New york crime prediction using comparaision between machine learning and deep learning models

Aya TRABELSI, Nour LABIDI, Yasmine MAHDOUI

***Abstract*—This paper presents a comparison between two machine learning models for predicting crime in New York. It presents the design of a user-friendly web application that enables individuals to provide personal data and specify a location within the city. Leveraging advanced algorithms, the system forecasts potential criminal activity in the selected area. The paper examines the methodology, the comparison and selection of models, and the web application's implementation. The findings emphasize the approach's effectiveness in raising crime awareness and supporting decision-making for both users and law enforcement within the New York setting.**

***Index Terms*—Crime Prediction, deep learning, Machine Learning, NYPD Dataset.**

I. Introduction

Crime analysis and prediction are critical components in modern urban safety and law enforcement strategies. Accurate predictions of crime characteristics can assist law enforcement agencies in resource allocation and crime prevention efforts.

In this paper, we propose a comparison between two machine learning models and then choosing the model that performs better and simultaneously predicts four key attributes of a crime: *crime type*, *suspect age group*, *suspect race*, and *suspect sex*. The model processes data with diverse feature types, including temporal, contextual, demographic, and geographic information. The primary objective of this work is to demonstrate the feasibility and effectiveness of such a models.

II. Related Work

Several studies have explored crime prediction using machine learning and deep learning techniques as in [1] they They carried out an extensive analysis of various crime prediction methods, including Support Vector Machines (SVM) and Artificial Neural Networks (ANN), and concluded that no single approach can universally overcome the challenges posed by diverse crime datasets. However, most approaches focus on single-task predictions such as crime type classification or geographic crime hotspots. Few studies address multi-task predictions where multiple outputs are predicted simultaneously. Our work bridges this gap by implementing a multi-output neural network tailored to the complex nature of crime prediction and comparing it to a traditional machine learning model.

III. Methodology

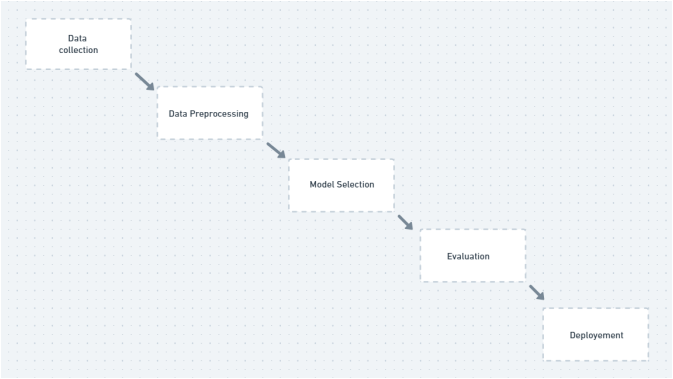

Fig. 1. Methodology workflow

A. Data collection

Data collection is the foundational step in the machine learning workflow, where relevant information is systematically gathered and recorded from various sources to create a dataset. This stage is critical to ensure the availability of sufficient and representative data, as the quality, quantity, and diversity of the data directly influence the performance and generalizability of the machine learning models.

B. Data Preprocessing

Data preprocessing is the process of preparing raw data for machine learning by addressing inconsistencies, missing values, and outliers, as well as ensuring uniformity and compatibility with algorithms. This step typically involves:

- **Cleaning**: Rectifying errors, handling missing or duplicated data.
- **Exploration:** Understanding the dataset's structure, distributions, and relationships through descriptive statistics and visualization.
- **Modifying:** Transforming features, encoding categorical variables, and scaling or normalizing numerical data to optimize the dataset for the chosen models.

C. Model selection

Model selection involves identifying and implementing machine learning algorithms, evaluating their performance, and choosing the model that best addresses the problem at hand. This step ensures that the chosen model not only fits the data

well but also generalizes effectively to unseen scenarios. In this work, two popular models, Random Forest and XGBoost, are used as candidates due to their proven robustness, ability to handle structured data efficiently, and high performance in predictive tasks.

- **Random Forest** Random Forest is an ensemble learning method that combines the predictions of multiple decision trees to improve accuracy and control overfitting. It operates by training several decision trees on randomly sampled subsets of the dataset, using a technique called bagging (bootstrap aggregating). The final prediction is determined by aggregating the outputs of all individual trees, either through majority voting (for classification) or averaging (for regression). Random Forest is well-suited for handling datasets with missing values, outliers, and non-linear relationships, making it a versatile choice for structured data.
- **XGBoost (Extreme Gradient Boosting)** XGBoost is a gradient-boosting framework designed for efficiency, speed, and scalability. Unlike Random Forest, which aggregates independent trees, XGBoost builds trees sequentially, with each tree correcting the errors of the previous one. This iterative process minimizes a predefined loss function, resulting in a model that focuses on the most challenging predictions. XGBoost employs techniques like regularization to reduce overfitting and parallel processing to enhance computational performance. It is particularly effective for complex structured datasets and is widely used in competitive machine learning tasks due to its superior accuracy and flexibility.

