Predictive Analysis of Crime Incidents Using Neural Networks

Siddhartha Dheer
Deep Learning (CSE 676-B)

1. Introduction

The dataset I selected for this project, `Crime_Incidents_20250604.csv`, has over 322,000 individual crime incident reports from the city of Buffalo, New York. I selected this dataset because it has a lot of detail in the data, it is a very granular dataset, and it is more representative of real-world public safety data that has been collected over years. Each row in the dataset is an incident for a single reported crime, maintained with multiple associated features that describe the what, when, where, and sometimes how of the crime.

Here's what's included:

1. **Incident Information:** Each row in the dataset contains a primary classification for the incident in a column called `Incident Type Primary`. This primary classification represents what type of crime occurred – theft or theft-related incidents, assault and battery, homicide (including murder or manslaughter), breaking and entering (including attempted and aggravated breaking and entering), robbery, and different sexual offenses.
2. **Date and Time:** The `Incident Datetime` column gives the exact date and time of the incident. This allows the extraction of derived features that show when crimes occurred, like `Hour of Day` and `Day of Week` to see if crime occurs periodically.
3. **Geo-location:** Each incident in the dataset was geotagged with a latitude and longitude (Geo-location), allowing us to map incidents across neighborhoods and areas. Geo-location preserves the spatial aspects of incident data and shows us how that spatial component relates to crime regarding classification and frequency.
4. **Census and Address Data:** Census Tract data (not used in this study) shows demographic or social economic details, as do the abundance of Address and sparse ZIP Code values. This information could be useful for designing evidence-based policies, urban downtown planning initiatives, or targeting programs or interventions.
5. **Description (Optional):** The `Incident Description` column documents the running narrative storyline or sub-category. Rather than model this field directly into our analysis, we simply removed it from consideration because of redundancy in sub-category levels and too much sparsity in many themes.

2. Dataset Description

Dataset Name: `Crime_Incidents_20250604.csv`
Total Records: 322,519 rows

Class Distribution

Class	Count
theft	133,330
assault	61,226
breaking & entering	54,738
theft of vehicle	30,151
robbery	18,711
sexual assault	2,398
other sexual offense	2,135
sexual offense	1,117
homicide	1,065

Table 1: Distribution of Target Classes

Basic Statistics

Feature	Mean	Std	Min	Max	Missing
Latitude	42.91	0.02	42.60	43.01	0
Longitude	-78.83	0.03	-79.00	-78.50	0
Hour of Day	13.0	6.5	0	23	0
Census Tract	–	–	0	170+	Few

Table 2: Key Feature Statistics

3. Preprocessing Techniques

To prepare the dataset for training and evaluation, the following preprocessing activities were performed:

1. **Datetime Feature Engineering:**
   The Incident Datetime column was parsed into two

temporal features: Hour of Day and Day of Week, in order to capture periodic cycles that might exist (e.g., spikes in the evening, weekends).

2. **Geolocation Cleanup:**
The latitude and longitude measurements were cleaned through the use of geographic boundaries, thereby discarding disqualified latitude and longitude measurements that were not located in viable coordinates located in Buffalo (i.e., invalid points, or incorrectly logged coordinates).

3. **Feature Normalization:**
Continuous numerical attributes, such as the latitude, longitude, and Hour of Day, were normalized using the MinMaxScaler to map all values to the [0, 1] range, which would improve model convergence and stability during training.

4. **Categorical Encoding:**
The target variable, Incident Type Primary was label-encoded to convert class names to integers as indices for those class names. While encoded categorical variables like Day of Week were set up in a One-Hot encoding style to preserve categorical individuality and avoid the representation of the day of week in ordinal fashion.

5. **Feature Selection and Cleaning:**
Incident ID, Description, and address fragments - which were proved confusing by their sparsity or high redundancy—were removed with non-informative variables. This served to strip futility and reduce total weight of features in the eventual model.

