Unlocking Toronto's Crime Trends: - A Journey through Data Science Capstone Project

Note: This report assumes you're comfortable with basic data analytics concepts such as collecting data, cleaning it, doing Exploratory Data Analysis (EDA), and using machine learning models for predictions. We've added links to delve deeper into these topics as you go through the report.

In today's data-driven world, the field of data science has emerged as a promising avenue for individuals keen on unraveling insights from vast amounts of data. Aspiring data scientists often find themselves navigating through a plethora of online resources, each emphasizing the importance of mastering fundamental skills and the value of hands-on experience through capstone projects.

While there's no shortage of tutorials and courses offering guidance on data cleaning, visualization, model building, and inference making, many beginners struggle to connect the dots and understand how to apply these skills in real-world scenarios. This is where capstone projects play a pivotal role, serving as a bridge between theoretical knowledge and practical application.

This report aims to demystify the process of tackling a capstone project focused on analyzing crime rates, particularly in a vibrant city like Toronto, Canada. Through this example, beginners can gain valuable insights into how to approach their own capstone projects in the future.

Our analytical journey of a crime rate analysis project divided into nine key steps, each playing a crucial role in unraveling the insights hidden within the data:

1. Define the Business Problem and Identify Stakeholders
2. Identify Data Sources/Collect Relevant Data
3. Clean/ Prepare data for Analysis.
4. Exploratory Data Analysis (EDA)
5. Feature Engineering/Data Preprocessing
6. Data Modeling
7. Interpretation
8. Model Deployment
9. Conclusion and next steps

Define the Business Problem and Identify Stakeholders:

In our case, defining the business problem involves understanding and addressing the dynamics of crime trends within the city. This includes identifying key stakeholders impacted by crime, such as law enforcement agencies, policymakers, community organizations, and residents. The business problem revolves around comprehensively analyzing crime data to identify patterns, hotspots, and factors influencing crime rates. Stakeholders are crucial in guiding the direction of the analysis, interpreting findings, and implementing data-driven strategies to enhance public safety and security in Toronto.

Identify Data Sources/Collect Relevant Data:

The most important step in any data science endeavor is obtaining the data. As we see many cases in the news over the past years highlighting that international students are being targeted for fatal shootings (ref. ***The Times of India***) or for violent crimes for unknown reasons and found dead without knowing the cause (ref. ***CBC***)These factors realizes that it is important for a comprehensive analysis of crime trends involving different types of crime incidents. It would be essential to gather information from reliable sources and provide the appropriate valuable insights for the investigation agencies.

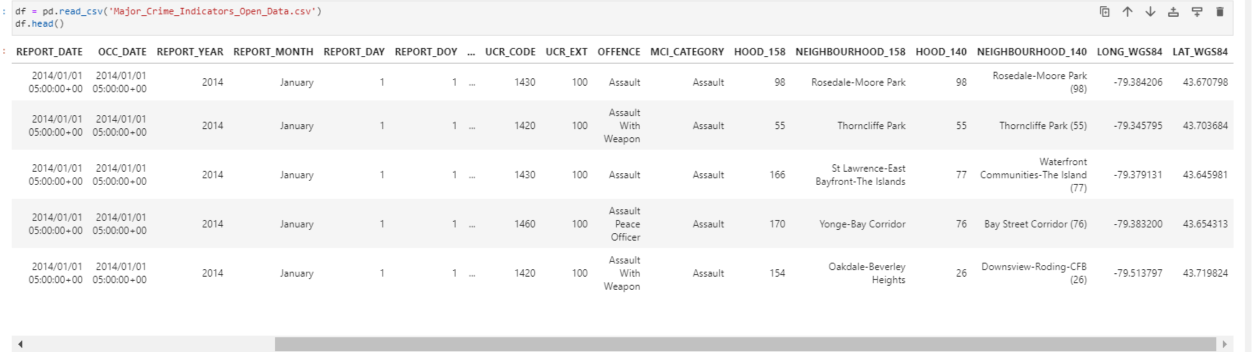
Due to this reason, we selected this topic which is the pervasive nature of crime as a social issue that has profound effects on individuals and communities globally. We delve into the ***Major Crime Indicators*** dataset, which is provided by the Toronto Police Service and can be accessed through the Toronto Public Safety Portal. The portal aims to promote transparency and public awareness regarding crime in the city, ensuring its reliability and accuracy. Toronto Police Service provides open analytics to aid in visualizing and understanding police information. These interactive visualizations provide trend analysis and important information briefly.

The dataset includes various variables that provide information about each crime incident such as date and related offences, categories include **Assault**, **Break and Enter**, **Auto Theft**, **Robbery** and **Theft Over**, Premises type, location, neighborhood, latitude, and longitude, etc.

Data Field Description:

Field	Field Name	Description
1	EVENT_UNIQUE_ID	Offence Number
2	REPORT_DATE	Date Offence was Reported
3	OCC_DATE	Date Offence Occurred
4	REPORT_YEAR	Year Offence was Reported
5	REPORT_MONTH	Month Offence was Reported
6	REPORT_DAY	Day of the Month Offence was Reported
7	REPORT_DOY	Day of the Year Offence was Reported
8	REPORT_DOW	Day of the Week Offence was Reported
9	REPORT_HOUR	Hour Offence was Reported
10	OCC_YEAR	Year (Toronto Police, n.d.)Offence Occurred
11	OCC_MONTH	Month Offence Occurred
12	OCC_DAY	Day of the Month Offence Occurred
13	OCC_DOY	Day of the Year Offence Occurred
14	OCC_DOW	Day of the Week Offence Occurred
15	OCC_HOUR	Hour Offence Occurred
16	DIVISION	Police Division where Offence Occurred
17	LOCATION_TYPE	Location Type of Offence
18	PREMISES_TYPE	Premises Type of Offence
19	UCR_CODE	UCR Code for Offence
20	UCR_EXT	UCR Extension for Offence
21	OFFENCE	Title of Offence
22	MCI_CATEGORY	MCI Category of Occurrence
23	HOOD_158	Identifier of Neighbourhood using City of Toronto's new 158 neighbourhood structure
24	NEIGHBOURHOOD_158	Name of Neighbourhood using City of Toronto's new 158 neighbourhood structure
25	HOOD_140	Identifier of Neighbourhood using City of Toronto's old 140 neighbourhood structure
26	NEIGHBOURHOOD_140	Name of Neighbourhood using City of Toronto's old 140 neighbourhood structure
27	LONG_WGS84	Longitude Coordinates (Offset to nearest intersection)
28	LAT_WGS84	Latitude Coordinates (Offset to nearest intersection)

Here's an overview of the dataset. We aim to analyze this data to identify which category experiences the highest crime rates and to uncover any seasonal patterns over time.



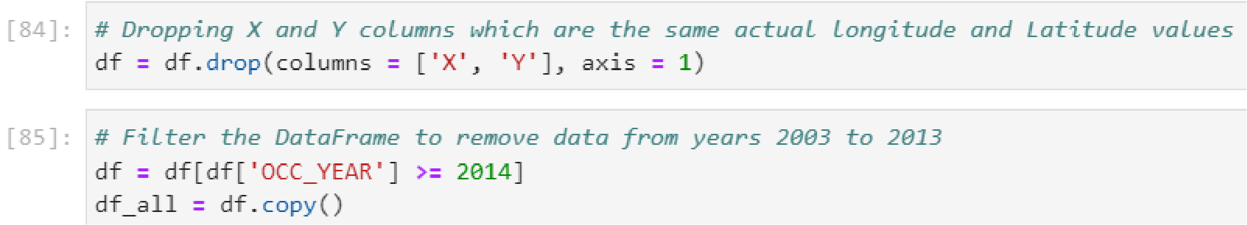
```
df = pd.read_csv('Major_Crime_Indicators_Open_Data.csv')
df.head()
```

	REPORT_DATE	OCC_DATE	REPORT_YEAR	REPORT_MONTH	REPORT_DAY	REPORT_DOY	...	UCR_CODE	UCR_EXT	OFFENCE	MCI_CATEGORY	HOOD_158	NEIGHBOURHOOD_158	HOOD_140	NEIGHBOURHOOD_140	LONG_WGS84	LAT_WGS84
	2014/01/01 05:00:00+00	2014/01/01 05:00:00+00	2014	January	1	1	...	1430	100	Assault	Assault	98	Rosedale-Moore Park	98	Rosedale-Moore Park (98)	-79.384206	43.670798
	2014/01/01 05:00:00+00	2014/01/01 05:00:00+00	2014	January	1	1	...	1420	100	Assault With Weapon	Assault	55	Thorncliffe Park	55	Thorncliffe Park (55)	-79.345795	43.703684
	2014/01/01 05:00:00+00	2014/01/01 05:00:00+00	2014	January	1	1	...	1430	100	Assault	Assault	166	St Lawrence-East Bayfront-The Islands	77	Waterfront Communities-The Island (77)	-79.379131	43.645981
	2014/01/01 05:00:00+00	2014/01/01 05:00:00+00	2014	January	1	1	...	1460	100	Assault Peace Officer	Assault	170	Yonge-Bay Corridor	76	Bay Street Corridor (76)	-79.383200	43.654313
	2014/01/01 05:00:00+00	2014/01/01 05:00:00+00	2014	January	1	1	...	1420	100	Assault With Weapon	Assault	154	Oakdale-Beverley Heights	26	Downsview-Roding-CFB (26)	-79.513797	43.719824

Clean/ Prepare data for Analysis:

With the dataset in hand, our next challenge is to clean and prepare the data for analysis. The quality of data analysis is directly influenced by the cleanliness of the data. Therefore, it's essential to approach this step with meticulous care that sets the foundation for meaningful insights. Cleaning the data involves various tasks such as checking for null values, handling missing data, identifying, and addressing outliers, and ensuring consistent column names. While this list isn't exhaustive and each dataset may present unique challenges during cleaning, it's imperative to address any discrepancies to ensure accurate analysis. Fortunately, the major crime indicators dataset we've selected for this example is relatively clean, with only minimal missing values that can be readily addressed for improved analysis.

In this step, we addressed missing values by removing them from the dataset. Additionally, we eliminated redundant columns, such as the X and Y columns, which duplicated information already present in the Longitude and Latitude columns. Furthermore, we filtered the dataset to include records only from the years 2014 to 2023, as there were very few records available from 2003 to 2013. This focused our analysis on the most relevant and recent data.



```
[84]: # Dropping X and Y columns which are the same actual longitude and Latitude values
      df = df.drop(columns = ['X', 'Y'], axis = 1)
```

```
[85]: # Filter the DataFrame to remove data from years 2003 to 2013
      df = df[df['OCC_YEAR'] >= 2014]
      df_all = df.copy()
```

Exploratory Data Analysis (EDA):

Once the data is cleaned and prepped, it's time to dive into exploratory data analysis (EDA). EDA is like peering through a magnifying glass, uncovering hidden patterns, correlations, and anomalies lurking within the data. Visualizations such as heatmaps, histograms, and time series plots breathe life into the numbers, providing valuable insights into crime hotspots, seasonal trends, and demographic influences.

The plots generated from the dataset offer comprehensive visualizations that provide deep insights into various aspects of the data. These visualizations include histograms, scatter plots, line charts, and more, each offering unique perspectives on different attributes and their relationships. By visually exploring the data, analysts can identify trends, patterns, outliers, and distributions, facilitating a better understanding of the underlying data structure.

The graph depicted in ***Figure. 1*** illustrates the trend of crime events over the period from 2014 to 2023. It shows that the overall trend has increased over time, with some fluctuations observed along the way.

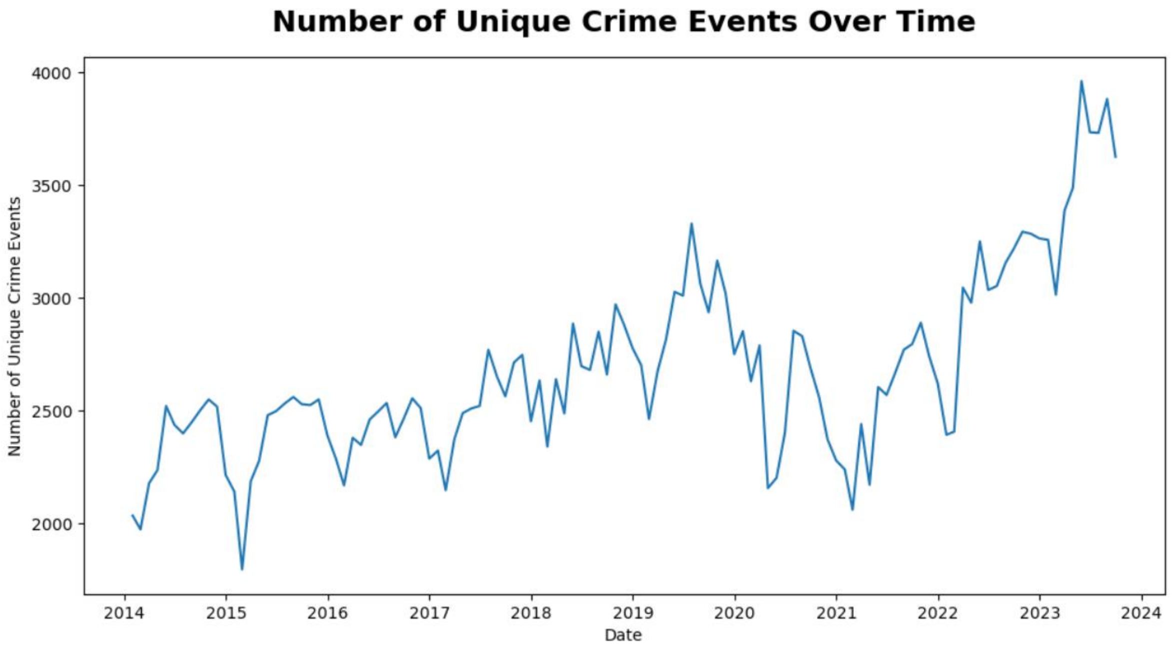


Figure. 1 Distribution of Crime Events over Time

From this observation, we can infer that there has been a general upward trend in the frequency of crime events over the years. This suggests a potential worsening of the crime situation in Toronto over time. However, the presence of fluctuations indicates that the trend is not entirely linear and may be influenced by various factors such as changes in law enforcement strategies, socio-economic conditions, or seasonal patterns.

