Comprehensive Crime Analysis and Prediction System: A Multi-Model Approach

December 6, 2024

Abstract

This report presents an integrated crime analysis and prediction system for Buffalo, NY, combining multiple machine learning approaches including Random Forest Classification, XGBoost Regression, and Facebook Prophet models. Additionally, an SVM classifier has been implemented to predict crime types, along with advanced visualization features such as 3D crime maps and statistical charts. The system provides comprehensive crime pattern analysis, threat assessment, and forecasting capabilities through an interactive web interface.

1 Introduction

Urban crime analysis and prediction have become essential tools for law enforcement and policymakers. The increasing availability of open data and advances in machine learning have made it possible to develop sophisticated systems for analyzing and predicting criminal activity. This project integrates multiple machine learning techniques to address various aspects of crime analysis and prediction, including:

- **Crime type prediction**: Utilizing Random Forest Classification to categorize crimes based on historical data.
- **Threat score assessment**: Implementing XGBoost Regression to assign a threat score to specific incidents or regions.
- **Time-series forecasting**: Employing Facebook Prophet to model and predict crime trends over time.
- **Crime likelihood prediction**: Using Support Vector Machines (SVM) to estimate the probability of different crime types in specific areas.

The system is designed to assist stakeholders by providing actionable insights through user-friendly visualizations and predictive capabilities.

2 Dataset and Preprocessing

The dataset utilized in this project is sourced from Buffalo's open data portal, encompassing crime incident reports from 2009 onward. The dataset includes a variety of features that provide comprehensive information about each incident. The key features are as follows:

Incident Details Information such as the type of crime, date, and time of occurrence provides a foundational understanding of criminal activity patterns. These attributes are critical for classification and regression tasks.

Location Geographic data, including coordinates and neighborhood identifiers, enable spatial analysis of crime distribution. This information is essential for generating 3D crime maps and understanding the geographic spread of incidents.

Temporal Features Temporal attributes, such as the hour of the day, day of the week, and month of the year, help in identifying trends and patterns that are time-dependent. These features are particularly important for time-series analysis.

2.1 Preprocessing Steps

To prepare the data for analysis and modeling, several pre-processing steps were undertaken:

Temporal Feature Extraction Date and time data were processed to derive additional features such as the day of the week, hour of the day, and month. These derived features enhance the predictive power of time-dependent models.

Coordinate Normalization Raw geographic coordinates were normalized to ensure consistency and accuracy in spatial analysis. This step is crucial for generating reliable spatial visualizations and location-based predictions.

Categorical Encoding Categorical variables, such as crime types and neighborhood identifiers, were converted into numerical representations using techniques like one-hot encoding. This transformation allows these features to be utilized effectively in machine learning models.

Data Aggregation The dataset was aggregated based on temporal and spatial features to identify high-level patterns. For example, crimes were grouped by neighborhood and time intervals to facilitate trend analysis and forecasting.

These preprocessing steps ensure that the data is clean, structured, and ready for use in the various machine learning models implemented in this project.

1) Context of the Question

This question pertains to the implementation of a machine learning model for crime risk prediction within the broader scope of a project focused on analyzing and mitigating crime patterns. The objective is to clarify how the model was trained, saved, and integrated into an interactive web application using Streamlit, and how it could process user inputs to generate meaningful predictions.

Details of the Implementation

The question involves a Random Forest Classifier model trained to predict the type of crime based on spatiotemporal and categorical inputs. The training process utilized features such as the day of the week, the hour of the day, and the neighborhood where the incident occurred. The target

variable was the primary type of crime. Both the features and the target variable were preprocessed using `LabelEncoder` to ensure compatibility with the machine learning pipeline.

The data was split into training and testing subsets, with 80% of the data allocated for training and 20% for testing. The Random Forest model was configured with 100 estimators and a fixed random state to ensure reproducibility. After training, the model demonstrated its capability to generalize patterns in the data effectively, making it suitable for deployment in an interactive environment.

