ML Model on Toronto Crime

Enje Godinez
Faculty of Business
Humber Institute of Technology and Advanced Learning
Toronto, Canada
n01510380@humbermail.ca

Pravin Anguraja
Faculty of Business
Humber Institute of Technology and Advanced Learning
Toronto, Canada
n01479757@humbermail.ca

Nguyen Minh Thu
Faculty of Business
Humber Institute of Technology and Advanced Learning
Toronto, Canada
n01511732@humbermail.ca

Prem Nishanth Navaneetha Krishnan
Faculty of Business
Humber Institute of Technology and Advanced Learning
Toronto, Canada
n01510436@humbermail.ca

Rahul Raghunandanan Nair
Faculty of Business
Humber Institute of Technology and Advanced Learning
Toronto, Canada
n01491273@humbermail.ca

***Abstract*—This case study focuses on the application of machine learning to predict the type of crime committed in a specific area using historical data from the Major Crime Indicators (MCI) dataset. The primary goal is to assist the police department in allocating resources and prioritizing activities.**

I. Introduction

Crime is a major concern for law enforcement agencies worldwide, and the ability to predict criminal activity is an essential component of effective policing. With the advent of machine learning and data analysis techniques, it has become possible to leverage vast amounts of data to develop predictive models for crime prevention and detection. In this case study, we will explore the application of machine learning tools and data preprocessing techniques to a real-world problem using a Major Crime Indicators (MCI) dataset.

The dataset includes all Major Crime Indicators (MCI) occurrences by reported date and related offenses in the City of Toronto[1]. Predicting the possible type of crimes committed in a particular area would help in the allocation of resources and planning of the police department. The case study will explore five machine learning algorithms - Linear Regression, K-Nearest Neighbors, Naive Bayes Classifier, Classification and Regression Trees, and Logistic Regression.

To achieve the objective of this case study, we will employ various data preprocessing techniques such as data cleaning, statistical analysis, knowledge base analysis, visualization, and model performance evaluation. We will also explore and identify the most effective algorithm and model.

II. Data Exploration and Preprocessing

The dataset is in a tabular form and includes chronological, geographical, and text data and contains incidents derived from the Toronto Police Open Data.

Data preparation and pre-processing are crucial steps in any data analysis project. In this case study, we started with an original dataset of 323,296 observations and 31 columns. This included all Major Crime Indicators (MCI) occurrences over five years. Before we can use the data to build machine learning models, we had to apply several data preparation and pre-processing techniques to obtain a clean and well-formatted dataset. One of the first steps in data preparation is to reduce the observation size. We did this by filtering the dataset to only include data within the year 2022. This reduced the original dataset to a size of 35863 observations and 31 columns.

A. Removal of unwanted columns

Based on our objectives and the dataset information, we identified columns to drop, and will not be used in our analysis. These columns included 'X', 'Y', 'OBJECTID', 'REPORT_DATE', 'REPORT_YEAR', 'REPORT_MONTH', 'REPORT_DAY', 'REPORT_DOY', 'REPORT_DOW', 'REPORT_HOUR', 'UCR_CODE', 'UCR_EXT', 'OFFENCE', 'HOOD_140', and 'NEIGHBOURHOOD_140'. The reasons for dropping these columns ranged from irrelevance to our analysis to redundancy with other columns.

B. Formatting and Transformation of columns

We dropped duplicates and checked for missing values in the dataset. Then formatted and transformed some columns in the dataset. For example, we converted categorical 'Month' (OCC_MONTH) and 'Day-of-week' (OCC_DOW) columns to their corresponding numeric values - OCC_MONTH_N, OCC_DOW_N. We also removed additional columns that we identified will not be used as inputs or predictors for our model. These included 'EVENT_UNIQUE_ID' and 'OCC_DATE'. Instead of 'OCC_DATE', we decided to use the breakdown values of Year, Day of Year, Month, Day of Week, Day, and Hour.