Below ***Figure.2*** bar graph distribution illustrates the frequency of different types of major crimes reported in Toronto area. The categories are as follows:

- **Assault:** This category has the highest frequency, with 190,630 reported incidents.
- **Break and Enter:** This category represents incidents involving breaking into and entering premises, with 67,789 reported incidents.
- **Auto Theft:** This category involves the theft of automobiles and has 55,283 reported incidents.
- **Robbery:** This category represents incidents involving theft or attempted theft with the use or threat of violence, with 33,016 reported incidents.
- **Theft Over:** This category includes incidents involving theft of items valued over a certain threshold and has 11,945 reported incidents.

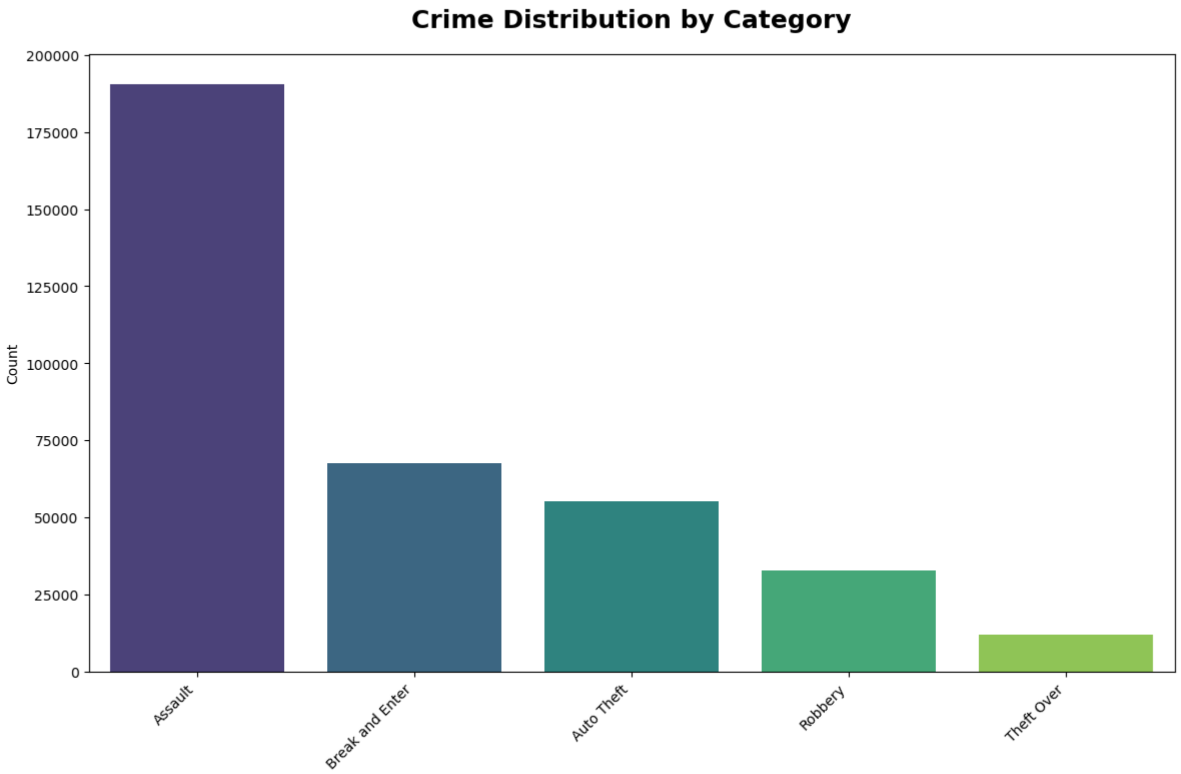


Figure. 2 Distribution of Major Crime categories

From this observation, it’s evident that **assault** is the most frequently reported type of major crime, followed by break and enter, auto theft, robbery, and theft over. This information provides valuable insights into the prevalence of different types of crimes and can help inform law enforcement strategies such as increased patrols, community engagement, and resource allocation to address prevalent types of crimes like assault, break and enter, auto theft, robbery, and theft over. These efforts aim to deter criminal activities, enhance investigation and prosecution, and improve public safety overall.

We have another factor which is Police divisions in Toronto Area, below ***Figure. 3*** illustrates that crime events occurred in each division wise and can be identify which division is highest crime rate, based on this analysis, police force can implement strategies focusing higher crime divisions to reduce the crime rate overall.

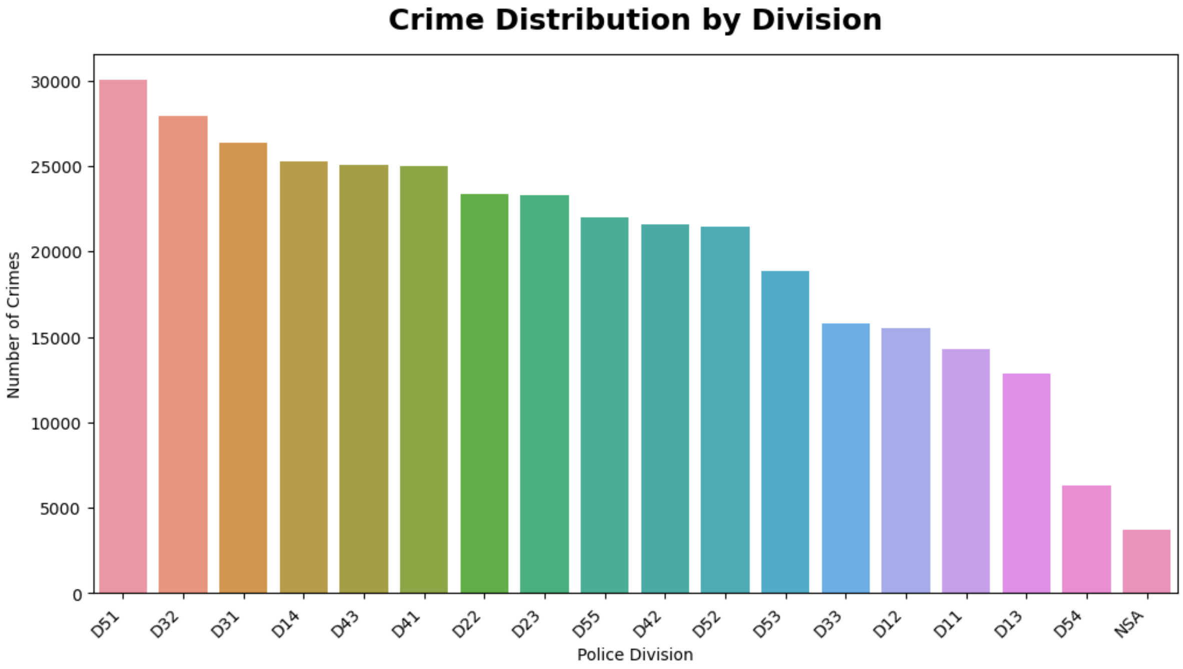


Figure. 3 Crime Distribution by Police Divisions

From this analysis of crime incidents by division, we observe that certain police divisions have significantly higher numbers of reported crimes compared to others. For instance, divisions D51, D32, D31, and D14 have the highest number of reported crimes, while divisions D54 and NSA has the lowest. To address this imbalance and ensure effective policing across all divisions, law enforcement agencies can take several actions like Resource Allocation, Targeted Interventions, Community Engagement, Data-Driven Policing.

The below ***Figure. 4*** illustrates the distribution of crime events based on the hour of the day provides valuable insights into the temporal patterns of criminal activity. From the data, it's evident that there are fluctuations in crime occurrence throughout the 24-hour period.

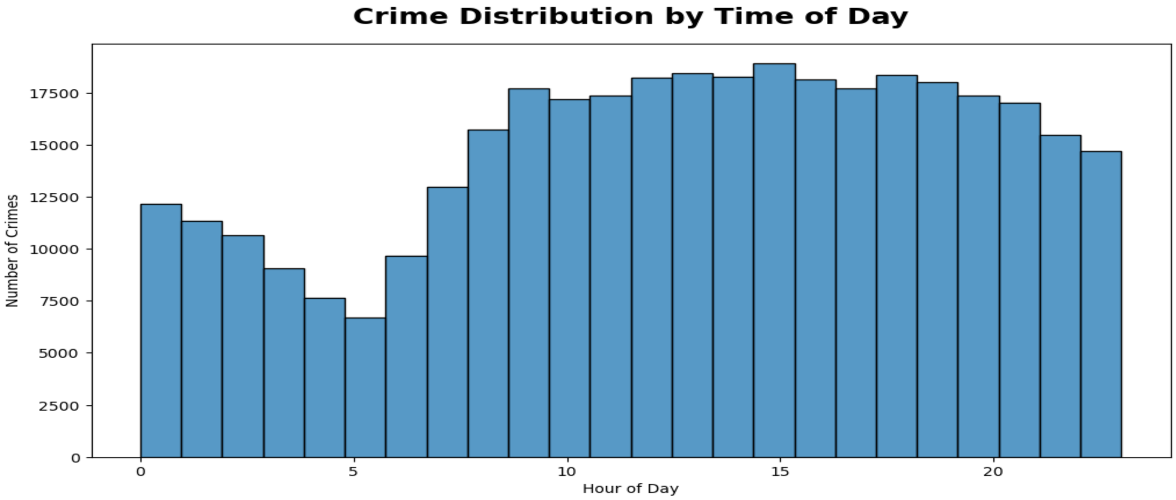


Figure. 4 Crime events distribution by Hour of the Day

The distribution of crime events by hour reveals peak times, like 3:00 PM and 1:00 PM, and off-peak times, such as 4:00 AM and 5:00 AM. Analyzing this distribution allows law enforcement agencies to discern peak hours of criminal activity, which are typically characterized by higher frequencies of reported incidents. By identifying these peak hours, authorities can strategically allocate resources such as police patrols, surveillance efforts, and response teams to areas with higher crime rates during specific time intervals. This proactive approach enables law enforcement to enhance public safety and security by effectively deterring criminal behavior and swiftly responding to incidents.

Moreover, understanding the hourly distribution of crime events facilitates the optimization of police shift scheduling. Law enforcement agencies can adjust officer deployment and shift rotations to ensure adequate coverage during periods of heightened criminal activity. By aligning police presence with the temporal patterns of crime, authorities can maximize their effectiveness in crime prevention and control efforts.

The below ***Figure. 5*** illustrates the crime rates of neighborhoods according to the City of Toronto's old 140 neighborhood structure, categorized by major crime types including Assault, Robbery, Break and Enter, Theft Over, and Auto Theft.

