Toronto crime prediction project

Ketao Li
School of Continuing Studies
York University
Toronto ON Canada
liketao@yahoo.com

ABSTRACT

Data mining and machine learning have become a vital part of crime detection and prevention. It is easier to determine the extent of vulnerability an individual is subjected to, at a specific geographic area on any occasion by use of machine learning algorithm. The main objective of the project is to anticipate if a particular neighborhood in the city, at a given duration of the day will be a crime hotspot or not, with an acceptable rate of accuracy. The project aims to exploit background criminal knowledge procured from Toronto Police Service's PSDP (Public Safety Data Portal). The PSDP is an ideal and reliable source for assimilating criminal occurrences in Toronto, where criminal investigation records are preserved since 2014. The project, with the combination of demographic information, establishes an approach of detecting crimes in a particular geographic area by analyzing and studying the criminal occurrences of the area and thus making deductions by employing well-founded and reliable learning algorithm. The analysis is further extended to incorporate the impact of housing and inhabitation, literacy rate, employment and socioeconomic status on the crime occurrence rate. The project also gives some recommendations to decrease the crime rate and enhance crime prevention of the selected MCI (Major Crime Indicators).

1 Introduction

Toronto is most populated city in Canada. It is of the most ethnically diverse cities in Canada. Crime is one of the biggest and dominating problem and its prevention is an important task. Even though Toronto known to be the safest city it is observed that assault and break & enter is still a problem.

There has been tremendous increase in machine learning algorithms that have made crime prediction feasible based on the past data. The main objective of the project is to anticipate if a particular neighborhood in the city, at a given duration of the day will be a crime hotspot or not, with an acceptable rate of accuracy.

The project aims to exploit background criminal knowledge procured from Toronto Police Service's PSDP (Public Safety Data Portal). The project conducts exploratory analysis on particular crime type from the MCI dataset (Robbery, Assault, Theft Over, Auto Theft, Break & Enter). The project gains the insights of main crime tendencies and correlations with demographics, other MCIs temporal data(time of day, day of week) , and other factors.

The project, with the combination of demographic information, establishes an approach of detecting crimes in a particular geographic area by analyzing and studying the criminal occurrences of the area and thus making deductions by employing well-founded and reliable learning algorithm. Here we apply various types of classifiers to model our dataset and compare their accuracies. Following are the methods I'll apply for classification:

- Logistic Regression

The binary logistic model is used to estimate the probability of a binary response based on one or more predictor (or independent) variables (features). It allows one to say that the presence of a risk factor increases the odds of a given outcome by a specific factor.

- Linear SVM

Support vector machines (SVMs) are a set of supervised learning methods which learn from the dataset and used for classification. Given a set of training examples, each marked as belonging to one of two classes, an SVM algorithm builds a model that predicts whether a new example falls into one class or the other.

- Decision Tree

Decision tree learning uses a decision tree (as a predictive model) to go from observations about an item (represented in the branches) to conclusions about the item's target value (represented in the leaves). It is one of the predictive modeling approaches used in statistics, data mining and machine learning. In data mining, a decision tree describes data (but the resulting classification tree can be an input for decision making).

- Naive Bayes model

Naive Bayes is a classification algorithm for binary (two-class) and multi-class classification problems. The technique is easiest to understand when described using binary or categorical input values.

- Random Forest

Random forests or random decision forests are 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 classes (classification) or mean prediction (regression) of the individual trees. Random decision forests correct for decision trees' habit of overfitting to their training set.

- kNN

kNN is considered among the oldest non-parametric classification algorithms. To classify an unknown example, the distance from that example to every other training example is measured. The k smallest distances are identified, and the most represented class by these k nearest neighbors is considered the output class label.

Ensemble methods are techniques that create multiple models and then combine them to produce improved results. Ensemble methods usually produce more accurate solutions than a single model would. This has been the case in a number of machine learning competitions, where the winning solutions used ensemble methods. The project uses some ensemble methods to get better model performance.

2 Problem Statement

In Toronto, crime is a big concern for the citizens. The residents expect to live in a safer place and the government wants to decrease crime rate. So the MCI dataset is very important for us to analyze.

