DATA ANALYTICS

PROJECT PRESENTATION:
CRIME DATA ANALYTICS

DIVYA K: PES1UG20CS135
DHANALAKSHMI K M : PES1UG20CS124
SYED AZFAR RAYAN : PES1UG20CS453

ABSTRACT AND SCOPE

- A crime is any unlawful action punishable by a state, and often creates a huge social impact.
-Our project aims on using data analytic tools for crime prediction and uncovering trends and patterns in crimes.
-By identifying crime hotspots and early warnings, police services can be increased in those areas to effectively prevent occurrences of further crimes.
-Various machine learning techniques will be used for trend identification, prediction, and visualization.

Our project aims to address questions such as

-Which are the major crime indicator categories?
-What types of crimes are most frequently committed and what are the trends in the crimes?
-Which hours of the day do these crimes occur and is there a pattern?
-Which months and which parts of the month are crimes likely to occur?
-Which are the crime hotspots in the city and the safest neighbourhoods in the city?
-Which days of the week are crimes most prevalent?
-Are there different trends observed for different categories of crime?

Our proposed approach is the supervised prediction technique of classification for building a predictive model that can predict the category of crimes. Algorithms such as Decision tree, KNN Classifier, Naïve Bayes, and Random Forest will be tested in order to identify the best performing model for crime prediction. We also propose a K-Means clustering model to outline the police districts according to crimes.

DATASET

- Toronto Crime dataset on major crime indicators of the year 2021 has been used .
- This dataset includes all Major Crime Indicators (MCI) occurrences by reported date and related offences.
- The MCI categories include Assault, Break and Enter, Auto Theft, Robbery and Theft Over. It is published on the Toronto police public safety data portal.

Intially there are 27 attributes in the dataset

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 34277 entries, 0 to 34276
Data columns (total 27 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   Index_              34277 non-null  int64
 1   event_unique_id     34277 non-null  object
 2   Division            34277 non-null  object
 3   occurrencedate      34277 non-null  object
 4   reporteddate        34277 non-null  object
 5   location_type       34277 non-null  object
 6   premises_type       34277 non-null  object
 7   ucr_code            34277 non-null  int64
 8   ucr_ext             34277 non-null  int64
 9   offence             34277 non-null  object
 10  reportedyear        34277 non-null  int64
```

```
 10  reportedyear        34277 non-null  int64
 11  reportedmonth       34277 non-null  object
 12  reportedday         34277 non-null  int64
 13  reporteddayofyear   34277 non-null  int64
 14  reporteddayofweek   34277 non-null  object
 15  reportedhour        34277 non-null  int64
 16  occurrenceyear      34277 non-null  int64
 17  occurrencemonth     34277 non-null  object
 18  occurrenceday       34277 non-null  int64
 19  occurrencedayofyear 34277 non-null  int64
 20  occurrencedayofweek 34277 non-null  object
```

```
 20  occurrencedayofweek 34277 non-null  object
 21  occurrencehour      34277 non-null  int64
 22  mci_category        34277 non-null  object
 23  Hood_ID             34277 non-null  object
 24  Neighbourhood       34277 non-null  object
 25  Longitude           34277 non-null  float64
 26  Latitude            34277 non-null  float64
dtypes: float64(2), int64(11), object(14)
memory usage: 7.1+ MB
```

In [4]:

```
data.isnull().sum()
```

Out[4]:

```
Index_                  0
event_unique_id         0
Division                0
occurrencedate          0
reporteddate            0
location_type           0
premises_type           0
ucr_code                0
ucr_ext                 0
offence                 0
reportedyear            0
reportedmonth           0
reportedday             0
reporteddayofyear       0
reporteddayofweek       0
reportedhour            0
occurrenceyear          0
occurrencemonth         0
occurrenceday           0
occurrencedayofyear     0
occurrencedayofweek     0
occurrencehour          0
mci_category            0
Hood_ID                 0
Neighbourhood           0
Longitude               0
Latitude                0
dtype: int64
```

There are no empty/missing fields in the data .

Irrelevant columns such as X,Y,Object_ID have been removed as they do not provide any information.

Dates have been converted to datetime and extra spaces have been stripped from the data.

In [5]:

```
data["event_unique_id"].value_counts()
```

Out[5]:

```
GO-20211545519    10
GO-2021967516     10
GO-2021684391      9
GO-20211470920     8
GO-20211176139     7
                  ..
GO-20211271443     1
GO-20211137072     1
GO-2021970958      1
GO-2021968615      1
GO-2022410748      1
Name: event_unique_id, Length: 30024, dtype: int64
```

In [6]:

```
data.event_unique_id.duplicated().sum()
```

Out[6]:

```
4253
```

-We notice that there are duplicate entries for certain event IDs

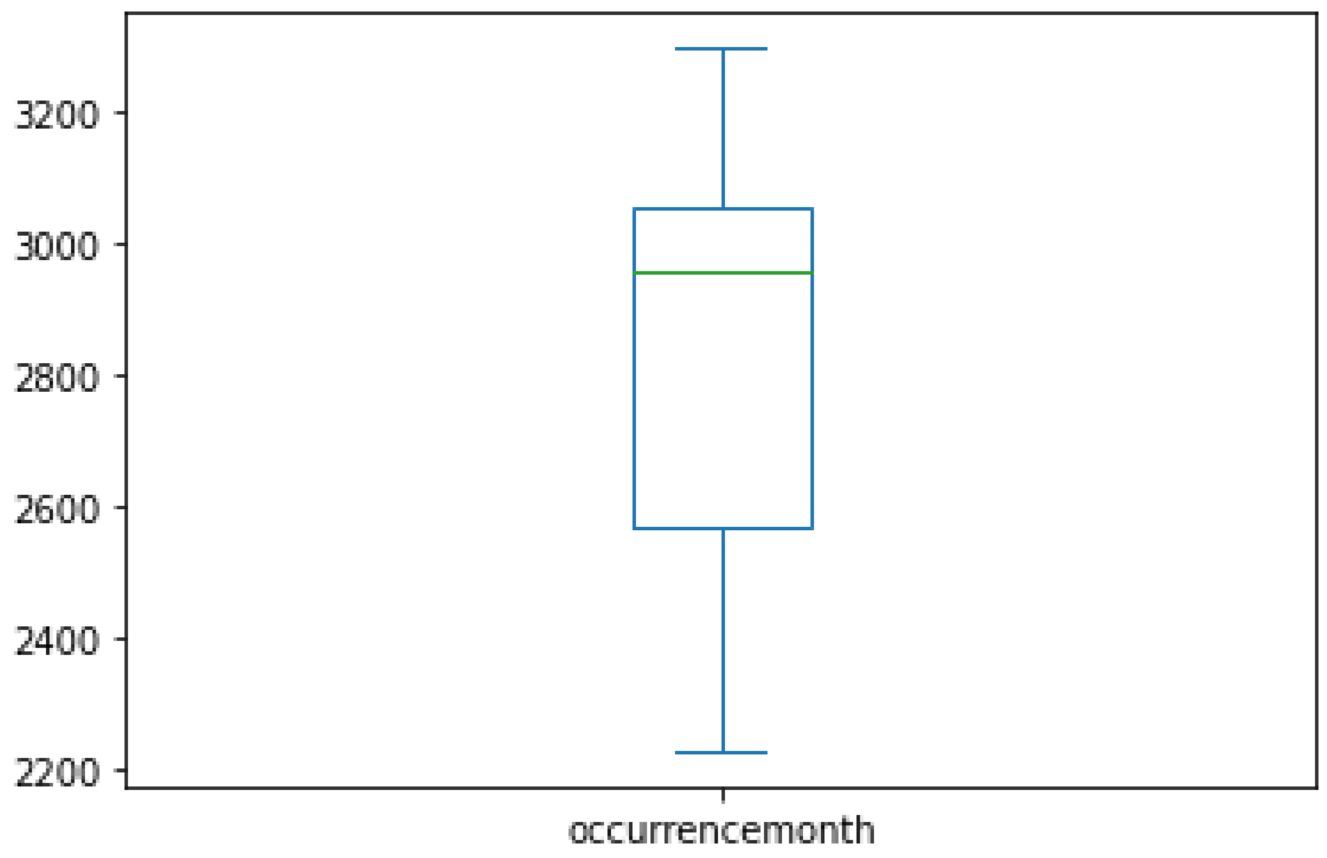
-We come to know that there are 3348 event_unique_ids which have duplicates.

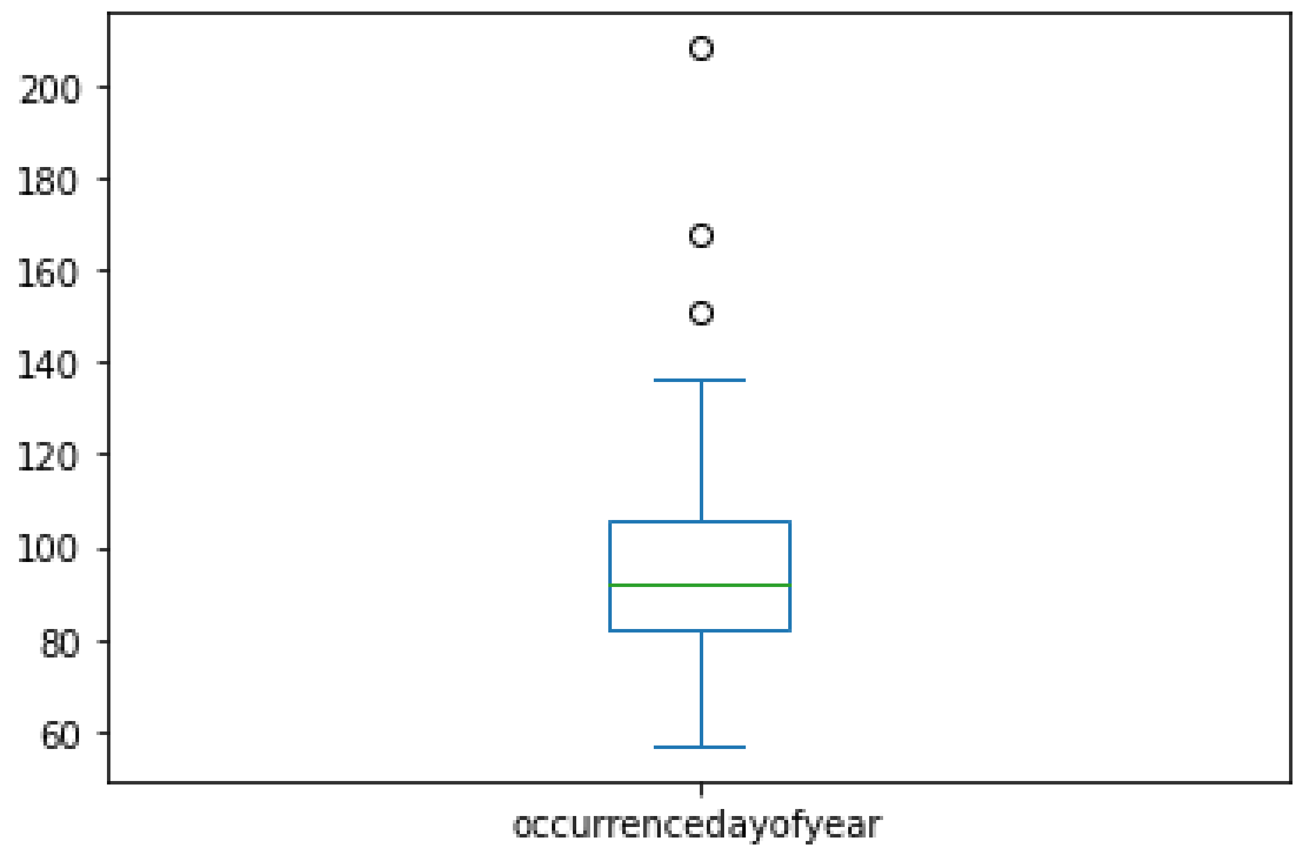
-This is because:

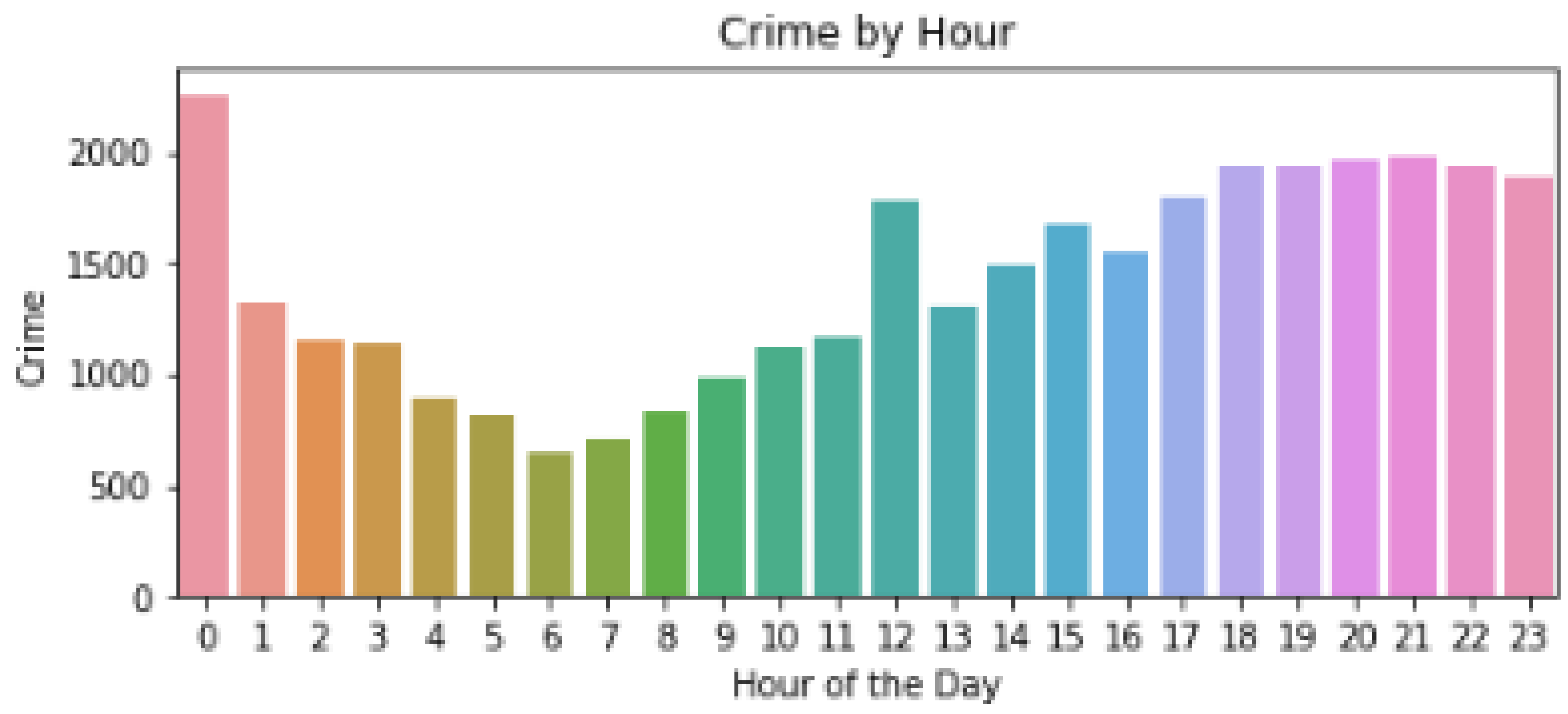
This data is provided at the offence and/or victim level, therefore one occurrence number may have several records associated to the various MCIs used to categorize the occurrence.

In Toronto, the average number of crimes per month is 2856

In Toronto, the average number of crimes per day is 93

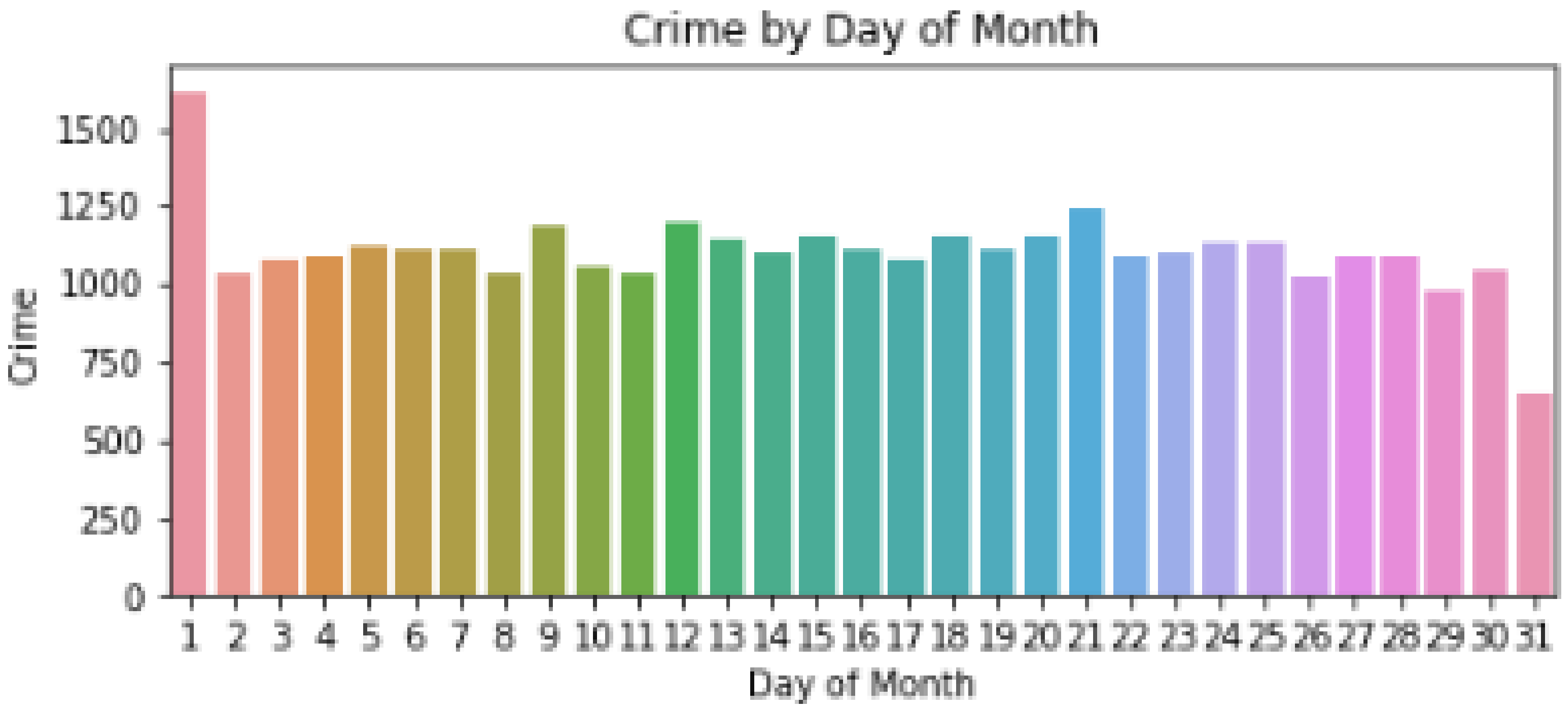


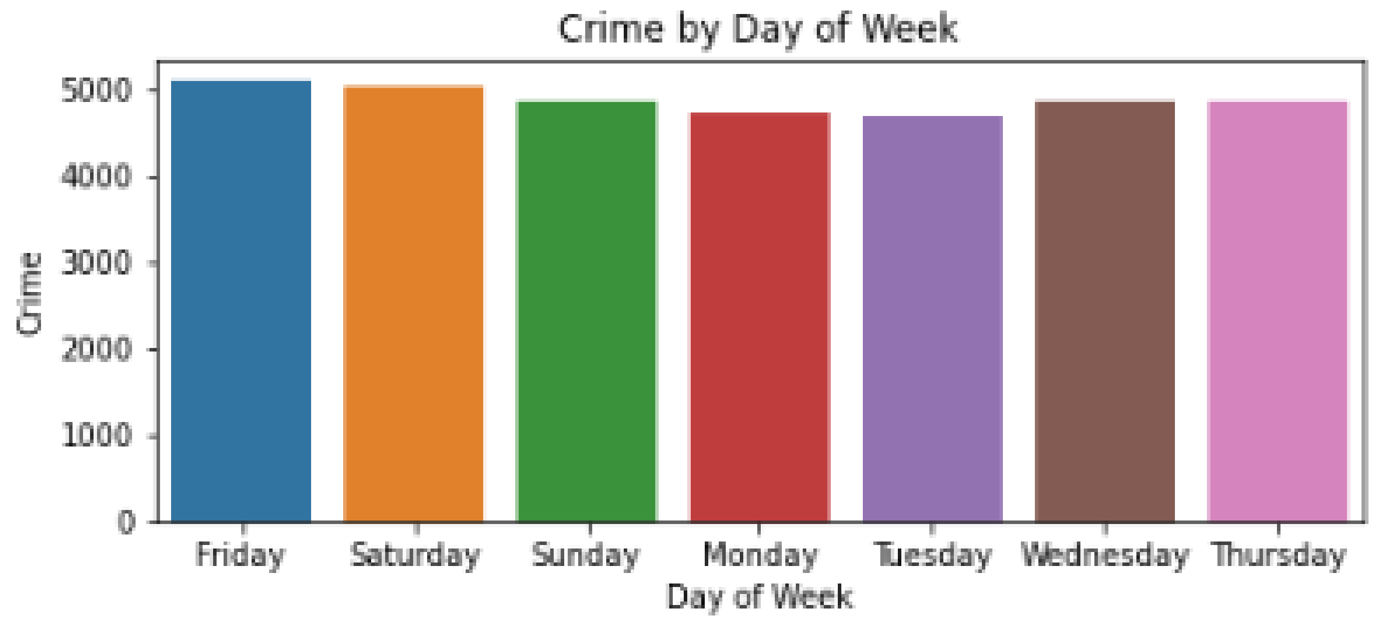




Monday and Tuesday see lower crime rates, while Friday and weekends see higher crime rates.

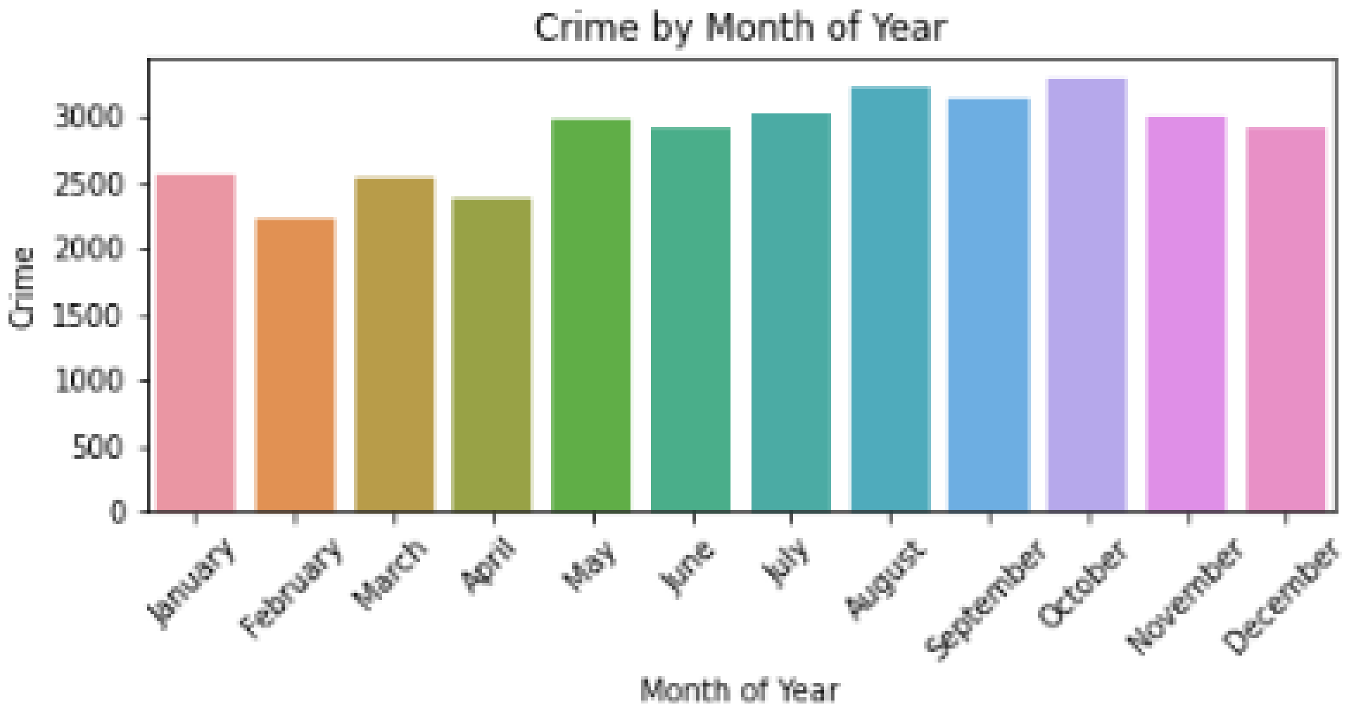
First day of the month sees a peak in the crime rates.

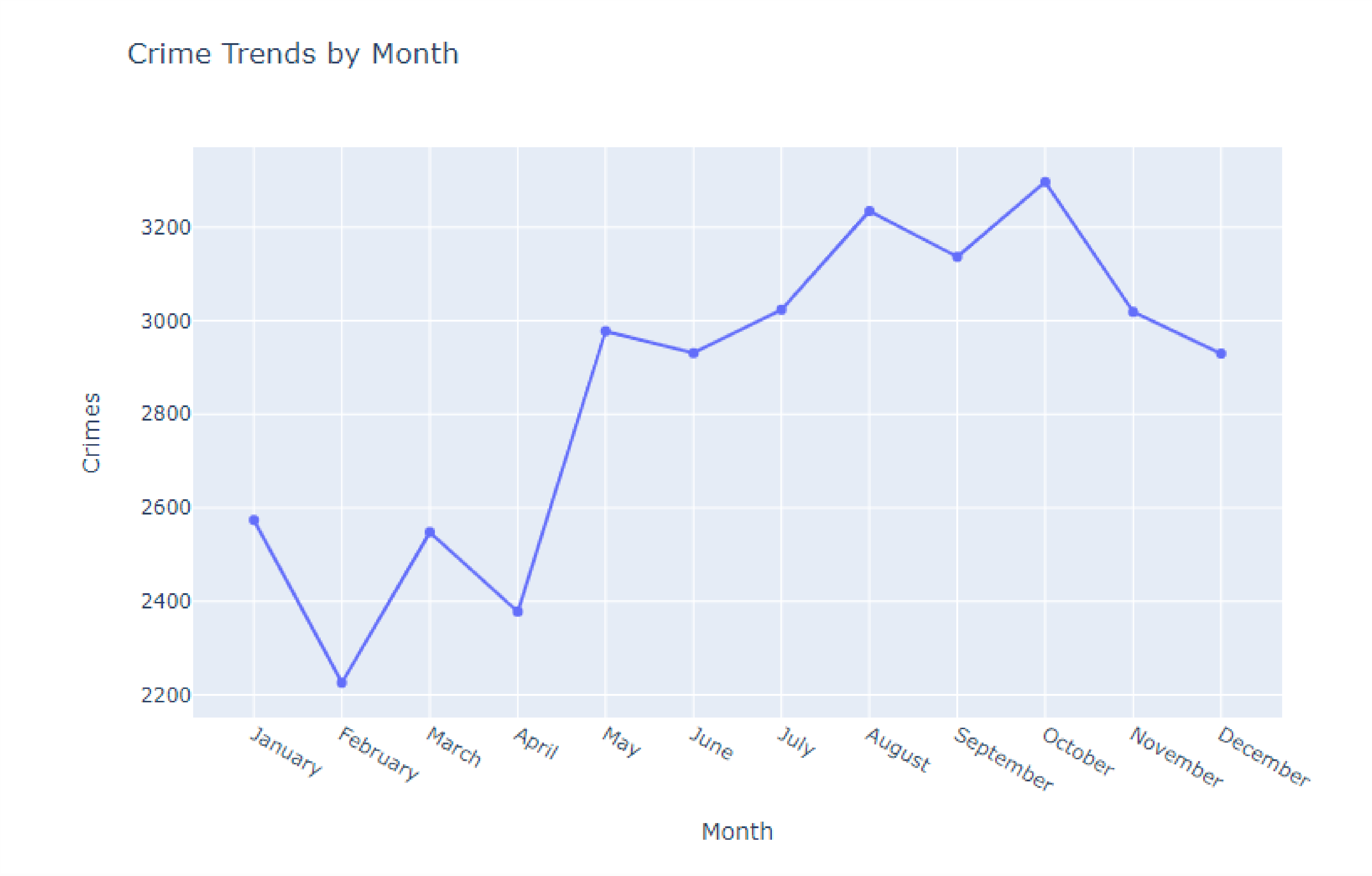




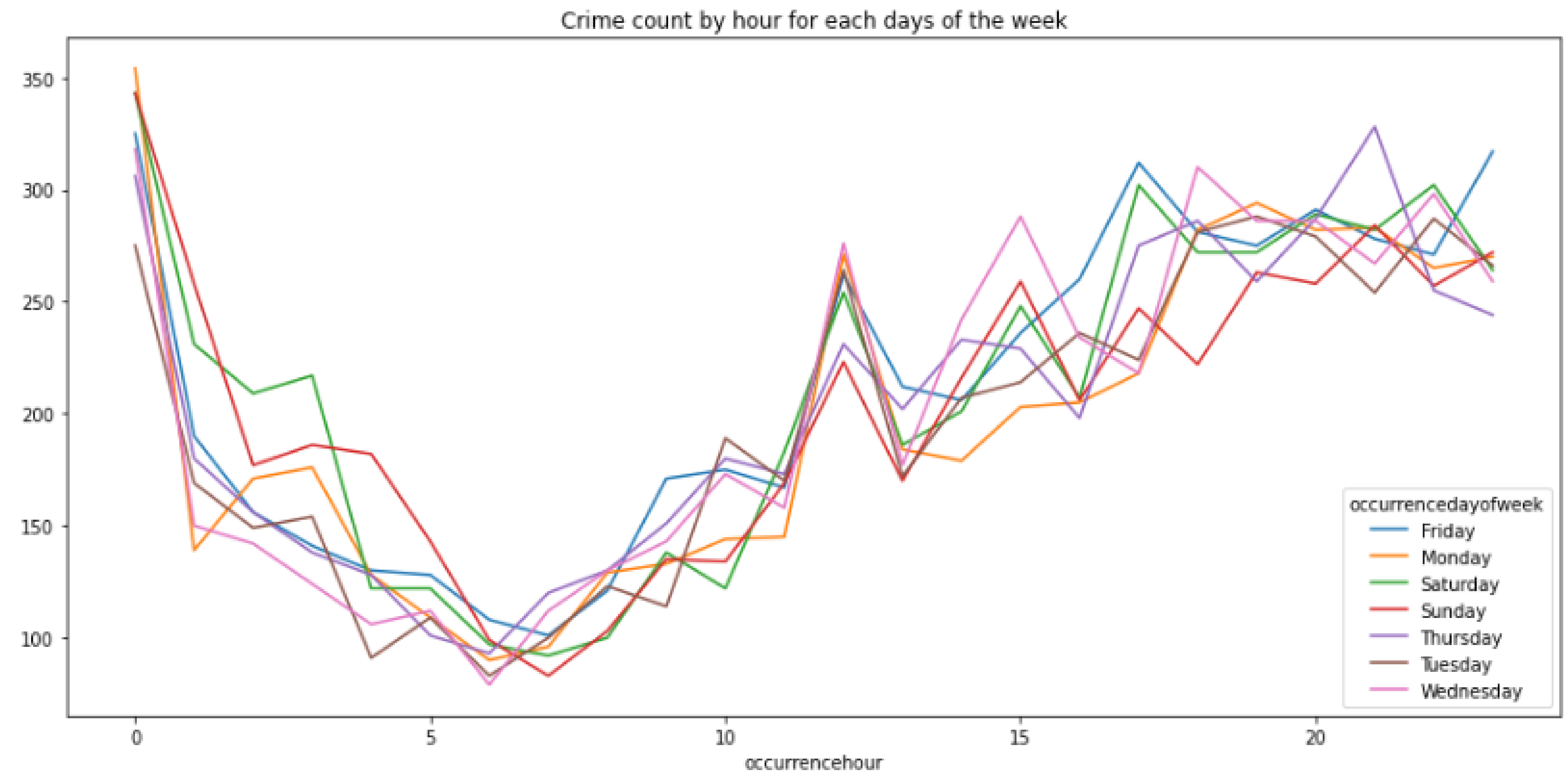