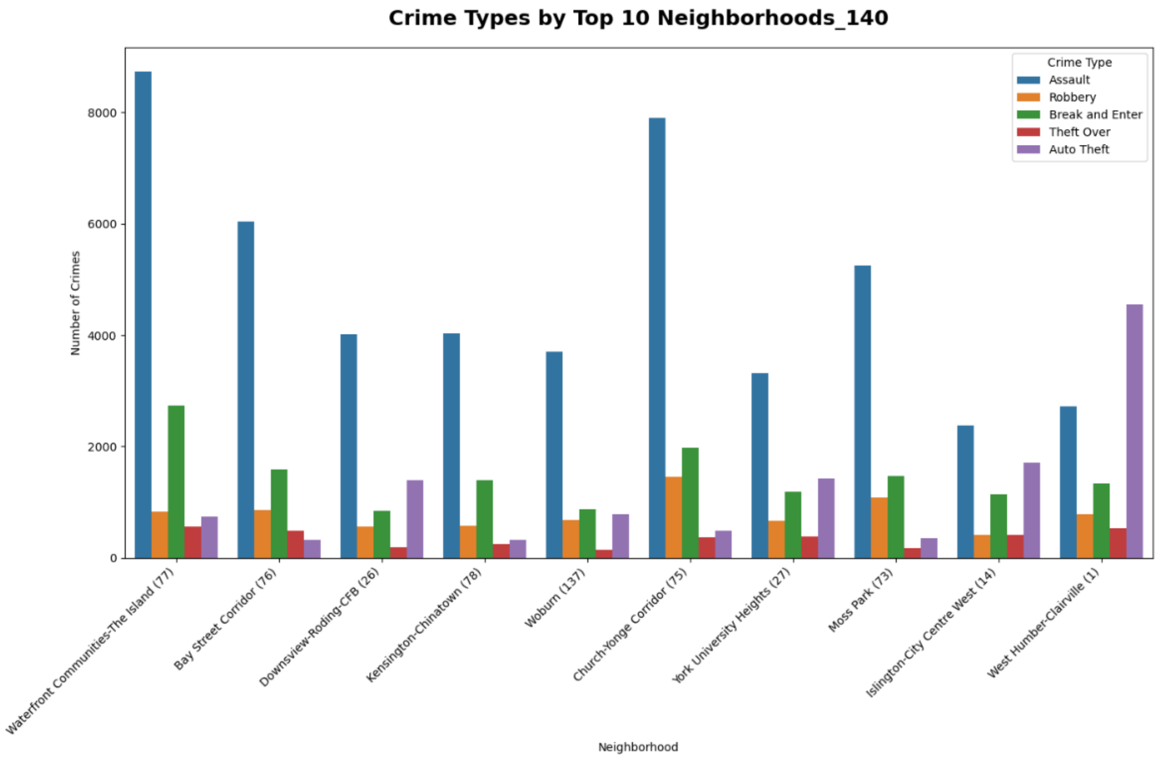


Figure. 5 Crime Rate by Top 10 Toronto's old 140 Neighborhood Structure

Based on the analysis, it's evident that Assault remains the dominant category even within the old Neighborhood 140 structure. Interestingly, among the top 10 neighborhoods, **West Humber – Clairsville (1)** stands out with the highest crime rate specifically in the Auto Theft category. This highlights the localized nature of certain crime types within the broader neighborhood context.

The distribution depicted in ***Figure. 6*** highlights that the majority of reported crimes took place outdoors, with apartments, commercial establishments, and houses following closely. This implies a higher prevalence of crimes in outdoor areas and residential settings compared to other premises types. Additionally, the "Other" category encompasses a notable number of incidents, likely representing a diverse range of locations not explicitly specified in the records. Furthermore, transit and educational premises experienced relatively fewer crime incidents, indicating lower crime rates in these environments.

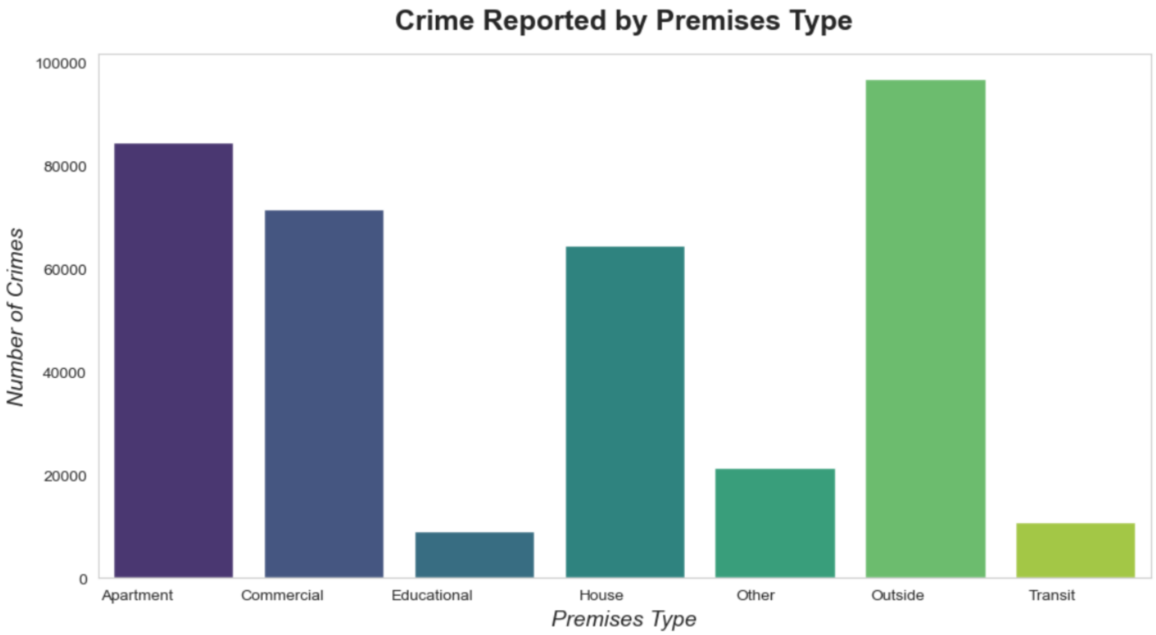


Figure. 6 Crime Distribution by Premises Type

From this analysis, we can infer that efforts to reduce crime rates should prioritize outdoor areas, apartments, commercial spaces, and residential buildings, where incidents are more frequent. Additionally, understanding the nature of crimes reported in the "Other" category could provide valuable insights into addressing less conventional crime locations.

From ***Figure. 7***, it's evident that the count of reported crimes varies across different days of the week. While there isn't a significant difference in the overall crime rate throughout the week, Fridays see the highest number of reported crimes, followed closely by Saturdays and Sundays. Thursdays, Wednesdays, and Mondays also exhibit considerable crime rates, albeit slightly lower than Fridays. Interestingly, Tuesdays consistently have the lowest count of reported crimes throughout the week.

This distribution provides valuable insights into the temporal patterns of criminal activities. The higher frequency of crimes on weekends, particularly Fridays and Saturdays, may be attributed to increased social activities and gatherings during these days, providing more opportunities for criminal incidents to occur. Conversely, the relatively lower crime rates on Tuesdays may be due to decreased social activities and stricter enforcement measures during weekdays.

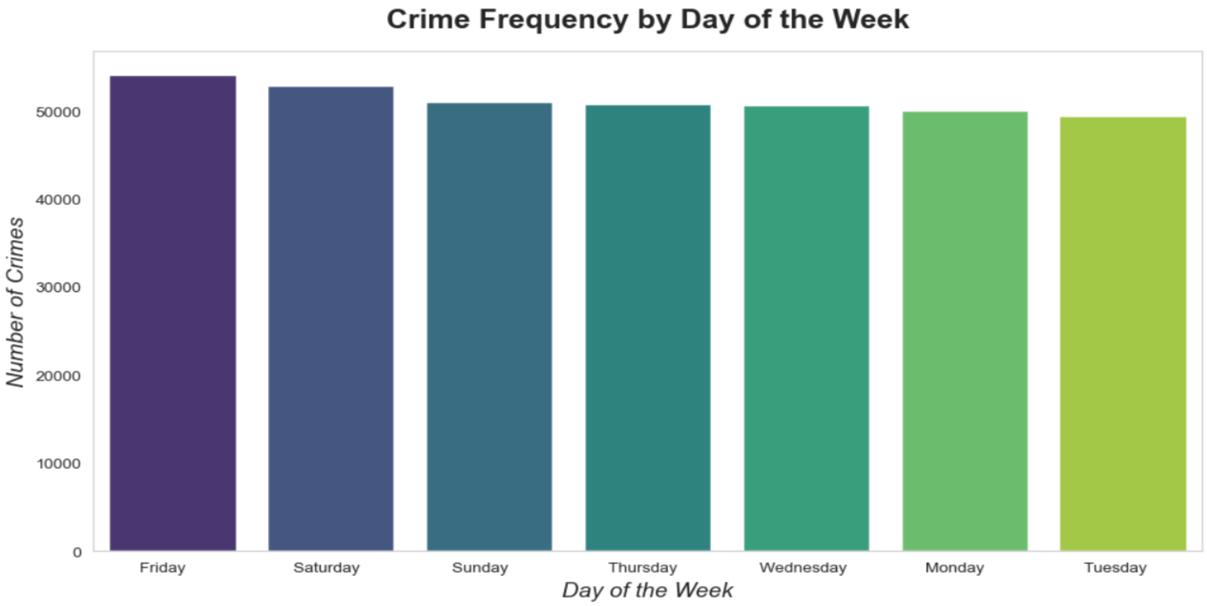


Figure. 7 Crime Rate by Day of the Week

Understanding these temporal patterns can inform law enforcement agencies and policymakers in allocating resources and implementing targeted interventions to address crime hotspots and mitigate criminal activities during peak periods. Additionally, businesses and community organizations can use this information to enhance security measures and ensure the safety of their premises and constituents, especially during high-risk periods.

Feature Engineering/Data Preprocessing:

After conducting exploratory data analysis (EDA), which involves understanding the data, identifying patterns, and gaining insights, the next step is feature engineering. Feature engineering is the process of creating new features or modifying existing ones based on the insights gained during EDA.

During EDA, we have discovered relationships between variables, identified important patterns, or uncovered hidden structures in the data. Feature engineering leverages this knowledge to create new or modify existing features that capture these relationships or patterns more effectively, making them easier for machine learning algorithms to understand and learn from.

In this step, we've employed various encoding techniques to convert categorical columns into numerical ones, essential for machine learning algorithms to process. For instance, we've mapped month names to numerical values (e.g., January: 1, February: 2, etc.) for both the **'REPORT_MONTH'** and **'OCC_MONTH'** columns using a predefined mapping dictionary.

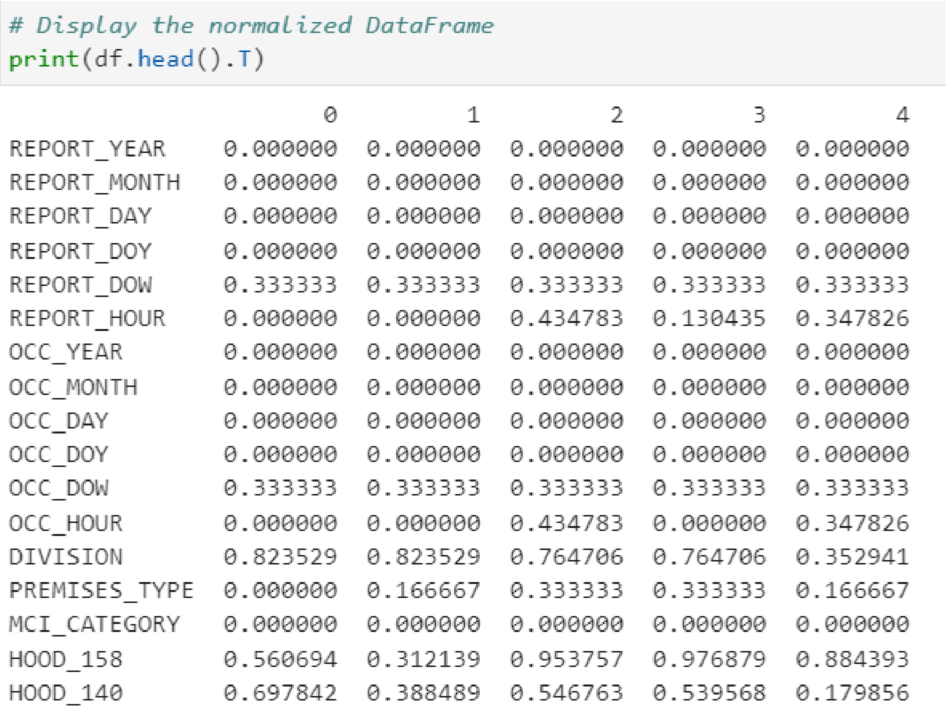
Additionally, we've converted other categorical columns such as **'OCC_DOW'** and **'REPORT_DOW'** to numerical values (e.g., Monday: 1, Tuesday: 2, etc.) to represent the seven days of the week. **'DIVISION'** column encoded with label encoder making DIVISION names each into numerical representation, while **'PREMISES_TYPE'** was mapped each premises type with numerical mapping from 0. **'MCI_CATEGORY'**, one of our target variables, was converted to numerical values, starting from 0 for 'Assault'. Rows with 'NSA' values, likely indicating non-stated areas or missing data, were removed from **'HOOD_158'** and

'HOOD_140'. These columns were then converted to numeric using pd.to_numeric (). **'REPORT_YEAR'** and **'OCC_YEAR'** were ordinal encoded to maintain chronological order.

Moreover, certain features deemed irrelevant or redundant were dropped from the dataset. For instance, the **'OFFENCE'** column was removed as it duplicated information already present in the 'MCI_CATEGORY' column. Similarly, the **'LOCATION_TYPE'** column was dropped as it duplicated information with the **'PREMISES_TYPE'** column. This pruning process streamlined the dataset, making it more manageable and focused for subsequent analysis.

After converting all categorical columns to numerical ones, we proceeded to scale the data for all columns except the target variable, MCI_CATEGORY. Minmax scaling is applied to ensure that all numerical features were on a similar scale, typically between 0 and 1. Normalizing the features ensures that each feature is treated equally during the modeling process. Without normalization, features with larger magnitudes could dominate the model's learning process, leading to biased results.

