PyTorch Intro, Data analysis & NN Models

CSE 676-B: Deep Learning, Spring 2025

Sharanya Nallapeddi
snallape@buffalo.edu

Understanding the pattern of crimes informs safer communities and decision-making processes. This report examines trends, statistical correlations, and points of possible intervention using a unique dataset of incidents of crime reported in Buffalo, New York. *https://data.buffalony.gov/Public-Safety/Crime-Incidents/d6g9-xbgu/about_data is a publicly available dataset which we are using in our Deep Learning project.*

318,735 criminal incidents that were reported in Buffalo, New York, are included in this dataset. These incidents included a variety of offenses, such as assault, larceny/ theft, and burglary. For each incident, the dataset provides an essential summary that includes the case number, date/time of occurrence, major criminal classification, description, location (address, latitude, and longitude), and police district managing the case. In addition, it contains other census-based data that may be helpful for demographic and geographic study of crime, such as the census block, tract, and neighborhood. The data is perfect for statistical and predictive analytics on the trend of criminal actions because it is tabular and contains both numerical and category information. The incident occurred during the hour of the day; the average is approximately 11.92, or noon, with a standard deviation of 7.19, indicating that although crimes occur throughout the day, they are most common at noon. Numerous fields have missing data; the most noticeable ones are Location (6,560 missing), Latitude and Longitude (1,281 missing), and Neighborhood (3,629 missing), all of which may have an impact on geospatial analysis. There is no unique identifier other than the case number because the Incident ID column is entirely absent. Despite the absence of these values, the dataset is still among the most complete for examining Buffalo's crime trends and patterns, which aids law enforcement and policymakers in making decisions based on facts.

Quantitative fields: Hour of Day is the time of occurrence, and the mean is 11.92 hours, approximately noon, with a standard deviation of 7.19 hours, thus indicating that crime occurs at a broad distribution in time during the day. However, there are missing values in several fields: Location is missing 6,560 values, Latitude and Longitude are missing 1,281 values, and Neighborhood is missing 3,629 values, which may affect geospatial analysis. Incident ID is missing completely, with 318,735 missing values, hence it does not contribute any usable data.

Different Preprocessing Techniques were applied to the dataset, please see below.

- **Handling Missing Data:** For ensuring the consistency and reliability of the data, we used several techniques to handle the missing values. We first dropped those rows using dropna() that contained more than five missing values because high missing values can result in the model being trained unreliably. We also dropped columns where the percentage of missing values exceeded a threshold. Where there were not enough missing values to make the removal of a column warranted, we have used imputation methods: non-numeric placeholders ("UNKNOWN", etc.) were changed to NaN, then filled up with the mean, median, or mode.
- **Data Cleaning**: Cleaning the dataset was done in several steps to standardize and make it more usable. Extra spaces in column names were removed for consistency across a variety of processing steps. All rows with "UNKNOWN" values in a large number of fields of a categorical nature were removed so as not to give misleading input to the machine learning models. Besides that, duplicate rows were identified and removed so that redundant information did not skew model training or evaluation.

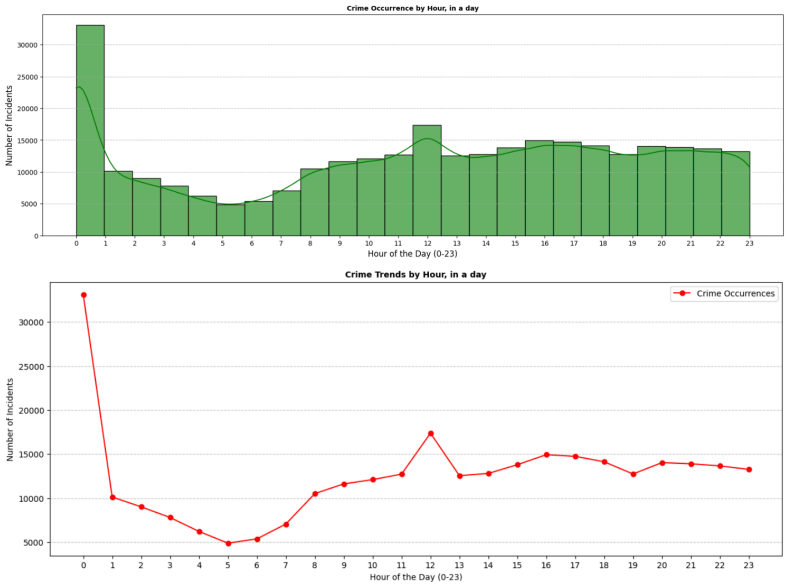
```
[ ]: dataset5 = dataset4[~(dataset4.apply(lambda row: (row.astype(str) == 'UNKNOWN').sum(), axis=1) > 2)]
     print(f"Rows are: {dataset5.shape[0]}, Columns are: {dataset5.shape[1]}")

     Rows are: 298583, Columns are: 29
```