It's quite helpful for crime prevention if we can predict crime occurrence in advance. The machine learning prediction model is expected to be established to predict the occurrence of a crime at a location at a specific time of a day and to anticipate if a particular neighborhood in the city, at any given duration of the day will be a crime hotspot or not, with an acceptable rate of accuracy. Furthermore, to incorporate the impact of housing and inhabitation, literacy rate, employment and socioeconomic status on the crime occurrence rate.

In this project, I plan to perform data modeling using classification models. There are mainly two types of classification: binary classification and multiclass classification. Both binary classification and multiclass classification will be used in the project. In the multiclass classifier, I'll use the temporal features, spatial features and demographic features to predict the crime type(one of crimes in Homicide, Robbery, Assault, Theft Over, Auto Theft, Break & Enter). In the binary classifier, I classify Robbery, and Assault into severe crime type. And I classify the other MCI types into not severe.

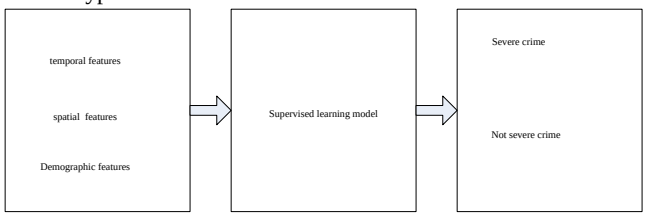


Figure 1 binary classifier

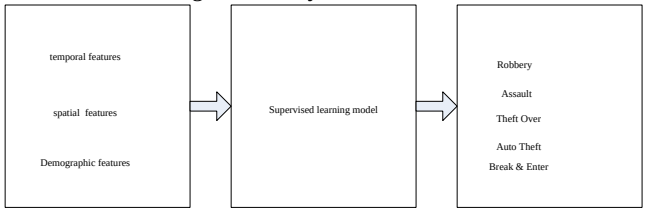


Figure 2 multiclass classifier

Besides classification models, I plan to use the unsupervised machine learning algorithms such as clustering to discover new relationship between various features.

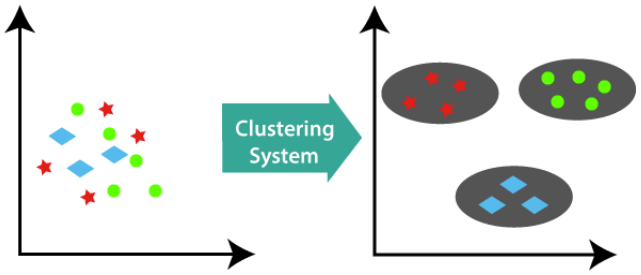


Figure 3 clustering

3 Dataset description

The Toronto Police Service (TPS) is undergoing continuous improvement efforts to enhance and strengthen ties with the society by providing access to open data for public safety in Toronto. The service provides Major Crime Indicator (MCI) dataset since 2014 on the Toronto Police Service's Public Safety Data Portal (PSDP).

variable	Description
X	The x coordinate of the location where the incident occurred in projection.
Y	The y coordinate of the location where the incident occurred in projection.
Index	Incident index
Event_unique_id	Event unique identificaiton
occurrencedate	The date when the incident occurred.
reporteddate	The date when the incident was reported.
premisetype	Premise type
ucr_code	Uniform Crime Reporting code
ucr_ext	Uniform Crime Reporting extension code
offence	An offence is a violation against the Criminal Code of Canada or other federal statute, provincial act, or municipal by-law.
reportedyear	The year when the incident was reported.
reportedday	The day when the incident was reported.
reporteddayofyear	The day of year when the incident was reported.
reporteddayofweek	The day of week when the incident was reported.
reportedhour	The hour when the incident was reported.
occurrenceyear	The year when the incident occurred.
occurrencemonth	The month when the incident occurred.
occurrenceday	The day when the incident occurred.
occurrencedayofyear	The day from the first day of the year when the incident occurred.
occurrencehour	The hour when the incident occurred.
MCI	Major Crime Indicators
Division	Toronto police division
Hood_ID	Neighbourhood identification

Neighbourhood	Neighbourhood name
Long	The latitude of the location where the incident occurred.
Lat	The longitude of the location where the incident occurred.
ObjectId	Object Identification

Table 1 dataset description

MCI	description
Homicide	The Homicide or Murder category includes first and second degree murder, and manslaughter. A Homicide or Murder occurs when a person directly or indirectly, by any means, causes the death of another human being.
Robbery	The act of taking property from another person or business by the use of force or intimidation in the presence of the victim.
Assault	The direct or indirect application of force to another person, or the attempt or threat to apply force to another person, without that person's consent.
Theft Over	The act of stealing property in excess of $5,000 (excluding auto theft).
Auto Theft	The act of taking or another person's vehicle (not including attempts). Auto Theft figures represent the number of vehicles stolen.
Break & Enter	The act of entering a place with the intent to commit an indictable offence therein.