	OCC_DATE	OCC_MONTH	OCC_MONTH_N	OCC_DOW	OCC_DOW_N	OCC_YEAR	OCC_DAY	OCC_DOY	OCC_HOUR
0	2022-01-01 05:00:00+00:00	January	1	Saturday	5	2022	1	1	15
1	2022-01-01 05:00:00+00:00	January	1	Saturday	5	2022	1	1	14
2	2022-01-01 05:00:00+00:00	January	1	Saturday	5	2022	1	1	19
3	2022-01-01 05:00:00+00:00	January	1	Saturday	5	2022	1	1	1
4	2022-01-01 05:00:00+00:00	January	1	Saturday	5	2022	1	1	19

Fig. 1. Formatting and Transformation column

After formatting and transforming the dataset, the dimensions of the new dataset were (35863, 13). We then performed some summary statistics and visualizations to gain insights into the data. We created a table of occurrence of the target variable values (MCI_CATEGORY) which showed the count of each crime type in the dataset. The table shows that 'Assault' was the most common crime type and occurs almost the same times as the other types combined. It is followed by 'Auto Theft', 'Break and Enter', 'Robbery', and 'Theft Over'.

C. *Visualization*

1) *Bar Graph*

We have created a bar graph representing the occurrences of MCI Categories, this shows us which kind of crime has the highest number of occurrences.

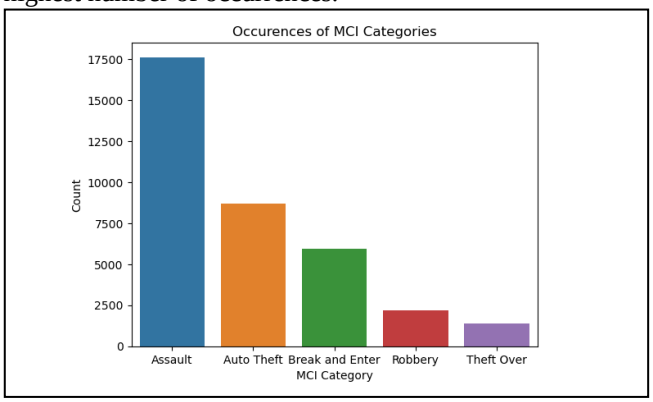

Fig. 2. Occurrences of crime bar plot

2) *Correlation Graph*

We have generated the Correlation matrix representation with the help of a heat map to identify the dependencies of the variables present in our dataset.

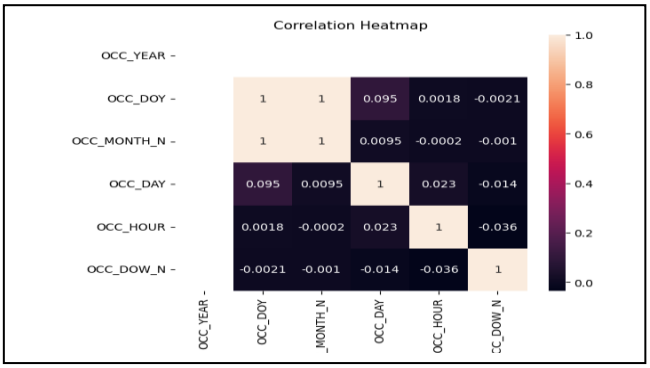

Fig. 3. Correlation heat map

From the matrix, we can conclude that the r-coefficient value is around 0, which represents that every independent variable is not correlated with each other.

3) *Line plot*

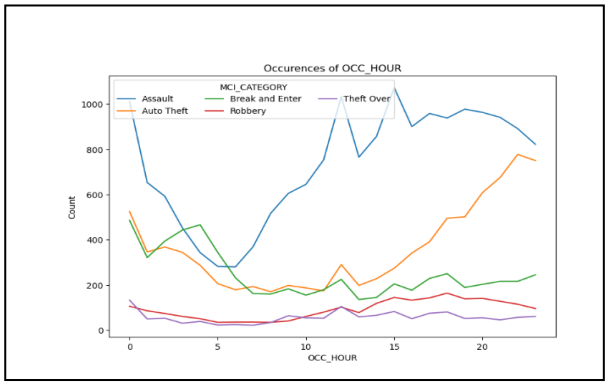

Fig. 4. Line plot Crime with respect to occurrence hour

We have generated the line plot to represent the occurrences of crime based on the hours. This gives us a good representation of different crime categories with respect to their occurrence hours. We can see that assault has higher precedence followed by Auto theft over the other crime categories.