4. Visualizations

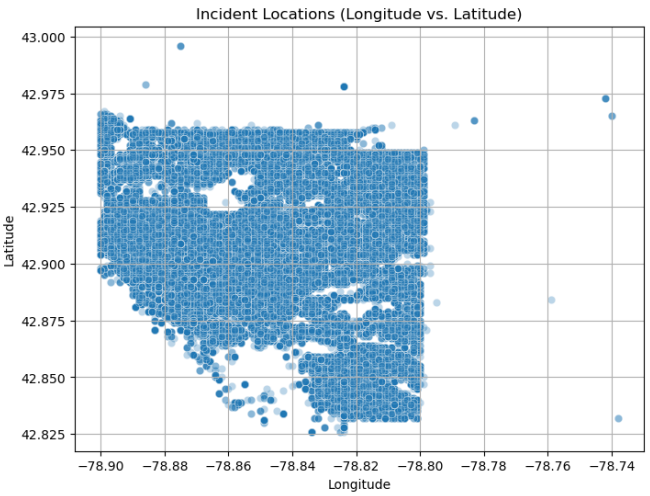

Figure 1: Cleaned Crime Locations

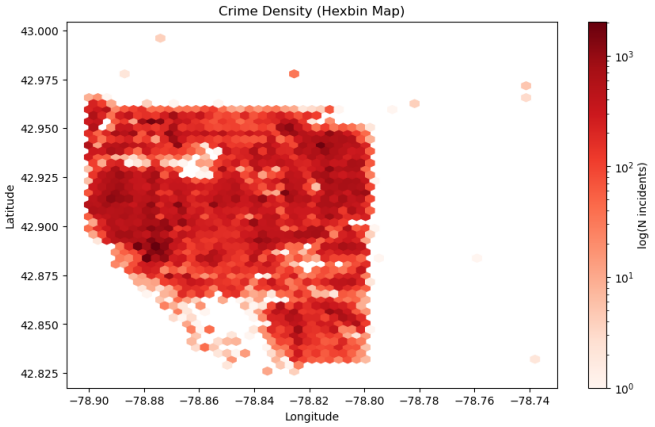

Figure 2: Crime Density (Hexbin Map)