The transformed dataset, after undergoing data preprocessing, feature engineering, and normalization, is shown below ***Figure. 8***, displaying the first five rows:

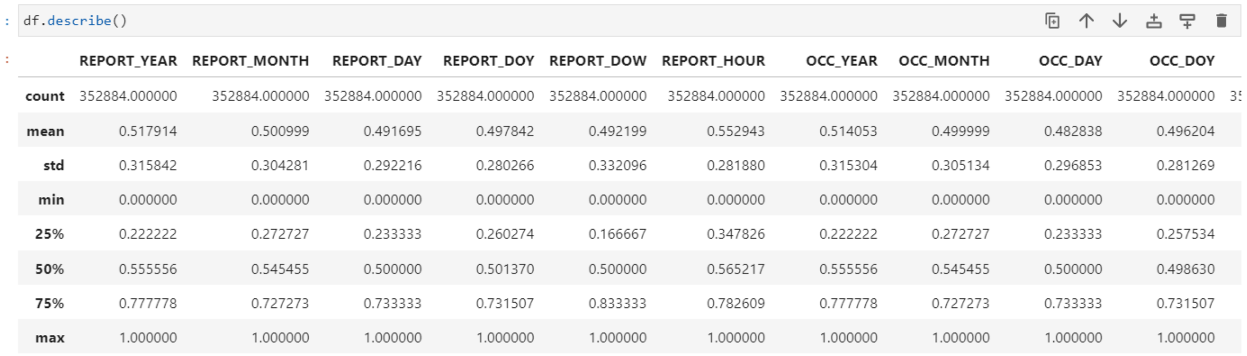


```
# Display the normalized DataFrame
print(df.head().T)
```

```
                      0         1         2         3         4
REPORT_YEAR    0.000000  0.000000  0.000000  0.000000  0.000000
REPORT_MONTH   0.000000  0.000000  0.000000  0.000000  0.000000
REPORT_DAY     0.000000  0.000000  0.000000  0.000000  0.000000
REPORT_DOY     0.000000  0.000000  0.000000  0.000000  0.000000
REPORT_DOW     0.333333  0.333333  0.333333  0.333333  0.333333
REPORT_HOUR    0.000000  0.000000  0.434783  0.130435  0.347826
OCC_YEAR       0.000000  0.000000  0.000000  0.000000  0.000000
OCC_MONTH      0.000000  0.000000  0.000000  0.000000  0.000000
OCC_DAY        0.000000  0.000000  0.000000  0.000000  0.000000
OCC_DOY        0.000000  0.000000  0.000000  0.000000  0.000000
OCC_DOW        0.333333  0.333333  0.333333  0.333333  0.333333
OCC_HOUR       0.000000  0.000000  0.434783  0.000000  0.347826
DIVISION       0.823529  0.823529  0.764706  0.764706  0.352941
PREMISES_TYPE  0.000000  0.166667  0.333333  0.333333  0.166667
MCI_CATEGORY   0.000000  0.000000  0.000000  0.000000  0.000000
HOOD_158       0.560694  0.312139  0.953757  0.976879  0.884393
HOOD_140       0.697842  0.388489  0.546763  0.539568  0.179856
```

Figure. 8 Normalized Data Frame after Data Preprocessing

Statistical Analysis: The descriptive statistics in ***Figure. 9*** summarize the distribution of values in the normalized dataset. They include measures like the mean, standard deviation, minimum, maximum, and percentiles, giving insights into the dataset's characteristics and variability.

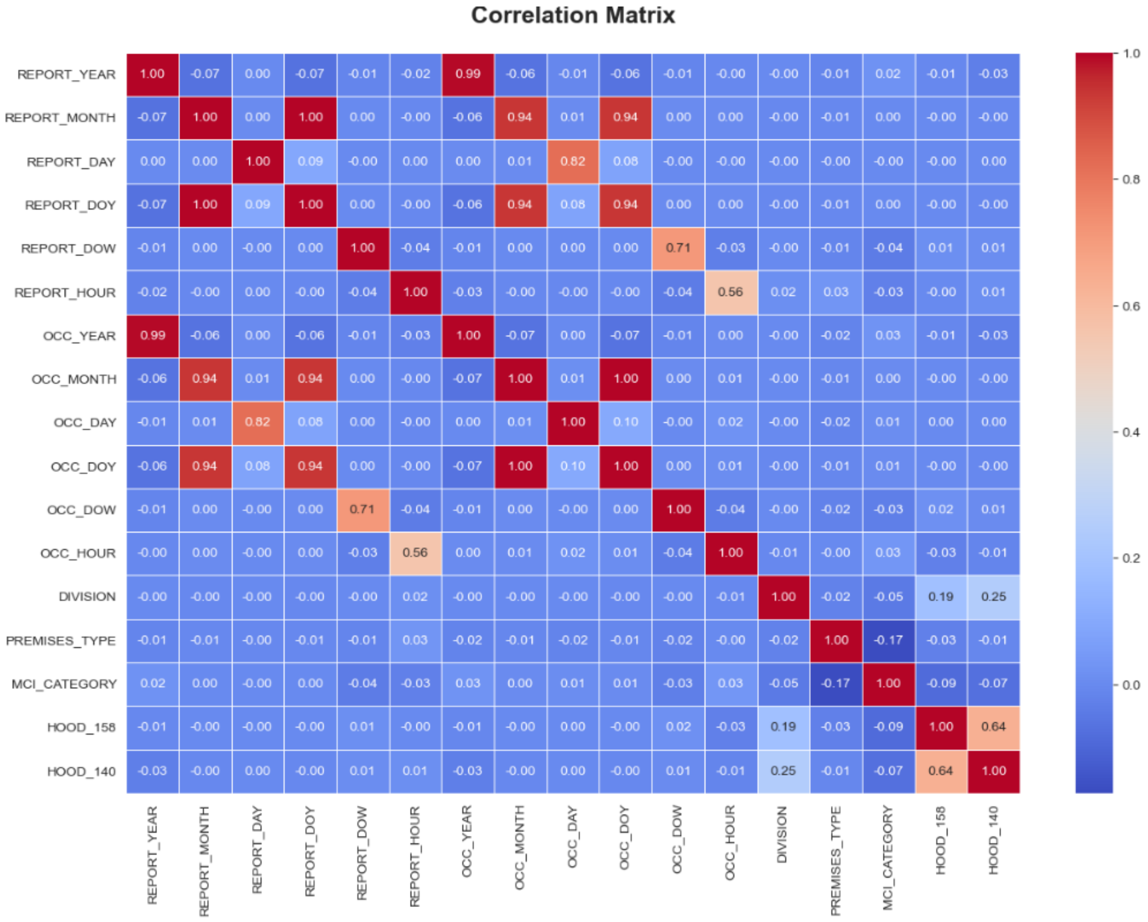


```
df.describe()
```

	REPORT_YEAR	REPORT_MONTH	REPORT_DAY	REPORT_DOY	REPORT_DOW	REPORT_HOUR	OCC_YEAR	OCC_MONTH	OCC_DAY	OCC_DOY
count	352884.000000	352884.000000	352884.000000	352884.000000	352884.000000	352884.000000	352884.000000	352884.000000	352884.000000	352884.000000
mean	0.517914	0.500999	0.491695	0.497842	0.492199	0.552943	0.514053	0.499999	0.482838	0.496204
std	0.315842	0.304281	0.292216	0.280266	0.332096	0.281880	0.315304	0.305134	0.296853	0.281269
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	0.222222	0.272727	0.233333	0.260274	0.166667	0.347826	0.222222	0.272727	0.233333	0.257534
50%	0.555556	0.545455	0.500000	0.501370	0.500000	0.565217	0.555556	0.545455	0.500000	0.498630
75%	0.777778	0.727273	0.733333	0.731507	0.833333	0.782609	0.777778	0.727273	0.733333	0.731507
max	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000

Figure. 9 Descriptive Statistics Summary

Here we utilized a heatmap to visualize the correlation matrix as below ***Figure. 10,*** A correlation matrix is a table showing correlation coefficients between variables. Each cell in the table represents the correlation between two variables. The value of the correlation coefficient ranges from -1 to 1. A correlation of 1 indicates a perfect positive correlation, -1 indicates a perfect negative correlation, and 0 indicates no correlation. The correlation matrix helps identify relationships between variables and can be used to assess multicollinearity in regression analysis or to identify patterns in data.



	REPORT_YEAR	REPORT_MONTH	REPORT_DAY	REPORT_DOY	REPORT_DOW	REPORT_HOUR	OCC_YEAR	OCC_MONTH	OCC_DAY	OCC_DOY	OCC_DOW	OCC_HOUR	DIVISION	PREMISES_TYPE	MCI_CATEGORY	HOOD_158	HOOD_140
REPORT_YEAR	1.00	-0.07	0.00	-0.07	-0.01	-0.02	0.99	-0.06	-0.01	-0.06	-0.01	-0.00	-0.00	-0.01	0.02	-0.01	-0.03
REPORT_MONTH	-0.07	1.00	0.00	1.00	0.00	-0.00	-0.06	0.94	0.01	0.94	0.00	0.00	-0.00	-0.01	0.00	-0.00	-0.00
REPORT_DAY	0.00	0.00	1.00	0.09	-0.00	0.00	0.00	0.01	0.82	0.08	-0.00	-0.00	-0.00	-0.00	-0.00	-0.00	0.00
REPORT_DOY	-0.07	1.00	0.09	1.00	0.00	-0.00	-0.06	0.94	0.08	0.94	0.00	0.00	-0.00	-0.01	0.00	-0.00	-0.00
REPORT_DOW	-0.01	0.00	-0.00	0.00	1.00	-0.04	-0.01	0.00	0.00	0.00	0.71	-0.03	-0.00	-0.01	-0.04	0.01	0.01
REPORT_HOUR	-0.02	-0.00	0.00	-0.00	-0.04	1.00	-0.03	-0.00	-0.00	-0.00	-0.04	0.56	0.02	0.03	-0.03	-0.00	0.01
OCC_YEAR	0.99	-0.06	0.00	-0.06	-0.01	-0.03	1.00	-0.07	0.00	-0.07	-0.01	0.00	-0.00	-0.02	0.03	-0.01	-0.03
OCC_MONTH	-0.06	0.94	0.01	0.94	0.00	-0.00	-0.07	1.00	0.01	1.00	0.00	0.01	-0.00	-0.01	0.00	-0.00	-0.00
OCC_DAY	-0.01	0.01	0.82	0.08	0.00	-0.00	0.00	0.01	1.00	0.10	-0.00	0.02	-0.00	-0.02	0.01	0.00	0.00
OCC_DOY	-0.06	0.94	0.08	0.94	0.00	-0.00	-0.07	1.00	0.10	1.00	0.00	0.01	-0.00	-0.01	0.01	-0.00	-0.00
OCC_DOW	-0.01	0.00	-0.00	0.00	0.71	-0.04	-0.01	0.00	-0.00	0.00	1.00	-0.04	-0.00	-0.02	-0.03	0.02	0.01
OCC_HOUR	-0.00	0.00	-0.00	0.00	-0.03	0.56	0.00	0.01	0.02	0.01	-0.04	1.00	-0.01	-0.00	0.03	-0.03	-0.01
DIVISION	-0.00	-0.00	-0.00	-0.00	-0.00	0.02	-0.00	-0.00	-0.00	-0.00	-0.00	-0.01	1.00	-0.02	-0.05	0.19	0.25
PREMISES_TYPE	-0.01	-0.01	-0.00	-0.01	-0.01	0.03	-0.02	-0.01	-0.02	-0.01	-0.02	-0.00	-0.02	1.00	-0.17	-0.03	-0.01
MCI_CATEGORY	0.02	0.00	-0.00	0.00	-0.04	-0.03	0.03	0.00	0.01	0.01	-0.03	0.03	-0.05	-0.17	1.00	-0.09	-0.07
HOOD_158	-0.01	-0.00	-0.00	-0.00	0.01	-0.00	-0.01	-0.00	0.00	-0.00	0.02	-0.03	0.19	-0.03	-0.09	1.00	0.64
HOOD_140	-0.03	-0.00	0.00	-0.00	0.01	0.01	-0.03	-0.00	0.00	-0.00	0.01	-0.01	0.25	-0.01	-0.07	0.64	1.00

Figure. 10 Correlation Matrix

Here, we notice some features that are strongly positively correlated with a correlation coefficient of 1, such as OCC_DOY and OCC_MONTH. Conversely, there are features showing a negative correlation with a coefficient of -1, like the LONG and LAT features.

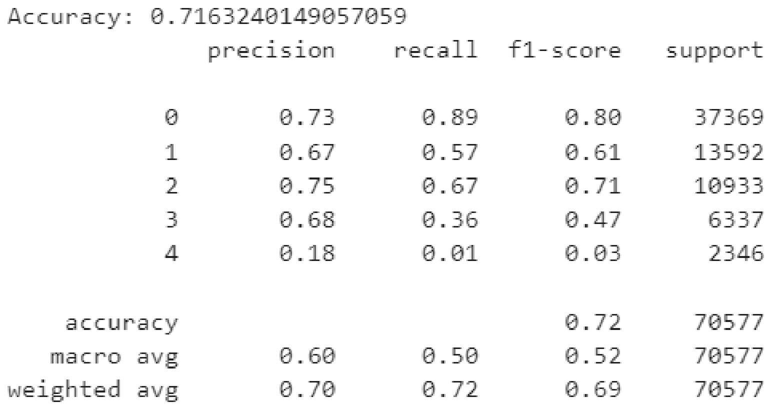
Data Modeling:

Armed with a deep understanding of the data, we embark on the modeling phase, where the magic of machine learning comes into play. In the Data Modeling phase, we harness the potential of machine learning algorithms to analyze the data and predict future crime rates based on historical patterns. This process involves selecting suitable algorithms, dividing the data into training, and testing sets, training the models, and evaluating their performance using various metrics. We fine-tune the models to enhance their accuracy and robustness against unseen data, aiming to identify the most effective model for deployment. Through iterative refinement and evaluation, we strive to uncover insights and patterns that can inform proactive measures for crime prevention and law enforcement efforts.