- **Feature Engineering**: We employed various encoding techniques to transform the categorical data into a usable form for machine learning models. Categorical features that had a lot of unique values were one-hot encoded into binary representation. Using label encoding, we mapped the ordinal categorical variable into numerical values. Feature selection was done through different techniques like variance thresholding, which allows filtering out irrelevant features that provide little information with minimal variance. Also, we have performed model-based feature selection ranking and then selecting the most relevant attributes to use during training.
- **Transformation of Data**: StandardScaler() was used to normalize numerical features in order to enhance model performance. Logarithmic transformations were used to correct for highly skewed distributions. Models that are sensitive to scale benefited from these measures, which guaranteed consistency across features.
- **Keeping the Dataset in Balance**: SMOTE was used to create synthetic samples for underrepresented classes in order to balance the dataset. This enhanced classification performance for unusual events and prevented models from favoring majority-class predictions.
- **Dividing the Dataset**: 70% of the dataset was used for training, 15% for validation, and 15% for testing. To prevent data leakage and skewed assessments, stratified sampling was employed to guarantee that class proportions stayed constant across all subsets.

Visualizations

1. Insights of Crime analysis by hour.



From the above graphs, we can understand the below details.

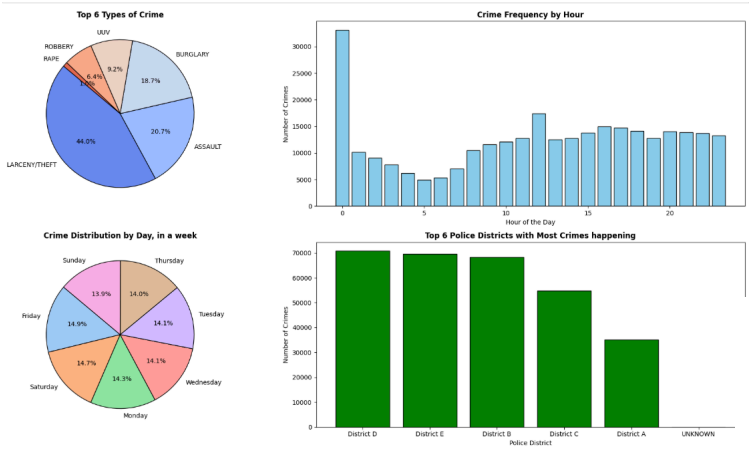
- Peak Crime Hours: At midnight (0:00), the number of crimes is at its maximum and then sharply declines. It makes sense that the secondary peak at 12:00 PM would be a sign of increased criminal activity at that time of day.
- Low Crime Hours: From 3:00 AM to 6:00 AM, there were few crimes reported. This makes sense given the lower level of public activity at the period.
- Gradual Rise During the Day: Following the early morning decline, crime begins to increase at 7:00 AM and peaks again between 12:00 PM and 8:00 PM. The pattern in the evening is rather steady.
- Possible causes: The midnight peak can be attributed to night life, booze-related incidents, and end-of-the-day activities, while the midday peak can be related to business increase, social activities, and crowded streets.
- Ramifications for Crime Prevention: Therefore, police patrolling should be increased during noon and midnight; resources should also be utilized most efficiently to give a faster response during peak time. These would help in effective strategic planning against crime and other anti-social illegal activities and a better management strategy for law enforcers or police.