Table 2 MCI description

4 ML solution

In this project, both supervised and unsupervised machine learning algorithms are used.

5.1 Supervised learning

There are mainly two types of classification: binary classification and multiclass classification. Both binary classification and multiclass classification will be used in the project. In the multiclass classifier, I'll use the temporal features, spatial features and demographic features to predict the crime type(one of crimes in Homicide, Robbery, Assault, Theft Over, Auto Theft, Break & Enter). In the binary classifier, I classify Homicide, Robbery, Assault into severe crime type. And I classify the other MCI types(Theft Over, Auto Theft, Break & Enter.) into not severe crime type.

In the project, supervised learning algorithms such as logistic regression, linear SVM, Decision Tree, Naïve Bayes model, random forest, KNN will be used to make the crime prediction.

In the project, ensemble methods are also be used to improve the model accuracy. Ensemble methods are techniques that create multiple models and then combine them to produce improved results. Ensemble methods usually produce more accurate solutions than a single model would. This has been the case in a number of machine learning competitions, where the winning solutions used ensemble methods. The project uses some ensemble methods to get better model performance.

Ensemble methods are meta-algorithms that combine several machine learning techniques into one predictive model in order to decrease variance (bagging), bias (boosting), or improve predictions (stacking).

Bagging

Bagging stands for bootstrap aggregation. One way to reduce the variance of an estimate is to average together multiple estimates. For example, we can train M different trees on different subsets of the data (chosen randomly with replacement) and compute the ensemble:

$$f(x) = 1/M \sum_{m=1}^{M} f_m(x)$$

Bagging uses bootstrap sampling to obtain the data subsets for training the base learners. For aggregating the outputs of base learners, bagging uses voting for classification and averaging for regression.

Boosting refers to a family of algorithms that are able to convert weak learners to strong learners. The main principle of boosting is to fit a sequence of weak learners− models that are only slightly better than random guessing, such as small decision trees− to weighted versions of the data. More weight is given to examples that were misclassified by earlier rounds.

The predictions are then combined through a weighted majority vote (classification) or a weighted sum (regression) to produce the final prediction. The principal difference between boosting and the committee methods, such as bagging, is that base learners are trained in sequence on a weighted version of the data.

The algorithm below describes the most widely used form of boosting algorithm called AdaBoost, which stands for adaptive boosting.

Algorithm AdaBoost

1: Init data weights $\{w_n\}$ to $1/N$
2: **for** $m = 1$ to M **do**
3: fit a classifier $y_m(x)$ by minimizing weighted error function J_m:
4: $J_m = \sum_{n=1}^{N} w_n^{(m)} 1[y_m(x_n) \neq t_n]$
5: compute $\epsilon_m = \sum_{n=1}^{N} w_n^{(m)} 1[y_m(x_n) \neq t_n] / \sum_{n=1}^{N} w_n^{(m)}$
6: evaluate $\alpha_m = \log\left(\frac{1-\epsilon_m}{\epsilon_m}\right)$
7: update the data weights: $w_n^{(m+1)} = w_n^{(m)} \exp\{\alpha_m 1[y_m(x_n) \neq t_n]\}$
8: **end for**
9: Make predictions using the final model: $Y_M(x) = \text{sign}\left(\sum_{m=1}^{M} \alpha_m y_m(x)\right)$

Stacking is an ensemble learning technique that combines multiple classification or regression models via a meta-classifier or a meta-regressor. The base level models are trained based on a complete training set, and then the meta-model is trained on the outputs of the base level model as features.

The base level often consists of different learning algorithms and therefore stacking ensembles are often heterogeneous. The algorithm below summarizes stacking.

