Crime Data Analysis

P Mounika, C Shrynika, V Sushmitha, Ch Venkata Vardhini
[1]Assistant Professor, Department Of Cse, Bhoj Reddy Engineering College For Women, India.
[2,3,4]B. Tech Students, Department Of Cse, Bhoj Reddy Engineering College For Women, India.

ABSTRACT

In today's rapidly evolving world, crime prevention and law enforcement agencies face the challenge of managing and analyzing vast amounts of crime data to identify trends and understand patterns of criminal activity. With the growing importance of data-driven decision- making, law enforcement agencies and government bodies are increasingly relying on innovative technology to address crime-related challenges. To meet these needs, we are developing a Flask-based web application integrated with machine learning algorithms, providing a robust solution for real-time crime data analysis and insights.

This project aims to build an interactive platform that leverages crime data analysis and machine learning models to identify potential crime hotspots and uncover meaningful trends. By integrating data from various sources and offering advanced analytics, the platform will empower law enforcement agencies to take informed measures in addressing crime. Users will be able to visualize crime trends, access interactive heatmaps, and generate actionable insights based on historical crime data.

Furthermore, by combining machine learning models with dynamic visualizations, the platform enables users to explore data in an intuitive and user-friendly manner, simplifying the interpretation of complex patterns. The system is designed to handle large datasets, ensuring scalability and reliable performance even as data volumes increase.

The platform's modular design ensures ease of maintenance and adaptability, allowing for the integration of new datasets and analytical models in the future. Ultimately, the goal of this project is to provide a powerful tool that supports law enforcement agencies in making data- driven decisions to enhance public safety and effectively reduce crime rates.

1-INTRODUCTION

In today's data-driven world, understanding and addressing crime patterns is crucial for ensuring public safety and shaping effective law enforcement strategies. With the rise in urbanization and technological advancements, crime trends have become increasingly complex, requiring innovative solutions to analyze criminal activities. Crime data analysis provides valuable insights into historical crime patterns, enabling law enforcement agencies, policymakers, and communities to make informed decisions and allocate resources effectively.

This project leverages machine learning and time-series forecasting techniques to analyze historical crime data, understand crime trends, and visualize the results through dynamic web interfaces. By transforming raw data into actionable insights, this solution aims to empower stakeholders with the tools needed to enhance public safety, prevent crime, and foster secure environments. The integration of advanced algorithms and interactive visualization bridges the gap between data and decision-making, making it a step toward more efficient and data-driven crime management.

Existing System

The existing system for crime data analysis primarily employs traditional statistical methods and manual analysis of historical crime data. Law enforcement agencies rely on static query- based reports to identify trends and allocate resources. While these approaches are functional, they have limitations in terms of accuracy, real-time insights, and scalability. The current systems often fail to provide proactive measures, relying instead on reactive responses to crime trends. Mentorship can be life-changing, and the ability to find a mentor online from the comfort of your house is pure bliss for many. Especially those that might not have a local network they can utilize or those that live in rural areas that lack specialized talent.

Proposed system

In today's data-driven world, the ability to predict and analyze crime patterns is crucial for effective law enforcement and resource management. The proposed system is a Flask-based Crime Data Analysis platform that integrates real-time data scraping with machine learning models to forecast crime trends. This system allows users to visualize crime patterns through dynamic, interactive dashboards and provides predictive insights to identify potential crime hotspots. By automating data analysis and offering an intuitive, user-friendly interface, it addresses current limitations in crime data tools, improving resource allocation and decision- making. This solution enables law enforcement agencies to act proactively, enhancing public safety and efficiency in crime management.

2-REQUIREMENT ANALYSIS

Functional Requirements

Functional requirements specify the actions or tasks a system must perform to meet user needs and achieve its objectives. These requirements outline what the system should do, including specific features, functionalities, and interactions. They describe the system's behavior, such as processing data, generating reports, executing calculations, or providing user interfaces for specific operations. Functional requirements are directly tied to the purpose of the system and ensure that it delivers the desired outcomes, aligning with user expectations and project goals.

Data Management Module

This module focuses on handling crime data efficiently and aligning with the project goals.