PyTorch Intro, Data analysis & NN Models

CSE 676-B: Deep Learning, Spring 2025

Sharanya Nallapeddi
snallape@buffalo.edu

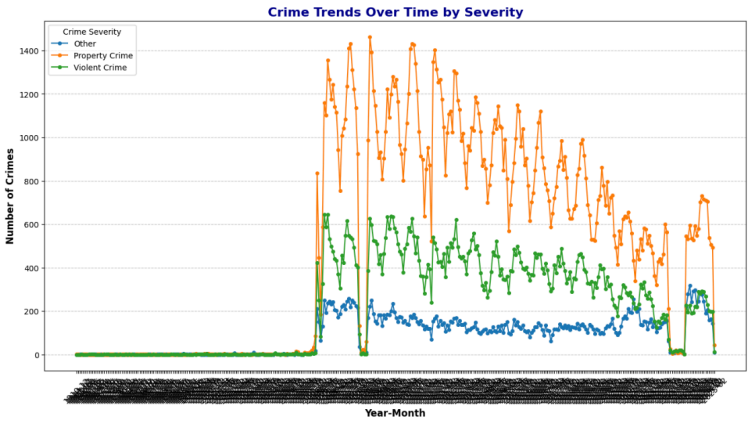
2. Crime Frequency and Police district with most crimes.



Things that we can depict from the above graph are listed below.

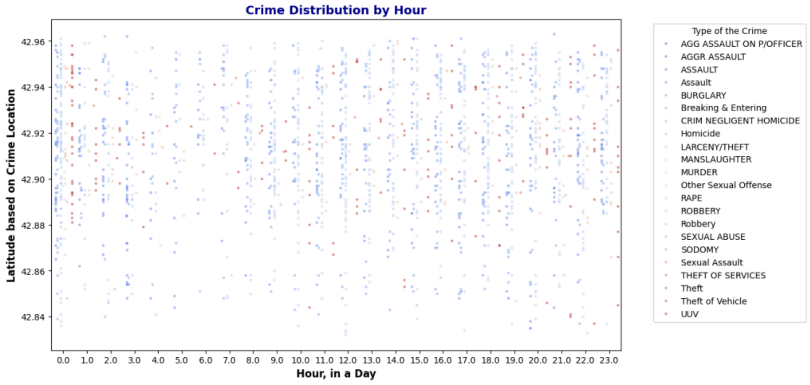
1. Theft (44%) is the most frequent crime, for almost half of all cases.
2. Assault is the 2nd most reported crime with 20.7%.
3. Burglary with 18.7% and Unauthorized Use of Vehicle (UUV - 9.2%) have the highest property crime rates with the data that is given.
4. Robbery at 6.4% and Rape at 1% are less frequent, but still has an impact.
5. Peak Time for Crime is Midnight, it is the most criminal time, mostly due to majority of thefts or violent fights.
6. Crime decreases after 2 AM but increases from 10 AM onwards, increasing again around afternoon to evening.
7. Late night & early morning see less crime, which are in sync with lower public presence.
8. Friday & Saturday have the highest, at 14.9% and 14.7%, respectively, due to the heightened activity on weekends. That doesn't mean we cant take breaks. But, the crime is 9. Also, it is actually relatively evenly distributed on Mondays to Thursdays, 14%. Sunday crimes are relatively lower at 13.9%.
10. District D & District E have the most crimes, close to 70,000+. Districts B & C report high crimes and need concentrated police enforcement.
11. District A has the least number of crimes, meaning that it has better security or it can also be because of lower population.

3. Crime Trends Over Time.



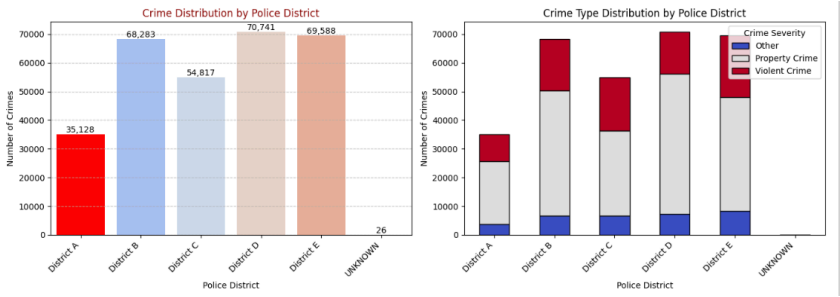
From the above graphs, we can understand the below details,
Graphically, one can ascertain that the property crime incidents contribute to nearly 60-65% of the total incidents reported with fluctuations on a monthly basis, peaking at 1,200-2,000 cases. The violent crimes constitute about 25-30% of total crimes. The values lie between 400 to 900 cases per month and depict sudden peaks. The graph indicates that there is a huge increase in crime around the midpoint, where the total incidents nearly tripled from about 500 to over 1,500 per month. Recently, the crime rates have shown a gradual decline, with about a 30-40% drop compared to peak values.