Monday and Tuesday see lower crime rates, while Friday and weekends see higher crime rates.

July-October, being the season of summer and fall in Toronto sees the highest crime rate.

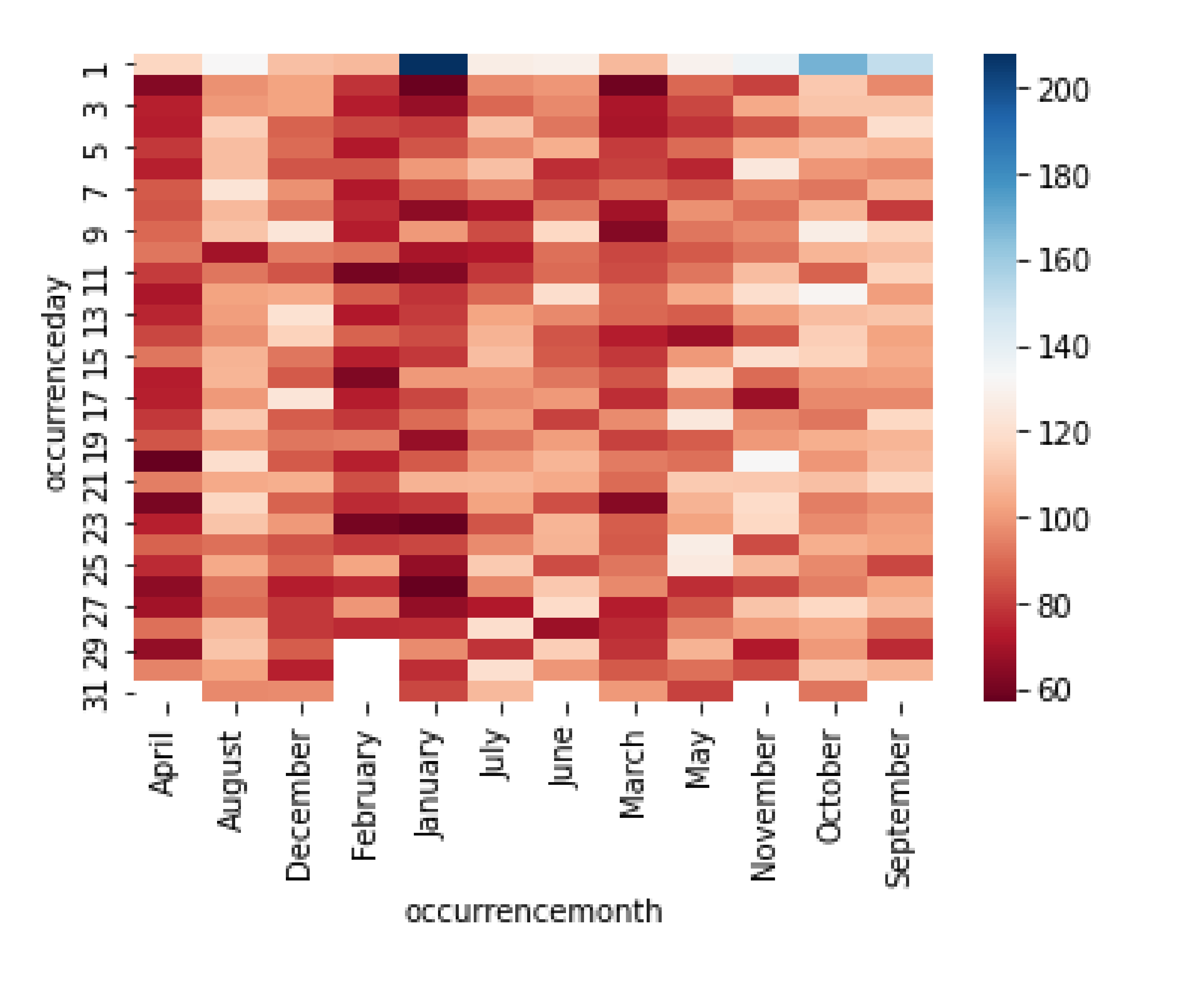




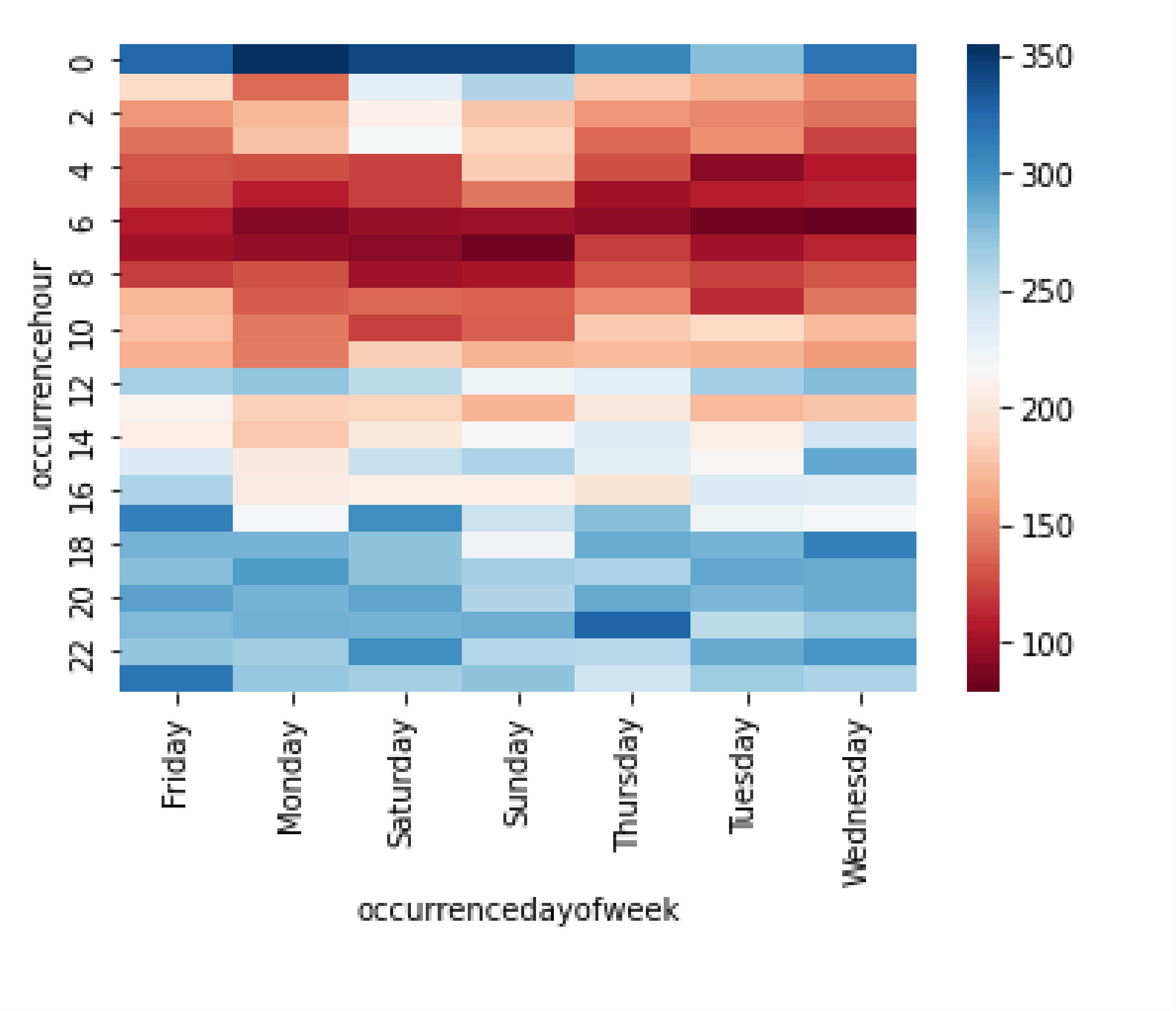
There is a exponential increase in from april to november in crimes.



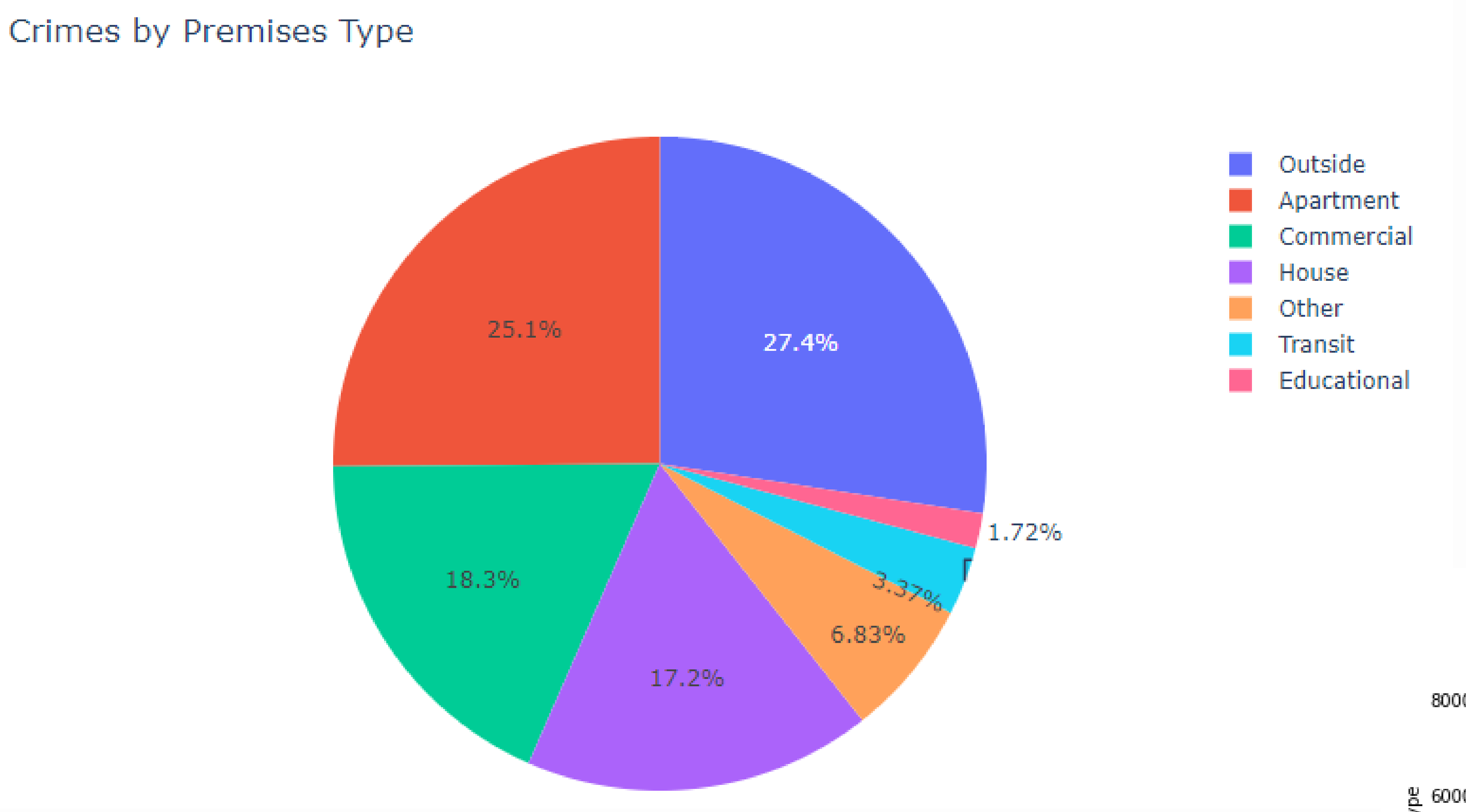
Crime rates show a peak at noon and increase through the evening and night, seeing a maximum at midnight hours

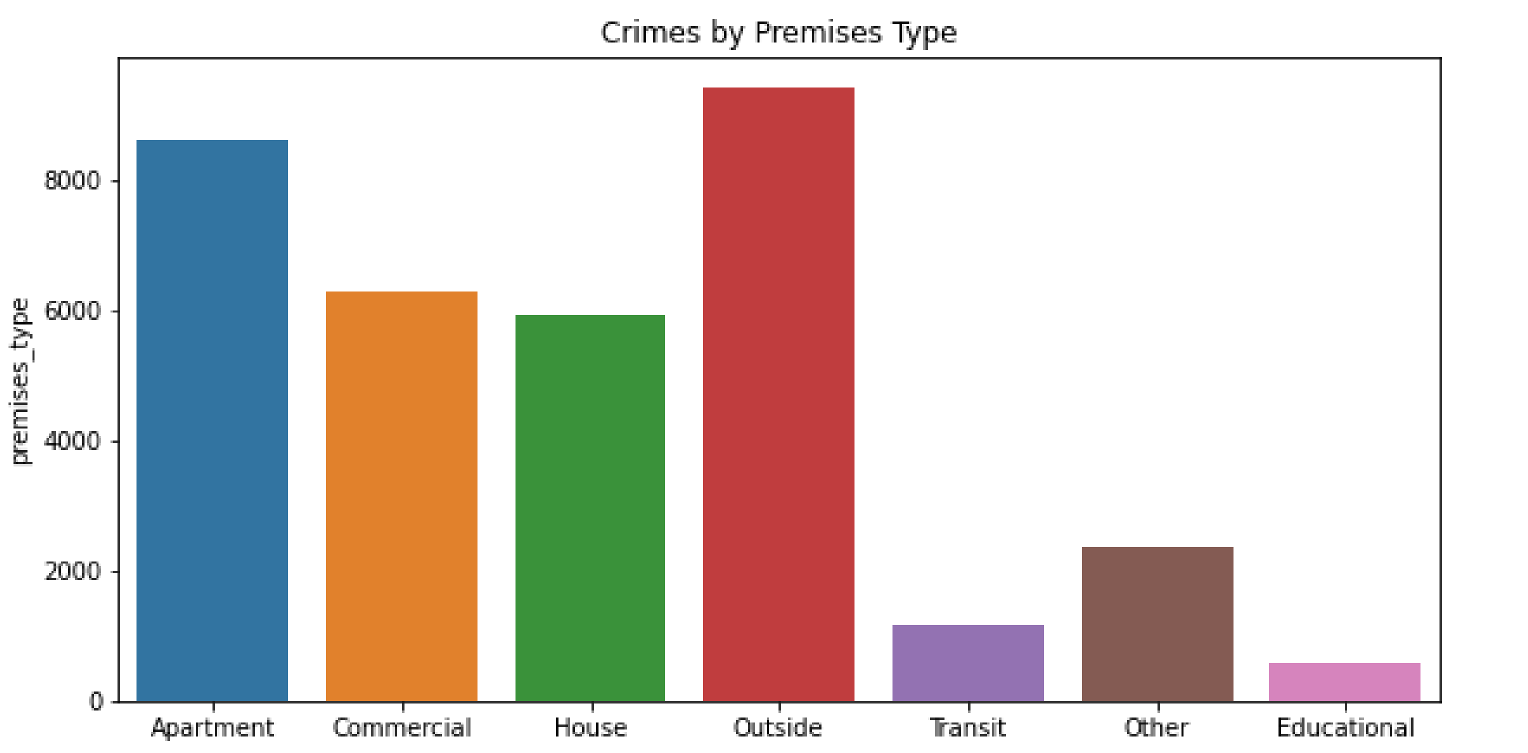


First few days of a month see higher crime rates, especially in the months of Aug-Oct

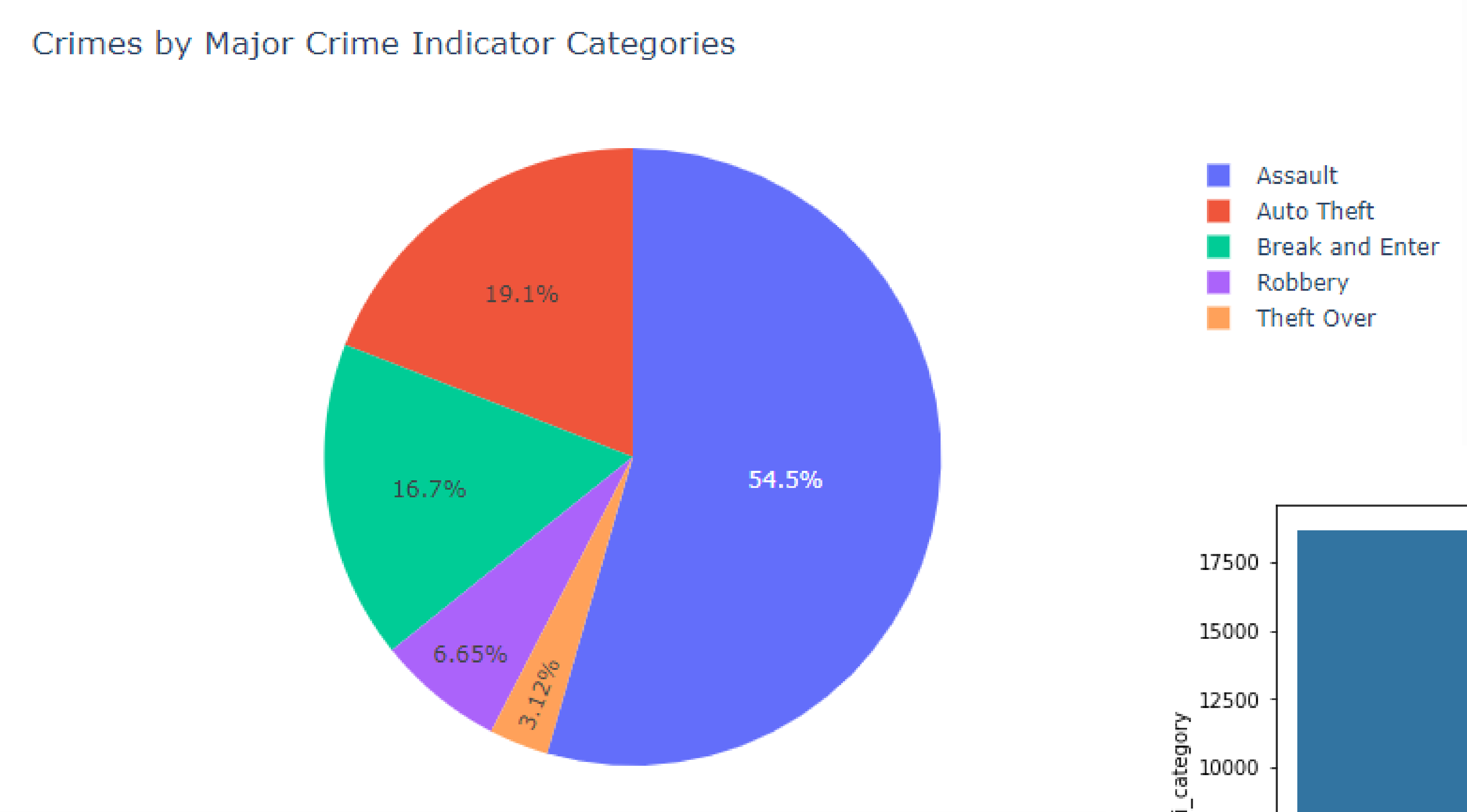


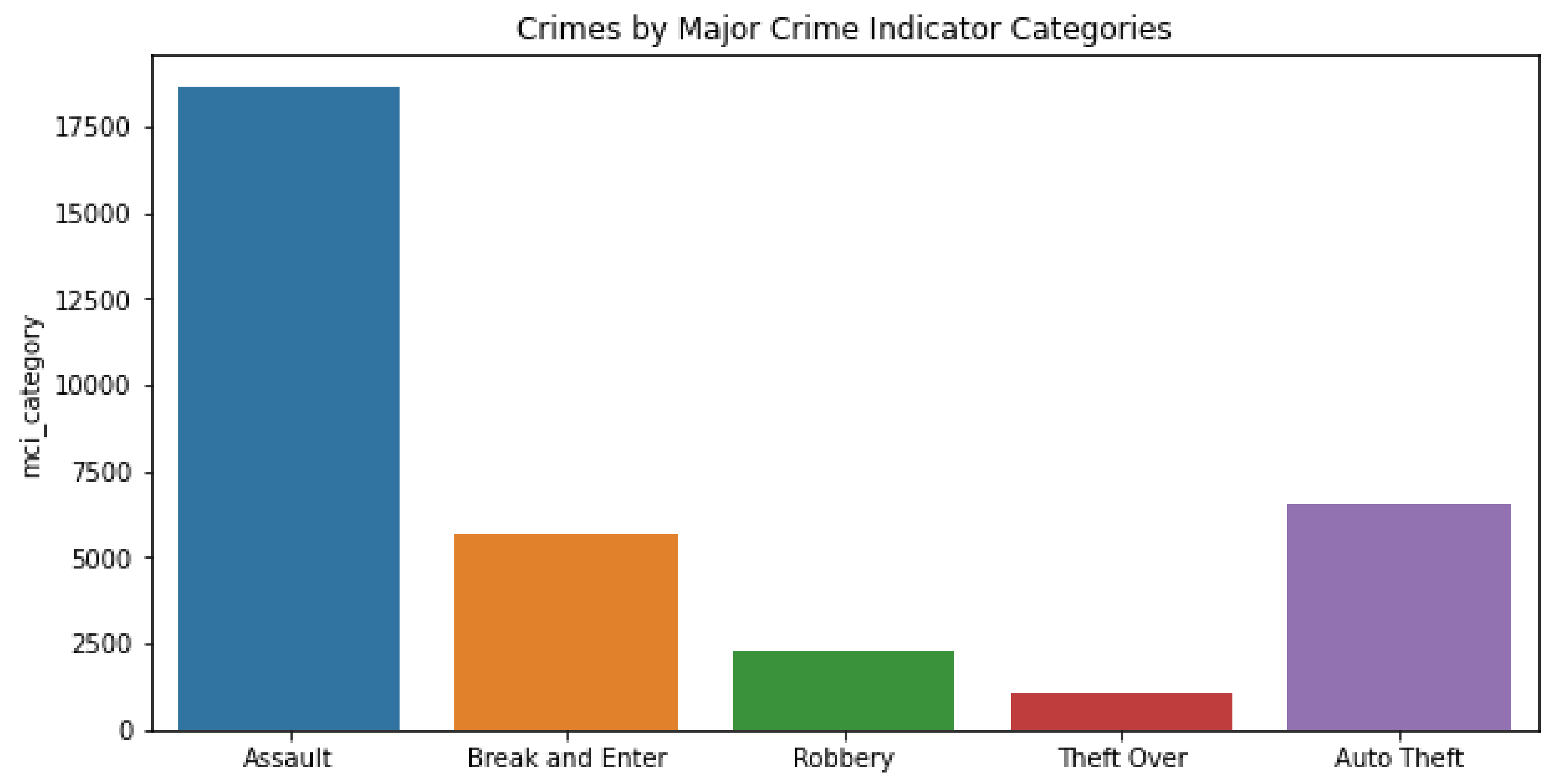
Maximum crime is at midnight, seeing a growing trend from 4pm in the evening, with a peak at 12 noon





Most crimes happen outside followed by apartments and commercial establishments

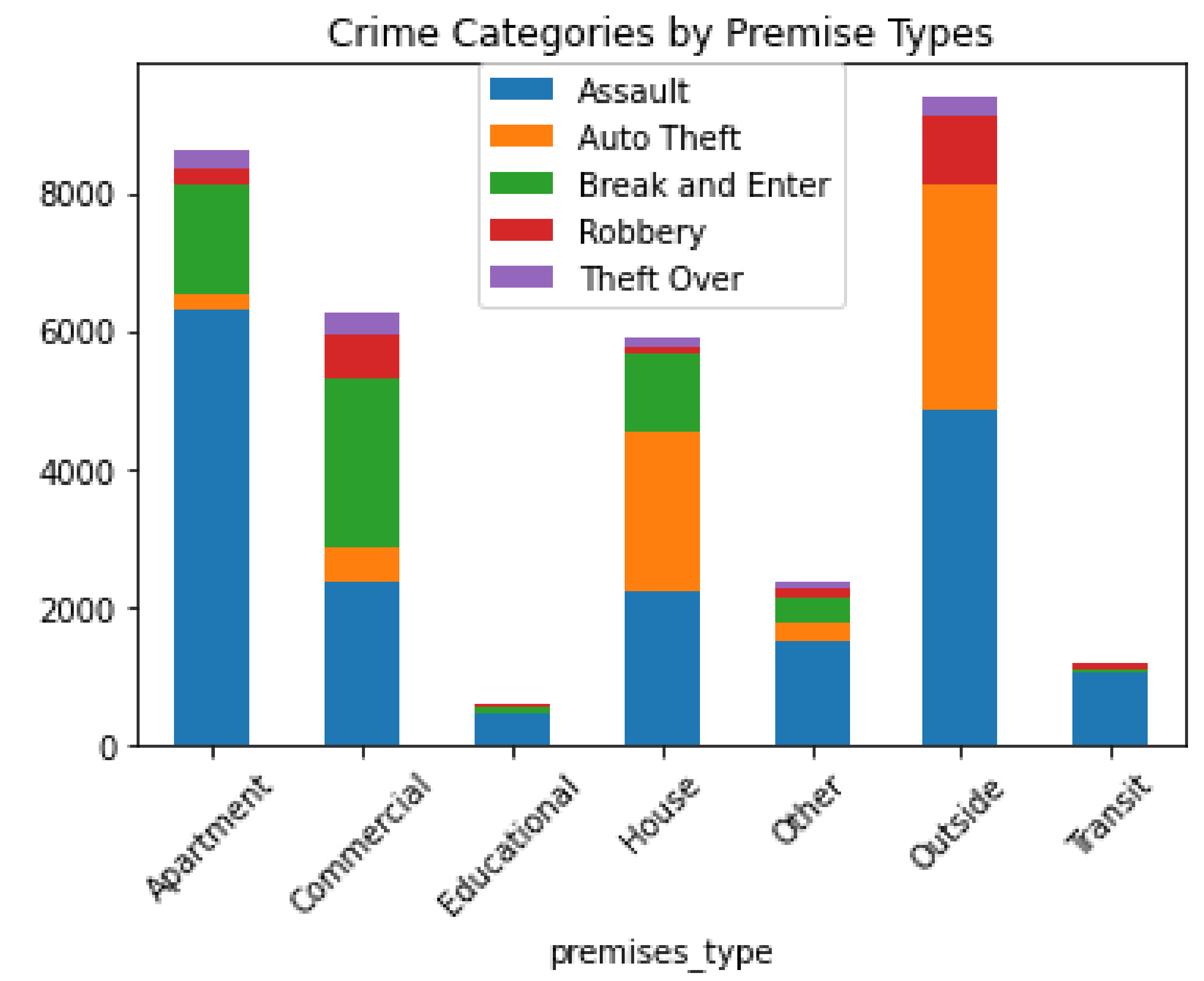