In this phase, we opted for a range of models to assess crime rate predictions, including the Random Forest Classifier, Logistic Regression, Neural Networks, and Time Series Analysis. This diverse selection allows us to compare the performance of different algorithms and choose the most suitable one based on factors like accuracy, interpretability, and computational efficiency.

Random Forest Classifier Model:

The Random Forest Classifier model, a popular ensemble learning method, was utilized for crime rate prediction. Initially, the dataset was split into features (X) and the target variable (y), representing the Major Crime Indicators (MCI) categories. Subsequently, the data was divided into training and testing sets using an 80:20 ratio, with the random_state parameter set to 42 for reproducibility.



```
Accuracy: 0.7163240149057059
              precision    recall  f1-score   support

           0       0.73      0.89      0.80     37369
           1       0.67      0.57      0.61     13592
           2       0.75      0.67      0.71     10933
           3       0.68      0.36      0.47      6337
           4       0.18      0.01      0.03      2346

    accuracy                           0.72     70577
   macro avg       0.60      0.50      0.52     70577
weighted avg       0.70      0.72      0.69     70577
```

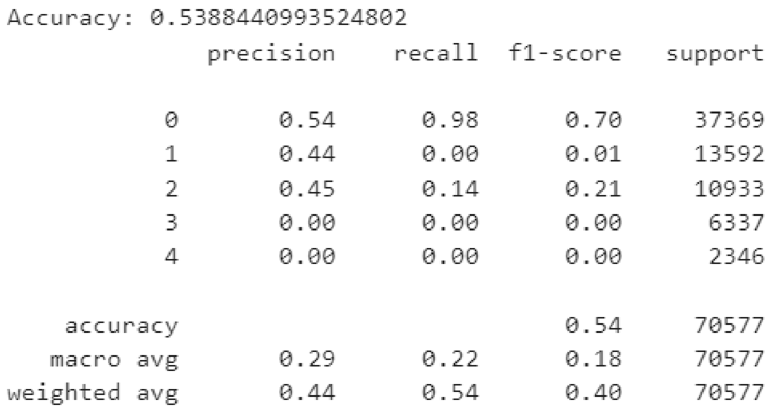
Figure. 11 Random Forest Classification Report

This model achieved a reasonable accuracy of approximately 72%. Looking at the classification report, it's evident that the model performed reasonably well across multiple metrics. For most classes, precision and recall scores are high, indicating that the model correctly classified instances of each crime category with minimal false positives and false negatives.

However, for the category **(Auto Theft)** labeled as 4, which likely corresponds to a less frequent type of crime, the precision and recall scores are comparatively lower. This suggests that the model struggled to accurately predict instances of this category, possibly due to its imbalanced representation in the dataset. Overall, the Random Forest Classifier demonstrates reasonable predictive capabilities for crime rate analysis, with room for further optimization, particularly in handling less prevalent crime categories.

Logistic Regression Model:

The Logistic Regression model was employed as another approach for crime rate prediction. Like the Random Forest Classifier, the dataset was split into training and testing sets with an 80:20 ratio using the train_test_split function from *Scikit-Learn*. The Logistic Regression classifier was then instantiated with a maximum iteration parameter (max_iter) set to 10,000 to ensure convergence during training.



```
Accuracy: 0.5388440993524802
              precision    recall  f1-score   support

           0       0.54      0.98      0.70     37369
           1       0.44      0.00      0.01     13592
           2       0.45      0.14      0.21     10933
           3       0.00      0.00      0.00      6337
           4       0.00      0.00      0.00      2346

    accuracy                           0.54     70577
   macro avg       0.29      0.22      0.18     70577
weighted avg       0.44      0.54      0.40     70577
```

Figure. 12 Logistic Regression Classification Report

After training the model on the training data, predictions were made on the testing set, and the accuracy of the model was evaluated. The accuracy of the Logistic Regression model was found to be approximately 54%, which is lower than that of the Random Forest Classifier.

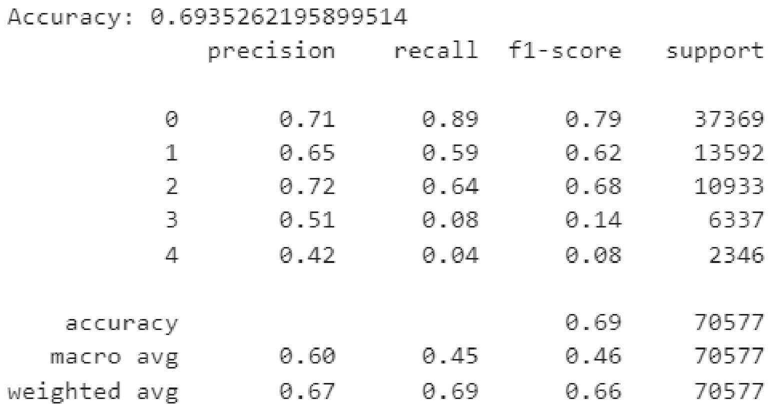
Further analysis was conducted using the classification report shown in ***Figure. 12****,* revealing precision, recall, and F1-score metrics for each MCI category. Compared to the Random Forest Classifier, the Logistic Regression model exhibited lower precision, recall, and F1-score values across most MCI categories. This suggests that the Logistic Regression model had a harder time accurately predicting instances of various crime categories compared to the Random Forest Classifier.

Gradient Boosting (XGBoost) Classifier Model:

The XGBoost classifier model was employed as another approach for crime rate prediction. Like the Random Forest Classifier, the XGBoost classifier was then instantiated with same splitting as above models. After training the model on the training data, predictions were made on the testing set, and the accuracy of the model was evaluated. The accuracy of the XGBoost classifier model was found to be approximately 69%, which is lower than that of the Random Forest Classifier but better performance than Logistic Regression Model.

Further analysis was conducted using the classification report shown in ***Figure. 13***, revealing precision, recall, and F1-score metrics for each MCI category. Compared to the Random Forest Classifier, the model exhibited lower precision, recall, and F1-score values across most MCI categories. This suggests that the

XGBoost Classifier model also had a harder time accurately predicting instances of various crime categories compared to the Random Forest Classifier.

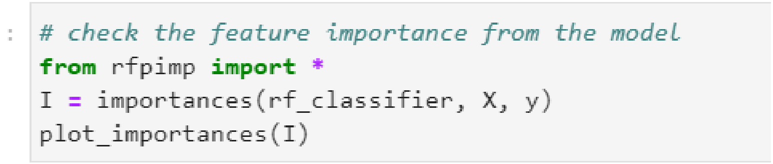


```
Accuracy: 0.6935262195899514
              precision    recall  f1-score   support

           0       0.71      0.89      0.79     37369
           1       0.65      0.59      0.62     13592
           2       0.72      0.64      0.68     10933
           3       0.51      0.08      0.14      6337
           4       0.42      0.04      0.08      2346

    accuracy                           0.69     70577
   macro avg       0.60      0.45      0.46     70577
weighted avg       0.67      0.69      0.66     70577
```

Figure. 13 Gradient Boosting (XGBoost) Classification Report

Enhancing Accuracy through Feature Importance Analysis:

In our pursuit of maximizing accuracy, we employed the feature importances analysis to identify the best features for optimal classifier models. Utilizing the **"rfpimp"** feature importances method, we discerned the most influential features from our selected set, as illustrated in the ***Figure. 14*** below.



```
# check the feature importance from the model
from rfpimp import *
I = importances(rf_classifier, X, y)
plot_importances(I)
```

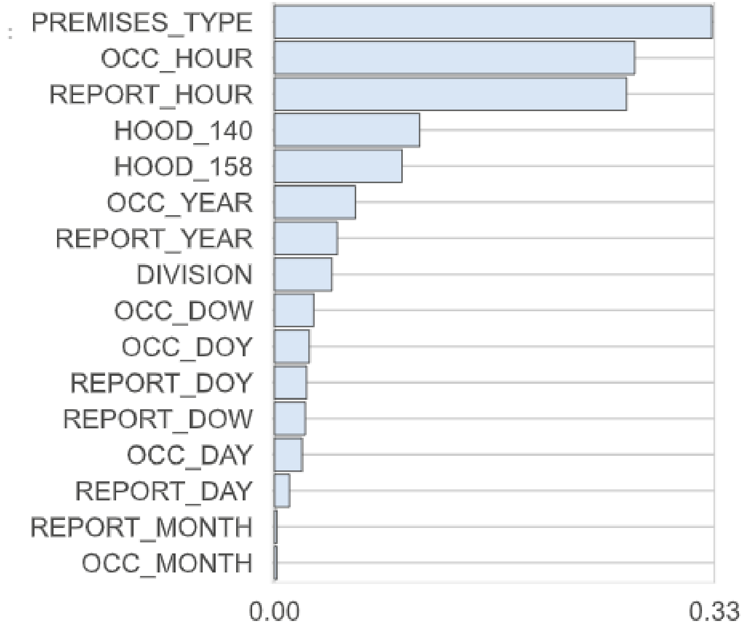
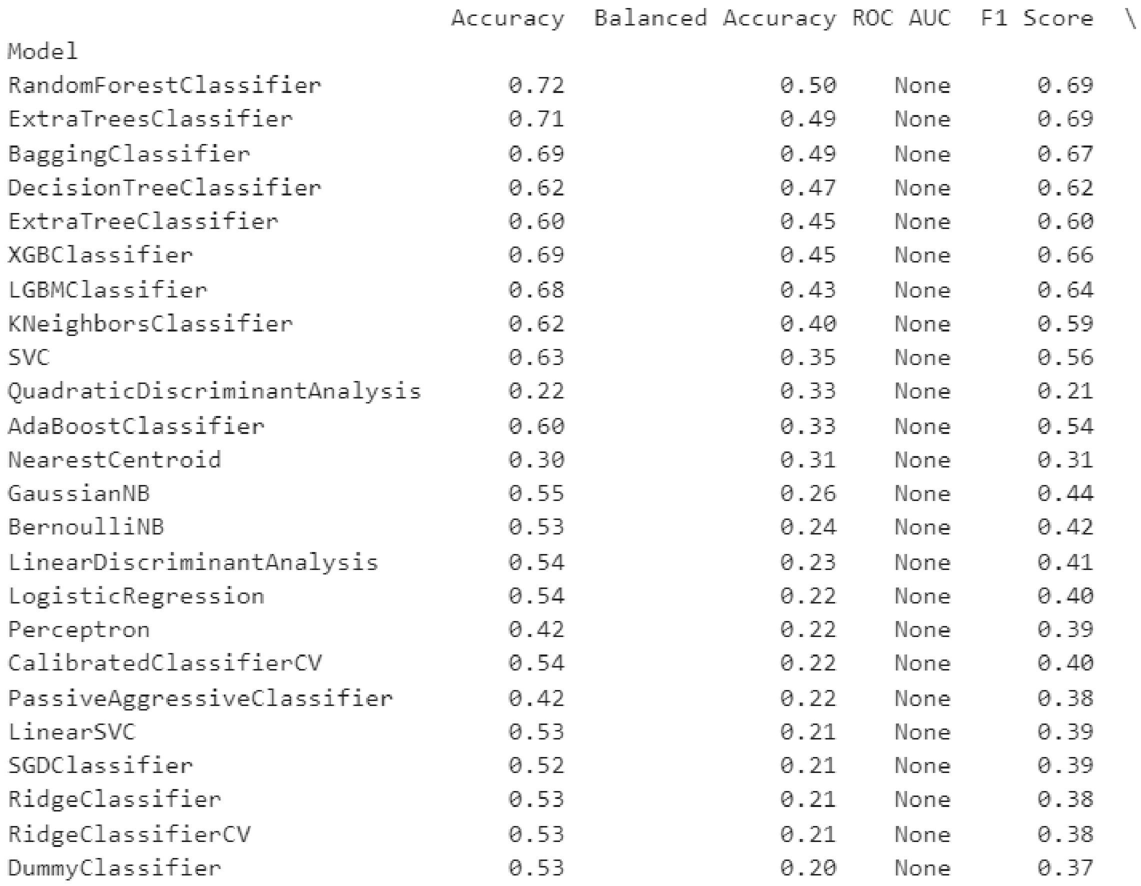


Figure. 14 Feature Importances

Lazy Predict Classifier Modeling:

Following the feature importance analysis, we meticulously selected highly influential features to enhance model accuracy. Subsequently, we applied *Lazy Predict* classifier modeling to identify the most promising model among the classifiers evaluated. However, contrary to expectations, as illustrated in ***Figure. 15*** below, we observed no discernible improvement in model accuracy, and the Random Forest Classifier remained the best-performing model.