D. Evaluation

Model evaluation is the process of assessing the performance of the selected machine learning model using a separate validation or test dataset. Key metrics, such as accuracy, precision, recall, F1-score, and others, are used to quantify the model's predictive ability and generalizability. This step ensures that the model meets the required standards and can reliably address the problem in practical applications.

IV. Implementation

A. Data Collection

The NYPD Complaint Data Historic dataset encompasses all recorded felony, misdemeanor, and violation incidents reported to the New York City Police Department (NYPD) from 2006 through the end of 2019. It comprises a total of 6,901,167 complaints distributed across 35 columns, providing spatial and temporal details about crime occurrences, along with their descriptions and penal codes.

B. Data preprocessing

1) ***Data Cleaning****:* The data cleaning step involves enhancing the dataset quality to achieve better prediction results.

CMPLNT_FR_DT	655
CMPLNT_FR_TM	48
CMPLNT_TO_DT	1668376
CMPLNT_TO_TM	1663829
ADDR_PCT_CD	2166
OFNS_DESC	18825
PD_CD	5865
PD_DESC	5865
CRM_ATPT_CPTD_CD	7
BORO_NM	10894
LOC_OF_OCCUR_DESC	1479992
PREM_TYP_DESC	39732
JURISDICTION_CODE	5865
PARKS_NM	6958159
HADEVELOPT	6637179
HOUSING_PSA	6446298
X_COORD_CD	24064
Y_COORD_CD	24064
SUSP_AGE_GROUP	4707573
SUSP_RACE	3339032
SUSP_SEX	3472346
TRANSIT_DISTRICT	6826123
Latitude	24064
Longitude	24064
Lat_Lon	24064
...	
VIC_AGE_GROUP	1638444
VIC_RACE	308
VIC_SEX	307
dtype: int64	

Fig. 2. Missing values in the dataset

The initial preprocessing phase focused on addressing missing values by either removing columns with substantial null entries or imputing binary indicators for certain categorical variables. Date and time columns were standardized by converting them into datetime objects, and rows lacking critical temporal information were excluded. To enhance temporal insights, new features such as year, month, day, hour, and weekday were extracted from the incident date.

Further data cleaning efforts targeted missing values and inconsistencies in demographic-related categorical variables. Missing or unknown entries were imputed appropriately to ensure a more complete dataset. Redundant or irrelevant columns were eliminated to streamline the data structure. Additionally, a new categorical feature was created to classify crimes into specific categories, enabling a more organized and interpretable analysis.

These preprocessing steps were designed to maintain the dataset's integrity, enhance its usability, and establish a robust foundation for exploratory data analysis and subsequent modeling.

2) ***Data Exploration****:* Data exploration plays a vital role in the data analysis process, focusing on examining and visualizing data to identify trends, patterns, and relationships. Using both statistical and graphical methods, this phase helps in understanding the dataset's structure, spotting any potential outliers, and guiding further analysis. It acts as an initial step for becoming acquainted with the data, which is essential

for making informed decisions, formulating hypotheses, and adjusting or removing features. The following plots offer a visual representation of the key patterns and trends discovered during the Exploratory Data Analysis, giving a detailed view of the dataset's characteristics.