Algorithm Stacking

Input: training data $D = \{x_i, y_i\}_{i=1}^{m}$
Ouput: ensemble classifier H
Step 1: learn base-level classifiers
for $t = 1$ to T **do**
    learn h_t based on D
end for
Step 2: construct new data set of predictions
for $i = 1$ to m **do**
    $D_h = \{x_i', y_i\}$, where $x_i' = \{h_1(x_i), ..., h_T(x_i)\}$
end for
Step 3: learn a meta-classifier
learn H based on D_h
return H

5.2 unsupervised learning

The unsupervised method (such as K-means) is used in the project. The unsupervised machine learning algorithm are used to discover new relationship between various features.

- K-means

K-means is one of the simplest unsupervised learning algorithms that solve the well-known clustering problem. The procedure follows a simple and easy way to classify a given data set through a certain number of clusters (assume k clusters) fixed a priori. The main idea is to define k centers, one for each cluster. These centroids should be placed in a smart way because of different location causes different result. So, the better choice is to place them as much as possible far away from each other. The next step is to take each point belonging to a given data set and associate it to the nearest centroid. When no point is pending, the first step is completed and an early groupage is done. At this point we need to re-calculate k new centroids as barycenters of the clusters resulting from the previous step. After we have these k new centroids, a new binding has to be done between the same data set points and the nearest new centroid. A loop has been generated. As a result of this loop we may notice that the k centroids change their location step by step until no more changes are done. In other words centroids do not move any more.

The k-means procedure can be viewed as a greedy algorithm for partitioning the n samples into k clusters so as to minimize the sum of the squared distances to the cluster centers. It does have some weaknesses:

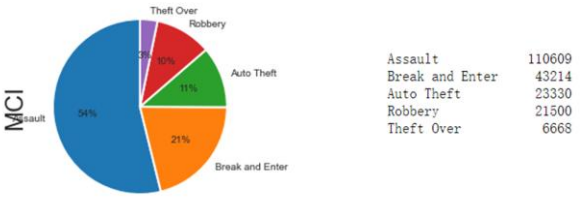
The way to initialize the means was not specified. One popular way to start is to randomly choose k of the samples.

Unfortunately there is no general theoretical solution to find the optimal number of clusters for any given data set. A simple approach is to compare the results of multiple runs with different k classes and choose the best one according to a given criterion, but we need to be careful because increasing k results in smaller error function values by definition, but also increases the risk of overfitting.

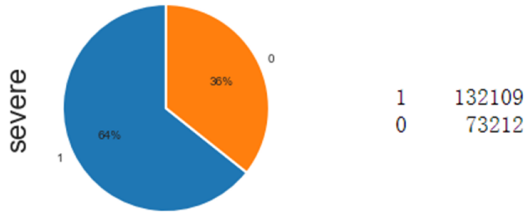
5 Project Experiment

5.1 Data preprocessing

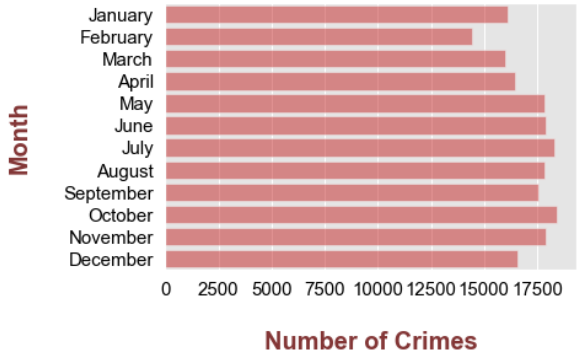
5.2 Data Exploration



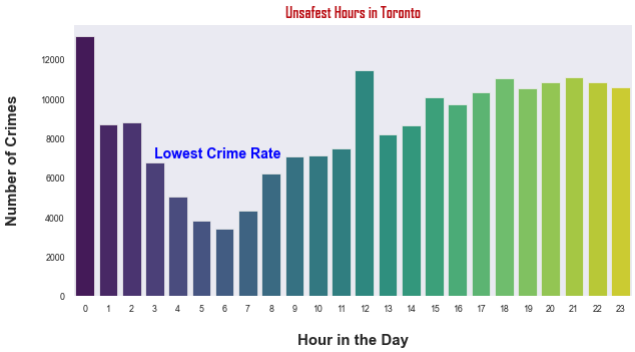
In the pie chart above, we can see that most crime types in the dataset are assault (54%). Only 3% of crime types are Theft over.



