THE FUTURE OF CRIME PREVENTION: POLICE CASE ANALYSIS USING MACHINE LEARNING.

(Criminal Case Analysis: Analysing and grouping commonalities among criminal cases and predicting the future crimes in terms of the nature of the crime.)

Project Id:

2023-224

Final Report Draft

Krishanthini.M- IT19980928

BSc (Hons) in Information Technology Specializing in Software Engineering

Department of Computer Science & Software Engineering

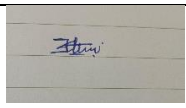
Sri Lankan Institute of Information Technology

Sri Lanka

DECLARATION

I declare that this is my own work, and this Thesis does not incorporate without acknowledgement any material previously submitted for a degree or diploma in any other University or institute of higher learning and to the best of my knowledge and belief it does not contain any material previously published or written by another person except where the acknowledgement is made in the text.

Also, I hereby grant to Sri Lanka Institute of Information Technology, the nonexclusive right to reproduce and distribute my Thesis, in whole or in part in print, electronic or other medium. I retain the right to use this content in whole or part in future works (such as articles or books).

Name	Student ID	Signature
Krishanthini.M	IT19980928	

The above candidate has carried out this research thesis for the Degree of Bachelor of Science (honours) Information Technology (Specializing in Software Engineering) under my supervision.

Signature of the supervisor

Ms. Hansika Mahaadikara

Date

Signature of the co supervisor

Ms. Sanjeevi Chandrasiri

ABSTRACT

The Analysing and grouping commonalities among criminal cases and predicting the trends in future crimes component marks a groundbreaking shift in crime analysis by embracing cutting-edge machine learning techniques, specifically the Decision Tree algorithms, to deliver precise and forward-thinking insights into criminal activities. Its core functionality revolves around the prediction of various critical crime attributes, encompassing the identification of prevalent crime types, the demographics most vulnerable to these crimes, the likely age groups of victims, the vehicles commonly involved, and the objects often targeted. With the use of historical crime data, the system equips law enforcement agencies with actionable intelligence, enabling them to proactively combat crime and allocate resources more effectively. Furthermore, it introduces the capacity to forecast crime rates over the next half-decade, a vital tool for long-term planning and mobility in the face of evolving criminal trends. The system doesn't just focus on specific crime categories and geographic regions; rather, it lays the groundwork for future expansion, which may involve a more extensive array of crime types and a broader dataset. The ongoing evolution of this component may also explore advanced machine learning methodologies, further refining predictive accuracy and analytical capabilities. In essence, this component represents a monumental stride toward data-driven crime analysis, enhancing the overall efficiency and effectiveness of law enforcement endeavours and setting the stage for a more secure and informed society.

Key Words: Crime pattern prediction, Machine Learning, Decision Tree Algorithm

ACKNOWLEDGEMENT

I am grateful to our project supervisor Ms. Hansika Mahaadikara and Co-Supervisor Ms.Sanjeevi Chandrasiri for their kind guidance, inspiration, motivation, and constructive suggestion that was helpful for me in my research project idea and preparation of this proposal. I also wish to express my gratitude to the project coordinator Mr. Jayantha Amararachchi for this support and opportunity. Also, I'd want to express my gratitude to all the educators, students, and parents who have been in touch with me for their insightful remarks and suggestions.

Table of Content

LIST OF FIGURES

Figure Page

LIST OF TABLES

Table **Page**

LIST OF ABBREVIATIONS

Abbreviations	**Description**
ML	Machine Learning
CSV	Comma-Separated Values.
OS	Operating System
NUMPY	Numerical Python
I/O	Input/Output
API	Application Programming Interface