Assault is the most prevelant crime followed by auto theft and break and enter

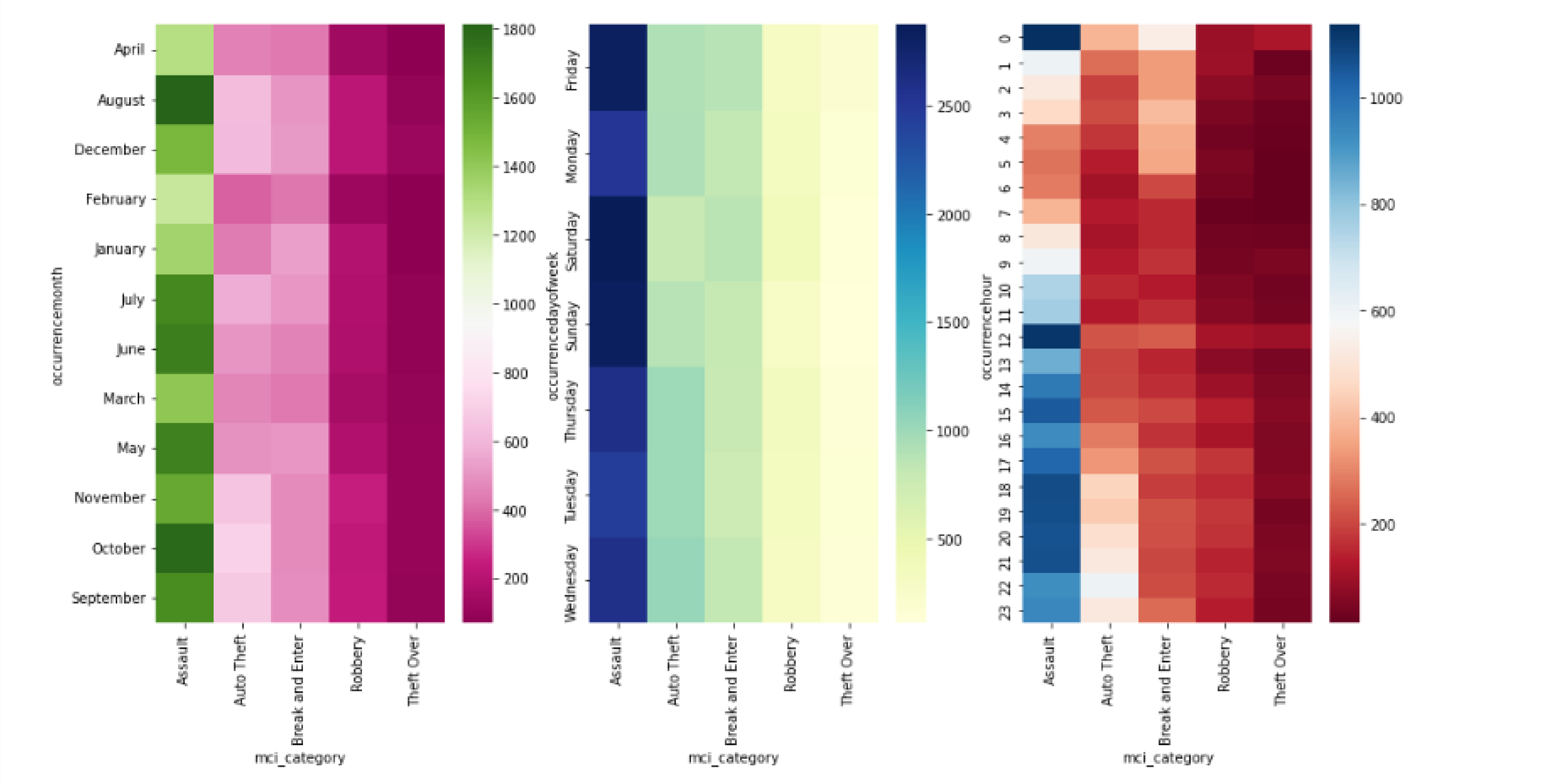
Crime Categories by Premise Types

mci_category	Assault	Auto Theft	Break and Enter	Robbery	Theft Over
premises_type					
Apartment	6286	222	1624	232	246
Commercial	2359	499	2424	675	317
Educational	429	6	94	53	9
House	2214	2303	1167	73	146
Other	1480	264	392	136	69
Outside	4866	3236	1	1021	278
Transit	1037	11	15	90	3



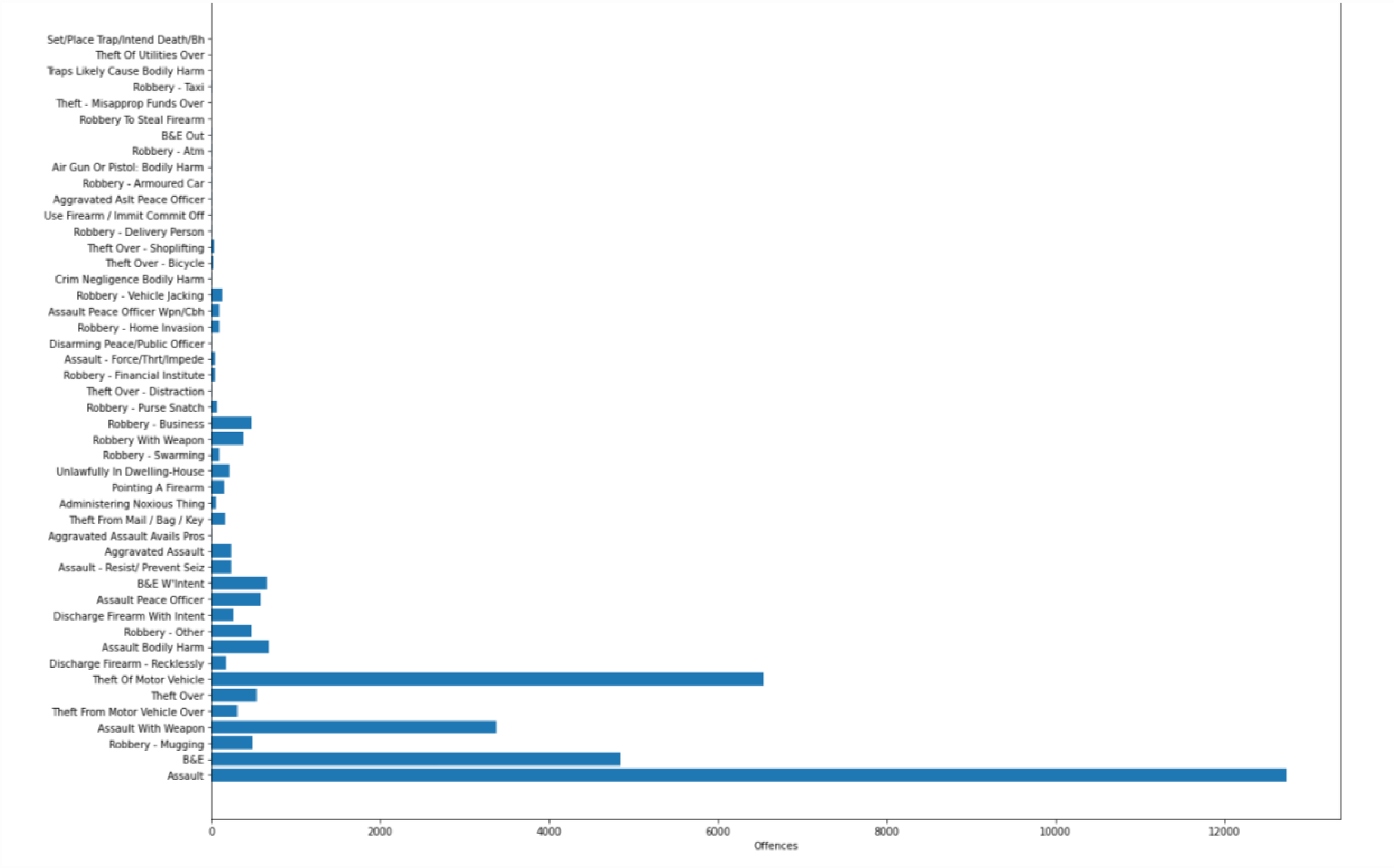
We see that assault is a prevelant crime especially in apartments. Auto thefts commonly occur outside

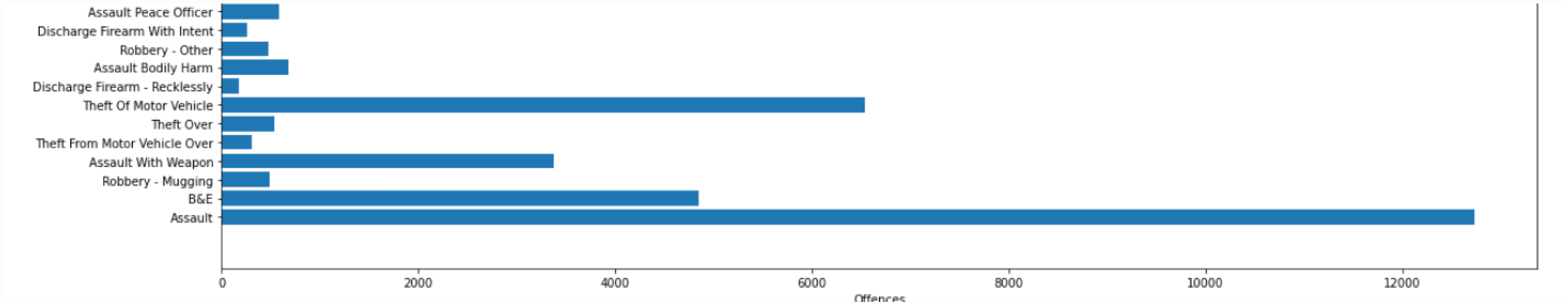
Heatmaps of MCI Categories by Month,Day and Hour



Assaults are higher from Aug-Oct, especially on weekends at noon and midnight

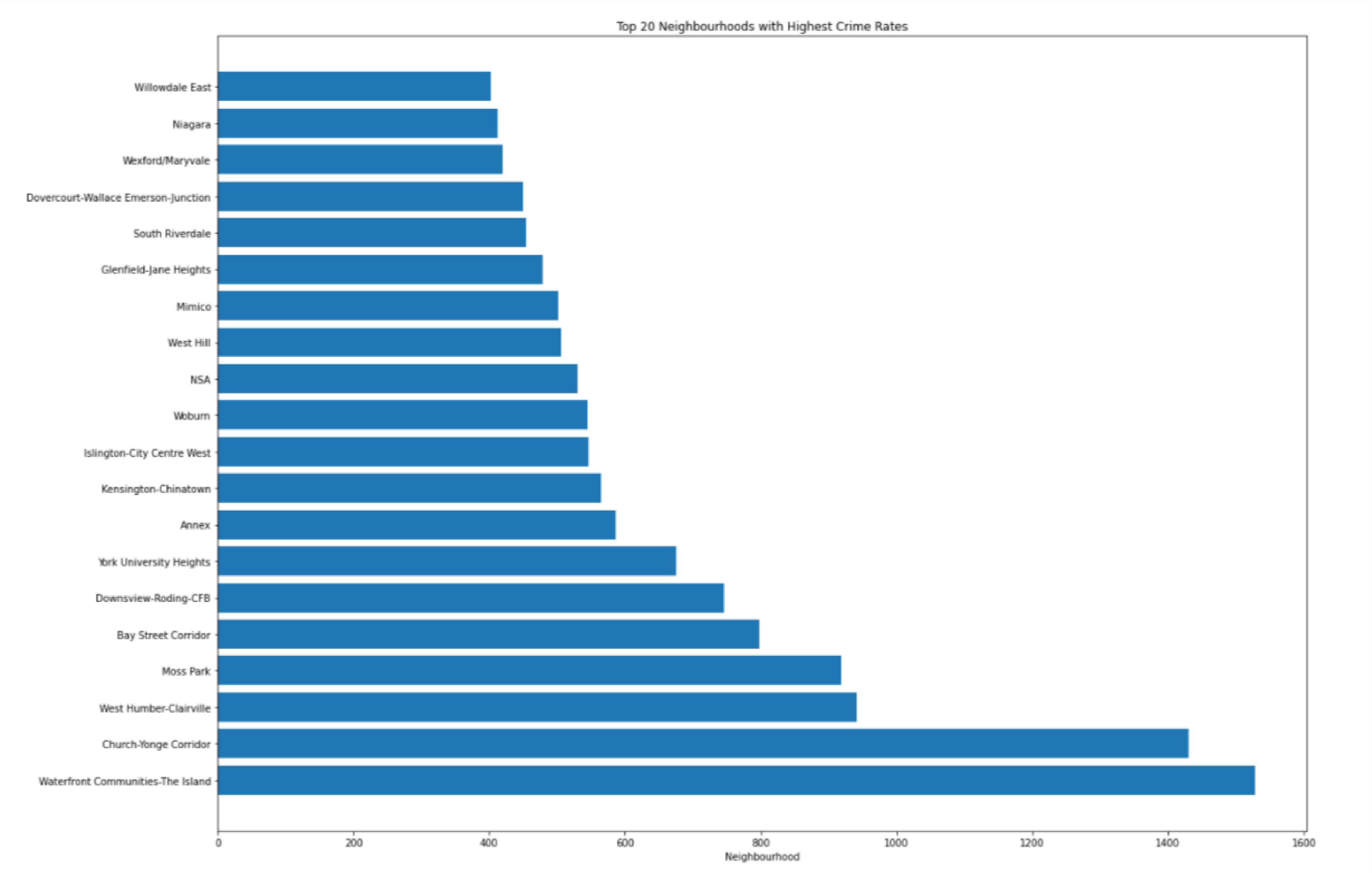
Top Offences





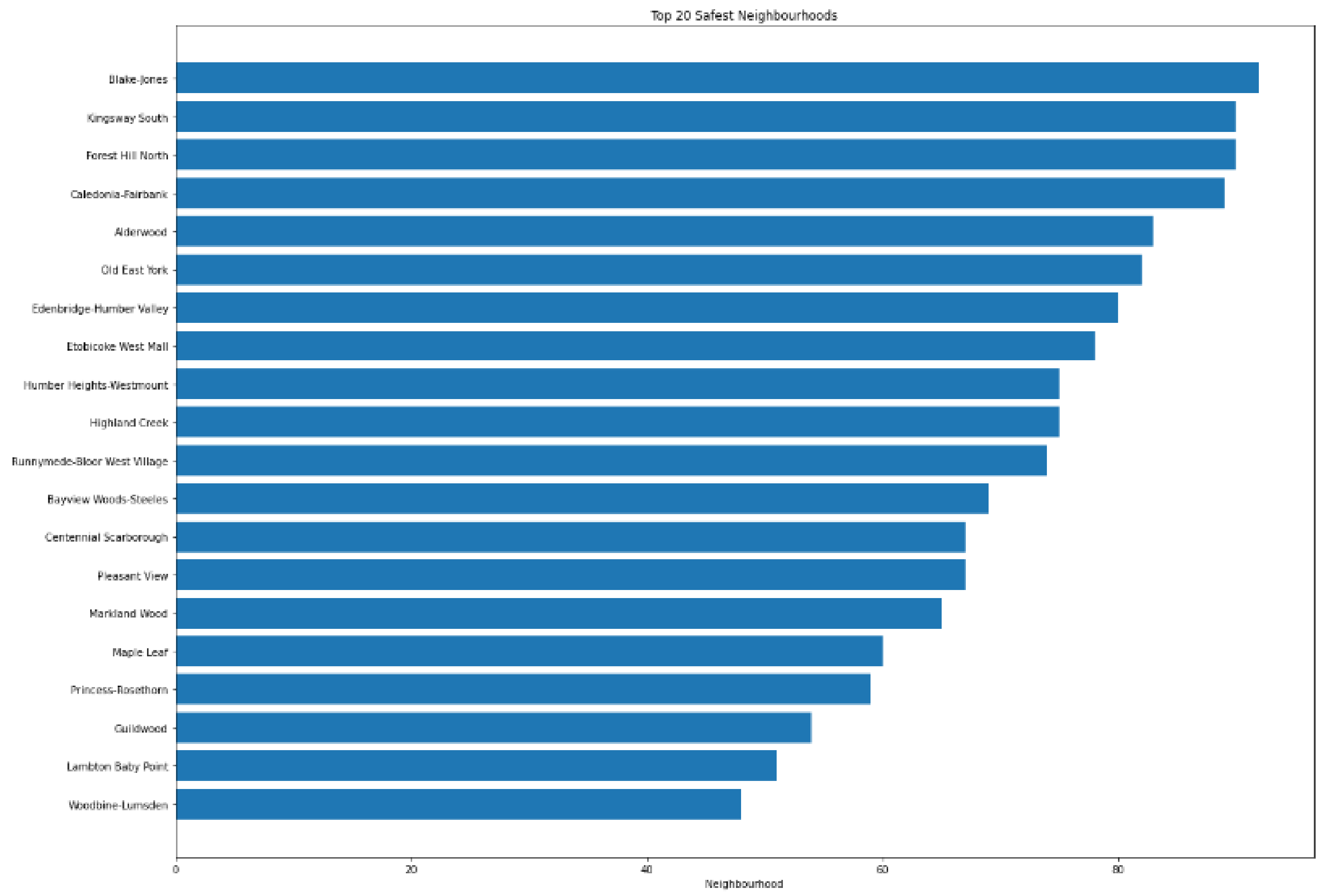
Top 3 offences are assault, Theft of Motor Vehicle and B&E(break and enter)