Saving and Integration

The trained model and associated encoders were serialized using the `joblib` library. Specifically, three files were generated:

1. `crime_risk_model.pkl`: The trained Random Forest model.
2. `le_neighborhood.pkl`: Encoder for the neighborhood feature.
3. `le_incident_type.pkl`: Decoder for the predicted crime type.

These files were then utilized in a Streamlit application to provide predictions based on user inputs. The application workflow involves processing the user inputs through the saved encoders, passing the transformed data to the model, and decoding the predicted numerical output into a human-readable crime type.

Real-Time User Interaction

The integration of the model with Streamlit allows users to interact with the system in real-time. Users specify the day of the week, hour, and neighborhood through the web interface. These inputs are preprocessed using the saved encoders to match the training data format. The transformed inputs are fed into the model, which generates a prediction. The output is then decoded back into a readable format and displayed to the user. This seamless workflow ensures that the system can provide actionable insights promptly and effectively.

Evaluation and Summary

Although not explicitly stated in the question, the model's performance can be evaluated using metrics such as accuracy, precision, recall, and F1-score. These metrics provide insights into the reliability of the predictions and the model's ability to generalize.

In summary, this question illustrates the implementation and deployment of a crime risk prediction model. The described system highlights how machine learning can be integrated into interactive applications to provide real-time, actionable insights based on historical data.

3 Model Selection and Data Preparation

The Facebook Prophet model was chosen for this crime forecasting project due to its robust handling of time series data with strong seasonal effects and its ability to manage missing data or outliers effectively. Prophet is particularly well-suited for crime data analysis as it can capture weekly patterns, yearly seasonality, holiday effects, and overall trends in crime rates. The data for this

project was sourced from a SQLite database containing incident reports from 2009 onwards. This historical data was preprocessed and aggregated by date and neighborhood to create daily crime counts for each area of Buffalo, NY.

3.1 Model Training

The training process involved creating separate Prophet models for each neighborhood in Buffalo, as well as an additional model for the entire city. This approach allows for the capture of both localized patterns specific to individual neighborhoods and broader trends across the city as a whole. Each model was fitted on the historical crime data extracted from the SQLite database. The models were then used to generate forecasts for the next 365 days, providing a year-long prediction of crime patterns for each neighborhood and the city overall. This granular approach enables a more nuanced understanding of crime trends across different areas of Buffalo.

3.2 Model and Forecast Storage

After the training process, both the trained models and their corresponding forecasts were saved for efficient retrieval and use in the Streamlit application. The Prophet models were serialized and saved as .pkl files using the joblib library, which provides utilities for saving and loading Python objects. Forecasts for each neighborhood and the entire city were saved as CSV files. A consistent naming convention was employed:

- `buffalo_crime_prophet_model.pkl` for the city-wide model,
- `buffalo_crime_forecast.csv` for the city-wide forecast,
- `prophet_model_neighborhood.pkl` for individual neighborhood models, and
- `forecast_neighborhood.csv` for neighborhood-specific forecasts.

This storage method facilitates quick loading of pre-trained models and forecast data in the Streamlit application, enhancing user experience and application performance.

3.3 Streamlit Application

The Streamlit application provides an intuitive and interactive interface for users to explore crime forecasts across different neighborhoods in Buffalo. Users can select a specific neighborhood from a dropdown menu, triggering the application to load the corresponding pre-trained model and forecast data. The application then displays a graph showing the predicted number of crimes over the next year, including both the forecast line and a shaded area representing the confidence interval. This visual representation allows users to easily interpret and compare crime forecasts across different neighborhoods, gaining insights into potential future crime trends in specific areas of the city. The interactive nature of the application empowers users to explore data dynamically, facilitating a deeper understanding of crime patterns in Buffalo.