INTRODUCTION

1.1 Background Study and Literature Review

1.1.1 Background Study

Crime is a pervasive issue that poses significant challenges to society, public safety, and law enforcement agencies all around the world. Understanding crime patterns, identifying trends, and predicting future criminal activities are essential aspects of effective crime prevention and law enforcement. In recent years, the integration of machine learning techniques into crime analysis has shown promise in providing valuable insights for crime reduction efforts. This background study explores the importance of crime analysis, the role of machine learning, and the specific relevance of Decision Tree models in addressing this critical issue. The adoption of machine learning techniques in crime analysis has unfold new avenues of exploration. Machine learning algorithms possess the capacity to process vast volumes of crime data, uncover intricate patterns, and generate precise forecasts, thereby reshaping the landscape of crime analysis. Within the spectrum of machine learning, Decision Tree models have emerged as particularly promising tools. They offer interpretability, allowing law enforcement professionals and policymakers to comprehend the factors influencing criminal activities intuitively. Furthermore, Decision Trees rank the importance of different features, such as time, location, and demographics, facilitating informed resource allocation decisions. Their versatility in handling diverse crime-related variables, coupled with efficiency in analysing large datasets, makes them an ideal choice for the task of predicting crime patterns. In essence, crime analysis, with its goal of reducing crime rates, enhancing public safety, and fostering community trust, has found a powerful combination in machine learning, particularly in the form of Decision Tree models. This research endeavours to control the potential of Decision Tree models to predict crime patterns, forecast future crime rates, and contribute significantly to the effectiveness of crime analysis. Ultimately, these efforts are directed toward supporting law enforcement agencies and policymakers in their mission to combat crime effectively and ensure the well-being of communities.

1.1.2 Literature Review

This section delves into prior research related to the topic of using machine learning algorithms for crime analysis, a subject that has gained increasing prominence in recent years. Numerous studies have shown that the utilization of diverse algorithms can greatly improve the effectiveness and precision of crime detection, prevention, and investigation, especially concerning the characteristics of the criminal incidents.

One noteworthy study in this domain was conducted by Neil Shah, Nandish Bhagat, and Manan Shah [2], which explores crime forecasting through a machine learning and computer vision approach. Their work underscores the significance of leveraging data such as facial recognition, number plate recognition, augmented and mixed realities, location tracking, and object identification in forecasting and averting criminal activities. Their proposal involves using the motive behind a crime as a determinant for classifying the nature of the crime, encompassing comprehensive crime aspects to facilitate efficient analysis.

Another notable contribution comes from Steven Walczak [3], who discusses the application of neural network models to predict specific crime types based on temporal and spatial data. Walczak's findings reveal that neural network models can predict the type of crime with an accuracy rate of 16.4% across 27 distinct crime types or 27.1% when similar crimes are grouped into seven crime categories. This research underscores the effectiveness of neural network models in aiding law enforcement decision-making for crime prevention.

Karabo Jenga, Cagatay Catal, and Gorkem Kar [4], in their research on machine learning in crime prediction, emphasize the potential of data mining and machine learning in tasks related to crime detection and prevention. They highlight the versatility of machine learning and data mining in the context of crime prediction.

Furthermore, Suhong Kim, Param Joshi, Paraminder Singh Kalsi, and Pooya Taheri [5] have contributed to the field by emphasizing the utility of decision trees in crime analysis. Their study involved the analysis of over 560,000 records to predict crimes in Vancouver, achieving a prediction accuracy ranging from 39% to 44% based on crime categories and time. This research underscores the significance of employing decision trees for the analysis of crime data to enhance crime prediction accuracy.

In summary, these research endeavours collectively underscore the importance of analysing criminal cases with respect to the nature of the crime within society. Furthermore, they emphasize the pivotal role of machine learning algorithms in improving the efficiency and accuracy of crime prevention and investigation. The present research seeks to build upon these contributions by leveraging machine learning algorithms to analyse historical crime data, identify recurring patterns, and discern commonalities in the pattern of criminal incidents, ultimately aiming to enhance the operational efficiency of law enforcement agencies.