4. Crime distribution by hour.



From the above graphs, we can understand the below details,
As illustrated above, the crime incidents span across all 24 hours of the day, though there is quite a noticeable density between midnight (00:00) and 02:00 AM, and then another in the afternoon to late evening hours, around 12:00 PM to 6:00 PM. Latitude ranges from 42.84 to 42.96, reflecting crime dispersion across geographical locations. In addition, the crime types range from thefts, assault, robbery, and homicides, indicating variation in the patterns of crimes. Larceny/theft and assault are the most frequent crimes, especially during late night and afternoon hours. Starting at 3:00 AM, the density of crime diminishes significantly until it starts rising again around 8:00 AM, with a strong relation to human activities in the city.

5. Crime and it's relevance with police



Above graph demonstrates the distribution of crimes by police district, with District A having the fewest crimes and District D having the highest (70,741 instances), closely followed by Districts E and B. Property crimes predominate in all districts, whereas violent crimes are comparatively high in Districts B, D, and E, according to the right graphic, which groups crimes by severity.

PyTorch Intro, Data analysis & NN Models

CSE 676-B: Deep Learning, Spring 2025

Sharanya Nallapeddi
snallape@buffalo.edu

Machine Learning Models

For Machine learning Models, we have used XGBoost Algorithm, LightGBM and Random Forest Classifier.

a. XGBoost Algorithm

- Gradient Boosting Framework: The prediction accuracy is increased by combining a number of weak learners in an optimum way.
- Regularization: L1 and L2 regularization reduces overfitting.
- Missing Value Handling: In training, missing values are handled automatically.
- Parallel Processing: Used for faster execution.
- Pros: High Accuracy: XGBoost is extremely efficient and proved to be 72.63% accurate on our dataset.
- Scalability: Performs really well with large datasets, like ours, having more than 300,000 records.
- Feature Importance: Helps in understanding big causes driving the trend for crime.
- Relevance to Dataset: The XGBoost algorithm efficiently classifies the categories of crime because of the size and categorization nature in the dataset.
- Crime incidents usually have complex dependencies, which the XGBoost captures using the boosting techniques in an efficient manner.

XGBoost Models accuracy, values related to precision, recall, f1-score and support are given below. It's accuracy is **72.70%.**

XGBoost Model Accuracy is given as: 72.70%

Classification Report is given as:

	precision	recall	f1-score	support
0	0.91	1.00	0.95	19320
1	0.38	0.24	0.30	19320
2	1.00	1.00	1.00	19321
3	0.36	0.51	0.42	19321
4	1.00	1.00	1.00	19320
5	0.99	1.00	0.99	19320
6	0.48	0.48	0.48	19321
7	1.00	1.00	1.00	19321
8	0.69	0.88	0.77	19320
9	0.66	0.58	0.62	19320
10	0.51	0.48	0.49	19321
11	0.69	0.65	0.67	19320
12	0.68	0.75	0.71	19320
13	1.00	1.00	1.00	19320
14	0.49	0.34	0.40	19320
accuracy			0.73	289805
macro avg	0.72	0.73	0.72	289805
weighted avg	0.72	0.73	0.72	289805

b. LightGBM Algorithm

- Histogram-Based Learning: Makes use of histograms to expedite calculations.
- Memory Usage Efficiency: Utilizes less RAM than XGBoost.
- Leaf-wise Growth Strategy: Instead of growing depth-wise, this strategy splits the leaf with the greatest loss reduction.
- Benefits
- Speed: Faster than XGBoost on training; therefore, it is suitable for big data.
- Accuracy: It achieved a 69.85% accuracy, proving that this is one of the strongest models for our current classification problem.
- Handles Large-Scale Data: This handles crime records and location-based variables with much efficiency.
- Relevant to Dataset: Since most of the crime datasets are imbalanced, LightGBM handles class imbalance nicely, improving the performance of classification. Its efficiency makes it suitable for real-time crime prediction models.

```
[LightGBM] [Info] Total Bins 4832
[LightGBM] [Info] Number of data points in the train set: 1352421, number of used features: 19
[LightGBM] [Info] Start training from score -2.708050
[LightGBM] [Info] Start training from score -2.708050
[LightGBM] [Info] Start training from score -2.708050
[LightGBM] [Info] Start training from score -2.708050
[LightGBM] [Info] Start training from score -2.708050
[LightGBM] [Info] Start training from score -2.708050
[LightGBM] [Info] Start training from score -2.708050
[LightGBM] [Info] Start training from score -2.708050
[LightGBM] [Info] Start training from score -2.708050
[LightGBM] [Info] Start training from score -2.708050
[LightGBM] [Info] Start training from score -2.708050
[LightGBM] [Info] Start training from score -2.708050
[LightGBM] [Info] Start training from score -2.708050
[LightGBM] [Info] Start training from score -2.708050
[LightGBM] [Info] Start training from score -2.708050
```