Real-Time Data Collection: Integrates web scraping to fetch and update crime data dynamically.

Data Preprocessing: Cleans and organizes datasets for accurate analysis.

Dataset Integration: Merges crime data from multiple sources into a unified format.

Analysis and Prediction Module

This module enables advanced data analysis and machine learning predictions.

Crime Trend Prediction: Uses machine learning models to forecast crime trends based on historical data.

Prediction Script Execution: Provides scripts (prediction.py) to process and generate predictions.

Output Visualization: Generates interactive visualizations for crime trends and predictions.

Non-Functional Requirements

Non-functional requirements define how a system operates rather than what it does. They focus on the quality attributes and performance of the system, such as usability, reliability, security, and scalability. These requirements ensure the system is intuitive to use, consistently available, and capable of handling varying workloads efficiently. Non-functional requirements also address aspects like compatibility with different platforms, ease of maintenance, and speed of execution. They are crucial for ensuring that the system not only fulfills its intended functions but does so in a manner that is effective, secure, and user-friendly.

- **Usability**: The system features an intuitive interface, allowing users to easily input crime data and understand prediction results without technical expertise.
- **Reliability**: The system consistently provides uninterrupted service, ensuring data integrity and operational availability even during high usage.
- **Scalability**: The system efficiently handles increasing user loads and larger datasets, maintaining optimal performance and functionality.
- **Security**: The system protects user data and crime datasets using secure authentication, encryption, and access controls, preventing unauthorized access or breaches.
- **Portability**: The system runs seamlessly on various platforms, including Windows, macOS, Linux, and major web browsers, without requiring significant modifications.
- **Maintainability**: The system is designed modularly, enabling easy updates, debugging, and integration of new features or datasets.

Hardware Requirements

Hardware Requirements are the most common set of requirements defined by any operating system or software application is the physical computer resources, also known as hardware.

- Processor : Intel i3
- RAM : 8GB
- SSD : 500GB

Software Requirements

The software requirements document is the specification of the system. It should include both definition and a specification of the requirements. It is a set of what the system should do rather than how it should do it. The software requirements provide a basis for creating the software requirements specification. It is useful in estimating cost, planning team activities, performing tasks and tracking the team's progress throughout the development activity.

- **Operating System** : Windows 10
- Integrated Development Environment (**IDE**): Visual Studio Code
- **Programming Language** : Python 3.6
- **Machine Learning Libraries**: NumPy, Pandas, Facebook Prophet
- **Front end Technologies**: HTML, CSS
- **Web Framework**: Flask
- **Database**: MySQL

3-DESIGN

Software Architecture

It describes the structure and behavior of technology infrastructure of an enterprise, solution or system. In other words, System architecture can be described as the flow of application which is represented below in the pictorial form. He purpose of system architecture activities is to define a comprehensive solution based on principles, concepts, and properties logically related to and consistent with each other. The solution architecture has features, properties, and characteristics which satisfy, as far as possible, the problem or opportunity expressed by a set of system requirements (traceable to mission/business and stake holders requirements).

System architecture is abstract, conceptualization-oriented, global, and focused to achieve the mission and life cycle concepts of the system. It also focuses on high-level structure in systems and system elements. It addresses the architectural principles, concepts, properties, and characteristics of the system-of-interest. It may also applied to more than one system, in some cases forming the common structure, pattern, and set of requirements for classes or families of similar or related systems.