Top 20 Neighbourhoods with Highest Crime Rates



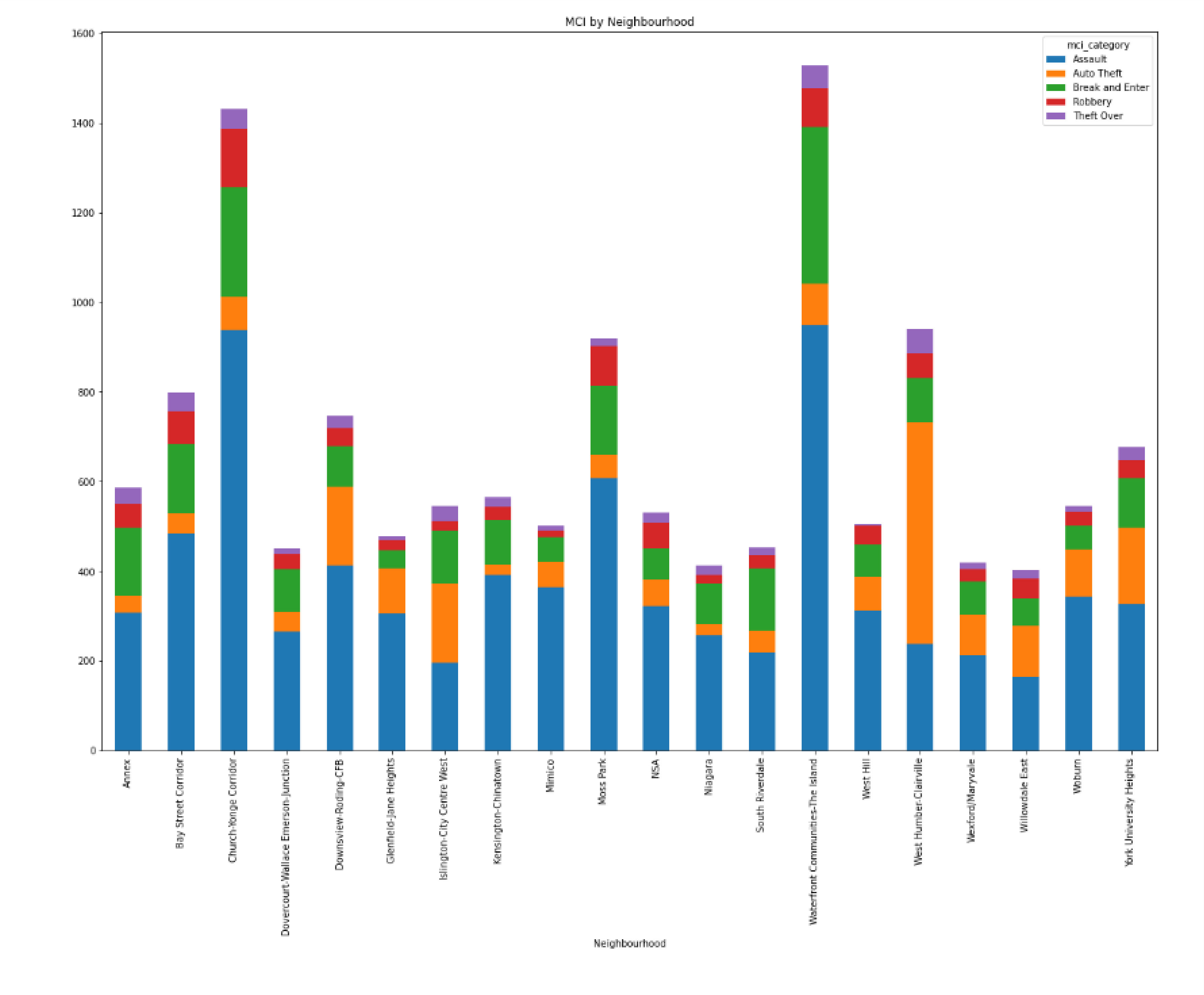
We notice that Waterfront Communities-The Island, Church-Yonge Corridor are neighbourhoods with the highest crime rate

Top 20 Safest Neighbourhoods



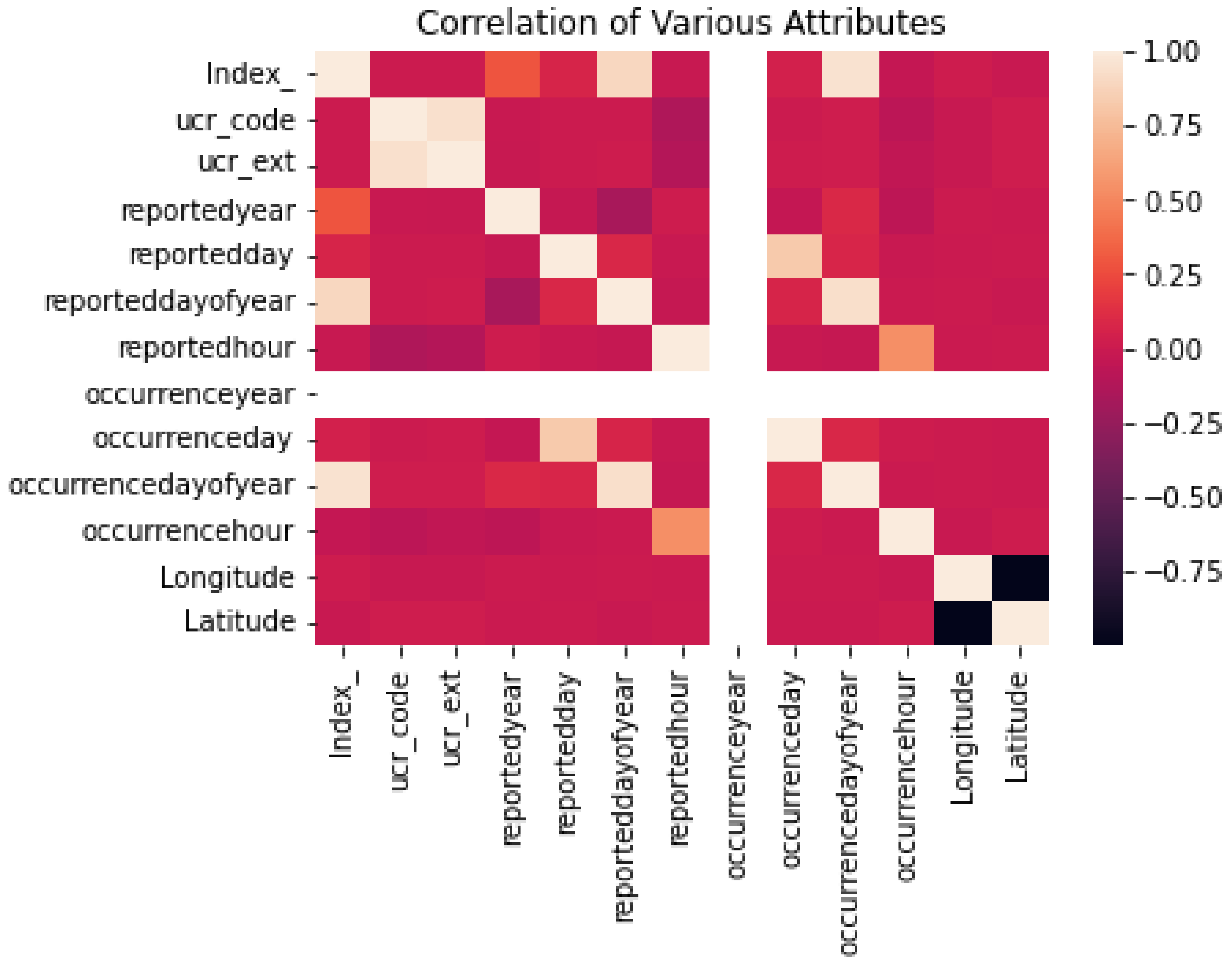
We notice that Woodbine-Lumsden, Lambton Baby Point, Guildwood are the safest neighbourhoods

Top 20 Neighbourhoods by MCI categories

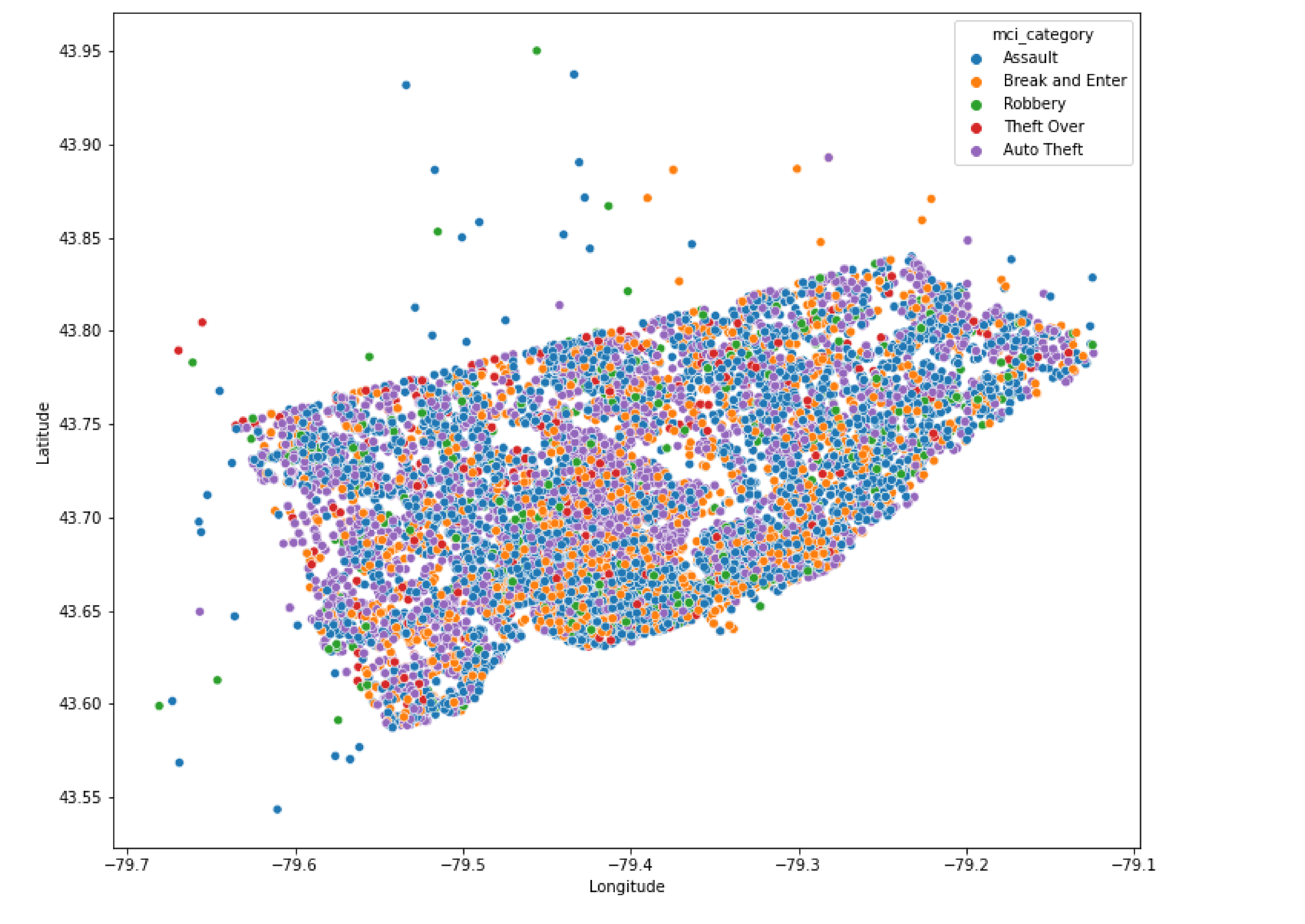


Besides assault, Waterfront Communities, Church-Yonge Corridor and Moss Park have high number of break and enter crimes, while West-Humber-Clairville has significant number of auto thefts

Correlation of Various Attributes



Latitude v/s Longitude



DATA PREPROCESSING

Feature selection is a data preprocessing technique for selecting a subset of the best variables prior to constructing a model.

SMOTE is an oversampling technique that generates synthetic samples from the minority class.

Multi-Column Encoder