In the pie chart above, we can see that severe crime account for 64% of the dataset.

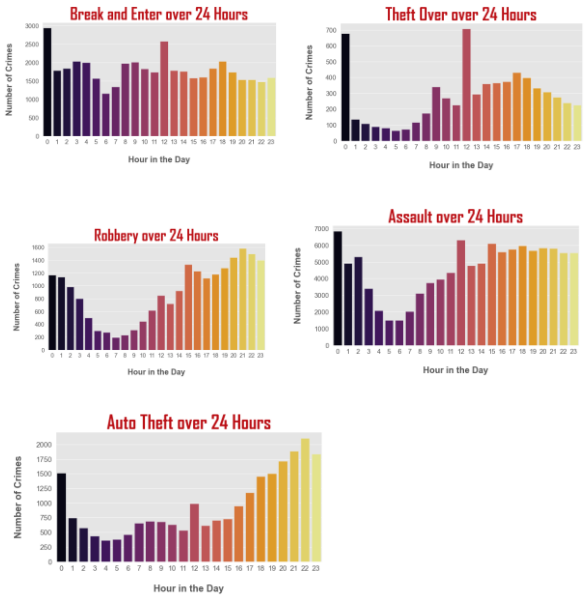


There are more crimes on July & October and least crimes on February.

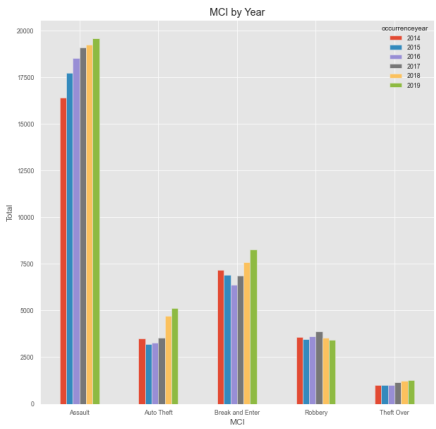


6 am is the safest hour in Toronto. Midnight is the most dangerous hour.

Insert Your Title Here



These figures above show the crime hour distribution for the specific crime.

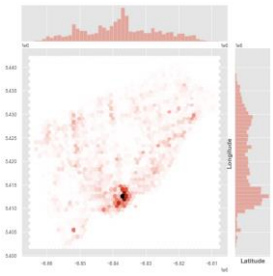


The figure above shows every MCI change by year. Assault increases year by year.

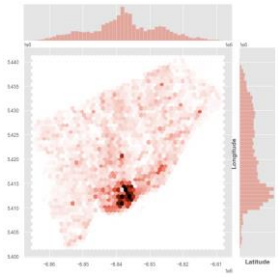
	Neighbourhood	Crime Count
122	Waterfront Communities-The Island	7707
6	Bay Street Corridor	6800
23	Church-Yonge Corridor	6217
124	West Humber-Clairville	5680
79	Moss Park	4768
138	York University Heights	3967
33	Downsview-Roding-CFB	3951
62	Kensington-Chinatown	3812
132	Woburn	3779
123	West Hill	3478

:

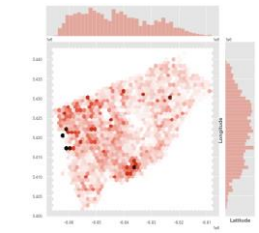
	Neighbourhood	Crime Cc
66	Lambton Baby Point	
134	Woodbine-Lumsden	
74	Maple Leaf	
47	Guildwood	
137	Yonge-St.Clair	
75	Markland Wood	
93	Old East York	
21	Casa Loma	
44	Forest Hill South	
64	Kingsway South	



