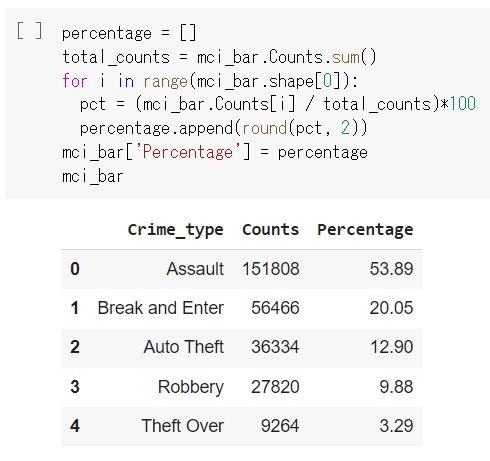
## 8.6. Step 6: Bar graph

In order to check distributions of categorical variables, bar graph was used. The code for was almost the same for each bar graph, so the one for ‘MCI’ data is shown.

### ‘MCI’ (Target variable)

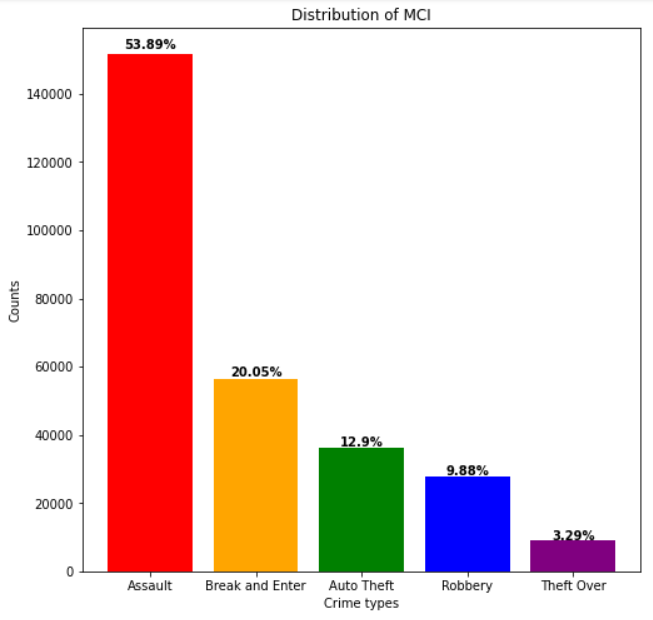
*Figures 7. ‘MCI’*





*Figures 7. ‘MCI’*

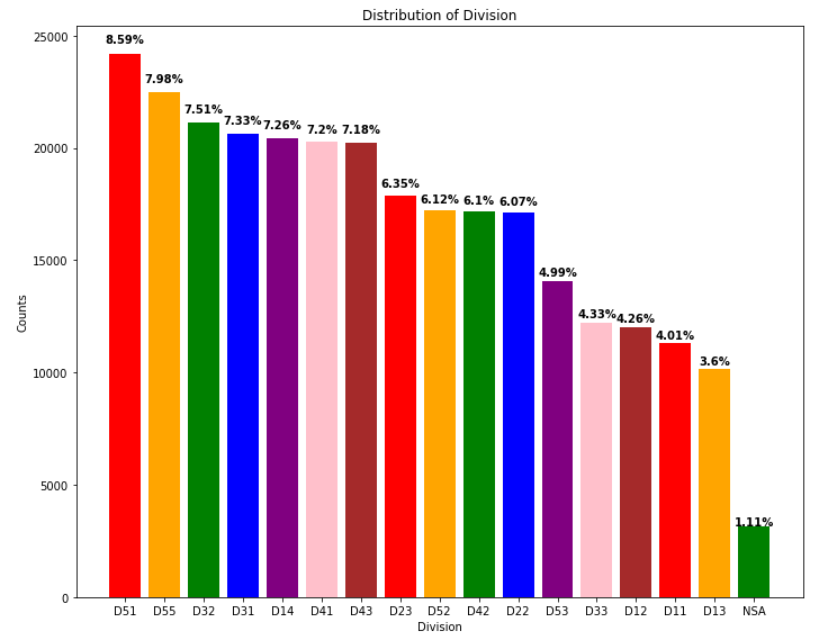




The distribution of ‘MCI’ is imbalanced. To handle this imbalance, SMOKE will be used.

### ‘Division’

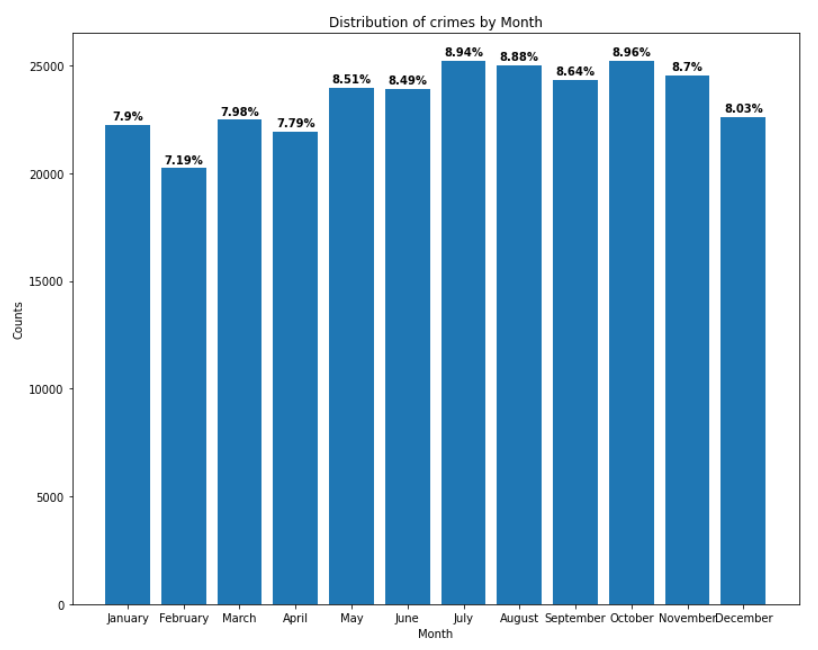
*Figures 8. ‘Division’*



‘Division’ is inbalanced.