1.2 Research Gap

The current landscape of crime case analysis research reveals several noticeable gaps that are at the forefront of this study's objectives. One of the most prominent gaps is the absence of a unified and user-friendly system that fully control the potential of machine learning, specifically the Decision Tree model, to predict future crime patterns. While numerous studies have explored the application of machine learning in crime prediction, a comprehensive and accessible system that seamlessly integrates advanced predictive modelling into crime analysis remains conspicuously absent. Furthermore, the existing body of research often adopts a fragmented approach, with a predominant focus on isolated aspects of crime analysis, such as location-based predictions or the forecasting of specific crime types. This narrow scope has created a significant research gap in the development of an all-encompassing crime analysis system. A factor notably lacking is, a versatile platform that not only identifies the highest occurring crimes but also elucidates the demographics most affected by them, unveils patterns involving vehicles and stolen objects, and offers reliable long-term crime rate predictions. This comprehensive view of crime patterns is fundamental for a comprehensive understanding of criminal activities and the formulation of effective crime prevention and law enforcement strategies. In essence, this research gap underscores the pressing need for a more inclusive, user-friendly, and predictive crime analysis system that combine advanced machine learning techniques and provides a multifaceted insight into the ever-evolving landscape of criminal activities.

Table 2.1.1: Comparison between existing research and the proposed component

RESEARCH	FUTURE PREDICTION	FINDING THE SIMILARITIES	GROUPING THE CRIMINAL ACTIVITIES	LOCATION DETECTION	CRIME PREVENTION
A MACHINE LEARNING AND COMPUTER VISIONAPPROACH TO CRIME PREDICTION AND PREVENTION [1]	✓	✗	✗	✗	✓
PREDICTING THE TYPES OF CRIMES COMMITTED IN THE CITY OF CHICAGO USING NEURAL NETWORKS. NEURAL COMPUTING AND APPLICATIONS [3]	✓	✗	✗	✗	✗
CRIME PREDICTION USING DECISION TREES. [5]	✓	✓	✗	✗	✗
PREDICTIVE MAPPING OF CRIME BY PROMAP: ACCURACY, UNITS OF ANALYSIS, AND THE ENVIRONMENTAL BACKCLOTH[7]	✓	✗	✗	✓	✓
CRIMINAL CASE ANALYSIS AND FUTURE PREDICTION	✓	✓	✓	✓	✓

1.3 Research Problem

The central research problem addressed in this study revolves around the critical need for a comprehensive and user-friendly crime analysis system that harnesses the potential of machine learning, particularly the Decision Tree model, to predict future crime patterns. Existing methods for crime analysis primarily rely on historical data and statistical methodologies, often falling short when it comes to real-time and proactive crime prediction. This research endeavours to bridge this evident gap by pioneering the development of a system that empowers users to input specific parameters, including the year, month, and location, and subsequently gain valuable insights into a spectrum of crime-related aspects. The core objectives encompass identifying the highest occurring crimes in a specified area, understanding the demographics most affected by these crimes, selective patterns related to vehicles involved and objects stolen, and crucially, predicting crime rates for the next five years. The paramount challenge at hand is to establish a strong and highly accurate predictive model that significantly enhances the field of crime analysis. Such an advanced system not only aids in understanding crime trends but also holds the potential to revolutionize crime prevention and law enforcement strategies. Ultimately, the research contributes to the overarching goal of bolstering public safety by providing actionable insights and foresight into criminal activities.

1.4 Research Objectives

1.4.1 Main Objective

The primary objective of this component is to create a cutting-edge crime analysis system that fills a crucial gap in the current landscape of crime analysis research. This system aims to control the full potential of machine learning, with a specific focus on the Decision Tree model, to provide accurate predictions of future crime patterns. To elaborate further, the main goal is to design a comprehensive and user-friendly platform that empowers users, particularly law enforcement agencies and analysts, to make informed decisions and take proactive measures in addressing criminal activities. This involves developing predictive models that can identify the highest occurring crimes, reveal the demographics most affected, unveil patterns related to vehicles involved and objects stolen, and provide reliable long-term crime rate predictions for the next five years. By achieving this objective, the component aims to enhance the effectiveness of crime analysis, prevention, and law enforcement strategies. It aspires to contribute significantly to improving public safety by offering actionable insights and foresight into criminal activities, thereby fostering more efficient allocation of resources and more targeted crime prevention efforts.