LightGBM Model Accuracy: 69.80%

Classification Report:

	precision	recall	f1-score	support
0	0.91	1.00	0.95	19320
1	0.35	0.22	0.27	19320
2	1.00	1.00	1.00	19321
3	0.33	0.49	0.40	19321
4	1.00	1.00	1.00	19320
5	0.99	1.00	1.00	19320
6	0.44	0.52	0.48	19321
7	1.00	1.00	1.00	19321
8	0.63	0.84	0.72	19320
9	0.60	0.48	0.53	19320
10	0.47	0.43	0.45	19321
11	0.63	0.56	0.59	19320
12	0.62	0.65	0.64	19320
13	1.00	1.00	1.00	19320
14	0.44	0.29	0.35	19320
accuracy			0.70	289805
macro avg	0.69	0.70	0.69	289805
weighted avg	0.69	0.70	0.69	289805

LightGBM Models accuracy, values related to precision, recall, f1-score and support are given to the right. It's accuracy is **69.80%.**

c. Random Forest Classifier Algorithm

- Feature Selection: Informs about the most influential features in crime.
- Robust against Overfitting: Averages many trees, reducing the chance of overfitting.
- Benefits: Good Generalization: Gave an accuracy of 72.44%, proving its robustness.
- Interpretable Model: Helps identify the most influencing features on the crime trend.
- Handles Noisy Data Well: Works well with missing or erroneous values.
- Relevance to Dataset: Certain factors governing the rate of crimes involve multiple dimensions: time, place, type, and severity among others. Thus, Random Forest can be an option when handling high-dimensional data.
- It is effective in feature importance analysis for focussing the concentration of law enforcement on very important crime indicators.

PyTorch Intro, Data analysis & NN Models

CSE 676-B: Deep Learning, Spring 2025

Sharanya Nallapeddi
snallape@buffalo.edu

Random Forest Models accuracy, values related to precision, recall, f1-score and support are given to the right. It's accuracy is **72.44%.**

```
Optimized Random Forest Model Accuracy: 72.44%

Classification Report:
              precision    recall  f1-score   support

           0       0.86      1.00      0.92     19320
           1       0.43      0.16      0.23     19320
           2       1.00      0.99      1.00     19321
           3       0.32      0.58      0.41     19321
           4       1.00      0.98      0.99     19320
           5       0.98      1.00      0.99     19320
           6       0.46      0.38      0.42     19321
           7       1.00      1.00      1.00     19321
           8       0.64      0.92      0.76     19320
           9       0.75      0.63      0.68     19320
          10       0.53      0.45      0.49     19321
          11       0.71      0.70      0.70     19320
          12       0.71      0.76      0.74     19320
          13       0.98      1.00      0.99     19320
          14       0.52      0.32      0.40     19320

    accuracy                           0.72    289805
   macro avg       0.73      0.72      0.71    289805
weighted avg       0.73      0.72      0.71    289805
```

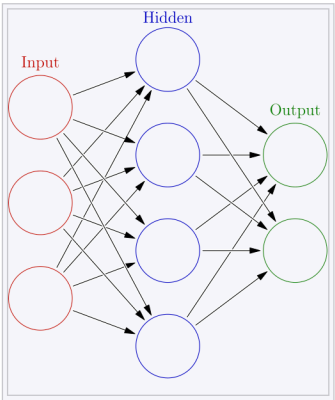
Conclusion

Out of the 3, both the accuracies of XGBoost and Random Forest were similar, Random Forest classifier algorithm gave an accuracy of 72.44%. While, XGBoost algorithm gave an accuracy of 72.77%. But, lightGBM stands at 69.80% accuracy, less compared to the rest 2.

Comparison report for Machine Learning Models

For the above XGBoost algorithm, Random Forest classifier and LightGBM algorithms, based on the comparisons related to accuracies, f-1scores and support factors, there was a visualization created, please see below.