### ‘reportedmonth’

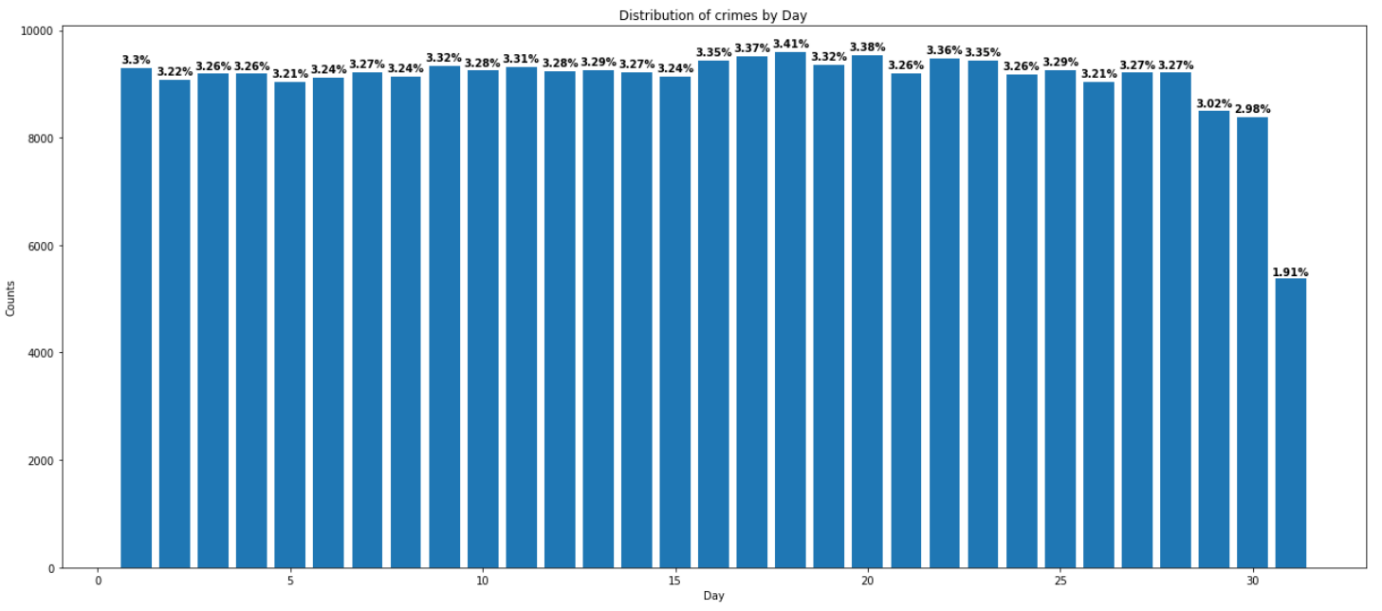
*Figures 9. ‘reportedmonth’*



Comparing to winter and spring seasons, summer and fall seasons have more crime occurrence.

### ‘reportedday’

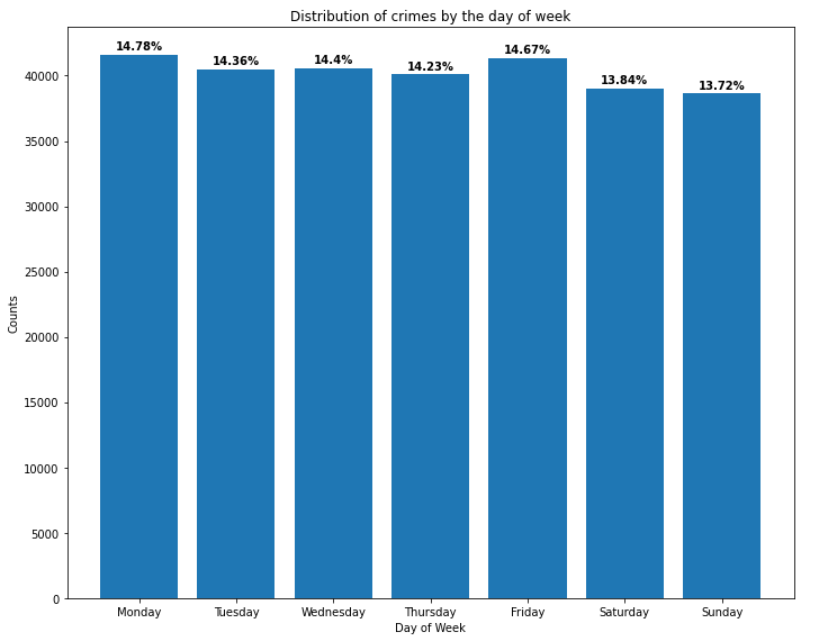
*Figures 10. ‘reportedday’*



The day of 31st has fewer crimes since it is in only seven months a year.

### ‘reporteddayofweek’

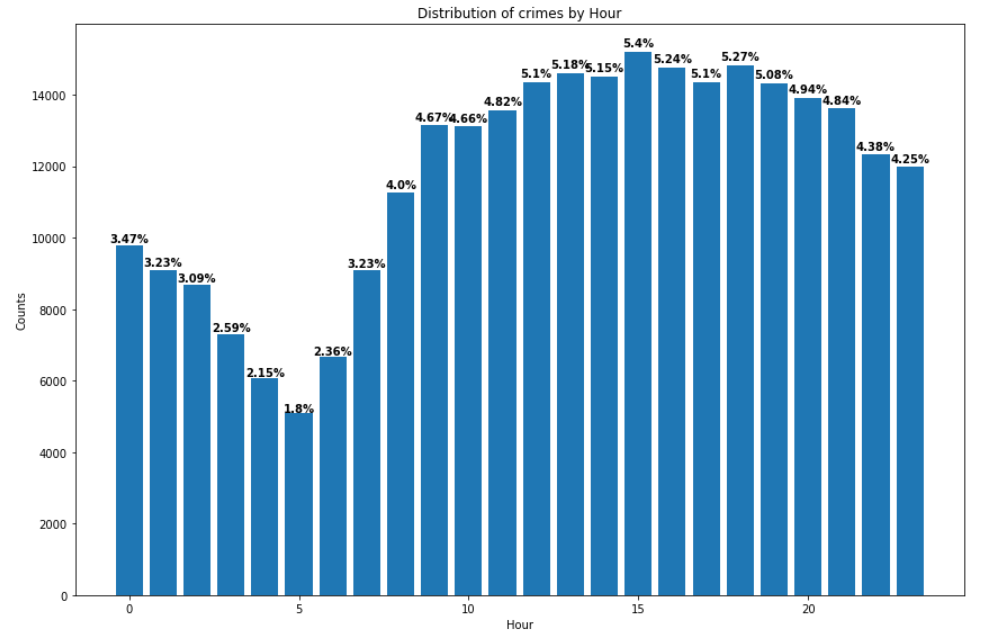
*Figures 11. ‘reporteddayofweek’*



Saturday and Sunday have less crimes.

### ‘reportedhour’

*Figures 12. ‘reportedhour’*



Crimes happen more during the day than after midnight.

### ‘Neighbourhood’

*Figures 13. ‘Neighbourhood’*

## 8.7. Step 7: Heat map

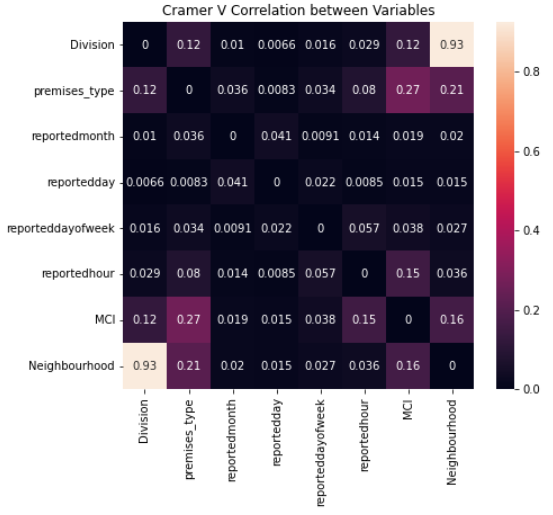
In order to see correlation among the categorical variables, Heat map was performed.

*Figures 14. ‘Heatmap’*





*Figures 14. ‘Heatmap’*



‘Division’ and ‘Neighbourhood’ are highly correlated.

# Data Preparation and Feature Engineering

## 9.0. Data Preparation Needs

After exploring each column, the following tasks need to be performed:

1. Removal of the ID-related columns, the unaccounted columns, the ‘occurrence’ columns, the similar columns (‘offence’ and ‘location\_type’), and unnecessary date columns
2. Cap and Floor on ‘Long’ and ‘Lat’
3. Data types conversion into category on ‘MCI’, ‘Division’, ‘premises\_type’, ‘reportedmonth’, ‘reportedday’, ‘reporteddayofweek’, ‘reportedhour’, and ‘Neighbourhood’

## 9.1. Removal of unnecessary columns

Unnecessary columns were dropped.