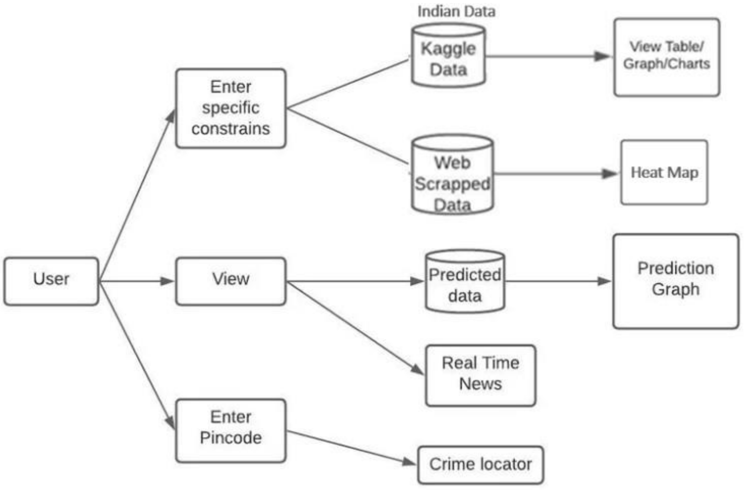


Fig. 3.1.1 Software Architecture

Technical Architecture

Technical Architecture refers to the structural process of designing and building system's architecture with focus on the users and sponsors view of the environment. Technology architecture associate's application components from application architecture with technology components representing software and hardware components. Its components are generally acquired in the market place and can be assembled and configured to constitute the enterprise's technological infrastructure. A technical architecture diagram provide a bird's eye view of the infrastructure of our project. The diagram illustrates how components in a system interact with one another in the large scale of things. Technical Architecture (TA) is a form of IT architecture that ids used to design computer system. It involves the development of a technical blueprint with regard to the arrangement, interaction, and interdependence of all elements so that system-relevant requirements are met.

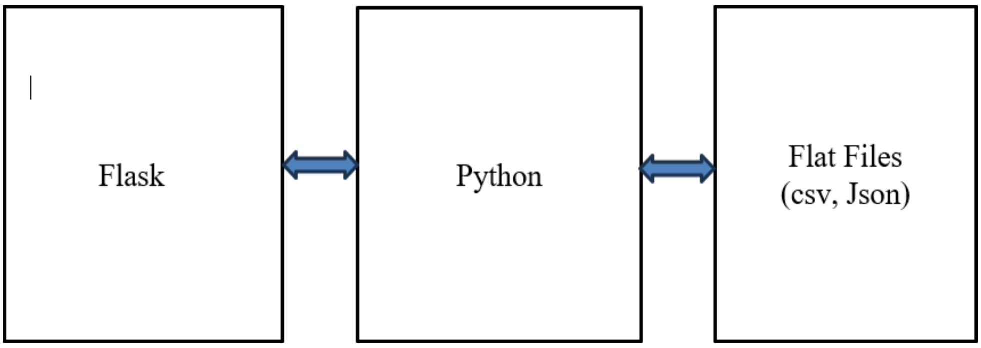


Fig. 3.1.2 Technical Architecture

4-IMPLEMENTATION

Algorithm

An algorithm is a step-by-step set of instructions or a defined procedure used to solve a specific problem or perform a task. It provides a clear and systematic way to process data, make calculations, or achieve a desired result. Algorithms are designed to be logical and efficient, ensuring they produce the correct outcome when followed correctly. They are fundamental in computer science and mathematics, forming the basis for programming, data analysis, and decision-making. A good algorithm is accurate and optimized for speed and resource usage, making it practical for solving real-world problems.

Algorithm Used: Facebook Prophet

Purpose: To analyze and predict future crime rates based on historical data.

Implementation of Algorithm

1. Data Loading

Load the historical crime data from a CSV file (actualdata2.csv) using Pandas. The dataset is expected to have two columns:

ds: Date of the observation.

y: The observed value (crime rate).

2. Data Preparation

Convert the dataset to a JSON format for record-keeping and future usage.

Ensure the dataset is formatted correctly for Prophet (date in ds and numeric target in y).

3. Model Initialization

Instantiate a Prophet object from the library. Default Settings:

Automatically detects and handles daily, weekly, or yearly seasonality. Fits a generalized additive model (GAM) for the data.

4. Model Training

Call fit() to train the model on the prepared dataset (actualdata2.csv).

The model learns the trend, seasonal components, and patterns from historical data.

Python

Python is one of the most popular programming languages, known for its versatility and ease of use. In this Crime Data Analysis project, Python is used for loading, preprocessing, and analyzing crime datasets. Libraries like Pandas and NumPy handle data manipulation, while Matplotlib visualizes trends and forecasts. Facebook Prophet is used to model crime trends and predict future occurrences based on time-series data.