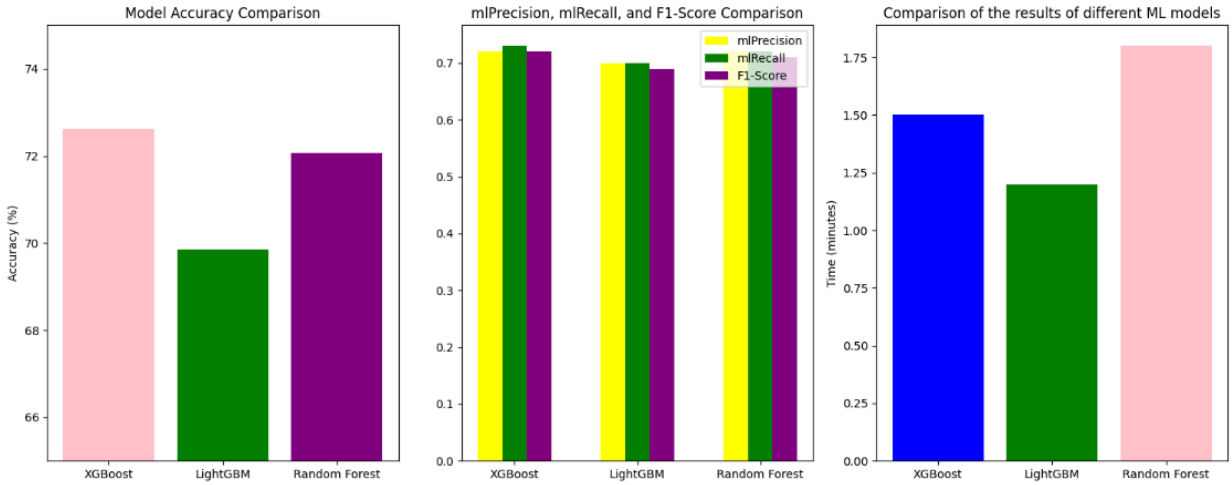
Neural Network:



An artificial neural network is an interconnected group of nodes, inspired by a simplification of neurons in a brain. Here, each circular node represents an artificial neuron and an arrow represents a connection from the output of one artificial neuron to the input of another.

According to the wikipedia, https://en.wikipedia.org/wiki/Neural_network_(machine_learning), In machine learning, a neural network (also artificial neural network or neural net, abbreviated ANN or NN) is a model inspired by the structure and function of biological neural networks in animal brains. Here, in our project, we have used, an NN based on a multi-class classification problem. In this case, it receives structured data as input, extracts the most relevant numerical features, and predicts categorical labels. This pipeline pre-processes date-time into Year, Month, Day, Hour, cleans missing values, standardizes numerical features, and finally encodes categorical target variables. The dataset is then divided into training, validation, and testing sets in the ratio 70%, 15%, and 15%, respectively, to maintain a balanced evaluation. The architecture of the neural network is deep feed-forward, comprising three hidden layers of 256, 128, and 64 neurons, respectively; each is followed by Batch Normalization and Dropout layers to avoid overfitting. The last output layer is Softmax-activated, allowing for multi-class classification, hence predicting one of the possible crime incident types in the dataset.

The performances of these models are evaluated in terms of accuracy scores, classification reports, and confusion matrices showing the detailed insights into the precision, recall, and F1-scores each for the different crime types. Visualization techniques also provide loss vs. epochs and accuracy vs. epochs plots to track model improvement. A structured approach has been followed with a neural network in such a manner that the accuracy remains greater than 75%, generalizes well, and does not overfit-a robust solution for real-world crime data classification.



PyTorch Intro, Data analysis & NN Models

CSE 676-B: Deep Learning, Spring 2025

Sharanya Nallapeddi

snallape@buffalo.edu

A Neural Network and trained using PyTorch and after running the code for it, the output is shown to the right.

Similarly, for details related to accuracy, precision, f-1 score and support, please see below.