*Figures 15. Removal of unnecessary columns*



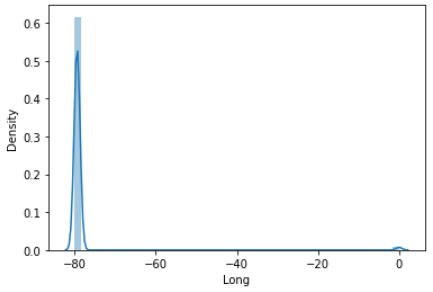
## 9.2. Cap and Floor on ‘Long’

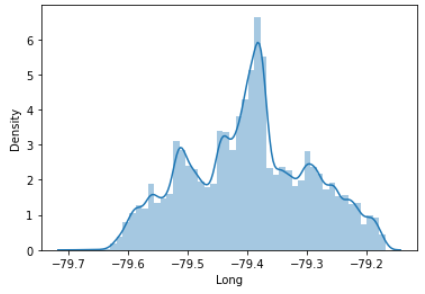
Cap and Floor was performed on ‘Long’ and ‘Lat’ to deal with outliers that affects models.

### ‘Long’

**Distribution plot (Before and After)**

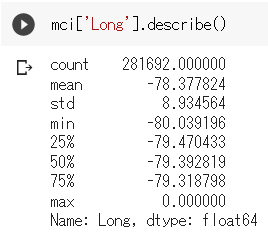
*Figures 16. Distribution plot*

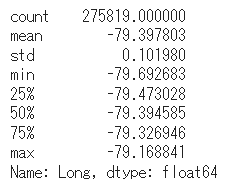
****Before:

After:

**Statistics (Before and After)**

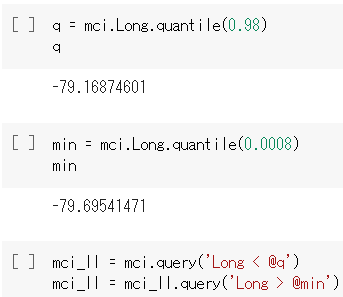
*Figures 17. Statistics*

Before:

After:

Code:

*Figures 18. Code for Cap and Floor*

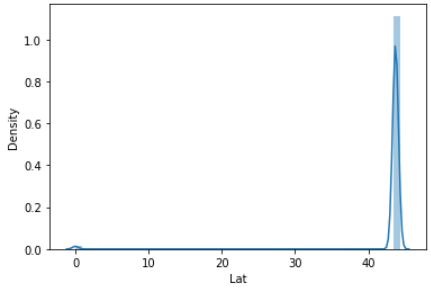


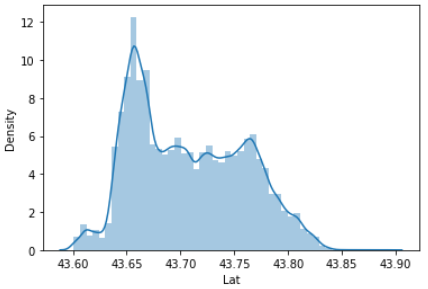


### ‘Lat’

**Distribution plot (Before and After)**

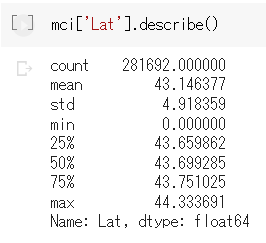
*Figures 19. Distribution plot*

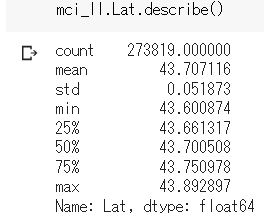
Before:

After:

**Statistics (Before and After)**

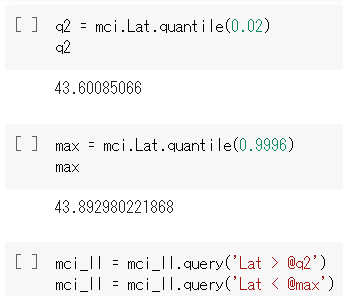
*Figures 20. Statistics*

Before:

After:

Code:

*Figures 21. Code for Cap and Floor*

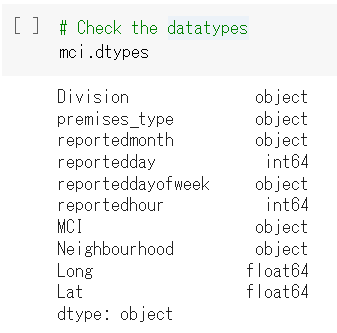




## 9.3. Data types conversion

After checking the current data types, ‘object’ features were converted into ‘category’. Also, ‘reportedhour’ should be treated as category, so it was transformed into ‘category’.

*Figures 22. Data types conversion*



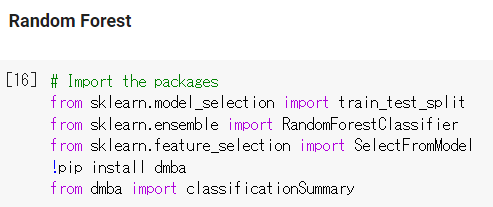
*Figures 22. Data types conversion*



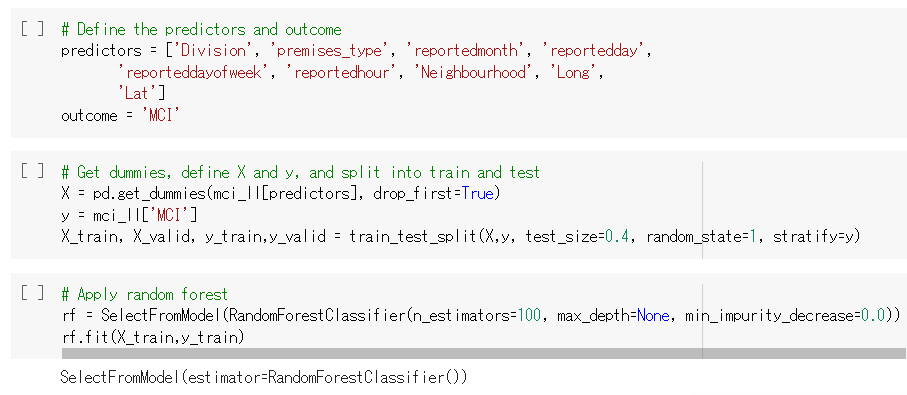
## 10.0. Feature selection

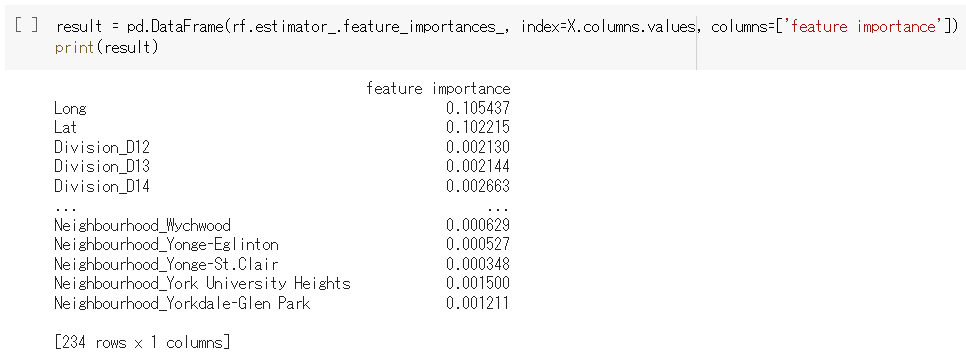
After the data preparation, Random Forest was conducted to select the important features and reduce demensionality. Before applying the model, One-Hot Encoding and train\_test\_split were performed.