4) *Finalizing predictors and the target variable*

Based on the visualization we have narrowed down our predictors and target variables which we are using to generate the ML model to work on the algorithms.

```
['OCC_YEAR',
 'OCC_DOY',
 'OCC_MONTH_N',
 'OCC_DAY',
 'OCC_HOUR',
 'OCC_DOW_N',
 'DIVISION',
 'LOCATION_TYPE',
 'PREMISES_TYPE',
 'HOOD_158',
 'MCI_CATEGORY']
```

Fig. 5. Predictors and Target variables

5) *Preparation of data for model building*

To prepare the dataset for model building, we converted categorical variables to numerical representations using two methods: 'get_dummies()' and 'LabelEncoder()'. We used the two methods to explore which one works best with the machine learning algorithms. In some algorithms, we also scaled the features using 'StandardScaler' to ensure that all variables were on the same scale and to improve the performance of the algorithms.

'get_dummies' is a method that converts categorical variables into numerical values by creating new binary columns for each unique category, with a value of 1 for the category and 0 for others.

'LabelEncoder' is a method that converts categorical variables into numerical values by assigning a unique integer value to each category, allowing algorithms to understand the relative importance of different categories.

III. ML ALGORITHMS

A. Linear Regression

Linear regression is a popular and widely used statistical algorithm used to model the relationship between a dependent variable (also known as the response or target variable) and one or more independent variables (also known as predictors or features). The goal of linear regression is to find the best-fitting straight line (i.e., a linear equation) that can predict the value of the dependent variable based on the values of the independent variables.

Initially, the linear regression model was developed without converting the significant amount of categorical variables in the dataset to numerical fields. The model's testing accuracy was only around 3%, which was unsatisfactory. Several attempts to improve the accuracy were made, such as feature selection, adjusting the test size, and tweaking the parameters of linear regression. Unfortunately, these attempts only resulted in a maximum accuracy of 4%.

After extensive research, we decided to convert the categorical variables to numerical variables using the 'LabelEncoder' [2]. We assigned the label encoder to the MCI category and premises type fields, taking into consideration the weight of the data distribution for each respective field.

While validating the trained data in our Linear Regression algorithm, we tried using the below mathematical function as a part of the classification summary code.

- **np.ceil() -** is the numpy function that returns the smallest integer greater than or equal to the input value. In this code, the predicted values are rounded up to the nearest integer using the ceil() function[5].

- **np.floor() -** is the numpy function that returns the largest integer less than or equal to the input value. In this code, the predicted values are rounded down to the nearest integer using the floor() function[6].

It is to be noted that after rounding and clipping the predicted values using ceil() or floor(), the classificationSummary() function is used to generate a confusion matrix for the test data, comparing the actual values (test_y) with the rounded and clipped predicted values (y_ceil_pred or y_floor_pred. The accuracy for the ceiling turned out to be 45.76%. While the accuracy for the floor function resulted be lesser compared to the ceiling function at 28.17%. From the results, we observe that the accuracy is greater by using the ceiling function as a part of our classification summary than the one used with the floor function. Hence, we choose the algorithm with the ceiling function.

TABLE I. LINEAR REGRESSION CONFUSION MATRIX

	Prediction				
	Assault	Break and Enter	Auto Theft	Robbery	Theft Over
Assault	0	0	10	172	81
Break and Enter	0	0	35	194	195
Auto Theft	0	0	123	1374	195
Robbery	0	0	18	654	532
Theft Over	3	0	20	1034	2458

B. KNN Algorithm

The k-nearest neighbors algorithm, sometimes referred to as KNN or k-NN, is a supervised learning classifier that employs proximity to producing classifications or predictions about the grouping of a single data point. Although it can be applied to classification or regression issues, it is commonly employed as a classification algorithm because it relies on the idea that comparable points can be discovered close to one another[3].

To find the best K value we ran a loop from k=5 to k=200 and we got the optimal value of K = 57. Using K= 57 we proceeded with the KNN Algorithm, we segregated the data as 60% training data set and 40% as the validation data set. We got an accuracy for validation of 63.51% with the help of the confusion matrix.