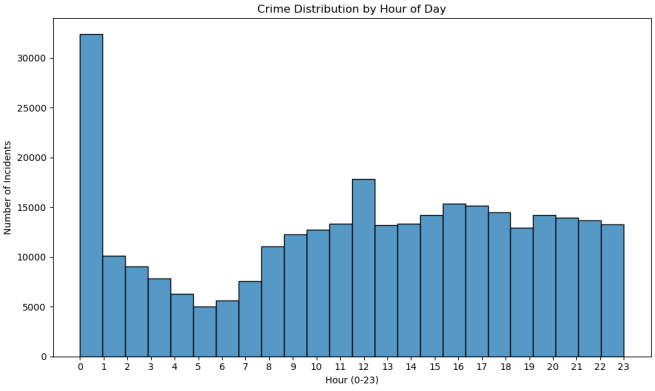

Figure 3: Crime Distribution by Hour

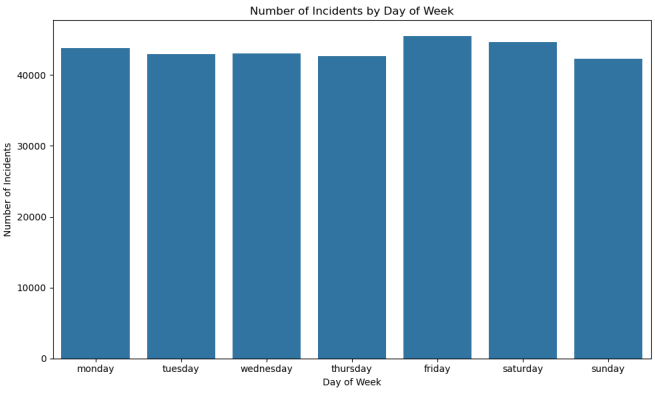

Figure 4: Crimes by Day of Week

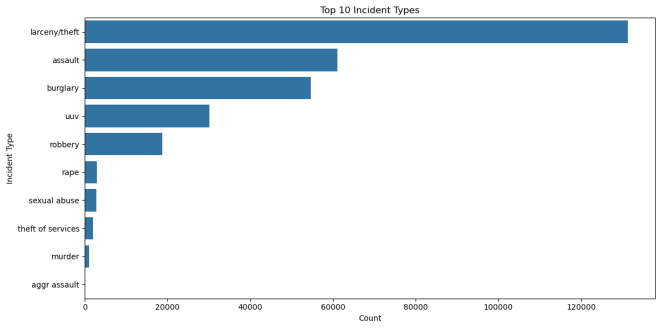

Figure 5: Top 10 Crime Types

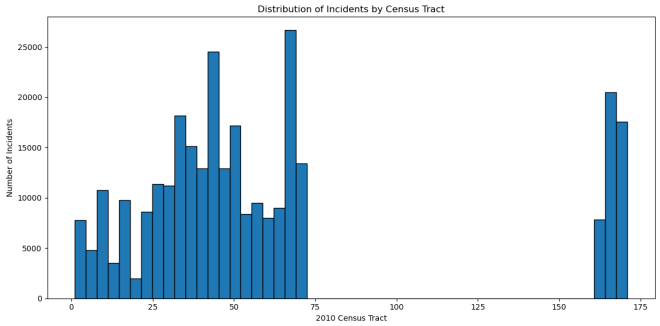

Figure 6: Incidents by Census Tract

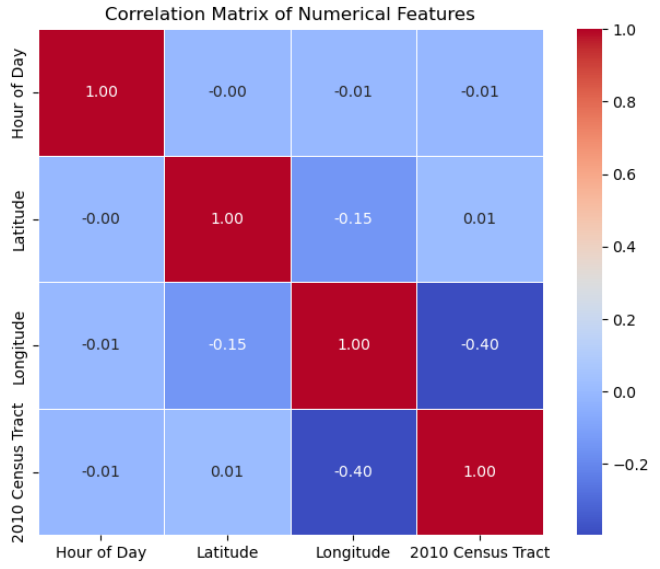

Figure 7: Correlation Heatmap

5. Model Description

Logistic Regression

Logistic Regression is used as the benchmark model for multi-class classification and implemented with Scikit-learn using the `LogisticRegression` class with the following specifications:

1. **Solver: lbfgs**
   The limited-memory Broyde-Fletcher-Goldfarb-Shanno (L-BFGS) algorithm which is a quasi-Newton optimization method was used since it is appropriate for multiclass problems and tends to converge efficiently with moderately sized datasets.

2. **Classification Type: Multinomial**
   The model was ensured true multiclass classification (based on softmax) through `multi_class='multinomial'` rather than through one-vs-rest.

3. **Maximum Iteration: 1500**
   The maximum number of training iterations was increased to 1500 to ensure convergence, especially in the context of the large amount of data and significant classes imbalance.