The model is trained on preprocessed data and evaluated for accuracy. Predictions involve preprocessing new inputs and running them through the Prophet model. For real-time user interaction, a Flask-based web application is developed, allowing users to upload datasets and view predictions, including graphical outputs. This approach ensures robust, scalable, and user- friendly crime trend analysis. A Flask-based web application allows users to upload datasets and view predictions through interactive visualizations, making the solution user-friendly and accessible.

1. **Data Loading**

Load the historical crime data from a CSV file (actualdata2.csv) using Pandas. The dataset is expected to have two columns:

ds: Date of the observation.

y: The observed value (crime rate).

2. **Data Preparation**

   Convert the dataset to a JSON format for record-keeping and future usage.

   Ensure the dataset is formatted correctly for Prophet (date in ds and numeric target in y).

3. **Model Initialization**

   Instantiate a Prophet object from the library. Default Settings:

   Automatically detects and handles daily, weekly, or yearly seasonality. Fits a generalized additive model (GAM) for the data.

4. **Model Training**

   Call fit() to train the model on the prepared dataset (actualdata2.csv).

   The model learns the trend, seasonal components, and patterns from historical data.

5. **Future Prediction**

   Generate a future DataFrame using make_future_dataframe():

   Specifies the number of days (periods) for which predictions are required. Use the predict() function to forecast future crime rates.

6. **Result Storage**

   Save the predictions (yhat, yhat_lower, yhat_upper) and the trend to a JSON file for easy access.