TABLE II. KNN CONFUSION MATRIX

	Prediction				
	Assault	Break and Enter	Auto Theft	Robbery	Theft Over
Assault	2996	426	91	0	0
Break and Enter	486	1174	32	0	0
Auto Theft	702	166	336	0	0
Robbery	347	66	10	1	0
Theft Over	168	63	32	0	0

C. Naïve Bayes Classifier

The supervised machine learning algorithm Nave Bayes classifier is employed for classification applications, including text classification. It also belongs to the family of generative learning algorithms, which model the input distribution of a certain class or category. Given that it is based on Bayes' Theorem, Nave Bayes is frequently referred to as a probabilistic classifier [8]. The family of straightforward "probabilistic classifiers" known as "naive

Bayes classifiers" in statistics is based on the application of Bayes' theorem with strong (naive) independence assumptions between the features [9]. Based on the algorithm, we forecasted those crime categories' probability distributions, given that features (OCC_YEAR', 'OCC_DOY', 'OCC_MONTH_N', 'OCC_DAY', 'OCC_HOUR', 'OCC_DOW_N') has occurred. The model's testing accuracy of validation data is 48.68%.

TABLE III. NAÏVE BAYES CONFUSION MATRIX

	Prediction				
	Assault	Break and Enter	Auto Theft	Robbery	Theft Over
Assault	1847	933	669	41	23
Break and Enter	445	933	271	38	5
Auto Theft	243	287	627	5	42
Robbery	199	110	66	38	11
Theft Over	76	78	97	3	9

D. Classification and Regression Trees (CART)

A decision tree is a non-parametric supervised learning algorithm, that can be used for both classification and regression tasks and models. It has a hierarchical, tree structure, which consists of different nodes such as root nodes, branches, internal nodes, and leaf nodes. The goal of a Classification or decision tree is to segregate the data into groups that are more similar and comparable to one another and include less unpredictability and impurity. Every split in the decision tree must lower randomness to do this[7]. Pre-proceed unnormalized data was used for the analysis and the scatter plot was created for the categories of crime, time(hours), and Premises (Numerical). Recursive partition was made in both the axis (hours and premises) to achieve non-overlapping multidimensional rectangles, but we cannot see the exact segregation because of the large dataset. However, the decision tree that was generated shows us what kind of crime took place with respect to the occurrence hour and premises type with achievable purity with the tree depth of 15. Moreover, the Classification tree algorithm was created with 43.3% accuracy for the validation dataset using the confusion matrix.

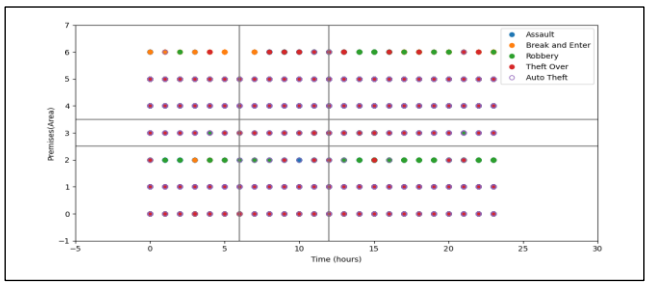

Fig. 6. Scatter plot for CART with Split

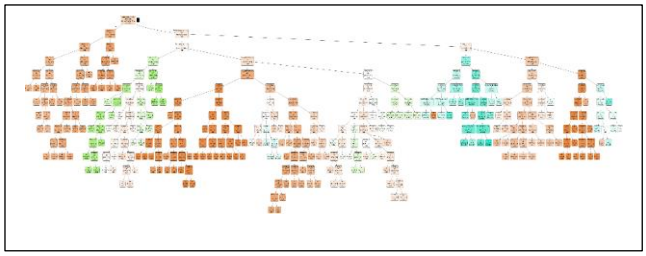

Fig. 7. Full-size Decision tree

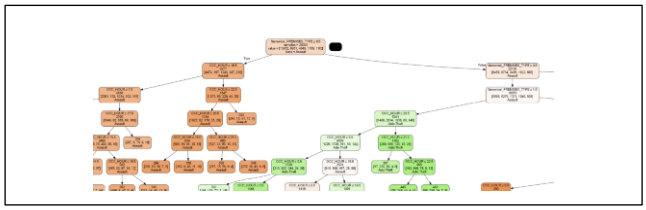

Fig. 8. Head Decision tree

TABLE IV. CART CONFUSION MATRIX

	Prediction				
	Assault	Break and Enter	Auto Theft	Robbery	Theft Over
Assault	2921	418	174	0	0
Break and Enter	1586	97	9	0	0
Auto Theft	626	522	56	0	0
Robbery	333	83	8	0	0
Theft Over	190	66	7	0	0