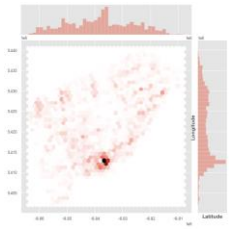
Assault distribution



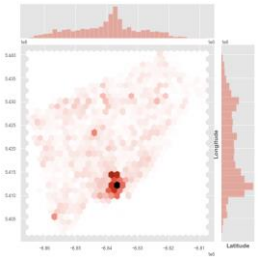
Break and Enter distribution



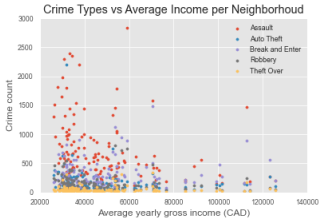
Auto Theft distribution

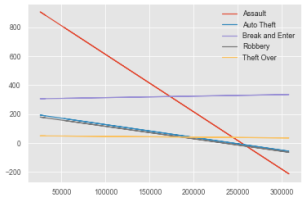


Robbery distribution

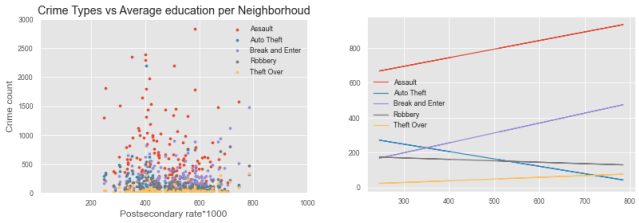


Theft over distribution

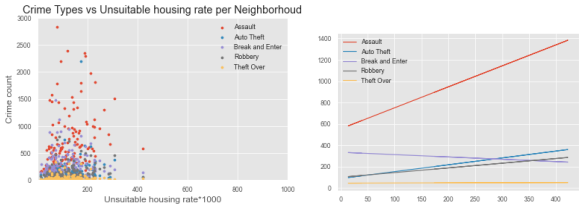




Crime vs income



Crime vs education



Crime vs housing

Correlations between MCI

5.3 Data Modelling

5.3.1 Multiple classification

5.3.1.1 Feature engineering

Recursive Feature Elimination

```
from sklearn import tree
from sklearn.feature_selection import RFE
model = tree.DecisionTreeClassifier()
rfe = RFE(model, 16)
fit = rfe.fit(X, y)
print("Num Features: %d" % fit.n_features_)
print("Selected Features: %s" % fit.support_)
print("Feature Ranking: %s" % fit.ranking_)
```

```
Num Features: 16
Selected Features: [ True  True  True  True  True  True  True  True False False  True
  True  True  True  True  True]
Feature Ranking: [1 1 1 1 1 1 1 1 1 2 3 1 1 1 1 1 1]
```

Feature Importance

```
from sklearn.ensemble import ExtraTreesClassifier
model = ExtraTreesClassifier(n_estimators=10)
model.fit(X, y)
print(model.feature_importances_)
```

```
[0.07514333 0.09955348 0.10194223 0.11402251 0.01475681 0.11672759
 0.11565578 0.01075627 0.01229054 0.01044473 0.00930637 0.00978005
 0.01366529 0.01453673 0.01344185 0.07777378 0.0846667  0.10553596]
```

We use the recursive feature elimination and feature importance methods to do the feature selection. Both methods show that population and Postsecondary are the least important feature. We remove these features.

5.3.1.2 Dataset balancing

Using SMOTE to balance the dataset

```
from imblearn import over_sampling
from imblearn.over_sampling import SMOTE
```

```
smote = SMOTE('minority')
```

```
X_sm, y_sm = smote.fit_sample(X, y)
```

```
unique, counts = np.unique(y_sm, return_counts=True)
dict(zip(unique, counts))
```

```
{0: 132109, 1: 132109}
```

5.3.1.3 Model performance

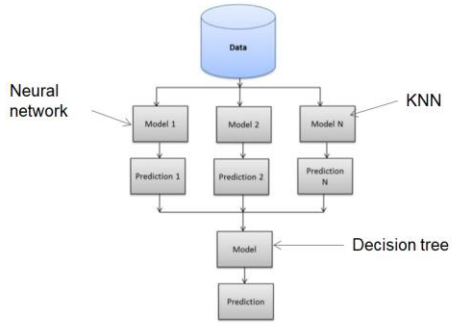
In the multiple classification model, we use many algorithms. The table below provides the performance.

Algorithms	Accuracy	Precision	Recall	F1 Score
Random forest	0.83	0.83	0.83	0.83
Decision tree	0.69	0.69	0.69	0.69
neural network	0.52	0.52	0.52	0.51
Support vector machines	0.50	0.49	0.50	0.49
k-nearest neighbours	0.52	0.52	0.52	0.51
Logistic regression	0.34	0.32	0.34	0.32
Naïve Bayes	0.33	0.32	0.32	0.30

Random forest has the best performance.