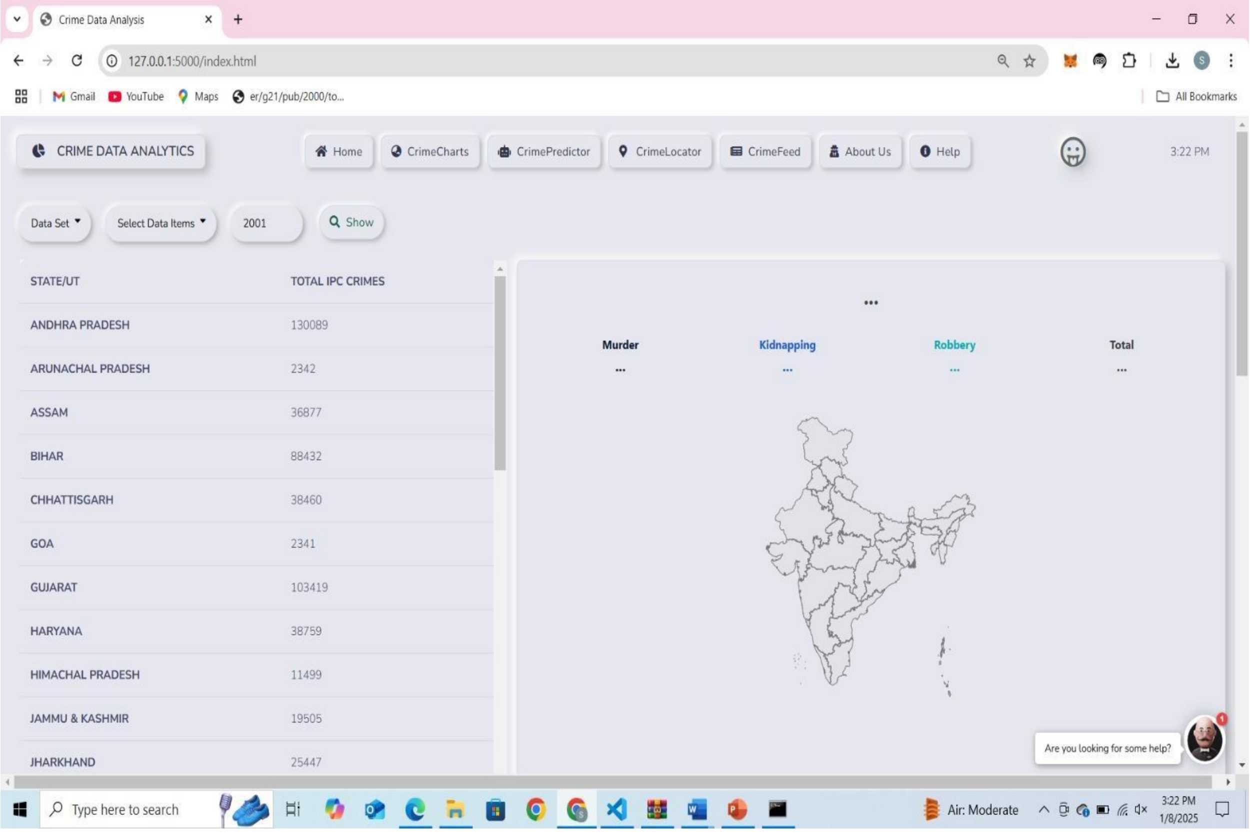
7. **Visualization**

   Use plot() to visualize the predictions, including historical data and forecasted crime rates.

Pseudo Code

Pseudocode is a detailed yet readable description of what a computer program or algorithm must do, expressed in a formally-styled natural language rather than in a programming language. It allows designers to express the design in great detail and provides programmers a detailed template for the next step of writing code in a specific programming language. Because pseudocode is detailed yet readable, it can be inspected by the team of designers and programmers as a way to ensure that actual programming is likely to match design specifications. Catching errors at the pseudocode stage is less costly than catching them later in the development process. Once the pseudocode is accepted, it is rewritten using the vocabulary and syntax of a programming language.

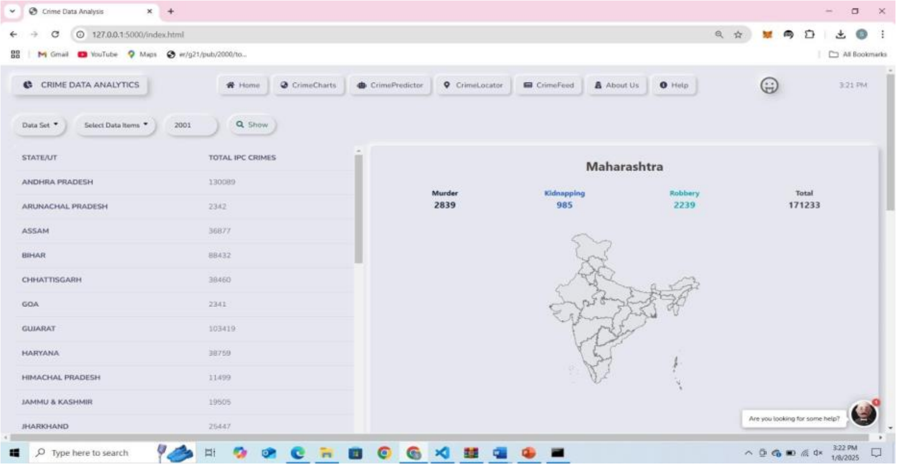
5-SCREENSHOTS



STATE/UT	TOTAL IPC CRIMES
ANDHRA PRADESH	130089
ARUNACHAL PRADESH	2342
ASSAM	36877
BIHAR	88432
CHHATTISGARH	38460
GOA	2341
GUJARAT	103419
HARYANA	38759
HIMACHAL PRADESH	11499
JAMMU & KASHMIR	19505
JHARKHAND	25447

Murder	Kidnapping	Robbery	Total
...	...	...	...