1.4.2 Specific Objectives

- Implement a Decision Tree machine learning model to analyze historical crime data and identify patterns.
- Combine the patterns in crime, including the highest occurring crimes, affected gender demographics, vulnerable age groups, vehicles involved, and objects stolen.
- Visualize the analyzed data in graphical forms, including charts and graphs, to provide users with intuitive and informative representations of crime trends and patterns.
- Design a predictive model to estimate crime rates for the next five years based on user-provided data.
- Develop visualization tools to generate graphs depicting crime trends for specific locations and years.
- Evaluate the accuracy and effectiveness of the Decision Tree machine learning model in crime prediction through rigorous testing and validation procedures.

2 METHODOLOGIES

2.1 Introduction

This section delineates a structured and systematic approach for effectively implementing the functions of the proposed system. It adopts a well-defined research methodology to ensure that every aspect of the system's development is executed with precision and purpose. The chosen methodology adheres to a recognized software lifecycle model, which serves as a guiding framework for the entire development process. This ensures that each phase, from initial concept to final deployment, is precisely planned and executed. Furthermore, this approach is underpinned by a substantial body of research within the field of study. Prior investigations and studies have yielded a wealth of knowledge and insights that are invaluable resources for the current research endeavour. This existing body of work not only informs the direction of the project but also provides a foundation of knowledge that will be leveraged to achieve both the primary objectives and the sub-objectives set forth in this research. In summary, the systematic approach outlined in this section underscores the rigor and thoroughness with which the proposed system will be developed. It aligns with established software development practices and leverages the collective wisdom of previous research efforts to ensure the successful attainment of the project's objectives.

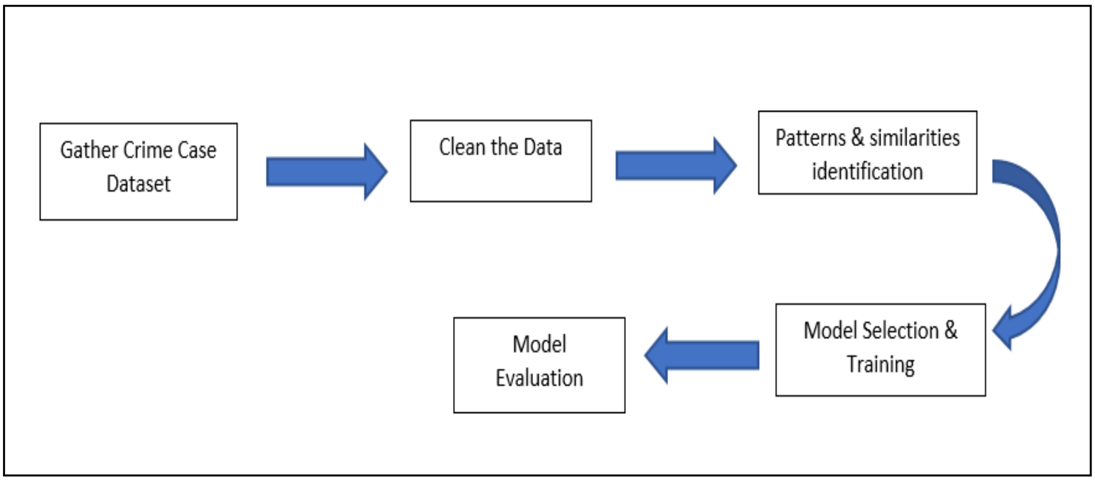


***Figure 2.1.1**:* Methodology

2.2 System overview

Analysing and grouping commonalities among criminal cases and predicting the future crimes in terms of the pattern of the crime. Considering the outcome from the literature review, most important was to decide the technologies, software solution and most appropriate tools for implementation phase. The proposed system can be divided as following components:

- Predicting the patterns and trends in the crime
- Predicting the crime rates for the next five years.
- Analyzing the crime rates for all crimes according to the Location and year.