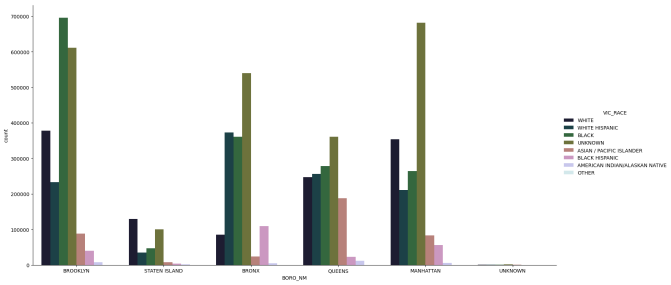

Fig. 3. Victime's caracteristics / BORO name

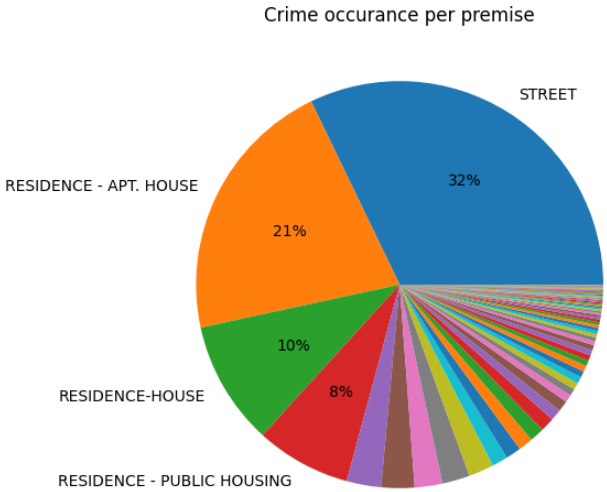

Fig. 5. Crimes zone of occurence

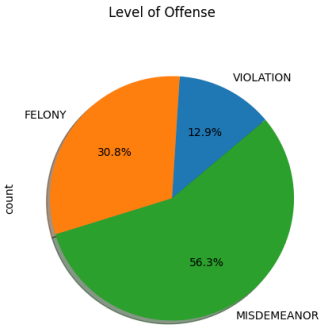

Fig. 6. level of offense

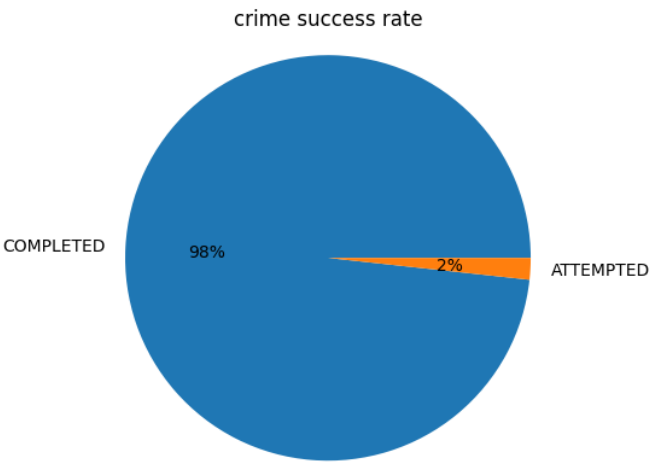

Fig. 4. Crime's success rate

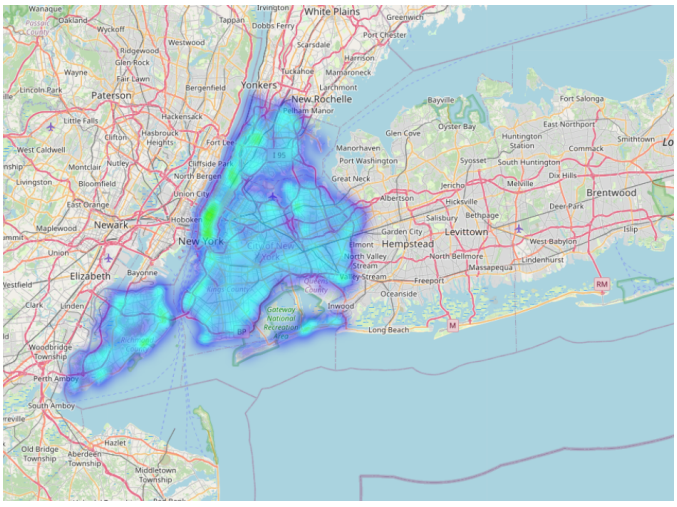

Fig. 7. crimes heatmap in New york

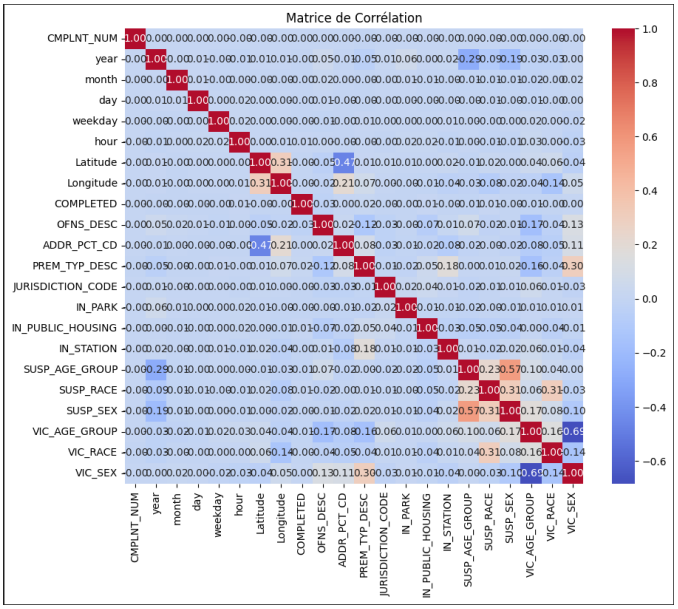

Fig. 8. Correlation matrix

*3) **Data modification**:* To better understand the relationships between the features in our dataset, we utilized a correlation matrix. This matrix allowed us to examine the strength and direction of the linear relationships between the variables. By identifying features with weak or no correlation to the target variable, we were able to determine which features might be redundant or irrelevant to the predictive model. This step was essential in streamlining the dataset and ensuring that only the most relevant features were retained for further analysis and modeling. After conducting the data exploration and analyzing the correlation matrix, we identified several features that were not contributing significantly to the model's performance. As a result, we eliminated these irrelevant features and retained the following important ones: year, month, day, hour, Latitude, Longitude, victime age group,victime RACE, suspect age group, suspect race, and suspect sex. To further enhance the dataset and improve the predictive capabilities of our model, we introduced a new column, codeplus, which is derived from the Plus Codes system. The Plus Code is a global addressing system that encodes geographic coordinates (latitude and longitude) into a short string of characters. This system enables the representation of locations with high precision, making it particularly useful for geospatial data.

To generate the codeplus column, we applied the Open Location Code (OLC) library to encode the latitude and longitude of certain locations into Plus Codes. Specifically, we set the precision level to 10, which corresponds to an area of approximately 14x14 meters. This transformation allows the model to leverage spatial proximity, making predictions more accurate by grouping nearby locations with similar codes.

This approach helps encode geographical information in a more structured manner, allowing our model to take spatial relationships into account when making predictions, ultimately improving its accuracy.