```
Outside        9402
Apartment      8610
Commercial     6274
House          5903
Other          2341
Transit        1156
Educational     591
Name: premises_type, dtype: int64
5    9402
0    8610
1    6274
3    5903
4    2341
6    1156
2     591
Name: premises_type, dtype: int64
```

```
Assault            18671
Auto Theft          6541
Break and Enter     5717
Robbery             2280
Theft Over          1068
Name: mci_category, dtype: int64
0    18671
1     6541
2     5717
3     2280
4     1068
Name: mci_category, dtype: int64
```

CLASSIFICATION MI MODELS

1) Logistic Regression
2) Gaussian Naïve Bayes
3) KNN
4) Random Forest Ensemble Model
5) Adaboost

Using Multi-Column Encoder

Logistic Regression

```
Accuracy of Logistic Regression :  0.28846403531550247
[[ 758 1413 1568 1379  943]
 [ 482 2004 1441 1765  640]
 [ 695  156 3016 1158 1147]
 [ 675 1665 1207 1747  925]
 [ 653  672 2049 1288 1362]]
              precision    recall  f1-score   support

           0       0.23      0.13      0.16      6061
           1       0.34      0.32      0.33      6332
           2       0.32      0.49      0.39      6172
           3       0.24      0.28      0.26      6219
           4       0.27      0.23      0.25      6024