*Figures 23. Feature selection*



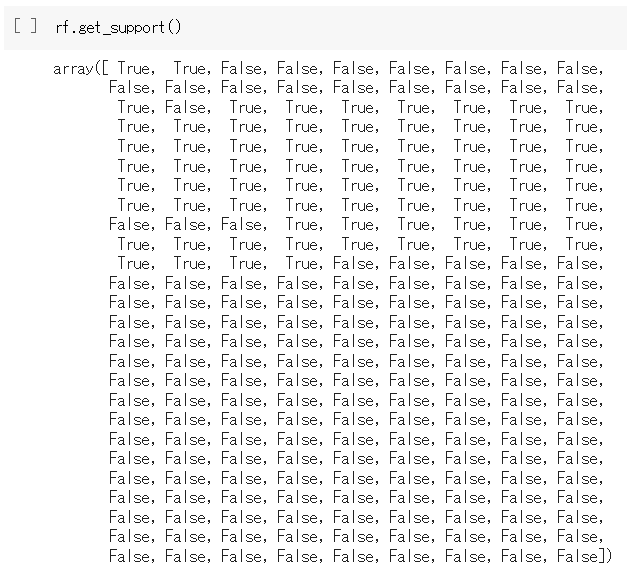
*Figures 23. Feature selection*





‘Long’ and ‘Lat’ are highly important compared to other features. Consequently, I used get\_support to obtain selected features.

*Figures 23. Feature selection*



*Figures 23. Feature selection*



In the result, 74 features were selected. Checking against the original columns, it was found that ‘Division’ and ‘Neighbourhood’ should be dropped. The following is the final selected columns.

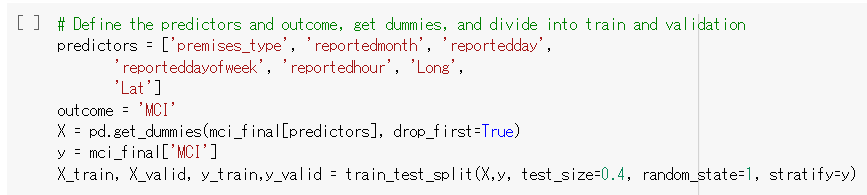
*Figures 23. Feature selection*

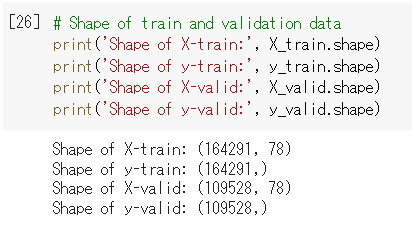


## 11.0. Train data and Validation data

The data with final selected columns were split into train data and validation data.

*Figures 24. Train data and Validation data*

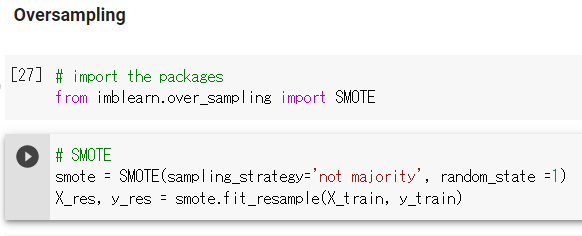




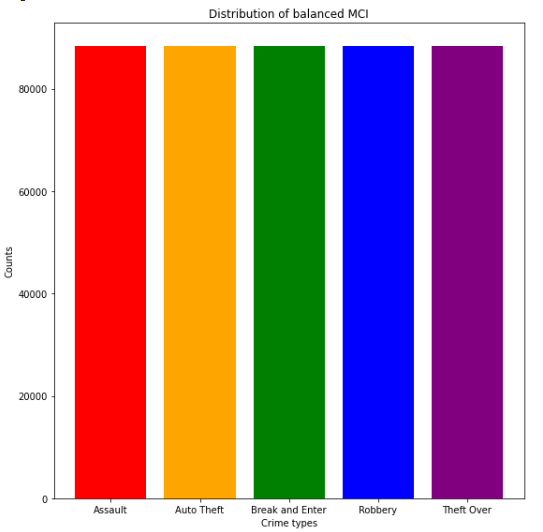
## 12.0. Upsampling (SMOTE)

Since the values in ‘MCI’ are inbalanced, resampling may be needed. Hence, I created resampled train data using SMOTE apart from the original train data created in the previous section. The reason why oversampling was chosen over undersampling is to keep the sufficient number of rows. Regarding the parameters of SMOTE, ‘not majority’ was selected for sampling\_strategy to match the values with the most frequent value which is ‘Assault’.

*Figures 25. Upsampling (SMOTE)*



‘MCI’ with oversampling



# Model Exploration

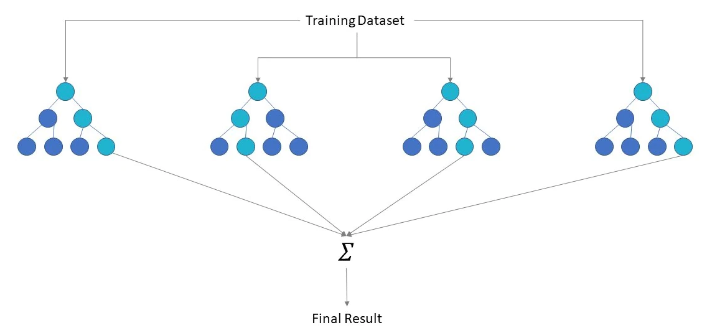
## 13.0. Modeling Approach/Introduction

The objective is to create a crime type prediction model, so it is necessary to build classification models. Although there are many classification algorithms, Random Forest, XGboost, Multilayer perceptron (MLP) were selected. Firstly, Random Forest was selected because most of features are categorical, and some outliers are retained. Secondly, XGboost was chosen because boosting method may show better accuracy than Random Forest. Lastly, I selected MLP because it is used in many articles for crime prediction (Walczak, 2021).

## 14.0. Model Technique #1: Random Forest

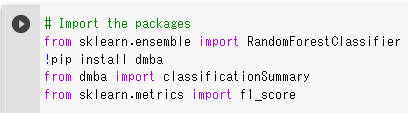
Random Forest is a machine learning algorithm that combines multiple decision trees to reach the best result (IBM Cloud Education, 2020). It is one of the ensemble learning methods using bagging. In the bagging method, several data samples are generated using bootstrapping, and each model is trained independenty. Subsequently, the best prediction is given. The benefit of Bagging is to minimize the overfitting.

*Figures 26. Random Forest (IBM Cloud Education, 2020)*



## 14.1. Execution of Random Forest

*Figures 27. Execution of Random Forest*



### Important hyper parameters

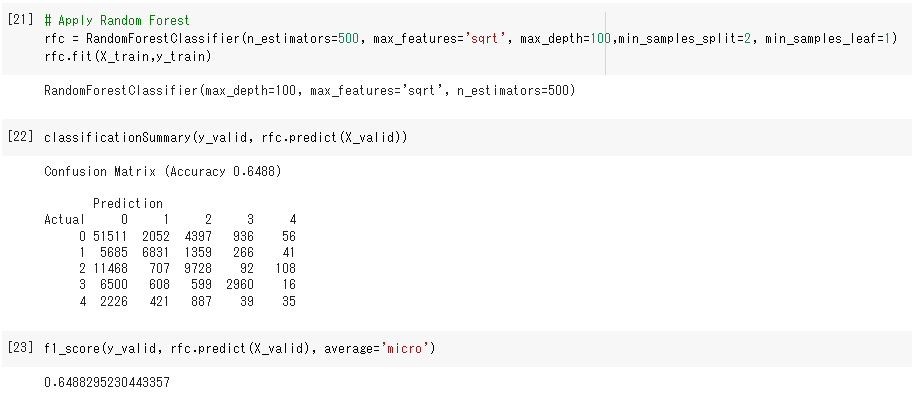
Although there are many hyper parameters, the ones in the following are more important (Koehrsen, 2018). These parameters were decided after try and error.