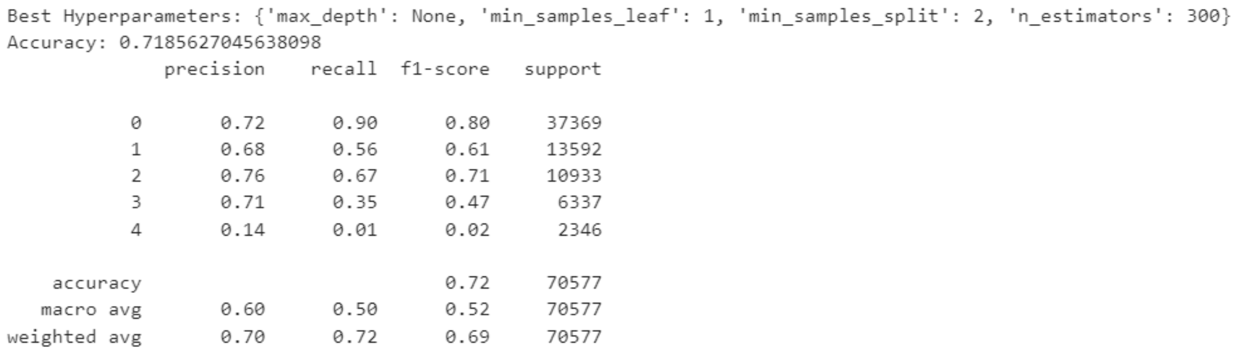


```
                               Accuracy  Balanced Accuracy ROC AUC  F1 Score  \
Model
RandomForestClassifier             0.72               0.50    None      0.69
ExtraTreesClassifier               0.71               0.49    None      0.69
BaggingClassifier                  0.69               0.49    None      0.67
DecisionTreeClassifier             0.62               0.47    None      0.62
ExtraTreeClassifier                0.60               0.45    None      0.60
XGBClassifier                      0.69               0.45    None      0.66
LGBMClassifier                     0.68               0.43    None      0.64
KNeighborsClassifier               0.62               0.40    None      0.59
SVC                                0.63               0.35    None      0.56
QuadraticDiscriminantAnalysis      0.22               0.33    None      0.21
AdaBoostClassifier                 0.60               0.33    None      0.54
NearestCentroid                    0.30               0.31    None      0.31
GaussianNB                         0.55               0.26    None      0.44
BernoulliNB                        0.53               0.24    None      0.42
LinearDiscriminantAnalysis         0.54               0.23    None      0.41
LogisticRegression                 0.54               0.22    None      0.40
Perceptron                         0.42               0.22    None      0.39
CalibratedClassifierCV             0.54               0.22    None      0.40
PassiveAggressiveClassifier        0.42               0.22    None      0.38
LinearSVC                          0.53               0.21    None      0.39
SGDClassifier                      0.52               0.21    None      0.39
RidgeClassifier                    0.53               0.21    None      0.38
RidgeClassifierCV                  0.53               0.21    None      0.38
DummyClassifier                    0.53               0.20    None      0.37
```

Figure. 15 Lazy Predict Classifier Models and Their Corresponding Accuracies.

Fine-tuning the Random Forest Model for Improved Performance:

Despite identifying the Random Forest Model as the most effective thus far, our pursuit of improved performance led us to fine-tune its hyperparameters. However, even with the utilization of the best hyperparameters—max_depth: None, min_samples_leaf: 1, min_samples_split: 2, and n_estimators: 300—we observed no significant enhancement in model performance.

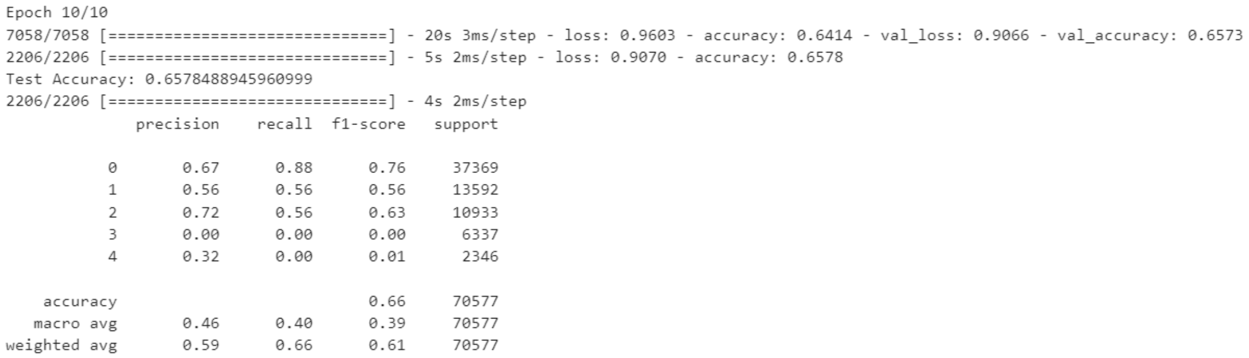


```
Best Hyperparameters: {'max_depth': None, 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 300}
Accuracy: 0.7185627045638098
              precision    recall  f1-score   support

           0       0.72      0.90      0.80     37369
           1       0.68      0.56      0.61     13592
           2       0.76      0.67      0.71     10933
           3       0.71      0.35      0.47      6337
           4       0.14      0.01      0.02      2346

    accuracy                           0.72     70577
   macro avg       0.60      0.50      0.52     70577
weighted avg       0.70      0.72      0.69     70577
```

Figure. 16 Refining the Random Forest Model

Neural Networks:

In our quest for enhanced performance, we delved into advanced modeling methodologies, specifically neural networks, and Time series Analysis. For Neural Networks model, leveraging ***TensorFlow*** and ***Kera's***, we constructed a Sequential model comprising several layers, including dense and dropout layers. The dataset was split into training and testing sets with a test size of 20%. To facilitate multi-class classification, we encoded the target variable using Label Encoder. The model architecture included an input layer with 64 neurons, followed by a dropout layer with a dropout rate of 0.5 to mitigate overfitting. Subsequently, another dense layer with 32 neurons was incorporated before the output layer, which employed SoftMax activation for multi-class classification.



```
Epoch 10/10
7058/7058 [==============================] - 20s 3ms/step - loss: 0.9603 - accuracy: 0.6414 - val_loss: 0.9066 - val_accuracy: 0.6573
2206/2206 [==============================] - 5s 2ms/step - loss: 0.9070 - accuracy: 0.6578
Test Accuracy: 0.6578488945960999
2206/2206 [==============================] - 4s 2ms/step
              precision    recall  f1-score   support

           0       0.67      0.88      0.76     37369
           1       0.56      0.56      0.56     13592
           2       0.72      0.56      0.63     10933
           3       0.00      0.00      0.00      6337
           4       0.32      0.00      0.01      2346

    accuracy                           0.66     70577
   macro avg       0.46      0.40      0.39     70577
weighted avg       0.59      0.66      0.61     70577
```

Figure. 17 Fully Connected Neural Network Accuracy and Classification Report

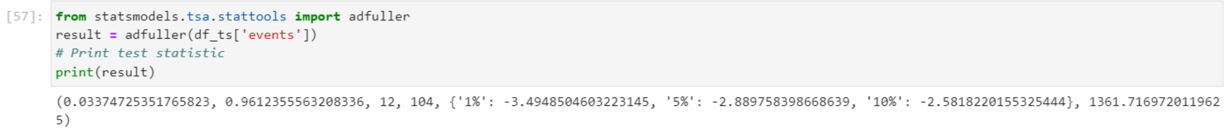
The model was compiled using the **Adam** optimizer and sparse **categorical cross-entropy** loss function. During training, the model underwent 10 epochs with a batch size of 32, achieving a validation accuracy of approximately **65%**. Evaluation on the test set yielded an accuracy of **65.78%.** The classification report revealed precision, recall, and f1-score metrics for each class, with notable challenges in classes 3 and 4. Though exhibiting potential as a predictive tool in crime analysis, further optimization may be warranted to enhance performance, particularly in handling classes with fewer instances, as evidenced by its performance compared to Random Forest and XGBoost models.

Time Series Analysis:

Time series modeling is a specialized technique within data analysis focused on comprehending and predicting data points gathered sequentially over time. Unlike conventional statistical models, time series models consider the chronological order of data points and strive to capture inherent patterns and trends in the time series data. This method proves beneficial for forecasting future values based on past observations, particularly in scenarios where understanding and predicting temporal patterns are paramount. Widely utilized across various domains such as finance, economics, sales forecasting, and weather forecasting, time series modeling leverages historical trends to glean insights into prospective outcomes.

Given the temporal nature of our data patterns, we've chosen to extend our analysis by integrating time series analysis techniques, including seasonal ARIMA (SARIMA) and long short-term memory (LSTM) networks.

We conducted an Augmented Dickey-Fuller test to assess the stationarity of our dataset and determine the suitable time series model. The test results, depicted in ***Figure. 18***, revealed a p-value exceeding 0.05. Typically, a p-value greater than 0.05 implies the failure to reject the null hypothesis, indicating non-stationarity in the time series data. The elevated p-value suggests the presence of seasonality and trend components within the data.

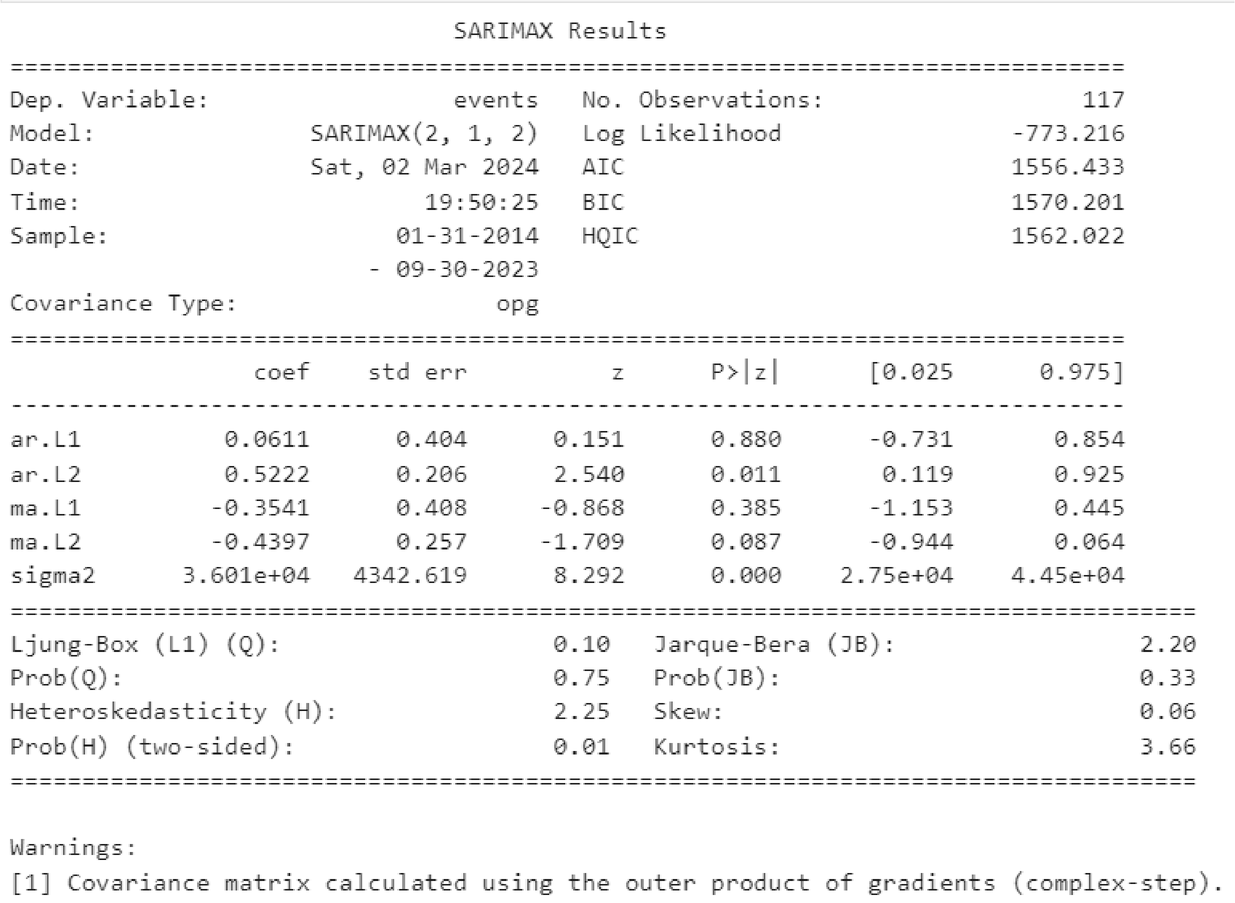


```
[57]: from statsmodels.tsa.stattools import adfuller
      result = adfuller(df_ts['events'])
      # Print test statistic
      print(result)

      (0.03374725351765823, 0.9612355563208336, 12, 104, {'1%': -3.4948504603223145, '5%': -2.889758398668639, '10%': -2.5818220155325444}, 1361.716972011962
      5)
```

Figure. 18 Augmented Dickey-Fuller test results

SARIMAX Model:

Based on the Augmented Dickey-Fuller test results indicating the presence of seasonality and trend components in the data, we opted for the SARIMAX (Seasonal Autoregressive Integrated Moving Average with Exogenous Variables) model for further analysis. SARIMAX models are effective for forecasting in time series data, especially when dealing with seasonal patterns. They incorporate parameters such as autoregression, differencing, moving average, and seasonal components, enabling the capture of complex temporal patterns. By fitting SARIMAX models to historical data, we can generate forecasts for future time points, offering valuable insights for decision-making and planning.



```
                               SARIMAX Results
==============================================================================
Dep. Variable:                  events   No. Observations:                  117
Model:                SARIMAX(2, 1, 2)   Log Likelihood                -773.216
Date:                 Sat, 02 Mar 2024   AIC                           1556.433
Time:                         19:50:25   BIC                           1570.201
Sample:                     01-31-2014   HQIC                          1562.022
                          - 09-30-2023
Covariance Type:                   opg
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
ar.L1          0.0611      0.404      0.151      0.880      -0.731       0.854
ar.L2          0.5222      0.206      2.540      0.011       0.119       0.925
ma.L1         -0.3541      0.408     -0.868      0.385      -1.153       0.445
ma.L2         -0.4397      0.257     -1.709      0.087      -0.944       0.064
sigma2      3.601e+04   4342.619      8.292      0.000    2.75e+04    4.45e+04
===================================================================================
Ljung-Box (L1) (Q):                   0.10   Jarque-Bera (JB):                 2.20
Prob(Q):                              0.75   Prob(JB):                         0.33
Heteroskedasticity (H):               2.25   Skew:                             0.06
Prob(H) (two-sided):                  0.01   Kurtosis:                         3.66
===================================================================================

Warnings:
[1] Covariance matrix calculated using the outer product of gradients (complex-step).
```

Figure. 19 SARIMAX Model (2,1,2) Summary Report

The SARIMAX model results shown in ***Figure. 19*** indicate that the selected model has an order of (2, 1, 2), which signifies two autoregressive terms, one differencing term, and two moving average terms. The coefficients for these terms, along with their standard errors and p-values, provide insights into the model's performance and significance. Notably, the coefficient values help understand the impact of each term on the time series data. Additionally, the log likelihood, AIC, BIC, and other information criteria provide measures of model fit and complexity. The diagnostic tests, such as Ljung-Box (Q) and Jarque-Bera (JB), assess the model's residuals for autocorrelation and normality assumptions. Overall, the SARIMAX model appears to adequately capture the underlying patterns and dynamics in the time series data, as indicated by significant coefficients and satisfactory diagnostic test results.

One Step Forecasting using above SARIMAX model (2,1,2)

The one-step forecast generated from above SARIMAX model provides predictions for the number of events for the next twelve months as shown in below ***Figure. 20***. The predicted mean values suggest an increasing trend in the number of events over time, with fluctuations observed in certain months. Additionally, the confidence intervals indicate the range within which the actual values are likely to fall, with lower and upper limits providing bounds for the forecasts. Analyzing these predictions and confidence intervals can help identify potential patterns, trends, and uncertainties in the data, assisting decision-making and planning processes. Moreover, comparing the forecasted values with actual observations can assess the accuracy and reliability of the SARIMAX model for future predictions.

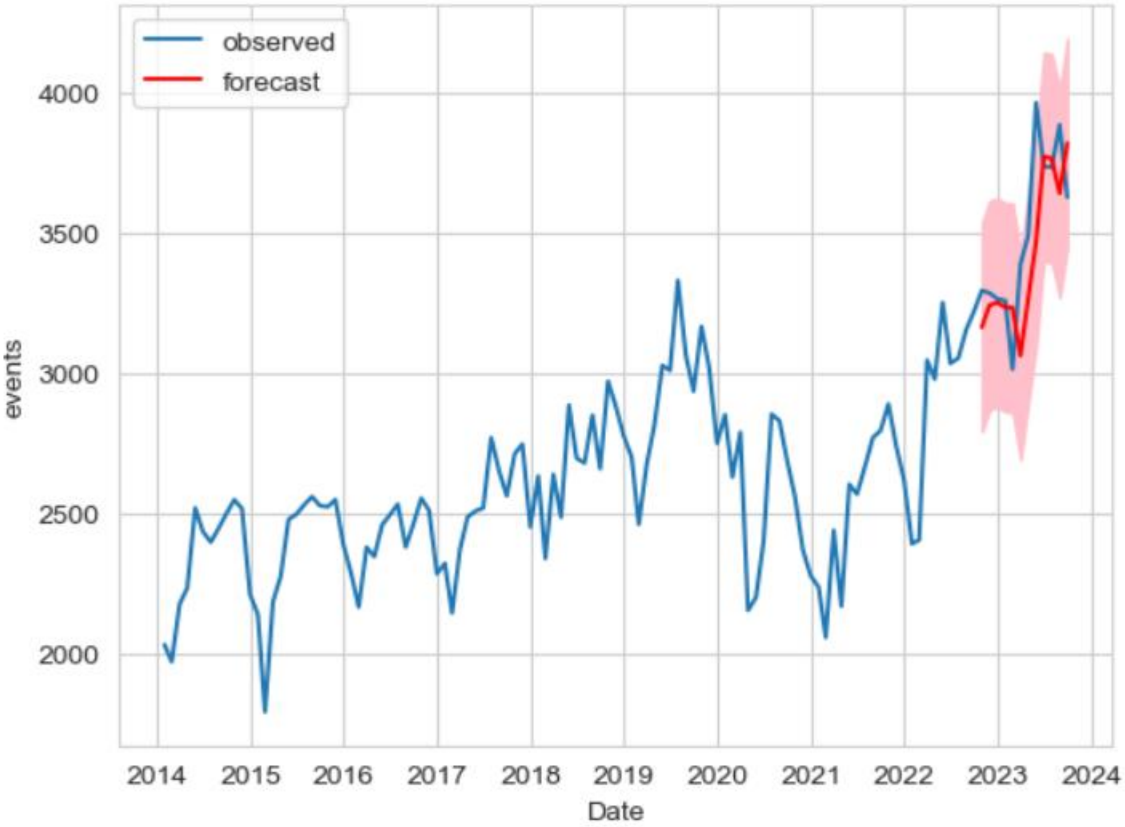


Figure. 20 One Step Forecasting

LSTM (Long Short-Term Memory) Model:

LSTMs are well-suited for capturing long-term dependencies and patterns in sequential data due to their ability to remember information over extended time intervals. By training an LSTM model on historical crime rate data, we aim to leverage its capacity to learn from past observations and make accurate predictions for future time points. This approach allows us to capture complex temporal patterns and dynamics inherent in the crime rate data, enabling us to generate forecasts that can assist in decision-making and resource allocation for crime prevention and law enforcement efforts. Additionally, evaluating the performance of the LSTM model through metrics such as mean squared error (MSE) or mean absolute error (MAE) can provide insights into its effectiveness in capturing and predicting temporal patterns in the data.

In this step, The LSTM (Long Short-Term Memory) model was trained and evaluated on the crime rate dataset using different optimizers (adam, rmsprop, adamax) and epochs (50, 80, 100). Among the configurations tested, the model with the adam optimizer and 50 epochs achieved the lowest mean squared error (MSE) on the test data, indicating better predictive performance. The MSE for this configuration was calculated to be 48402.71 for the test dataset.

The LSTM model effectively captured temporal dependencies and patterns in the crime rate data, offering valuable insights for predicting future crime rates. Lower MSE values signify better model accuracy, and comparable MSE values for training and testing data indicate the model's robustness without overfitting.

This suggests the LSTM model's ability to generalize well to unseen data, demonstrating its effectiveness in capturing underlying crime rate patterns.

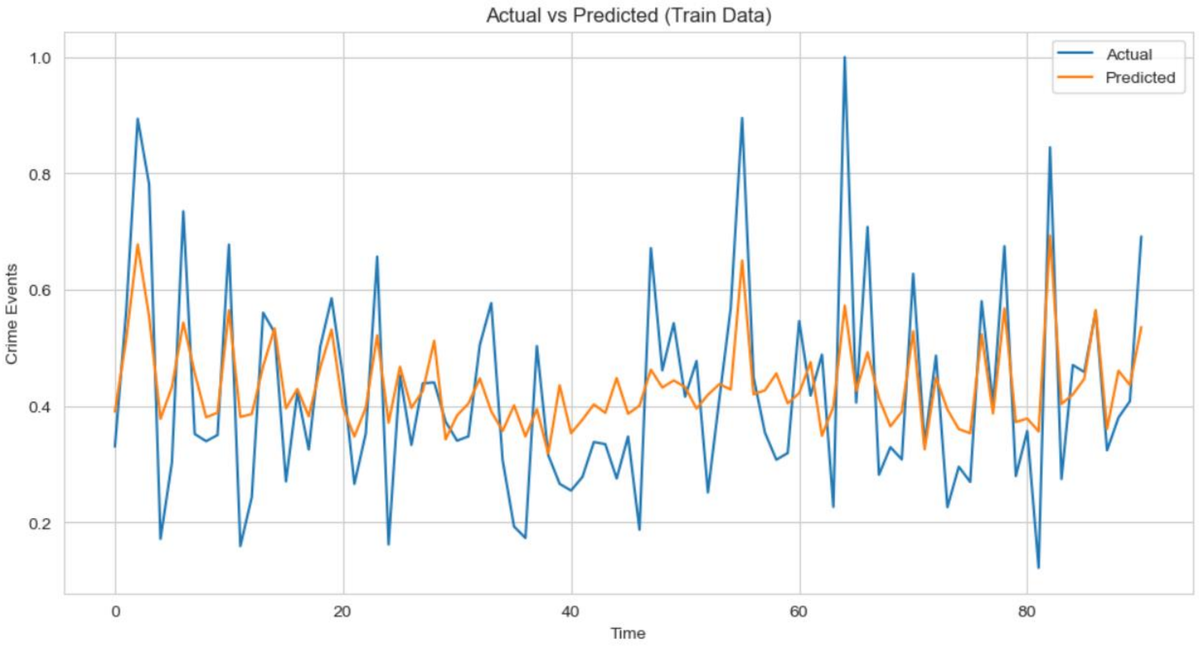


Figure. 21 LSTM Model prediction with Train Data

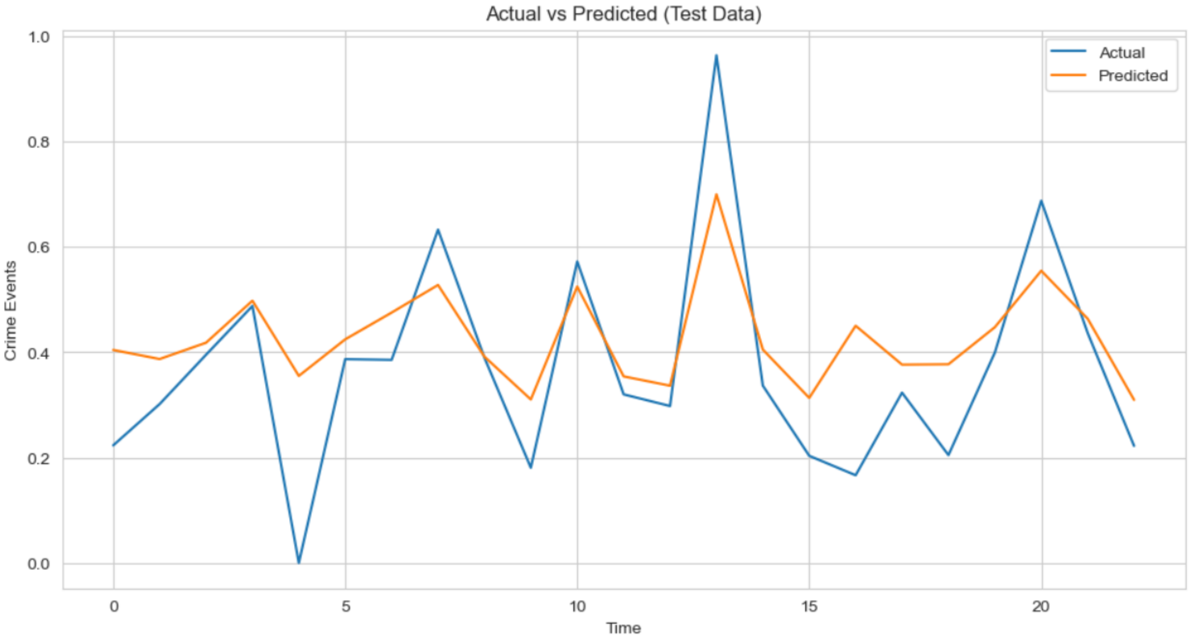


Figure. 22 LSTM Model prediction with Test Data

Above both plots ***Figure. 21*** and ***Figure.22*** display the "Actual" line representing actual crime events and the "Predicted" line showing the model's predictions. The first plot compares training dataset results, while the second depicts the testing dataset. Ideally, the "Predicted" line should closely align with the "Actual" line, indicating accurate predictions.

Deviations between them highlight areas of model underperformance or high prediction errors, offering insight into the model's effectiveness in capturing crime analysis patterns. While our model appears to perform reasonably well in capturing the overall trends. However, there are some instances where the "Predicted" line deviates from the "Actual" line, particularly during periods of fluctuating crime rate. This indicates potential areas where the model may be less accurate in capturing the variability of crime rate.