Screenshot 1 Home Page

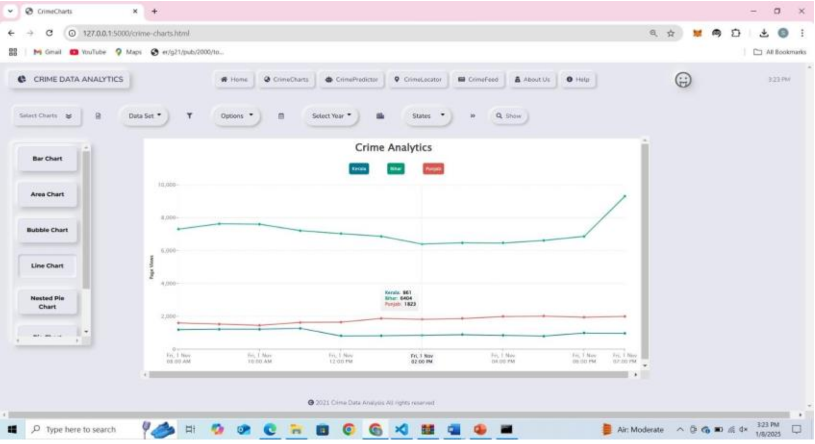


STATE/UT	TOTAL IPC CRIMES
ANDHRA PRADESH	130089
ARUNACHAL PRADESH	2342
ASSAM	36877
BIHAR	88432
CHHATTISGARH	38460
GOA	2341
GUJARAT	103419
HARYANA	38759
HIMACHAL PRADESH	11499
JAMMU & KASHMIR	19505
JHARKHAND	25447

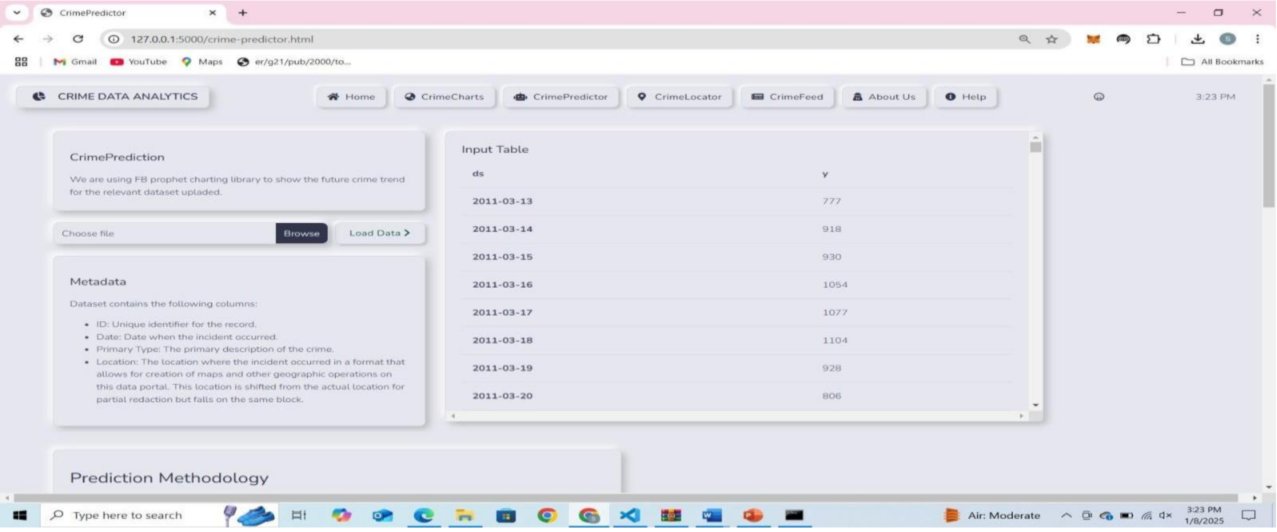
Maharashtra

Murder	Kidnapping	Robbery	Total
2839	985	2239	171233

Screenshot 2 Crime Data Analysis



Screenshot 3 Crime Charts



CrimePrediction

We are using FB prophet charting library to show the future crime trend for the relevant dataset upladed.

Metadata

Dataset contains the following columns:

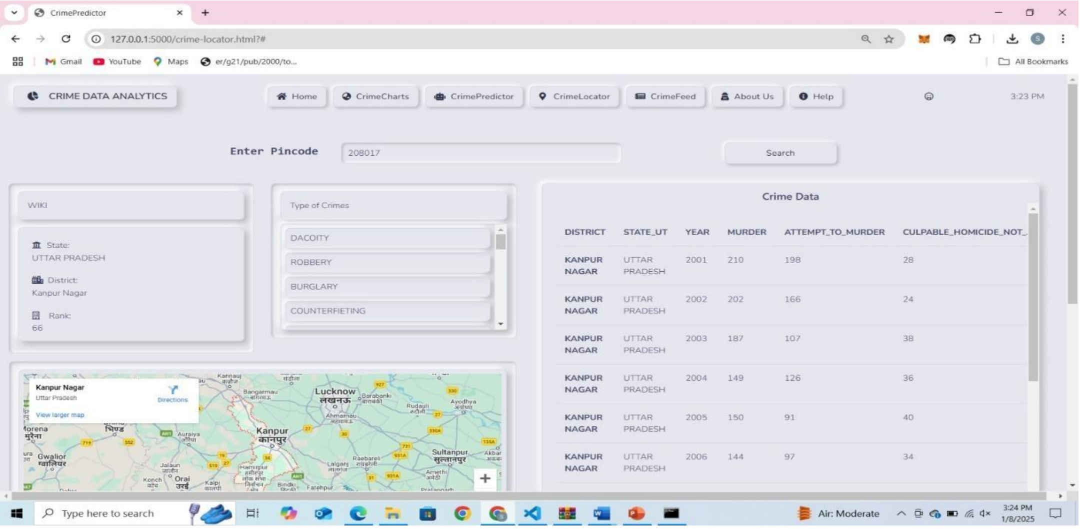
- ID: Unique identifier for the record.
- Date: Date when the incident occurred.
- Primary Type: The primary description of the crime.
- Location: The location where the incident occurred in a format that allows for creation of maps and other geographic operations on this data portal. This location is shifted from the actual location for partial redaction but falls on the same block.

Input Table

ds	y
2011-03-13	777
2011-03-14	918
2011-03-15	930
2011-03-16	1054
2011-03-17	1077
2011-03-18	1104
2011-03-19	928
2011-03-20	806

Prediction Methodology

Screenshot 4 Crime Predictor



Enter Pincode 208017

State: UTTAR PRADESH

District: Kanpur Nagar

Rank: 66

Crime Data

DISTRICT	STATE_UT	YEAR	MURDER	ATTEMPT_TO_MURDER	CULPABLE_HOMICIDE_NOT_
KANPUR NAGAR	UTTAR PRADESH	2001	210	198	28
KANPUR NAGAR	UTTAR PRADESH	2002	202	166	24
KANPUR NAGAR	UTTAR PRADESH	2003	187	107	38
KANPUR NAGAR	UTTAR PRADESH	2004	149	126	36
KANPUR NAGAR	UTTAR PRADESH	2005	150	91	40
KANPUR NAGAR	UTTAR PRADESH	2006	144	97	34

Screenshot 5 Crime Locator

6-CONCLUSION

In conclusion, the increasing crime rates in India, caused by factors like poverty and corruption, require effective solutions to address the issue. Using data analysis and advanced tools can help officials understand and manage crime trends better. By identifying high-risk areas through visual tools and actionable insights, law enforcement can take the necessary steps to improve safety and allocate resources more effectively. By adopting these technologies, stakeholders can take proactive measures to enhance safety and improve resource allocation for law enforcement efforts.

REFERENCES

[1]. J. Wang, J. Hu, S. Shen, J. Zhuang and S. Ni, "Crime risk analysis through big data algorithm with urban metrics", Phys. A Stat. Mech. Appl., vol. 545, May 2020.

[2]. N. A. S. Zaidi, A. Mustapha, S. A. Mostafa and M. N. Razali, "A classification approach for crime prediction".

[3]. F. Yi, Z. Yu, F. Zhuang, X. Zhang and H. Xiong, "An integrated model for crime prediction using temporal and spatial factors".

[4]. U. Thongsatapornwatana, "A survey of data mining techniques for analyzing crime patterns", Proc. 2nd Asian Conf. Defence Technol. (ACDT*)*, pp. 123-128, Jan. 2016.

[5]. H. Berestycki and J.-P. Nadal, "Self-organised critical hot spots of criminal activity", Eur.

J. Appl. Math., vol. 21, no. 4, pp. 371-399, Oct. 2010.

[6]. Haining, "Mapping and analysing crime data: Lessons from research and practice", Int. J. Geogr. Inf. Sci., vol. 16, no. 5, pp. 203-507, 2002.

[7]. A. Joshi, A. S. Sabitha and T. Choudhury, "Crime analysis using K-means clustering",2017. [8]. X. Zhao and J. Tang, "Exploring transfer learning for crime prediction", Nov. 2017.