3.4 Benefits of this Approach

This comprehensive approach to crime forecasting and visualization offers several key advantages. The pre-training of models and saving of forecasts allows for rapid loading and display in the Streamlit app, ensuring a responsive user experience. By using separate models for each neighborhood, the system can capture local patterns and trends that may differ significantly across various

areas of the city. The inclusion of a city-wide model provides an overall view of crime trends in Buffalo, offering a broader perspective alongside the neighborhood-specific forecasts. Additionally, the use of Prophet allows for easy updates as new data becomes available, ensuring that the forecasts can be kept current with minimal effort. This combination of granular neighborhood-level insights and city-wide trends, presented through an interactive interface, creates a powerful tool for analyzing and visualizing crime patterns in Buffalo, potentially aiding in resource allocation and crime prevention strategies.

4 Introduction to Threat score assessment

Crime and safety concerns in urban neighborhoods have become a significant issue for residents and authorities. This project focuses on developing a threat assessment system that predicts safety levels based on historical incident data by leveraging machine learning and advanced analytics. The system evaluates threat scores using parameters like neighborhood, time, and day of the week. This can assist law enforcement agencies, city planners, and residents in understanding and mitigating safety risks.

4.1 Dataset

The data is collected from historical incident reports stored in an SQLite database. Each record includes:

- **Case Details:** Descriptive information about the reported incident, such as type and severity.
- **Location:** Identified by neighborhood and corresponding geographic coordinates.
- **Time:** Date and time of occurrence, offering temporal patterns in criminal activity.

The dataset spans multiple years, providing a comprehensive view of crime trends and allowing for temporal and spatial threat assessment. By analyzing these records, the system calculates a composite threat score tailored to specific conditions, enabling meaningful and actionable predictions.

5 Methods

5.1 Modeling

The system employs an XGBoost Regressor (`XGBRegressor`) due to its efficiency and accuracy in handling structured datasets. The modeling process involved the following steps:

- **Data Splitting:** The dataset was divided into training and test sets using the `train_test_split` function, ensuring a balanced distribution of the features.
- **Model Training:** The model was trained with hyperparameters optimized for performance:
  - `n_estimators`=100
  - `learning_rate`=0.1
  - `max_depth`=6
- **Feature Importance:** The model evaluated the contribution of features such as neighborhood, day of the week, and hour, ranking them by their impact on predictions.

- **Model Deployment:** The trained model and supporting encoders were serialized for seamless integration with the web application.

5.2 Web Application

An intuitive web application was developed using Streamlit, providing users with real-time interaction to predict threat levels. Key features of the application include:

- **Input Interface:** Users specify parameters like neighborhood, day of the week, and time of day.
- **Visualization:** Threat levels are displayed on an interactive gauge chart powered by Plotly, offering clear and immediate insights.
- **Threat Categories:** Predictions are categorized into low, moderate, and high threat levels, with contextual guidance for safety precautions.

5.3 Results

The model demonstrated strong predictive performance on the test data. Evaluation metrics include:

- **Mean Squared Error (MSE):** 86.60
- **Mean Absolute Error (MAE):** 6.70
- **R^2 Score:** 0.90

The threat scores are normalized to a scale of 0-100 and categorized as follows:

- **Low Threat (0-33):** Indicates a safe environment suitable for travel or activities.
- **Moderate Threat (33-66):** Suggests exercising caution, particularly during less populated hours.
- **High Threat (66-100):** Warns of significant risk, advising individuals to avoid the area if possible.

6 Conclusion

This project successfully demonstrates the application of machine learning for predicting safety levels in urban neighborhoods. The system provides actionable insights for stakeholders by quantifying and categorizing threats.

6.1 Future Work

To improve the system further, the following enhancements are proposed:

- **Integration of Real-Time Data:** Incorporating live feeds of incident reports to enable dynamic threat assessments.
- **Mobile App Development:** Extending functionality to mobile platforms for wider accessibility.