*Table 4. Important hyper patameters for Random Forest (Koehrsen, 2018)*

|  |  |
| --- | --- |
| **Parameters** | **Description** |
| **n\_estimators** | How many trees required |
| **max\_features** | How many features required |
| **max\_depth** | How much depth required at the maximum |
| **min\_samples\_split** | How many samples required to split an internal node at the minimum |
| **min\_samples\_leaf** | How many samples required to be at a leaf node at the minimum |

### Result

*Figures 28. Result*



### Grid Search

I tried to execute Grid Search to find out the best parameters for more than 30 minutes. However, it did not finish running even using GPU. The codes are shown in the following:

*Figures 29. Grid Search*

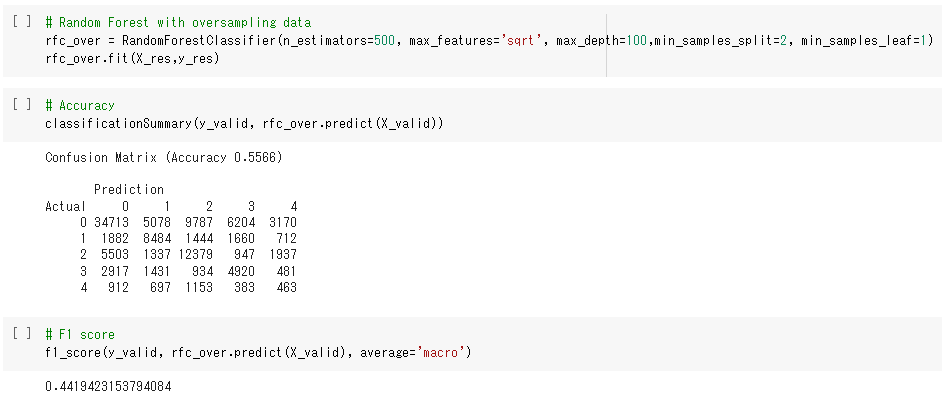


## 14.2. Execution of Random Forest with Oversampling

Random Forest with Oversampling was performed next to try to obtain better accuracy and F1 score.

### Result

*Figures 30. Execution of Random Forest with Oversampling*

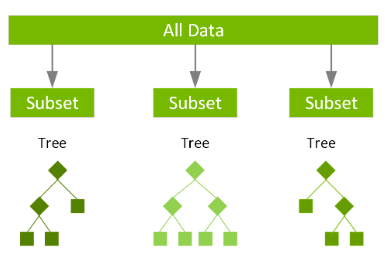


Both accuracy and F1 score got worse.

## 15.0. Model Technique #2: XGboost

XGboost is also one of the ensemble methods (XGBoost, n.d.). As with Random Forest, XGboost generates multiple decision trees, but how to build and combine trees differs. It used gradient boosting which creates weak models in order compensating for weaknesses. Subsequently, it yields the best judge as an output. Underfitting can be minimized by boosting.

*Figures 31.* XGboost *(XGBoost, n.d.)*



## 15.1. Execution of XGboost

*Figures 32. Execution of XGboost*



### Important hyper parameters

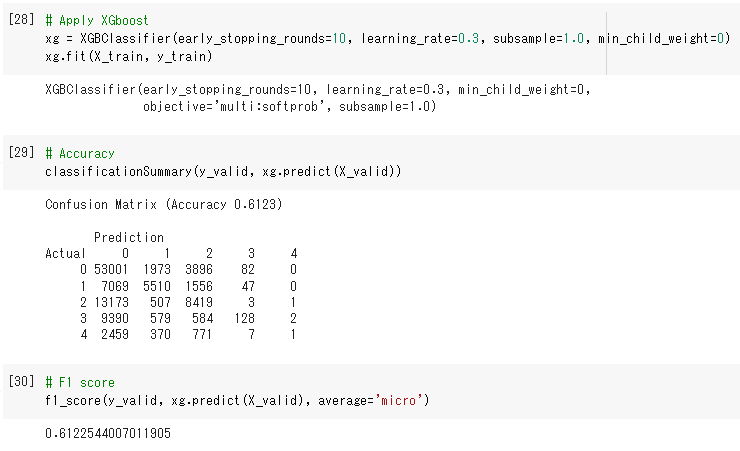
The parameters in the following table are more important (Coelho, 2020)

*Table 5. Important hyper patameters for XGboost (Coelho, 2020; How to avoid model overfitting with early stopping rounds, n.d.; Brownlee, 2016)*

|  |  |
| --- | --- |
| **Parameters** | **Description** |
| **early\_stopping\_rounds** | When to stop running to avoid overfitting |
| **learning\_rate** | Control of weighting trees |
| **subsample** | Proportion of samples extracted in trees |
| **min\_child\_weight** | The minimum weight of a leaf |

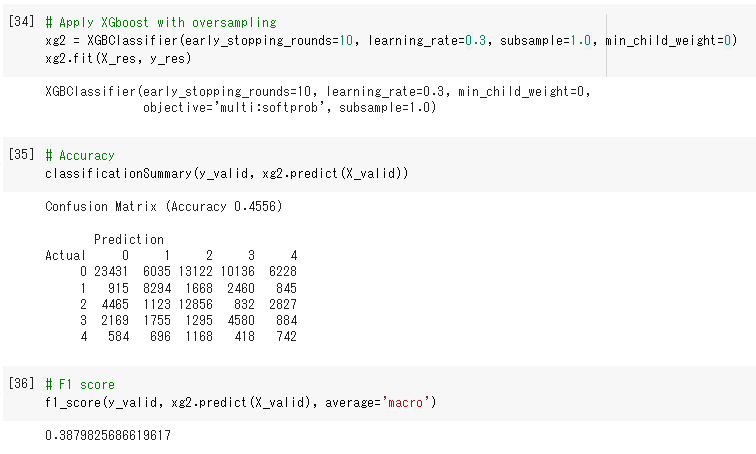
### Result

*Figures 33. Result*



## 15.2. Execution of XGboost with Oversampling

*Figures 34. Execution of XGboost with Oversampling*

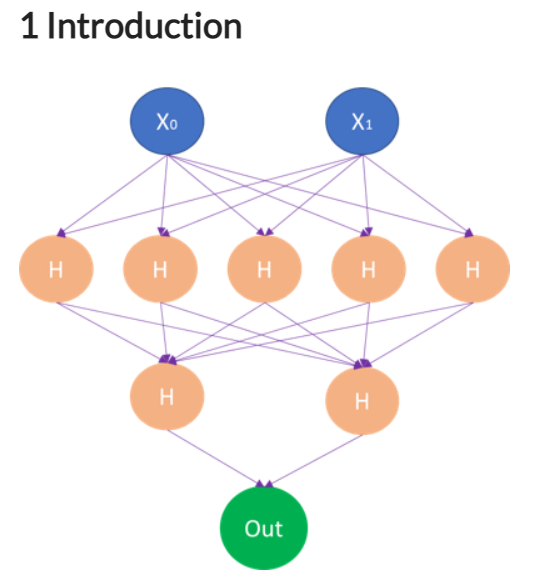


It got worse with Oversampling data.

## 16.0. Model Technique #3: Multilayer Perceptron

MLP is one of newral networks used for classification (Fuchs, 2021). It consists of layers fully connected to the following layer, and neurons are the nodes except for the input layer. Also, there may be some hidden layers between input and output layers. It uses backpropogation to train the network.