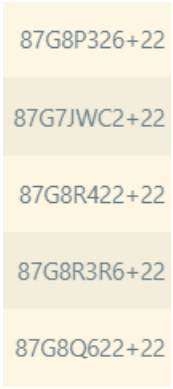

Fig. 9. code plus example

C. Model selection

This section presents the two machine learning models utilized for predicting crime characteristics: **Random Forest** and **XGBoost**. These models were specifically applied to a **multi-output classification task**, where multiple interrelated target variables are predicted simultaneously.

1) Multi-Output Classification: An Overview: In traditional machine learning tasks, a model is often designed to predict a single target variable. However, in our case, we aim to predict **four distinct but interrelated target variables**:

1) `OFNS_DESC`: The offense category (e.g., property, personal).
2) `SUSP_AGE_GROUP`: The suspected age group of the offender.
3) `SUSP_RACE`: The suspected race of the offender.
4) `SUSP_SEX`: The suspected gender of the offender.

This introduces a *multi-output classification problem*, where each instance in the dataset is associated with multiple labels $y = \{y_1, y_2, \ldots, y_T\}$, each corresponding to a specific target variable. The goal is to build a model that predicts all target variables simultaneously, leveraging potential correlations between them.

2) Input-Output Structure: **Input Features** (X)

The model is trained on a combination of numerical and categorical input features that describe the circumstances of reported crimes:

$$X = \{x_1, x_2, \ldots, x_{13}\}$$

where each x_i corresponds to a feature, such as:

- `year`, `month`, `day`, and `hour`: Temporal attributes.
- `weekday`: Categorical feature indicating the day of the week.
- `PREM_TYP_DESC`: Type of premises (e.g., residence, public space).
- `IN_PARK`, `IN_PUBLIC_HOUSING`, `IN_STATION`: Binary indicators for location-specific attributes.
- `VIC_AGE_GROUP`, `VIC_RACE`, `VIC_SEX`: Attributes of the victim.

- `code_plus`: A geospatial identifier representing the approximate crime location.

Target Variables (y)

The targets represent crime details to be predicted:

$$y = \{y_1, y_2, y_3, y_4\}$$

where:

- $y_1 =$ `OFNS_DESC`
- $y_2 =$ `SUSP_AGE_GROUP`
- $y_3 =$ `SUSP_RACE`
- $y_4 =$ `SUSP_SEX`

3) Random Forest for Multi-Output Classification: **Methodology**

Random Forest handles multi-output classification by independently building decision trees for each target variable. Each tree in the forest predicts a single target variable y_t, and the predictions from all trees are aggregated to make the final prediction.