The cornerstone of the crime analysis system lies in the gathering and preprocessing of data sourced from various law enforcement agencies. This critical phase ensures that the data is transformed into a format conducive to comprehensive analysis. During preprocessing, a significant portion of the data is encoded, optimizing its suitability for model training, thereby enhancing efficiency in subsequent phases. Decision Tree algorithm strategically selected for its distinctive attributes and applicability in predicting crime patterns. Decision Trees are renowned for their exceptional interpretability, offering a clear and intelligible decision-making path. This feature is instrumental in providing transparency and aiding in understanding why a specific prediction was made, which is crucial for both law enforcement and stakeholders. Furthermore, Decision Trees demonstrate versatility in handling a combination of categorical and numerical features, a common scenario in real-world datasets like crime data. This flexibility is paramount as it allows the model to effectively learn from a diverse range of input variables and find out the patterns and connections in between the crimes. Following the analysis, the outputs are not only insightful but also presented in a visually intuitive format. The crime analysis system employs visualization tools to convert the findings into charts, graphs, and other visual representations. This approach ensures that the results are not only accurate but also accessible, enabling law enforcement and other users to quickly grasp and act upon the insights gleaned from the data. In essence, our component harmoniously integrates data collection, preprocessing, Decision Tree modelling, and visual representation of the results. The transparency and adaptability of Decision Trees, combined with intuitive visualization, offer a holistic system for predicting and comprehending criminal activities. This integrated approach is poised to revolutionize crime analysis, providing actionable insights in an easily digestible format.