*Figures 35. Multilayer Perceptron (Fuchs, 2021)*



## 16.1. Execution of Multilayer Perceptron

*Figures 36. Execution of Multilayer Perceptron*



### Important hyper parameters

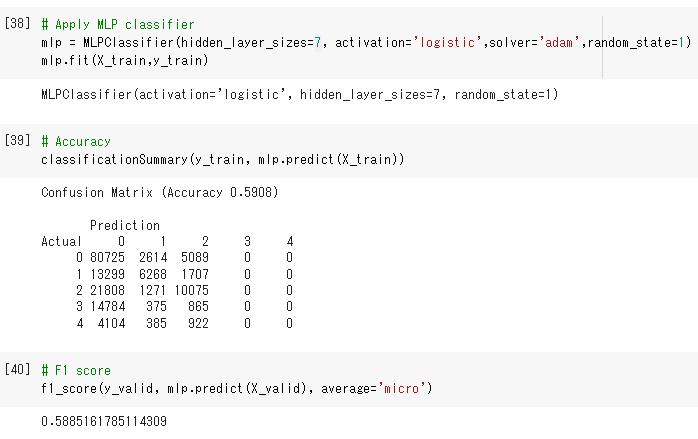
More important parameters are the following:

*Table 6. Important hyper patameters for MLP (Fuchs, 2021)*

|  |  |
| --- | --- |
| **Parameters** | **Description** |
| **hidden\_layer\_sizes** | The number of hidden layers |
| **activation** | How to activate each neuron |
| **solver** | How to optimize learning |

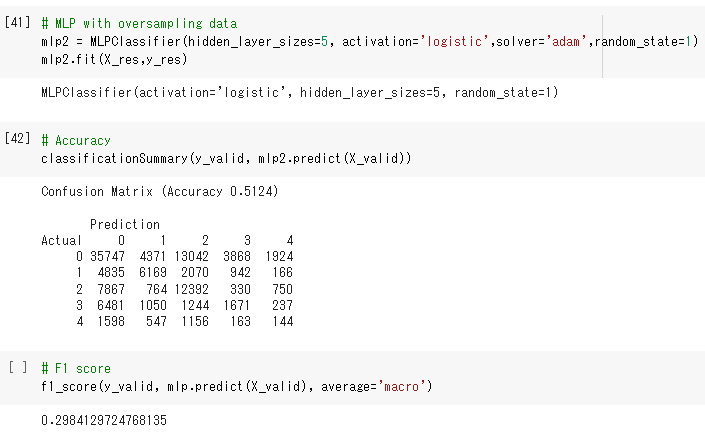
### Result

*Figures 37. Result*



## 16.2. Execution of Multilayer Perceptron with Oversampling

*Figures 38. Execution of Multilayer Perceptron with Oversampling*



It got worse with Oversampling.

## 17.0. Model Comparison

Using Accuracy and F1 score, Random Forest was the best.

*Table 7. Model Comparison*

|  |  |  |
| --- | --- | --- |
| **Models** | **Accuracy** | **F1 score** |
| **Ramdom Forest** | **0.6488** | **0.6488** |
| Ramdom Forest with Oversampling | 0.5566 | 0.4419 |
| XGboost | 0.6123 | 0.6123 |
| Xgboost with Oversampling | 0.4556 | 0.3879 |
| MLP | 0.5908 | 0.5885 |
| MLP with Oversampling | 0.5124 | 0.2984 |

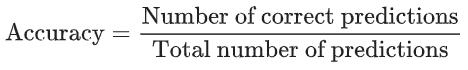
# Model Recommendation

## 18.0 Model Selection

Among the six models that I created for crime type prediction, Random Forest was the best. It was selected based on Accuracy and F1 score since the models were for classification. Accuracy can determine the best model, and F1 score can measure the accuracy (Classification: Accuracy, n.d.; Wood, n.d.). Both formulas are the following:

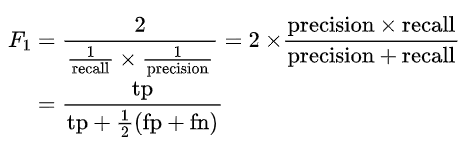
### Accuracy

*Figures 39. Accuracy(Classification: Accuracy, n.d.)*



### F1 score

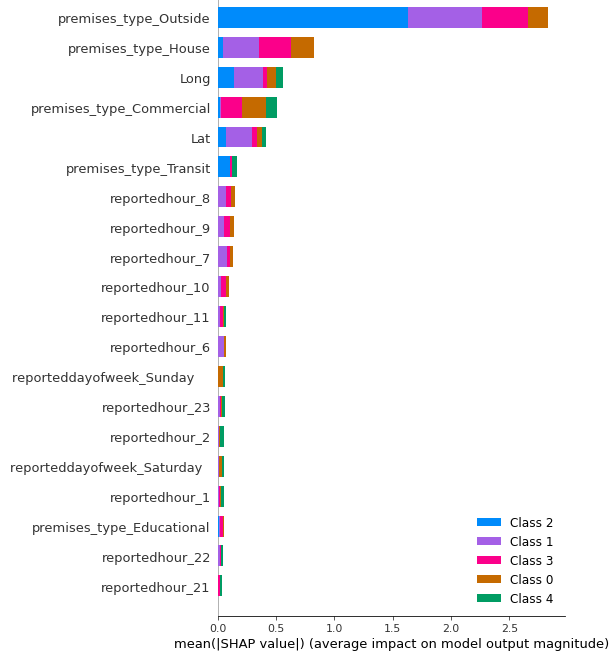
*Figures 40. F1 score (Wood, n.d.)*



## 19.0 Model Theory

This Random Forest model was built with 500 decision trees, max\_features of 8, max depth of 100, min\_sample\_splt of 2, and min\_sample\_leaf of 1. The important features are the following:

*Figures 41. SHAP*



## 19.1 Model Assumptions and Limitations

This model’s accuracy is not high because there are five classes in the target variable, and data size is relatively small. Therefore, officers could refer the result of this model, but they have keep in mind that the results are not always accurate. However, this model could still be useful to optimize the response because it could offer the possible crime types.

## 20.0 Model Sensitivity to Key Drivers

This model was trained based on the current dataset. Hence, when there are drifts in the future dataset, the same accuracy of this model cannot be guaranteed. The ditails for drifts are provided in the next section.

# Validation and Governance

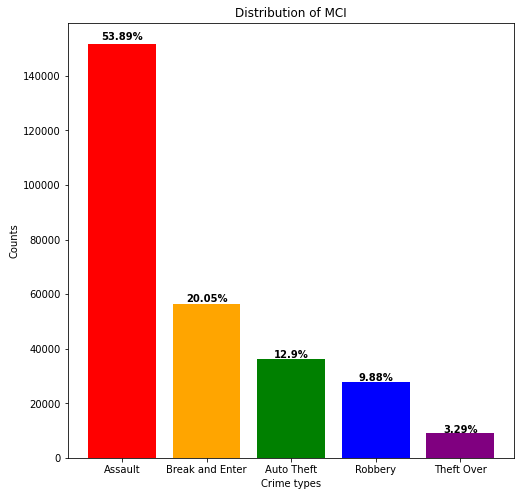
## 21.0. Categorical Variable Level Monitoring

The following data is the accepted ranges of the inputs. These ranges are correspond to the historical dataset.

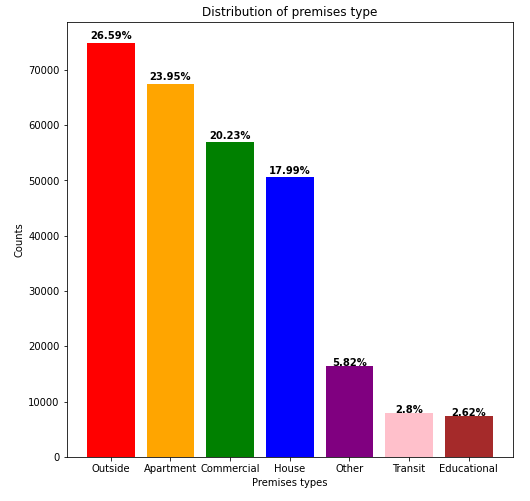
### Acceptable ranges for Categorical Variables

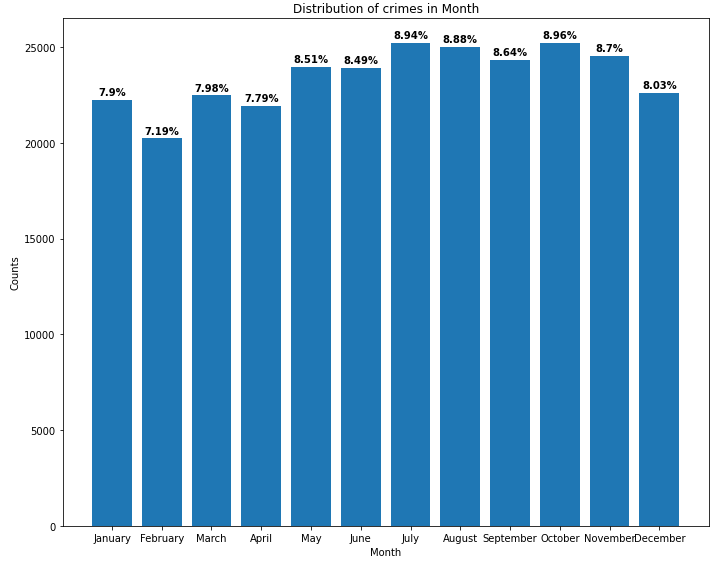
The distributions of proportions are used to show the acceptable ranges for the categorical variables.