Neural Network Architecture

Layer	Configuration
Input	Encoded input features
Hidden 1	Linear →128 + ReLU + Dropout(0.3)
Hidden 2	Linear 128→64 + ReLU + Dropout(0.3)
Hidden 3	Linear 64→32 + ReLU
Output	Linear 32→9 classes (softmax)

Table 3: Neural Network Layers

Training Setup:

- Loss: `CrossEntropyLoss`
- Optimizer: `Adam`, LR = 0.001
- Epochs: 15
- Batch Size: 256

Training Metrics

Epoch	Loss	Val Accuracy	Time
1	273.24	0.9702	0.70s
5	43.13	0.9959	0.50s
10	19.67	0.9964	0.50s
15	13.83	**0.9964**	0.50s

Table 4: NN Training Log (Selected Epochs)

6. Evaluation and Results

In this section, we will discuss the evaluation of the basic logistic regression model and the suggested neural network classifier. Both sets of models were evaluated/assessed on validation accuracy, test accuracy, and

F1-score to not only evaluate overall performance, but also to see how well the models deal with the class imbalance.

Logistic Regression

Validation Accuracy: 97.04%
Test Accuracy: 97.11%
Macro F1 Score: 0.54

Although the logistic regression model is a good fit with a high accuracy score, the macro-averaged F1 score shows the model is unable to accurately classify the minority classes. The performance in the classification among incident types is heavily weighted toward the majority classes such as theft and assault.

Neural Network

Validation Accuracy: 99.64%
Test Accuracy: 97.11%

The neural network shows excellent generalization performance, and has much better validation accuracy when compared to the logistic regression model. It handles class imbalance more effectively because of its nonlinear decision boundaries and deeper feature learning.

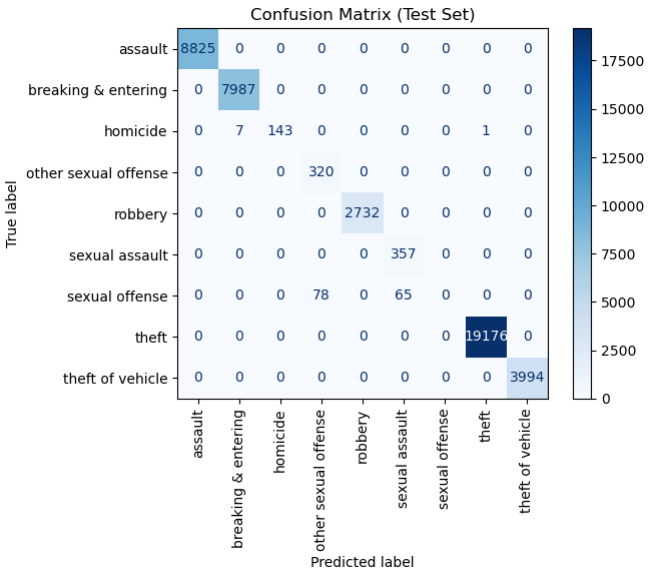

Figure 8: Confusion Matrix - Neural Network Predictions on Test Set

Model Comparison

Metric	Logistic Regression	Neural Network
Val Accuracy	97.04%	99.64%
Test Accuracy	97.11%	97.11%
Minority Recall	Poor	Slightly Better

Table 5: Performance Comparison

7. Conclusion and Future Work

Conclusion:
The neural network model showcased significant predictive power with high accuracy for the most common instances of crime and a modest improvement in identifying the less common instances of crime. In general, the neural network model outperformed a logistic regression baseline by discovering deeper representations and nonlinear patterns in the data.

Limitations:

1. The model's performance for the minority classes is limited by the data imbalance. The major impact of this class imbalance is lower recall for the final model for the rare types of crime that we were interested in predicting.
2. The model sometimes struggles to separate the low-frequency crimes into their own categories, basically because there is not enough data for the model to distinguish the categories.

Future Work:

1. Address the issue of data imbalance by attempting to implement methods like SMOTE or a loss function class-weighted for the different forensic types, this way improving minority recall.
2. Investigate more complex architecture like LSTMs or transformers to use the temporal sequences in the data to dirt deeper into creating temporal patterns for the forensics.
3. Pull in more external contextual features like weather, holidays, and special events to increase the contextual awareness that the model has when predicting.

Appendix

- **Dataset:** `Crime_Incidents_20250604.csv`
- **Libraries Used:** PyTorch, Scikit-learn, Matplotlib, Seaborn
- **Code Repository:** Full implementation and scripts are included in the project submission folder.