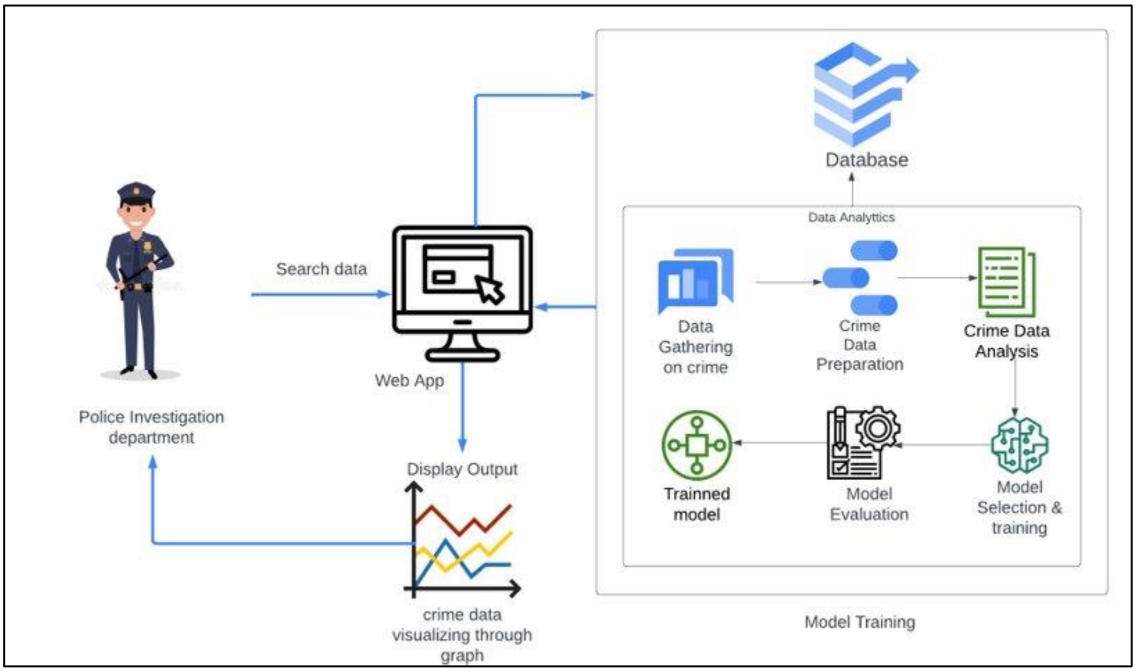


***Figure 2.2.1**:* System high-level architecture diagram

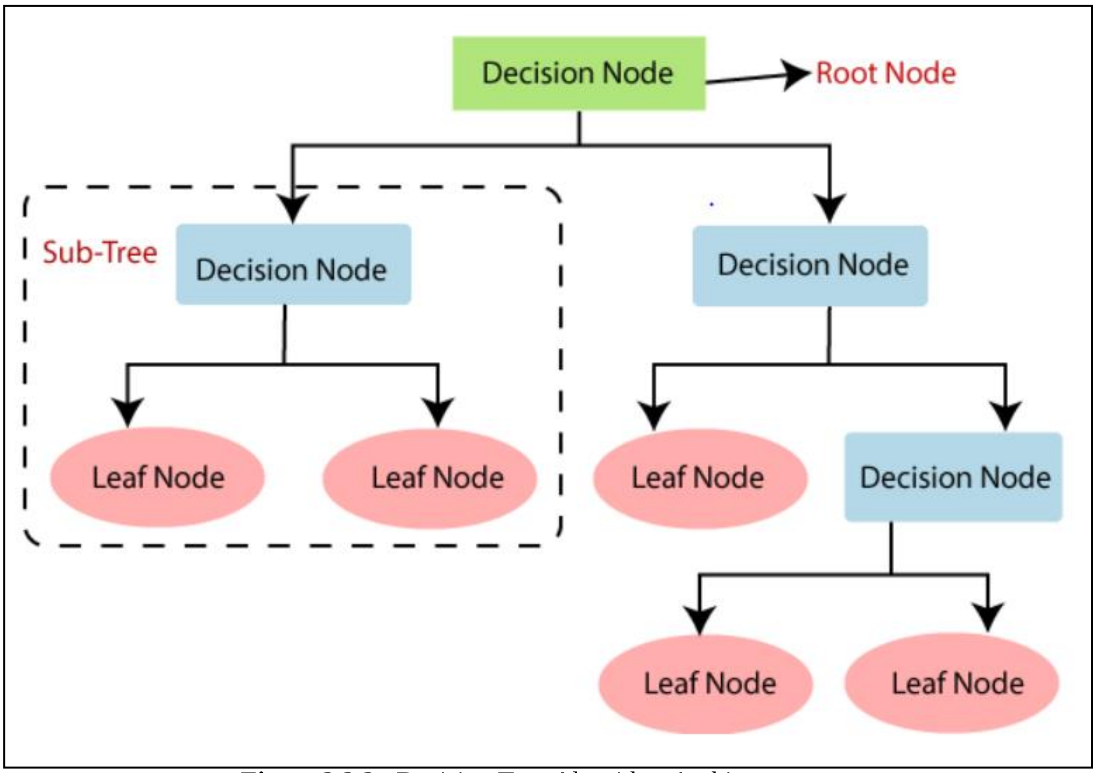


Figure 2.2.2: Decision Tree Algorithm Architecture

Data Processing for the model training.

Preprocessing plays a pivotal role in the successful analysis and prediction of crime patterns within our component. As the foundation upon which our machine learning models are built, preprocessing encompasses a series of essential data preparation steps that ensure the accuracy and reliability of our predictive models. From handling missing data and transforming categorical variables to extracting meaningful features, this preprocessing stage serves as the crucial gateway between raw crime data and actionable insights. In the following sections, we will delve into the intricacies of each preprocessing step, shedding light on how these data transformations enhance our system's capabilities to discern crime patterns, demographics, and predictive trends. Through meticulous data preprocessing, we pave the way for more accurate and effective crime analysis, ultimately contributing to improved public safety and law enforcement strategies.