*Figures 42. Acceptable ranges for Categorical Variables*

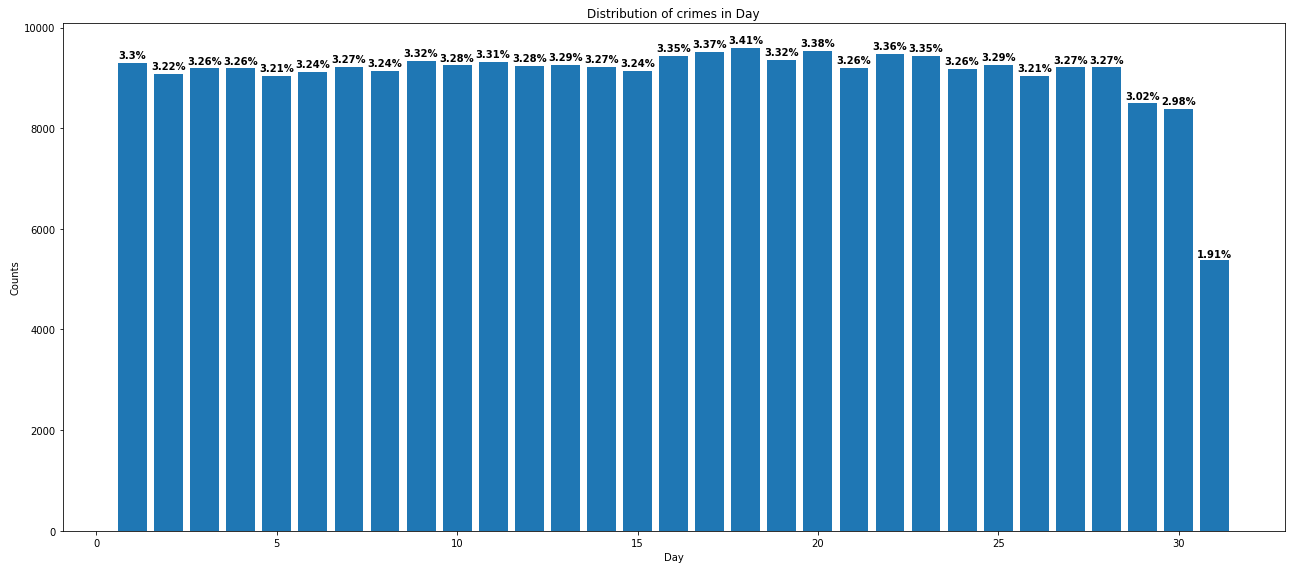


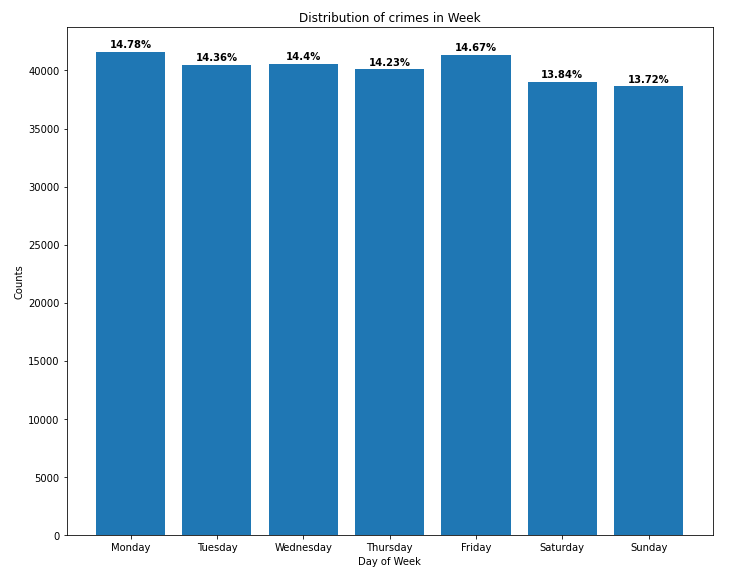
*Figures 42. Acceptable ranges for Categorical Variables*



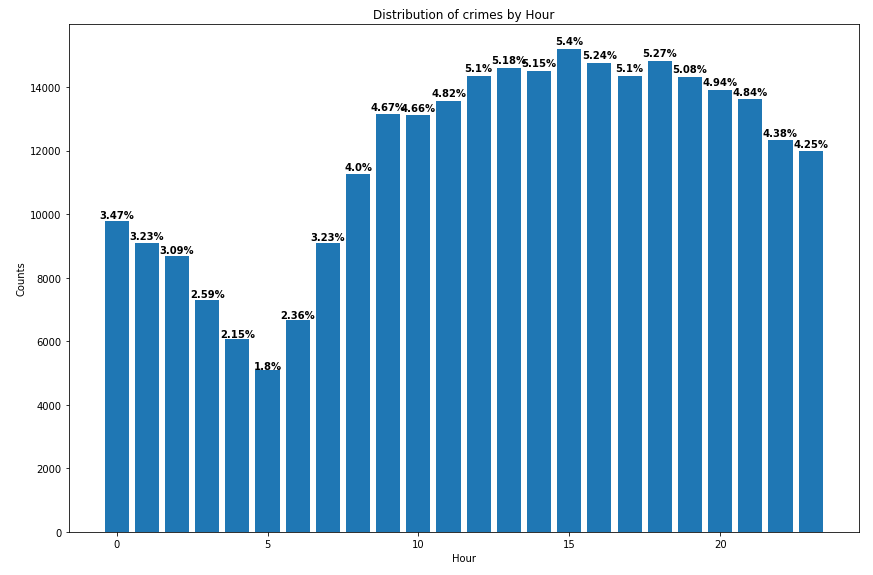


*Figures 42. Acceptable ranges for Categorical Variables*





*Figures 42. Acceptable ranges for Categorical Variables*



### Acceptable ranges for Numerical Variables

For the numerical variables, statistics are used to show the average ranges.

*Table 8. Acceptable ranges for Numerical Values*

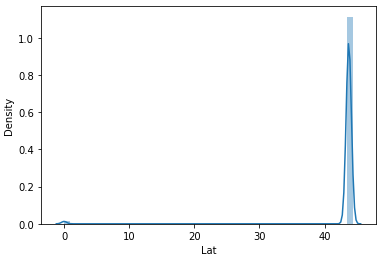
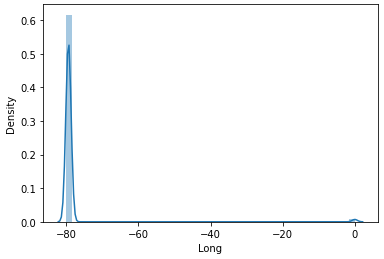


## 21.1. Caps & Floors

Outliers were detected in Long, Lat, reportedday, reportedhour, and premises\_type. Since every points are very important in reportedday, reportedhour, and premises\_type, outliers in these variables were retained. On the other hand, outliers in Long and Lat were removed because they were clearly abnormal. The distributions of Long and Lat before and after deleting the outliers are the following.