5.3.1.4 Multiple classification ensemble method

We performed the stacking method to get the better performance but the result is not ideal.



	precision	recall	f1-score	support
0	0.41	0.41	0.41	22273
1	0.45	0.45	0.45	22347
2	0.41	0.41	0.41	21841
3	0.51	0.51	0.51	22086
4	0.48	0.49	0.48	22062
accuracy			0.45	110609
macro avg	0.45	0.45	0.45	110609
weighted avg	0.45	0.45	0.45	110609

5.3.1.4 Interpretation

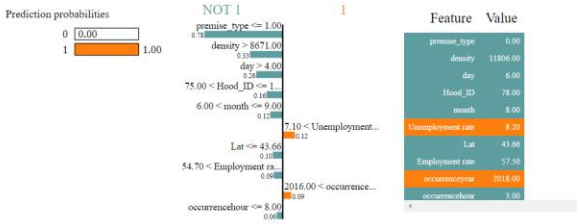
5.3.1.4.1 Feature importance

```
model = RandomForestClassifier() # XGBClassifier()
model.fit(X, y)
pd.DataFrame({'Variable':X.columns,
              'Importance':model.feature_importances_}).sort_values('Importance', ascending=False)
```

	Variable	Importance
5	Long	0.130183
2	occurrencedayofyear	0.128208
6	Lat	0.127950
1	occurrenceday	0.103347
3	occurrencehour	0.098030
17	premise_type	0.085107
15	month	0.077065
16	day	0.068593
0	occurrenceyear	0.062272
14	density	0.017185
4	Hood_ID	0.015886
12	Postsecondary rate	0.014035
13	income	0.012812
9	Unsuitable rate	0.012307
7	Employment rate	0.012188
11	Postsecondary	0.011826
8	Unemployment rate	0.011612
10	population	0.011393

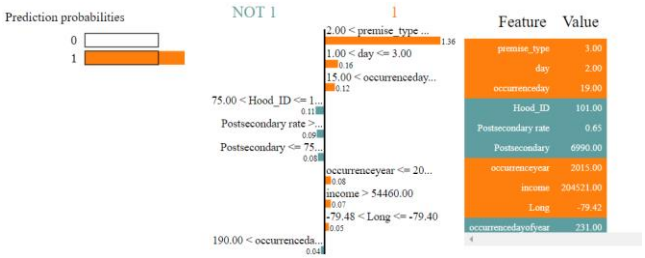
5.3.1.4.1 LIME for random forest

y_test.iloc[0] = 1(assault)



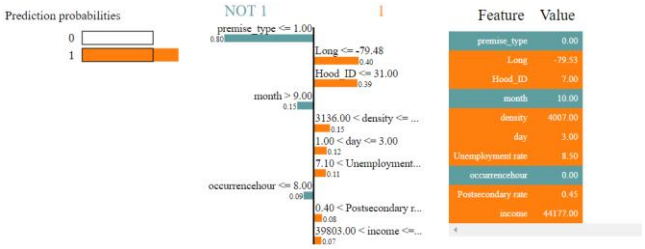
Model predicted whether the test case 0 is assault. Biggest effect is premise type; this has decreased the probability of assault significantly. Next, density also decreases the probability of assault. Unemployment increases the probability of assault.

y_test.iloc[1010] = 4(Auot theft)



Model predicted whether the test case 1010 is auto theft. Biggest effect is premise type; this has increased the probability of auto theft significantly. Next, day also increases the probability of assault. Hood_ID decreases the probability of auto theft.

y_test.iloc[5001] = 3(Theft over)



Model predicted whether the test case 5001 is theft over. Biggest effect is premise type; this has decreased the probability of theft over significantly. Next, Longitude increases the probability of theft over. Hood_ID increases the probability of theft over.