Simple Imputer with 'Mean' Strategy: The first step involves handling missing data in the 'victim_age' column. By using a SimpleImputer from the scikit-learn library, specifying the 'mean' strategy. This means that missing values in the 'victim_age' column are replaced with the mean (average) age of all available data points. This is a common technique to address missing numerical data while maintaining the dataset's statistical properties.

Filling Missing Values in 'crime_type' and 'area_code': Next, addressing missing values in the 'crime_type' and 'area_code' columns. For 'crime_type', filling missing values with -1, providing a placeholder value to indicate missing data. Similarly, for 'area_code', missing values are also filled with -1. These values can later be handled appropriately during analysis.

Mapping 'victim_sex': Mapping the 'victim_sex' column, which originally contains 'M' for male and 'F' for female, into numerical values. 'M' is mapped to 0, and 'F' is mapped to 1. This transformation converts categorical data into a format that machine learning models can work with.

Extracting Date Components: Extracting additional features related to the date of the crime. Specifically, you're creating three new columns: 'day', 'month', and 'year' by splitting the 'date_occurred' column into its constituent parts. This allows the model to consider the day, month, and year as separate features, which can be relevant for understanding patterns in crime data over time.

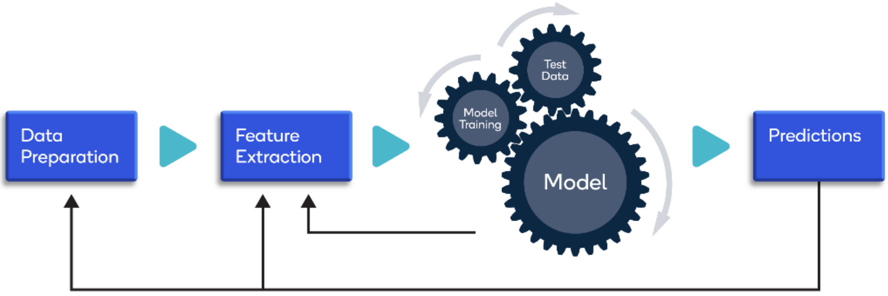


Figure 2.2.3: Feature Extraction used for date.

Mapping Text to Numerical codes to predict the pattern in the Future Crimes.

Mapping text to numerical codes is a fundamental preprocessing technique employed within this component to predict future crime patterns using machine learning models. In the context of this specific project, which focuses on crime case analysis, this technique plays a pivotal role in converting various categorical attributes into numeric representations. These attributes encompass not only crime types but also area codes, victim characteristics (such as age and gender), and other relevant factors. The process commences by creating a mapping table that establishes a clear association between each unique textual category and a corresponding numeric code. For example, crime types like "burglary," "robbery," and "assault" are mapped to numeric codes for subsequent analysis. Furthermore, this approach extends to area codes, victim sexes (coded as 'M' for male and 'F' for female), and more.

Once the mapping is complete, the original textual data within the dataset is transformed into these numeric representations. This transformation is pivotal for the effective utilization of machine learning algorithms, particularly the Decision Tree model, which requires numeric input. By applying these numerical codes, the component can construct predictive models that enhance crime analysis, offering valuable insights into future crime patterns.