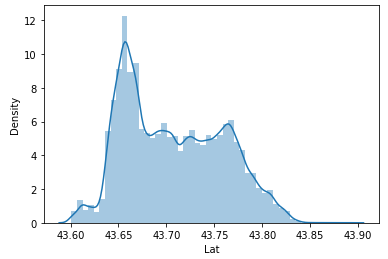
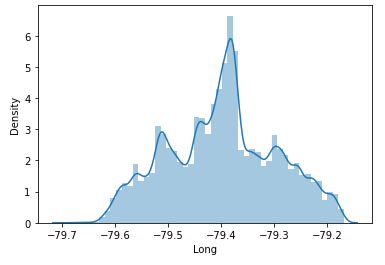
### Before Cap and Floor

*Figures 43. Before Cap and Floor*



### After Cap and Floor

*Figures 44. After Cap and Floor*



## 21.2. Missing Values

The features with missing values were not taken as inputs. Therefore, handling missing value might be needed in the future dataset if necessary. Firstly, if there is a small amount of missing values which is less than 10%, delete them. Secondly, if there is a large amount of missing values in the categorical variables, create a new category for the missing values. Lastly, if there is a large amount of missing values in the numerical variables, impute the values that are difined using k-nearest neighbor algorithm.

## 21.3. Variable Drift Monitoring

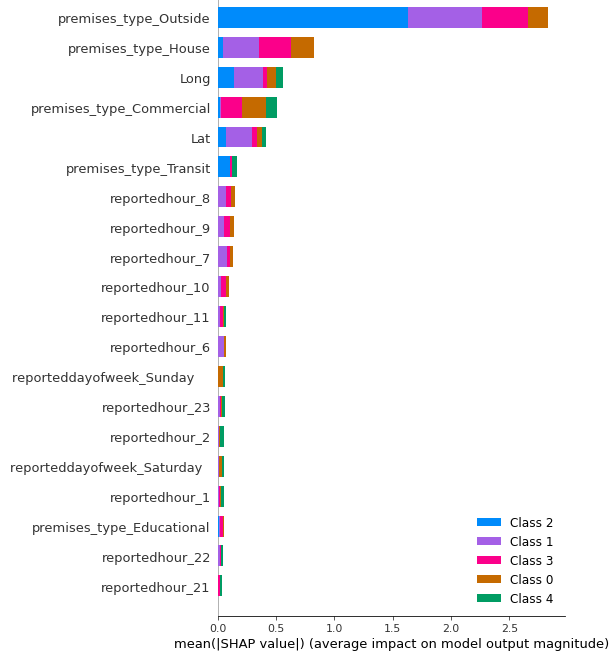
Crimes have changed over time (Changing Times, changing crimes, n.d.). Therefore, the followng drifts could be possible:

*Table 9. Variable Drift Monitoring*



In the Data drift, each impact was determined based on future importance analysis using SHAP. The results are shown in the following.

*Figures 45. SHAP*



## 22.0. Model Health & Stability

The model’s accuracy could shifts over time as well. To check the model health and stability, Accuracy and F1 score will be used because these parameters were used in model comparison.

### Initial Model Fit Statistics: 1st model ‘Random Forest’

*Table 10. Initial Model Fit Statistics*

|  |  |
| --- | --- |
| **Fit statistics** | |
| Accuracy | 0.6488 |
| F1 score | 0.6488 |

### Acceptable ranges:

Since both accuracy and F1 score are not high enough, the acceptable ranges should not be much lower than the initial values.

*Table 11. Acceptable Ranges*

|  |  |
| --- | --- |
| **Acceptable range** | |
| Accuracy | More than 0.63 |
| F1 score | More than 0.63 |

## 23.0. Risk Tiering (e.g., no action, report, refit, rebuild)

*Table 12. Risk Tiering*



# Conclusion and Recommendations

## 24.0. Impacts on Business Problem (Scope of the recommended model)

This model is a crime type prediction model that can classsify assault, auto theft, Break and Enter, robbery, and Theft in Toronto over with the information of premises, hour, day, day of week, month, longitude, and latitude. Using this model. 911 callers and officers could predict the possible crime type that could optimize the response time and preparation, which could improve civilian safety in Toronto.

## 25.0. Recommended Next Steps

Since the accuracy of this model is not high enough, the effort for creating more accurate model are needed. Also, it could classify only five categories, so it would be great to have the model that can predict one from more categories. Furthermore, the crimes have always been changing over time. Therefore, the model creation should adopt any changes in the future.

# References

Brownlee, J. (2016, September 16). Tune Learning Rate for Gradient Boosting with XGBoost in Python. Machine Learning Mastery. <https://machinelearningmastery.com/tune-learning-rate-for-gradient-boosting-with-xgboost-in-python/>

Changing Times, changing crimes. (n.d.). Cololado College History Department. <https://sites.coloradocollege.edu/hip/criminality-and-criminalization/changing-times-changing-crimes/>

Classification: Accuracy. (n.d.) Machine Learning. <https://developers.google.com/machine-learning/crash-course/classification/accuracy>

Coelho, A. (2020, June 17). Narrowing the Search: Which Hyperparameters Really Matter?. dataiku. <https://blog.dataiku.com/narrowing-the-search-which-hyperparameters-really-matter>

Fuchs, M. (2021, February 3). NN - Multi-layer Perceptron Classifier (MLPClassifier). <https://michael-fuchs-python.netlify.app/2021/02/03/nn-multi-layer-perceptron-classifier-mlpclassifier/>

How to avoid model overfitting with early stopping rounds. (n.d.). Practical Data Science. <https://practicaldatascience.co.uk/machine-learning/how-to-avoid-model-overfitting-with-early-stopping-rounds>

IBM Cloud Education. (2020, December 7). What is Random Forest? IBM. <https://www.ibm.com/cloud/learn/random-forest>

Koehrsen, W. (2018, January 9). Hyperparameter Tuning the Random Forest in Python. Medium. <https://towardsdatascience.com/hyperparameter-tuning-the-random-forest-in-python-using-scikit-learn-28d2aa77dd74>

Major Crime Indicator. (2022, April 4). Public Safety Data Portal. <https://data.torontopolice.on.ca/search?q=crime>

Major Crime Indicators (MCI) . PUBLIC SAFETY DATA PORTAL: OPEN DATA DOCUMENTATION. (2022, August 9). Retrieved August 19, 2022, from <https://torontops.maps.arcgis.com/apps/dashboards/6d4cace4645c4b759619bc82d306ad0a>

McKenzie, A. (2019, April 4). Calling 911: Improving safety and response times for emergency workers. Fleet Complete Blog. <https://blog.fleetcomplete.com/improving-safety-response-times-for-emergency-workers#:~:text=Complete%2C%20can%20help.-,Urgency,literally%20putting%20out%20a%20fire>.

Toronto Police Service – 2022 Operating Budget Request. (n.d.). Toronto.ca. <https://www.toronto.ca/legdocs/mmis/2022/ex/bgrd/backgroundfile-175060.pdf>

Walczak, S. (2021, October 7). Predicting crime and other uses of neural networks in police decision making. Frontiers. <https://www.frontiersin.org/articles/10.3389/fpsyg.2021.587943/full#ref20>

Wood, T. (n.d.). F-Score. DeepAI. <https://deepai.org/machine-learning-glossary-and-terms/f-score>

XGBoost. (n.d.). NVIDIA. <https://www.frontiersin.org/articles/10.3389/fpsyg.2021.587943/full#ref20>