E. Logistic Regression

Logistic regression is a popular classification algorithm that is widely used in machine learning. It models the probability of a binary target variable given a set of input features. The goal is to find a linear relationship between the input features and the target variable by minimizing the logistic loss function. The resulting model can be used to predict the probability of the target variable taking a particular value[4].

Based on the exploration, the best-performing logistic regression model achieved an accuracy of 0.6391, with a set of parameters that included a solver of 'liblinear', a penalty of 'l2', max iterations of 1500, and a random state of 42. The model was tested with around 10 variations of the parameters.

The assault had the highest accuracy rate at 0.8181, while Theft Over had an accuracy rate of 0.00, indicating that the model was unable to predict this category effectively. This result suggests that further exploration may be required to improve the model's performance in this category.

TABLE V. LOGISTIC REGRESSION CONFUSION MATRIX

	Prediction				
	Assault	Break and Enter	Auto Theft	Robbery	Theft Over
Assault	2874	475	151	13	0
Break and Enter	412	1226	54	0	0
Auto Theft	577	221	392	14	0
Robbery	290	62	29	43	0
Theft Over	125	78	57	3	0

IV. CONCLUSION

We did our finest to analyze the crime pattern and its variations using the machine learning model we constructed. We employed a variety of algorithms that might aid the Toronto Police in reducing crime in the future. According to our machine learning model, Logistic regression accuracy is 63.91%, followed by K-NN at 63.51%, Naive Bayes at 48.68%, Linear Regression at 45.59%, and Classification Tree at 43.32%. Based on the confusion matrices for each algorithm, we can conclude that the model we have developed can be used primarily to study only the Assault Category when compared to others because it has high accuracy. We can explore more algorithms, such as neural networks or deep learning, to drill down into further categories to move forward with the analysis.

REFERENCES

[1] Major Crime Indicators Open Data. (n.d.). Data.torontopolice.on.ca. Retrieved April 12, 2023, from https://data.torontopolice.on.ca/datasets/TorontoPS::major-crime-indicators-open-data/about

[2] sklearn.preprocessing.LabelEncoder — scikit-learn 0.22.1 documentation. (2019). Scikit-Learn.org. https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.LabelEncoder.html

[3] IBM. (n.d.). What is the k-nearest neighbors algorithm? | IBM. Www.ibm.com. https://www.ibm.com/topics/knn#:~:text=The%20k%2Dnearest%20neighbors%20algorithm%2C%20also%20known%20as%20KNN%20or

[4] sklearn.linear_model.LogisticRegression — scikit-learn 0.21.2 documentation. (2014). Scikit-Learn.org. https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html

[5] numpy.ceil — NumPy v1.24 Manual. (n.d.). Numpy.org. https://numpy.org/doc/stable/reference/generated/numpy.ceil.html

[6] numpy.floor — NumPy v1.24 Manual. (n.d.). Numpy.org. https://numpy.org/doc/stable/reference/generated/numpy.floor.html

[7] Gunjal, S. (2020, June 10). Decision Tree. Quality Tech Tutorials. https://satishgunjal.com/decision_tree/#:~:text=The%20objective%20of%2 0decision%20tree

[8] What is Naïve Bayes | IBM. (n.d.). Www.ibm.com. https://www.ibm.com/topics/naive-bayes#:~:text=The%20Na%C3%AFve%20Bayes%20classifier%20is

[9] *Naive Bayes classifier*. (2023, March 3). Wikipedia. https://en.wikipedia.org/wiki/Naive_Bayes_classifier#:~:text=In%20statistics%2C%20naive%20Bayes%20classifiers