5.3.2 Binary classification

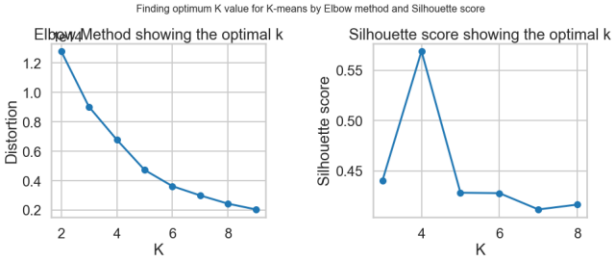
In the binary classification model, we use many algorithms. The table below provides the performance.

Algorithms	Accuracy	Precision	Recall	F1 Score
Random forest	0.83	0.83	0.83	0.83
Decision tree	0.77	0.77	0.77	0.77
neural network	0.71	0.71	0.71	0.71
Support vector machines	0.67	0.67	0.67	0.67
k-nearest neighbours	0.65	0.65	0.65	0.65
Logistic regression	0.57	0.58	0.57	0.57
Naïve Bayes	0.57	0.58	0.57	0.57

Random forest has the best performance.

5.3.4 clustering

For K means, selection of K is an important factor. For this project, Elbow method and Silhouette score is used to decide the optimum value of k.



Observations based on above graphs:

Using Elbow method, we can see that local optima can be found at K=4 or K=5.

Using Silhouette score, we got more confidence to select K=5 as the Silhouette score is highest at that point.

Hence, we will make 5 clusters in this process.

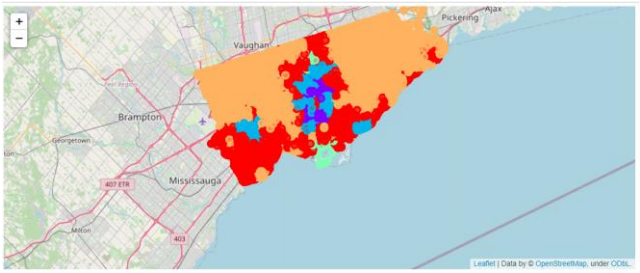


Figure cluster distribution

cluster	count	Employment rate	Unemploy rate	Unsuitable rate	Postsec Rate	income
0	67655	64.39	7.4297	8.5	0.5858	55763
1	3424	60.25	6.6257	3.73	0.65199	210656
2	11309	63.70	6.60	5.5974	0.6144	107292
3	9598	73.8	5.85	8.5594	0.75	65620
4	113335	55.2188	9.7214	16.9239	0.4088	33743

cluster	observations
0	This cluster has the second highest number of MCI cases. This cluster has the second highest unemployment rate. The cluster has the third highest unsuitable housing rate. This cluster has the second lowest postsecondary rate. This cluster has the second highest income.
1	This cluster has the least number of MCI cases. This cluster has the third lowest unemployment rate. The cluster has the lowest unsuitable housing rate. This cluster has the second highest postsecondary rate. This cluster has the highest income.
2	This cluster has the third highest number of MCI cases. This cluster has the third highest unemployment rate. The cluster has the fourth highest unsuitable housing rate. This cluster has the third highest postsecondary rate. This cluster has the second highest income.
3	This cluster has the second least number of MCI cases. This cluster has the lowest unemployment rate. The cluster has the third lowest unsuitable housing rate. This cluster has the highest postsecondary rate. This cluster has the third highest income.
4	This cluster has most number of MCI cases. This cluster has the highest unemployment rate. The cluster has the highest unsuitable housing rate. This cluster has the lowest postsecondary rate. This cluster has the lowest income.

6 Conclusion

In this project, we establish binary classification model and multiple classification model to predict the crime. Actually the crime occurrence depends on many factors and sometimes it's only random event. Our models have considerate accuracy. The spatial factors such as longitude and latitude, and the temporal factors such as occurrence day of year, occurrence day play the important role in the predicting crime. We can deploy more polices on the specific location and time to lower the crime rate.

We also use unsupervised algorithm to cluster. There are 5 clusters after the calculation. By analyzing the characteristics of every cluster, we can conclude that in the area with the higher income and better housing condition, there is less criminal cases and Higher post-secondary rate and lower unemployment help to decrease the criminal rate.

7 Reference

[1]**https://data.torontopolice.on.ca/datasets/mci-2014-to2019**
[2]**https://open.toronto.ca/dataset/neighbourhood-profiles/**
[3]**https://arxiv.org/pdf/1602.04938.pdf**