    accuracy                           0.29     30808
   macro avg       0.28      0.29      0.28     30808
weighted avg       0.28      0.29      0.28     30808
```

Gaussian Naive Bayes

```
Accuracy of Gaussian Naive Bayes :  0.3066086730719293
[[   5 3018 2313  169  556]
 [   1 4289 1427  259  356]
 [   0 1292 4228  220  432]
 [   1 3635 1642  293  648]
 [   0 2205 2982  206  631]]
              precision    recall  f1-score   support

           0       0.71      0.00      0.00      6061
           1       0.30      0.68      0.41      6332
           2       0.34      0.69      0.45      6172
           3       0.26      0.05      0.08      6219
           4       0.24      0.10      0.15      6024

    accuracy                           0.31     30808
   macro avg       0.37      0.30      0.22     30808
weighted avg       0.37      0.31      0.22     30808
```

KNN

```
Accuracy of KNN :  0.3290703713321215
[[ 803  924  999 1496 1839]
 [ 653 1491 1080 1474 1634]
 [ 537  631 2160 1024 1820]
 [ 721  895  769 2208 1626]
 [ 346  527  809  866 3476]]
              precision    recall  f1-score   support

           0       0.26      0.13      0.18      6061
           1       0.33      0.24      0.28      6332
           2       0.37      0.35      0.36      6172
           3       0.31      0.36      0.33      6219
           4       0.33      0.58      0.42      6024

    accuracy                           0.33     30808
   macro avg       0.32      0.33      0.31     30808
weighted avg       0.32      0.33      0.31     30808
```

Random Forest Ensemble Model

```
Accuracy of Random Forest Ensemble Model :  0.4975006491820306
[[2466 1209  764  737  885]
 [ 402 4474  327  668  461]
 [ 717  578 3141  533 1203]
 [ 824 1279  612 2517  987]
 [ 465 1125 1117  588 2729]]
              precision    recall  f1-score   support

           0       0.51      0.41      0.45      6061
           1       0.52      0.71      0.60      6332
           2       0.53      0.51      0.52      6172
           3       0.50      0.40      0.45      6219
           4       0.44      0.45      0.44      6024

    accuracy                           0.50     30808
   macro avg       0.50      0.50      0.49     30808
weighted avg       0.50      0.50      0.49     30808
```

Adaboost

```
Accuracy of Adaboost :  0.429044040508959
[[1793 1211 1327  962  768]
 [ 342 3981  509  973  527]
 [ 513  576 3673  393 1017]
 [ 602 1601 1187 1798 1031]
 [ 387 1041 1905  718 1973]]
              precision    recall  f1-score   support

           0       0.49      0.30      0.37      6061
           1       0.47      0.63      0.54      6332
           2       0.43      0.60      0.50      6172
           3       0.37      0.29      0.33      6219
           4       0.37      0.33      0.35      6024

    accuracy                           0.43     30808
   macro avg       0.43      0.43      0.42     30808
weighted avg       0.43      0.43      0.42     30808
```

Using One-Hot Encoding

Random forest

```
Accuracy of Random Forest with OneHotEncoder :  0.6872812135355892
```

	precision	recall	f1-score	support
Assault	0.69	0.90	0.78	4634
Break and Enter	0.64	0.36	0.46	1380
Robbery	0.80	0.34	0.48	573
Theft Over	0.04	0.00	0.01	302
Auto Theft	0.70	0.61	0.65	1681
accuracy			0.69	8570
macro avg	0.57	0.44	0.47	8570
weighted avg	0.67	0.69	0.66	8570

Gaussian Naive Bayes