For each target y_t, the prediction is computed as:

$$\hat{y}_t = \text{argmax}_c \frac{1}{N} \sum_{i=1}^{N} \mathbb{I}(h_i(x) = c)$$

where N is the number of decision trees, $h_i(x)$ is the prediction of the i-th tree, and $\mathbb{I}$ is an indicator function.

Advantages

- **Simplicity:** Handles each target variable independently, simplifying implementation.
- **Robustness:** The ensemble approach reduces variance and enhances stability.
- **Correlation Exploitation:** While built independently, the shared feature space allows the model to capture implicit correlations between targets.

4) XGBoost for Multi-Output Classification: **Methodology**

XGBoost does not natively support multi-output classification, requiring separate models for each target variable. For each target y_t, XGBoost minimizes a loss function $\ell(y, \hat{y})$ defined as:

$$\mathcal{L}(\Theta) = \sum_{i=1}^{n} \ell(y_i, \hat{y}_i) + \sum_{k=1}^{K} \Omega(f_k)$$

where $\Omega(f_k)$ is a regularization term to penalize model complexity, and f_k represents the k-th tree.

Advantages and Drawbacks

- **Advantages:** Optimized for single-target tasks, with high predictive accuracy and regularization to prevent overfitting.
- **Drawbacks:** Separate models increase complexity, computational overhead, and memory usage.

5) Performance Comparison: The models were evaluated based on accuracy, recall, F1 score, and training time, with the results summarized in Table I.

TABLE I
PERFORMANCE COMPARISON BETWEEN RANDOM FOREST AND XGBOOST.

Model	Accuracy (%)	Recall (%)	F1 Score (%)
Random Forest	**78.5**	**76.8**	**77.2**
XGBoost	75.0	73.2	74.1

6) Evaluation: The results in Table I indicate that:

- **Random Forest** achieves higher accuracy, recall, and F1 score, making it better suited for tasks where minimizing false negatives is critical.
- **XGBoost** trains faster, but the requirement for separate models increases implementation complexity.

Given its superior performance on key metrics and simpler deployment for multi-output tasks, Random Forest was selected as the final model.

D. deployement on streamlit

This section introduces the web application designed to predict crime characteristics based on user-provided inputs and geographic selection. The application was developed using the **Streamlit** framework, enabling interactive and user-friendly functionalities. It leverages the trained Random Forest model to make predictions.

1) Application Workflow: The workflow of the application is divided into three main steps:

1) **Region Selection:** Users can interact with an interactive map centered on New York City to select a specific region. By clicking on the map, the latitude and longitude of the chosen location are captured and used as part of the prediction inputs.
2) **User Input:** Users provide additional details about the crime, such as:
   - Date and time (*year, month, day, hour*),
   - Day of the week,
   - Type of location (e.g., street, park, residence),
   - Victim's demographic details (age group, race, gender),
   - Indicators for whether the crime occurred in a park, public housing, or a station.
3) **Prediction and Decoding:** The application encodes user inputs into a format compatible with the trained model. Once the prediction is made, the results are decoded back into their original human-readable categories using mappings derived from the dataset.

2) Implementation Details: The application integrates several technologies and libraries:

- **Streamlit:** Provides an intuitive interface for user interaction.
- **Folium:** Enables map visualization and coordinate selection.
- **Scikit-learn:** Loads the trained Random Forest model for prediction.
- **Pandas:** Handles data transformation and encoding/decoding processes.

3) Example of Prediction: Once the user provides the required inputs and clicks the "Predict" button, the application processes the data and returns the following outputs:

- **Type of Crime (OFNS_DESC):** Describes the category of the crime, e.g., theft, assault.
- **Suspect's Age Group (SUSP_AGE_GROUP):** Provides an estimated age range of the suspect.
- **Suspect's Race (SUSP_RACE):** Indicates the likely racial group of the suspect.
- **Suspect's Gender (SUSP_SEX):** Predicts the gender of the suspect.

4) User Interface: The application's user interface. The interface comprises:

- An interactive map for region selection,
- Input fields for date, time, and crime details,
- A button to trigger the prediction process,
- A result section displaying decoded predictions in a clear and concise format.

5) Significance and Practical Use: The application demonstrates the integration of machine learning models into an interactive web tool for practical use in crime analysis. By allowing users to visualize and input specific details, it provides actionable insights that can assist law enforcement and researchers in understanding crime patterns.

REFERENCES

[1] Shiju Sathyadevan, Devan M. S., Surya S. Gangadharan, "Crime Analysis and Prediction Using Data Mining", International Conference on Networks Soft Computing (ICNSC), 2014.