7 Application Features

The Buffalo Crime Analysis Dashboard provides a comprehensive suite of tools and visualizations designed to analyze and forecast crime patterns within neighborhoods. The application leverages advanced predictive modeling and interactive data representations to empower users with actionable insights. Below, the primary features of the dashboard are elaborated upon in detail:

7.1 Neighborhood Crime Analysis and Incident Forecasting

The dashboard incorporates a predictive modeling feature to analyze neighborhood crime trends and forecast potential incidents. A Support Vector Machine (SVM) classifier forms the backbone of the prediction system. This model predicts the likelihood of specific crime types by analyzing various temporal features such as the hour of the day, day of the week, and month. The data preprocessing pipeline ensures high accuracy, utilizing a StandardScaler to normalize feature values and improve the model's performance.

Key capabilities of this feature include visualizing crime hotspots within neighborhoods using Pydeck's Hexagon Layer, which represents crime density in a clear and intuitive format. The dashboard also presents a pie chart to illustrate the distribution of different crime types in the selected neighborhood, offering users a snapshot of local crime dynamics. To enhance interactivity, sliders and dropdowns allow users to input specific time parameters and forecast incident probabilities accordingly. The likelihood of various crime types is presented with the top three categories ranked by their predicted probabilities. Terminology within this feature is carefully designed for clarity, replacing generic terms like "crime risk" with more precise labels such as "Incident Probability Analysis" and "Crime Pattern Analysis."

7.2 3D Crime Map

The 3D crime map is a standout feature of the dashboard, utilizing Pydeck to present a visually immersive and informative representation of crime data. The Hexagon Layer provides a bird's-eye view of crime density, where color-coded columns signify areas of varying crime intensity. Dark red columns represent the highest crime density, while light yellow columns denote the lowest. In addition to the Hexagon Layer, the Scatterplot Layer highlights individual crime incidents, complete with interactive tooltips that provide detailed information about each event. To aid interpretation, a comprehensive map legend is included, ensuring users can easily understand the color codes and their corresponding crime densities.

7.3 Crime Statistics

Crime statistics are presented through two primary visualizations that help users understand the broader patterns and trends:

- The first visualization is a pie chart depicting the distribution of various crime types. This chart allows users to quickly grasp the proportional representation of each type of crime in the dataset.
- The second visualization is a bar chart that highlights the distribution of crimes across different days of the week. This analysis reveals temporal patterns, such as the prevalence of certain crimes on specific days, aiding in strategic decision-making.

7.4 Peak Hours Analysis

To delve deeper into the temporal dimensions of crime, the dashboard features a heatmap that visualizes crime intensity by hour and day of the week. This tool helps identify peak crime hours and days, enabling law enforcement agencies to optimize patrol schedules and allocate resources more effectively. By providing a granular view of when crimes are most likely to occur, the heatmap serves as a critical component for strategic planning.

8 Conclusion

The Buffalo Crime Analysis Dashboard integrates data-driven visualizations and predictive modeling to offer valuable insights into urban crime patterns. By enabling users to explore historical crime data, identify high-crime areas and peak hours, and forecast potential incidents, the application serves as a powerful tool for both public safety officials and community members. Its intuitive design and advanced analytical features make it a versatile solution for crime analysis and prevention.

9 Future Work

To further enhance the capabilities of the dashboard, several potential improvements are proposed. Incorporating real-time crime data would enable live updates, allowing users to monitor ongoing incidents and adjust strategies dynamically. Adding demographic overlays could provide a socio-economic context to the analysis, shedding light on underlying factors contributing to crime patterns. Additionally, experimenting with advanced machine learning models, such as Gradient Boosting, could improve the accuracy and reliability of crime predictions, ensuring even better decision support for users.

10 References

- Buffalo Open Data Portal
- Pydeck Documentation
- Plotly Documentation
- `xgboost` Documentation: `https://xgboost.readthedocs.io/`
- Streamlit Documentation: `https://docs.streamlit.io/`
- Plotly Visualization: `https://plotly.com/`