```
Accuracy of Gaussian Naive Bayes with OneHotEncoder :  0.5408401400233372
```

	precision	recall	f1-score	support
Assault	0.54	0.98	0.70	4634
Break and Enter	0.00	0.00	0.00	1380
Robbery	0.00	0.00	0.00	573
Theft Over	0.00	0.00	0.00	302
Auto Theft	0.40	0.06	0.10	1681
accuracy			0.54	8570
macro avg	0.19	0.21	0.16	8570
weighted avg	0.37	0.54	0.40	8570

Logistic Regression

Accuracy of Logistic Regression with OneHotEncoder : 0.661610268378063

	precision	recall	f1-score	support
Assault	0.70	0.84	0.76	4634
Break and Enter	0.54	0.46	0.50	1380
Robbery	0.47	0.17	0.26	573
Theft Over	0.26	0.03	0.06	302
Auto Theft	0.64	0.63	0.63	1681
accuracy			0.66	8570
macro avg	0.53	0.43	0.44	8570
weighted avg	0.63	0.66	0.64	8570

Decision Trees

Accuracy of Decision Tree Classifier with OneHotEncoder : 0.5861143523920653

	precision	recall	f1-score	support
Assault	0.64	0.81	0.71	4634
Break and Enter	0.37	0.40	0.38	1380
Robbery	1.00	0.01	0.02	573
Theft Over	0.00	0.00	0.00	302
Auto Theft	0.59	0.43	0.50	1681
accuracy			0.59	8570
macro avg	0.52	0.33	0.32	8570
weighted avg	0.59	0.59	0.55	8570

	precision	recall	f1-score	support
Assault	0.58	0.97	0.73	4634
Break and Enter	0.71	0.04	0.08	1380
Robbery	0.00	0.00	0.00	573
Theft Over	0.00	0.00	0.00	302
Auto Theft	0.69	0.32	0.44	1681
accuracy			0.59	8570
macro avg	0.40	0.27	0.25	8570
weighted avg	0.56	0.59	0.49	8570

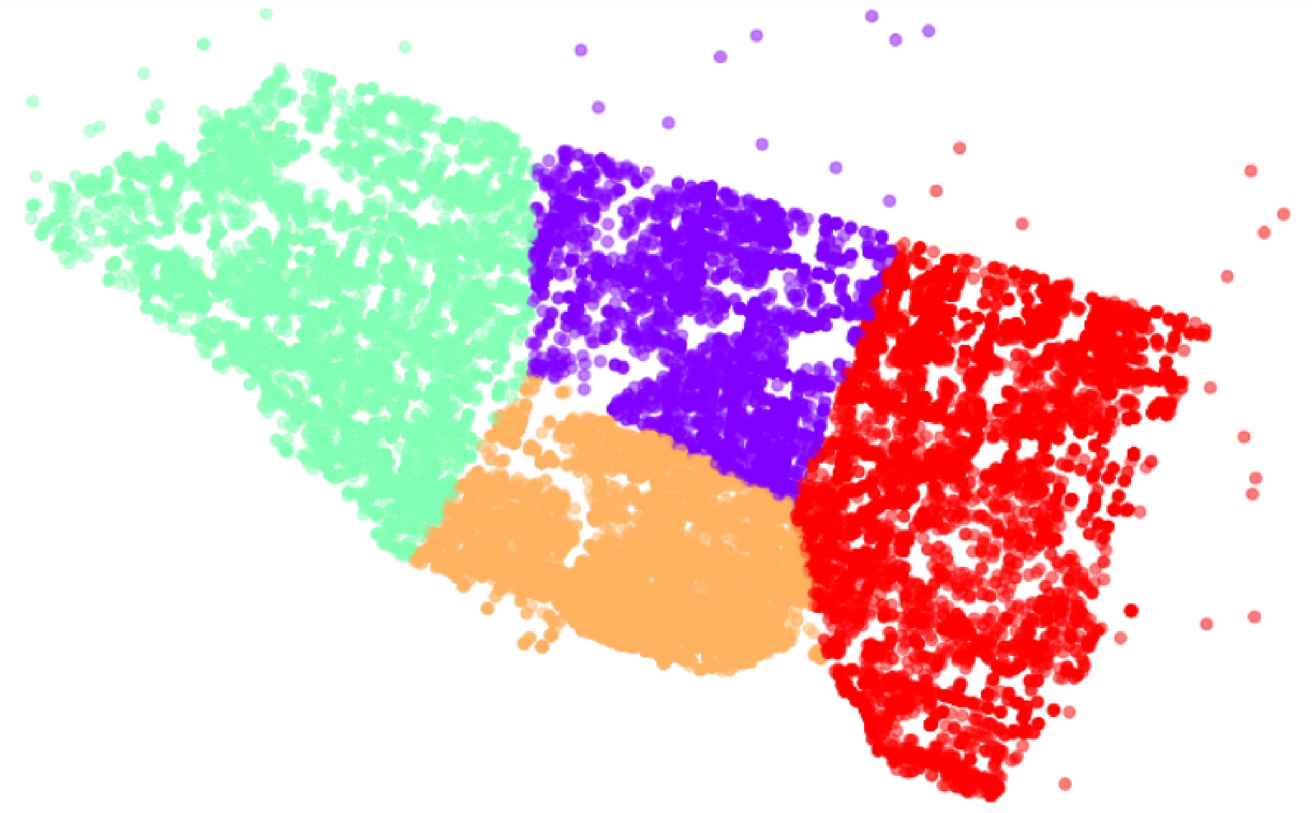
KNN

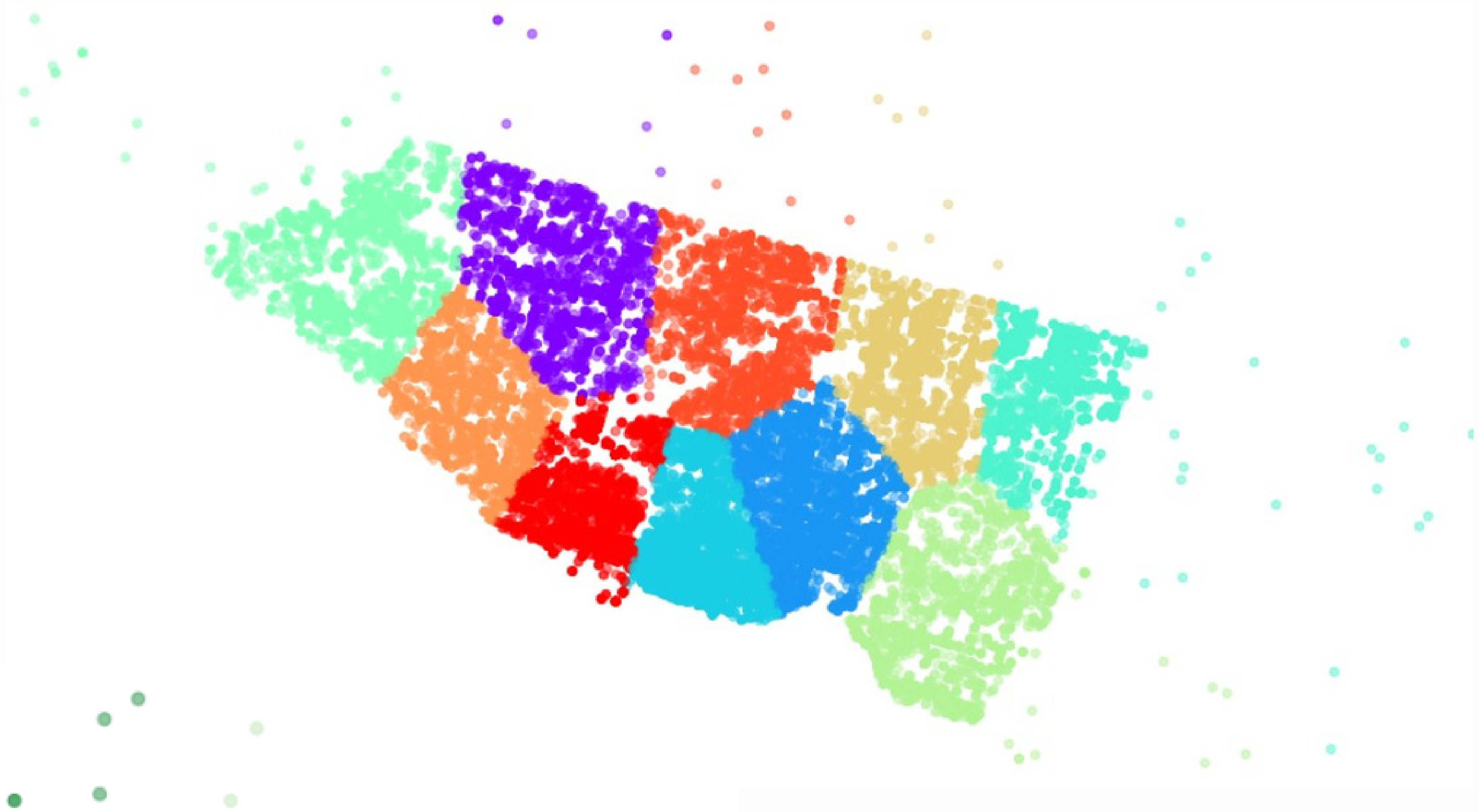
Accuracy of KNN with OneHotEncoder : 0.5929988331388565

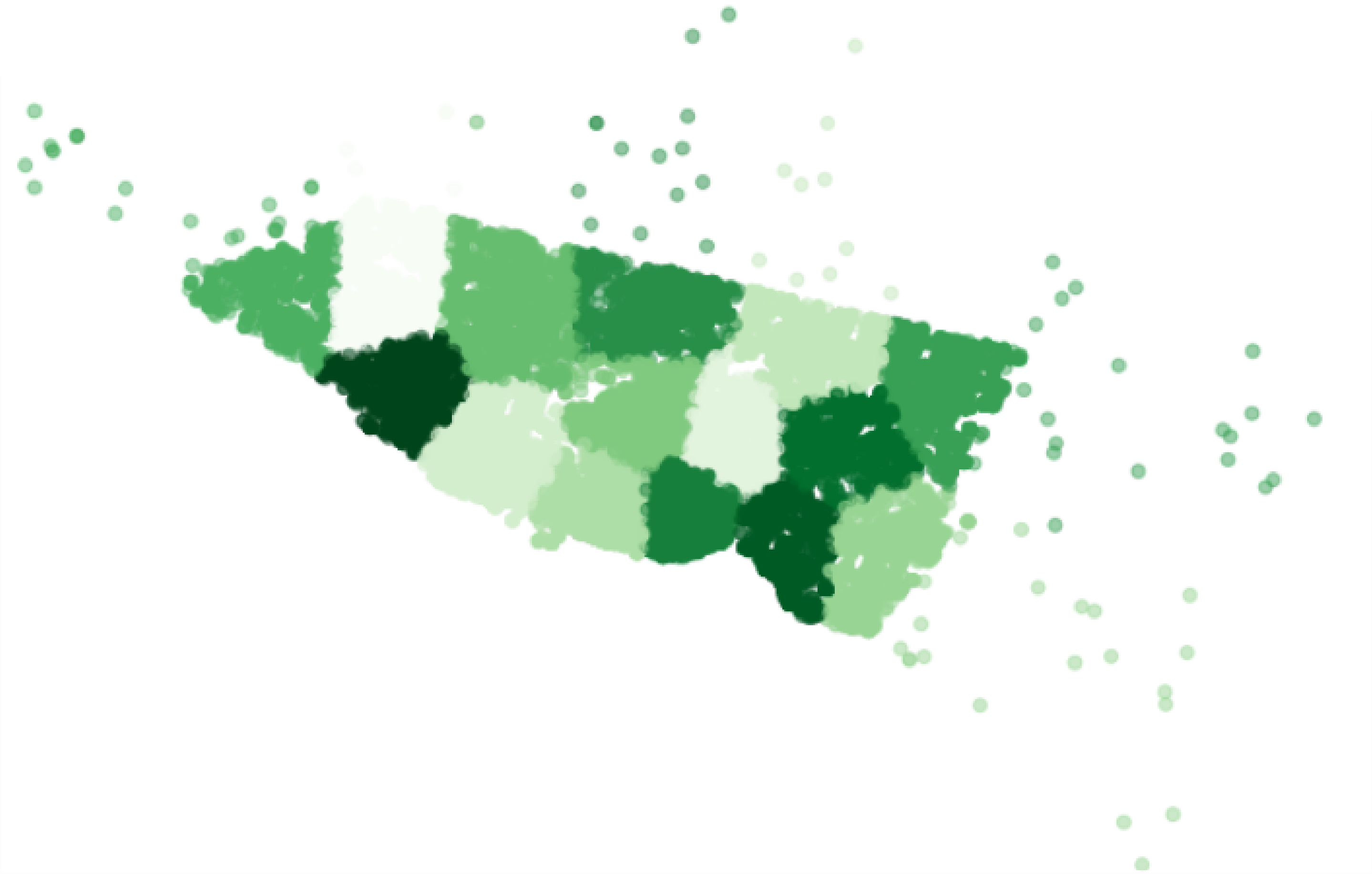
CLUSTERING

- It is a technique which involves the grouping of simialr datapoints together. Here, we have used K Means Clustering.
- We use the 'KMeans' module from the sklearn.cluster library.
- The input data is fitted into this model for getting the result.
- Then we use the 'matplotlib' library to plot the clusters in K Means.
- Alternatively, we can make a user defined function to implement the KMeans module and call it to perform the clustering.

CLUSTERING







TESTING

-FUNCTIONAL TESTING
1) UNIT TESTING: Each one of us did unit testing in the models we implemented.
2) INTEGRATION TESTING: While combining the different models, we did integration testing code using bottom-up approach
3)UI TESTING: All the team mates tested the UI that is interface and working.

-Gradio is the fastest way to demo the machine learning model with a friendly web interface hence we gradio to test our model through a interface .

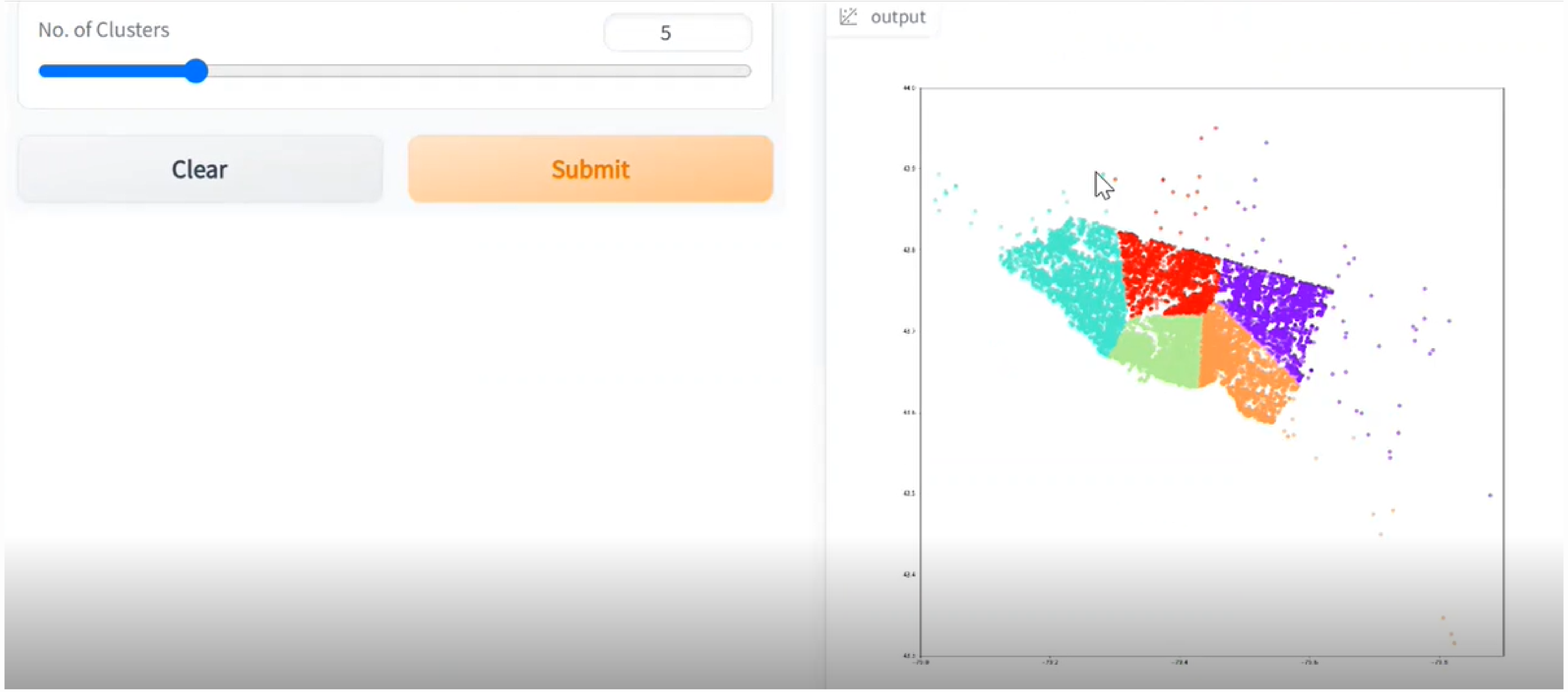
-To test the models we divide the data into test and train data which were evaluated with python library to get accuracy and other performance metrics.

-Metrics used include confusion matrix which returns a table layout that helps to visualize the performance of an algorithm rather than producing a numerical value that indicates the goodness of the algorithm. Precision and recall is found from this. Accuracy measures how many predictions are matched exactly with the actual or true label of the testing dataset and returns the percentage of correct results.Log loss is used to measure performance of classifiers by penalizing false classifications.

ROLE OF TEAM MEMBERS:- Each one of us did unit testing as well did pair programming. Each one of tested other's codes as well.

UI for the Project

K-Means Clustering



Classification

CRIME ANALYSIS

premises_type

3

occurrenceyear

2021

month_id

7

day_id

7

occurrencedayofyear

192

occurrencehour

20

Neighbourhood

130

Latitude

43.712973

Longitude

-79.455704

Clear

Submit

output

[1]

Flag

RESULTS USING MULTICLASS ENCODER

PERFORMANCE METRICS	Logistic Regression	Gaussian Naïve Bayes	KNN	Random Forest	Adaboost
Accuracy	28.9%	30.6%	32.9%	49.8%	42.9%
Precision	0.28	0.37	0.32	0.50	0.43
Recall	0.29	0.31	0.33	0.50	0.43
F1 Score	0.29	0.22	0.31	0.49	0.42

RESULTS using ONE HOT ENCODING

PERFORMANCE METRICS	Logistic Regression	Gaussian Naïve Bayes	KNN	Random Forest	Decision Tree
Accuracy	66%	54%	59.3%	68.7%	58.6%
Precision	0.63	0.37	0.56	0.67	0.59
Recall	0.66	0.54	0.59	0.69	0.59
F1 Score	0.64	0.40	0.49	0.66	0.55

Results and Conclusion

- Our goal of the project was to compare various classification as well as clustering models to check which model would work better.
- We did check lot of classification as well as clustering models which gave a conclusion that RANDOM FOREST and ADABOOST gave good accuracy compared to other models such as KNN, LOGISTIC REGRESSION, NAIVE BAYES .
- Data preprocessing was followed by splitting the dataset into training and testing sets, and later the performance parameters were examined.

- The exploratory data analysis exhibited extensive visualizations regarding crime particulars, including crime rates in different periods from daily to yearly trends, crime types , and high-intensity areas based on historical patterns.

FUTURE WORK

- Other factors like population and employment data can be analyzed to understand the trends better
- For future work, we want to expand this study to implement different learning techniques with corresponding visual data for different crime datasets using the latitude and longitude and show the regions which would be more safe for the people to travel in maps so the customer can choose whether or not to travel in that area.

REFERENCES

[1] Public Safety Data Portal. [Online]. Available: https://data.torontopolice.on.ca/.

[2] T. Dayara, F. Thabtah, H. Abdel-Jaber, and S. Zeidan, “Crime analyses using data analytics,” International Journal of Data Warehousing and Mining, vol. 18, no. 1, pp. 1–15, 2022.

[3] Mahmud, S., Nuha, M., & Sattar, A. (2020). Crime rate prediction using machine learning and Data Mining. Advances in Intelligent Systems and Computing, 59–69. https://doi.org/10.1007/978-981-15-7394-1_5

[4] Safat, W., Asghar, S., & Gillani, S. A. (2021). Empirical analysis for crime prediction and forecasting using machine learning and Deep Learning Techniques. IEEE Access, 9, 70080–70094. https://doi.org/10.1109/access.2021.3078117

[5] Sathyadevan, S., Devan, M. S., & Gangadharan, S. S. (2014). Crime analysis and prediction using data mining. 2014 First International Conference on Networks & Soft Computing (ICNSC2014). https://doi.org/10.1109/cnsc.2014.6906719

[6] Paper on Different Approaches for Crime Prediction system, International Journal of Engineering Research & Technology (IJERT) 2017.

[7] CRIME RATE PREDICTION, Journal of Engineering Sciences (JES), 2020.

Thank You