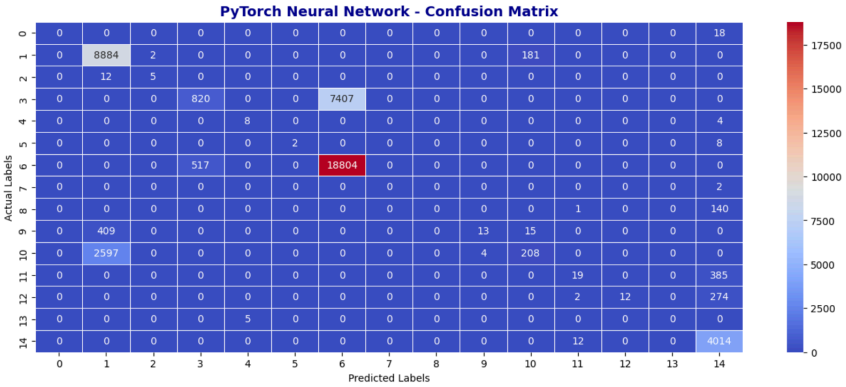
```
Classification Report is given by:
                              precision    recall  f1-score   support

                AGGR ASSAULT       0.00      0.00      0.00        18
                     ASSAULT       0.75      0.98      0.85      9067
                     Assault       0.71      0.29      0.42        17
                    BURGLARY       0.61      0.10      0.17      8227
     Breaking & Entering           0.62      0.67      0.64        12
CRIM NEGLIGENT HOMICIDE            1.00      0.20      0.33        10
              LARCENY/THEFT        0.72      0.97      0.83     19321
                MANSLAUGHTER       0.00      0.00      0.00         2
                      MURDER       0.00      0.00      0.00       141
                        RAPE       0.76      0.03      0.06       437
                     ROBBERY       0.51      0.07      0.13      2809
              SEXUAL ABUSE         0.56      0.05      0.09       404
     THEFT OF SERVICES             1.00      0.04      0.08       288
                       Theft       0.00      0.00      0.00         5
                         UUV       0.83      1.00      0.90      4026

                    accuracy                           0.73     44784
                   macro avg       0.54      0.29      0.30     44784
                weighted avg       0.70      0.73      0.65     44784
```

```
Epoch 1/50, Loss: 2133.0858, Train Acc: 0.7188, Val Acc: 0.7279
Epoch 2/50, Loss: 1979.4999, Train Acc: 0.7258, Val Acc: 0.7312
Epoch 3/50, Loss: 1962.0031, Train Acc: 0.7266, Val Acc: 0.7293
Epoch 4/50, Loss: 1953.8784, Train Acc: 0.7273, Val Acc: 0.7305
Epoch 5/50, Loss: 1948.2874, Train Acc: 0.7286, Val Acc: 0.7304
Epoch 6/50, Loss: 1940.7181, Train Acc: 0.7281, Val Acc: 0.7313
Epoch 7/50, Loss: 1937.8453, Train Acc: 0.7273, Val Acc: 0.7312
Epoch 8/50, Loss: 1932.0628, Train Acc: 0.7282, Val Acc: 0.7318
Epoch 9/50, Loss: 1930.9088, Train Acc: 0.7289, Val Acc: 0.7317
Epoch 10/50, Loss: 1927.5639, Train Acc: 0.7288, Val Acc: 0.7308
Epoch 11/50, Loss: 1923.4437, Train Acc: 0.7292, Val Acc: 0.7304
Epoch 12/50, Loss: 1920.6853, Train Acc: 0.7289, Val Acc: 0.7317
Epoch 13/50, Loss: 1919.8859, Train Acc: 0.7297, Val Acc: 0.7279
Epoch 14/50, Loss: 1917.8583, Train Acc: 0.7293, Val Acc: 0.7322
Epoch 15/50, Loss: 1916.5316, Train Acc: 0.7295, Val Acc: 0.7335
Epoch 16/50, Loss: 1916.1493, Train Acc: 0.7293, Val Acc: 0.7326
Epoch 17/50, Loss: 1913.9606, Train Acc: 0.7295, Val Acc: 0.7325
Epoch 18/50, Loss: 1912.1236, Train Acc: 0.7296, Val Acc: 0.7316
Epoch 19/50, Loss: 1908.7061, Train Acc: 0.7300, Val Acc: 0.7330
Epoch 20/50, Loss: 1906.7362, Train Acc: 0.7303, Val Acc: 0.7320
Epoch 21/50, Loss: 1905.2929, Train Acc: 0.7302, Val Acc: 0.7311
Epoch 22/50, Loss: 1904.1399, Train Acc: 0.7309, Val Acc: 0.7313
Epoch 23/50, Loss: 1905.4037, Train Acc: 0.7308, Val Acc: 0.7321
Epoch 24/50, Loss: 1902.3716, Train Acc: 0.7298, Val Acc: 0.7324
Epoch 25/50, Loss: 1902.2251, Train Acc: 0.7308, Val Acc: 0.7317