Interpretation:

In our next pivotal phase, we delve into the realm of interpretation. Here, we don our detective hats once more, deciphering the model's predictions, extracting actionable insights, and formulating data-driven strategies to address crime prevention and law enforcement efforts effectively.

Based on the extensive analysis and results obtained from our crime rate analysis project, several key interpretations emerge:

1. **Feature Importance**: Through feature analysis and correlation studies, we identified influential factors contributing to crime rates. Certain features showed strong correlations with crime occurrence, highlighting areas of focus for law enforcement and crime prevention strategies.

2. **Predictive Modeling Performance**: The machine learning models, including Random Forest Classifier, Time series Models - SARIMAX, and LSTM, demonstrated varying levels of performance in predicting crime rates. While some models achieved reasonable accuracy and precision, others exhibited limitations in capturing the complexity of the data.

3. **Temporal Patterns**: The time series analysis revealed significant temporal patterns and trends in the crime rate data. These patterns include seasonality, trends, and potentially cyclic behavior, providing valuable insights into the underlying dynamics of criminal activities over time.

4. **Model Interpretability**: The interpretability of models such as Random Forest Classifier and Time Series models allowed us to understand the underlying factors driving crime rates. This insight can aid policymakers and law enforcement agencies in developing targeted interventions and allocating resources effectively.

Model Deployment:

As we embarked on the culminating phase of our crime analysis project, the spotlight turned to deployment, the final act in our data science journey. With meticulous consideration, we selected the Random Forest Classifier model for its exceptional accuracy, outshining its counterparts in our exhaustive exploration.

In our quest to bring our model to life, we commenced the deployment process with purposeful steps. Setting the stage, we carved out a dedicated ***Model Directory*** as shown in ***Figure. 23*** for our deployment files, laying the groundwork for what was to come. Harnessing the power of PyCharm Professional via Anaconda Navigator, we navigated the terrain of deployment with confidence and precision. Within this directory, we meticulously crafted a model file housing the Random Forest model, complete with all essential preprocessing steps for the input features. Subsequently, we encapsulated the trained model into a pickle file, ensuring seamless integration into our ***Flask*** application.

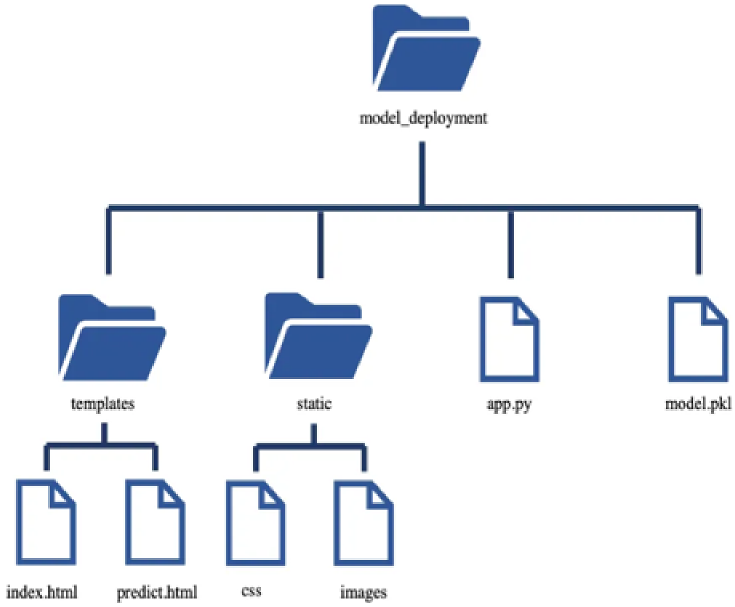
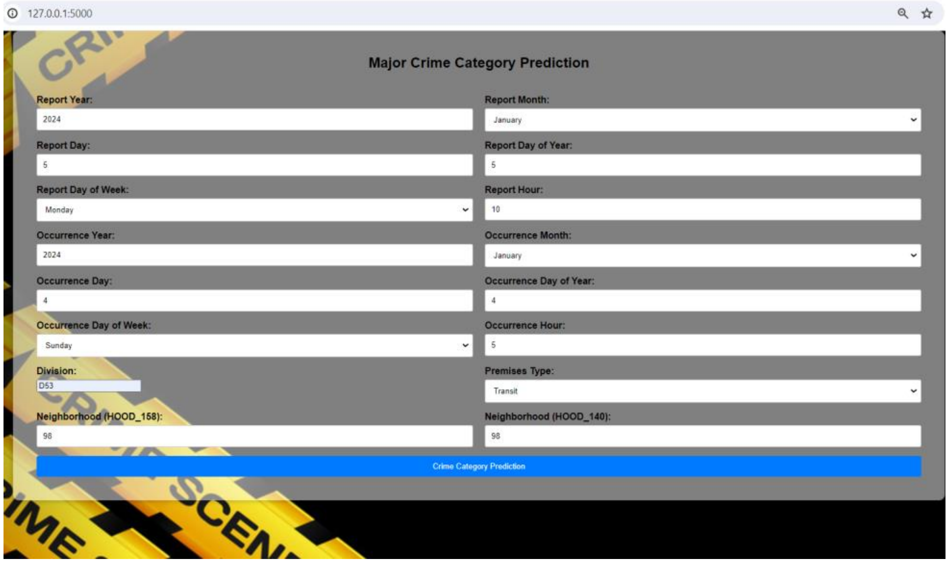


Figure. 23 Model Deployment Directory Structure

The development of the Flask application file followed suit, incorporating the functionality for loading the model alongside an aesthetically designed HTML web page, meticulously styled with CSS. This interactive web page empowers users to input values corresponding to the input features, facilitating the model to predict crime classes efficiently.

Behold, a sneak peek of our model deployment web page as shown in ***Figure. 24***, meticulously crafted using the Flask application. This user-friendly interface offers a glimpse into crime class predictions based on user input. With its sleek design and intuitive functionality, users can input relevant values, triggering the model to swiftly generate predictions. This seamless integration of the Random Forest Classifier model ensures a smooth and efficient user experience for those seeking insights into crime classification.



Major Crime Category Prediction

Report Year: 2024

Report Month: January

Report Day: 5

Report Day of Year: 5

Report Day of Week: Monday

Report Hour: 10

Occurrence Year: 2024

Occurrence Month: January

Occurrence Day: 4

Occurrence Day of Year: 4

Occurrence Day of Week: Sunday

Occurrence Hour: 5

Division: D53

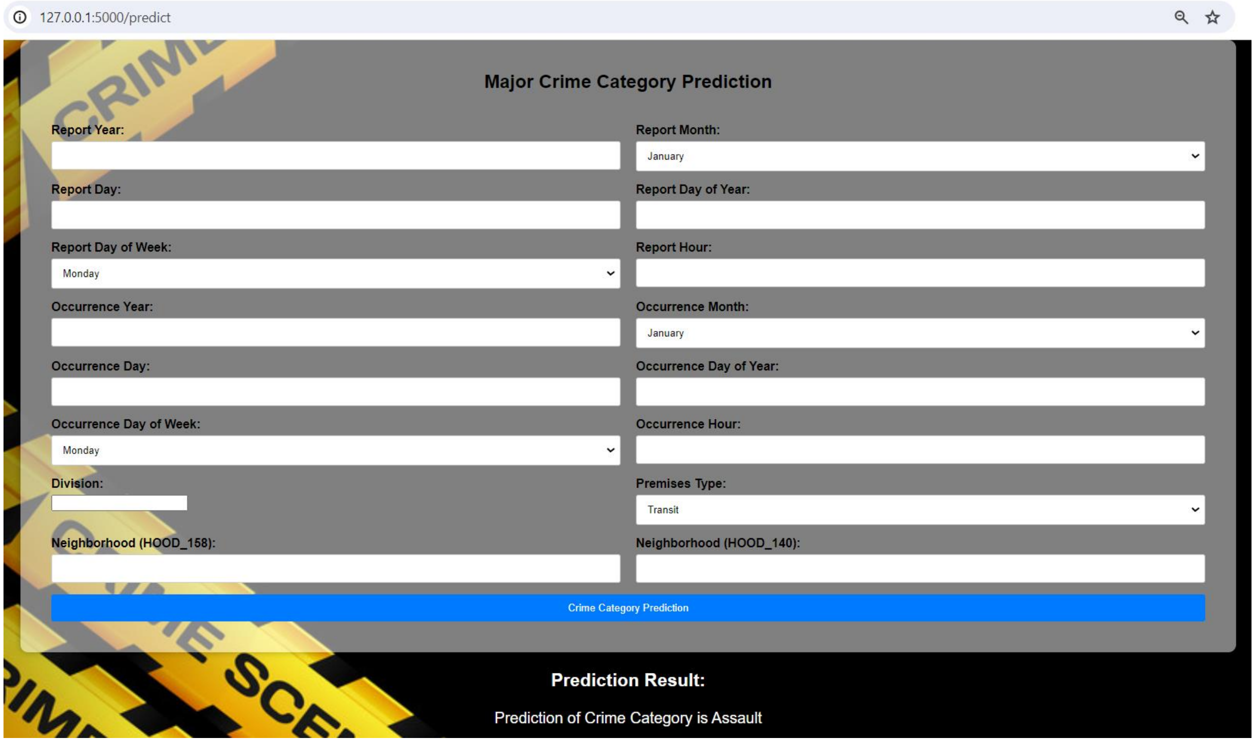
Premises Type: Transit

Neighborhood (HOOD_158): 98

Neighborhood (HOOD_140): 98

Figure. 24 Model Deployment User Interface

Based on the input data provided, we've generated prediction results as shown in ***Figure. 25*** to anticipate the major crime class category. This process involves feeding the input values into our model to derive the predicted outcome.



127.0.0.1:5000/predict

Major Crime Category Prediction

Report Year:

Report Month: January

Report Day:

Report Day of Year:

Report Day of Week: Monday

Report Hour:

Occurrence Year:

Occurrence Month: January

Occurrence Day:

Occurrence Day of Year:

Occurrence Day of Week: Monday

Occurrence Hour:

Division:

Premises Type: Transit

Neighborhood (HOOD_158):

Neighborhood (HOOD_140):

Crime Category Prediction

Prediction Result:

Prediction of Crime Category is Assault

Figure. 25 User Interface with Predicted Result

Conclusions:

In essence, our journey through the crime rate analysis capstone project encapsulates the essence of data science—a blend of curiosity, methodical exploration, and analytical prowess—all aimed at unraveling the mysteries hidden within the data and empowering decision-makers to create safer and more secure communities.

Comprehensive analysis yielded valuable insights into crime rate patterns and influential factors. Models like Random Forest Classifier, SARIMAX and LSTM showed promising predictive performance. Feature importance analysis highlighted key factors driving crime occurrence. Actionable insights were generated for targeted crime prevention strategies.

Our Journey Continues….

Our journey continues with a steadfast commitment to engaging stakeholders at further stages of the process. Following the deployment, we prioritize comprehensive training sessions to ensure proficient utilization of the model among all stakeholders. Moreover, ethical considerations remain paramount, and we remain vigilant in implementing safeguards to guarantee fair deployment and usage. To uphold effectiveness over time, we establish continuous evaluation and monitoring mechanisms, allowing us to adapt and refine strategies as needed. These post-deployment activities underscore our dedication to maximizing the impact of our crime analysis efforts while upholding ethical standards and ensuring ongoing effectiveness.

Upon reviewing the enclosed material, you should gain a comprehensive understanding of the appropriate direction for a capstone project and the essential elements it should encompass. The complete notebook containing code is accessible via the provided link ***here***. Should you require further clarification, wish to suggest enhancements, or desire to engage in discourse regarding potential ideas, please do not hesitate to reach out.

References:

1. Supervised learning. (n.d.). *Scikit-learn.* https://scikit-learn.org/stable/supervised_learning.html#supervised-learning
2. lazypredict. (2022b, September 28). *PyPI*. https://pypi.org/project/lazypredict/
3. Team, K. (n.d.-b). *Keras documentation: The Sequential model*. https://keras.io/guides/sequential_model/
4. *Welcome to Flask — Flask Documentation (3.0.X)*. (n.d.). https://flask.palletsprojects.com/en/3.0.x/
5. Indian international student killed in Toronto "polite, humble, a sweet child," father says. (2022, April 11). *CBC*. https://www.cbc.ca/news/canada/toronto/kartik-vasudev-indian-intenrational-student-1.6413878#:~:text=21%2Dyear%2Dold%20Kartik%20Vasudev%20gunned%20down%20Thursday%20night&text=Police%20have%20identified%20the%20victim,consulate%20general%20in%20Toronto%20Friday.
6. Tnn. (2023, May 14). Gujarat student found dead in Toronto. *The Times of India*. https://timesofindia.indiatimes.com/nri/us-canada-news/gujarat-student-found-dead-in-toronto/articleshow/100220123.cms?from=mdr

7. *Major Crime Indicators Open Data*. (n.d.). https://data.torontopolice.on.ca/datasets/TorontoPS::major-crime-indicators-open-data/about

8. *Introduction to the Keras Tuner*. (n.d.). TensorFlow. https://www.tensorflow.org/tutorials/keras/keras_tuner

9. Gulyamova, A. (2022, January 22). Deploying Classification Model with Flask - Aziza Gulyamova - Medium. *Medium*. https://medium.com/@agulyamova/deploying-classification-model-with-flask-1a694d3534a2