Epoch 26/50, Loss: 1899.8565, Train Acc: 0.7306, Val Acc: 0.7329
Epoch 27/50, Loss: 1901.6828, Train Acc: 0.7304, Val Acc: 0.7332
Epoch 28/50, Loss: 1900.7097, Train Acc: 0.7313, Val Acc: 0.7322
Epoch 29/50, Loss: 1898.1519, Train Acc: 0.7301, Val Acc: 0.7324
Epoch 30/50, Loss: 1897.3766, Train Acc: 0.7302, Val Acc: 0.7328
Epoch 31/50, Loss: 1896.5571, Train Acc: 0.7305, Val Acc: 0.7330
Epoch 32/50, Loss: 1894.4299, Train Acc: 0.7306, Val Acc: 0.7341
Epoch 33/50, Loss: 1894.4883, Train Acc: 0.7311, Val Acc: 0.7333
Epoch 34/50, Loss: 1894.0322, Train Acc: 0.7305, Val Acc: 0.7332
Epoch 35/50, Loss: 1893.9074, Train Acc: 0.7316, Val Acc: 0.7338
Epoch 36/50, Loss: 1891.0725, Train Acc: 0.7305, Val Acc: 0.7330
Epoch 37/50, Loss: 1890.5481, Train Acc: 0.7313, Val Acc: 0.7338
Epoch 38/50, Loss: 1890.5536, Train Acc: 0.7314, Val Acc: 0.7332
Epoch 39/50, Loss: 1890.9813, Train Acc: 0.7319, Val Acc: 0.7325
Epoch 40/50, Loss: 1890.6376, Train Acc: 0.7316, Val Acc: 0.7313
Epoch 41/50, Loss: 1887.9570, Train Acc: 0.7318, Val Acc: 0.7326
Epoch 42/50, Loss: 1886.7965, Train Acc: 0.7319, Val Acc: 0.7330
Epoch 43/50, Loss: 1888.4429, Train Acc: 0.7315, Val Acc: 0.7339
Epoch 44/50, Loss: 1887.9584, Train Acc: 0.7317, Val Acc: 0.7334
Epoch 45/50, Loss: 1887.0943, Train Acc: 0.7320, Val Acc: 0.7330
Epoch 46/50, Loss: 1886.5213, Train Acc: 0.7316, Val Acc: 0.7335
Epoch 47/50, Loss: 1885.6022, Train Acc: 0.7314, Val Acc: 0.7303
Epoch 48/50, Loss: 1886.3249, Train Acc: 0.7322, Val Acc: 0.7323
Epoch 49/50, Loss: 1883.4355, Train Acc: 0.7322, Val Acc: 0.7332
Epoch 50/50, Loss: 1883.3748, Train Acc: 0.7322, Val Acc: 0.7325
Training Time: 910.12 seconds
```

After the above mathematical implementation, a confusion matrix for the PyTorch Neural Network was generated.



For the Neural Network implemented, the PyTorch-developed neural network has been trained with 70% of the data, validated on 15%, and then tested on the last 15% for a balanced evaluation. Architecture: It is composed of three hidden layers containing 256, 128, and 64 neurons, respectively, all with Batch Normalization and Dropout after each one, to prevent overfitting, with a ratio of 30% on the first two and 20% on the last layer. The final layer is the output layer soft-max with N classes. Here, N equals the unique number of categories of crimes available in the dataset. Also, for the models performance, the confusion matrix was utilized. And the accuracy for it is **73.29%.**

"Artificial Intelligence, deep learning, machine learning — whatever you're doing if you don't understand it — learn it. Because otherwise you're going to be a dinosaur within 3 years." ~Mark Cuban.

References

• City of Buffalo. (n.d.). *Crime Incidents*. Buffalo Open Data. Retrieved February 7, 2025, from https://data.buffalony.gov/Public-Safety/Crime-Incidents/d6g9-xbgu/about_data

• Zhang, Z., & Barr, A. (2024). *Gentrification and crime in Buffalo, New York*. *PLOS ONE*, 19(6), e0302832. https://doi.org/10.1371/journal.pone.0302832