However, the significance of this mapping process doesn't end there. After the predictive models generate outcomes, a reverse mapping process is initiated. This process translates the numeric predictions back into their original categorical meanings, ensuring that the results are interpretable and actionable. These outcomes include identifying the highest occurring crimes, affected demographics, vehicles involved, objects stolen, and predictions for crime rates in the coming years. In essence, this component exemplifies how the systematic transformation of textual data into numerical codes not only facilitates efficient machine learning but also enhances the efficiency and accuracy of crime analysis. It represents a proactive approach to crime prevention and law enforcement, underscoring the practical utility of machine learning in the realm of crime case analysis**.**

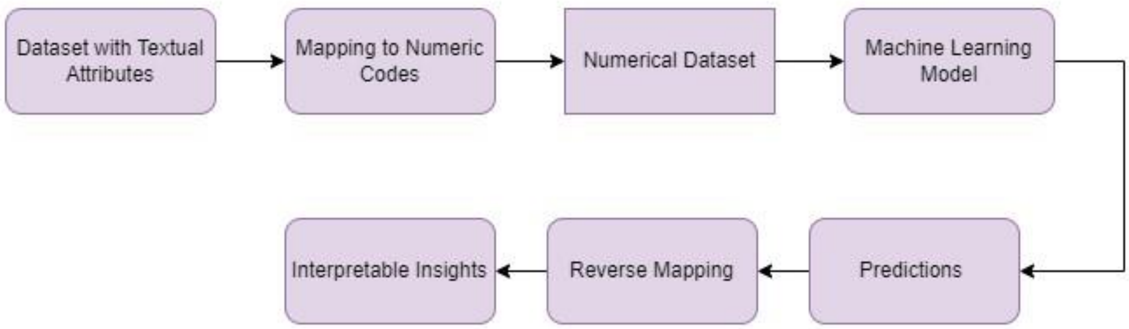


Figure 2.2.4: Mapping and Prediction process of Crime Pattern and Trends

Decision Tree Algorithm Model

In the realm of crime prediction and analysis, decision trees stand out as invaluable tools. These machine learning models, including Decision Tree Classifier and Decision Tree Regressor, enable law enforcement agencies and analysts to make data-driven predictions and gain deeper insights into criminal activities. The Decision Tree Classifier is tailored for categorical classification tasks. It operates by recursively dividing the dataset into subsets based on categorical attributes, effectively classifying data into different categories. For instance, it can predict the type of crime (e.g., burglary, robbery) based on various attributes such as location, date, and victim characteristics.

On the other hand, the Decision Tree Regressor specializes in predicting continuous numerical values. It differs from the classifier by making numeric predictions at each node in the tree. For crime analysis, this regressor can forecast continuous variables like victim age, crime rates, or the number of incidents. For instance, it can predict the average age of victims in a particular area or forecast future crime rates based on historical data. Decision trees are favoured in crime analysis due to their interpretability and versatility in handling mixed types of data features, often encountered in real-world crime datasets. They provide a transparent decision-making process, aiding in understanding the rationale behind predictions. Nevertheless, it's crucial to guard against overfitting by applying techniques such as pruning and optimizing hyperparameters to ensure the accuracy and reliability of the predictive models. In sum, decision trees play a vital role in enhancing crime prediction, investigation, and prevention strategies by offering actionable insights derived from data-driven analysis.

Finding the Crime rate for next Five years.

The model training logic within this Django view function plays a pivotal role in predicting future crime rates based on historical data. It starts by meticulously preparing the dataset, handling missing values, and structuring the information. After filtering the data for the selected crime type, it aggregates and calculates total crime counts for different time periods, thereby creating a comprehensive historical context. The feature variables, representing years and months, are isolated from the target variable, which holds historical crime counts. A Decision Tree Regressor model is thoughtfully selected for training, as it's well-suited for predicting continuous numeric values like crime counts. The heart of the logic lies in training this model, allowing it to discern patterns and relationships within the historical data. Subsequently, the model is employed to forecast crime counts for the next five years, iteratively constructing feature vectors and retrieving predictions. Ultimately, the results are presented through an intuitive webpage, facilitating user understanding of projected crime trends for their chosen crime category. This model training logic is the linchpin of the entire predictive process, enabling informed decision-making and enhanced crime analysis capabilities.

Predict Crime rates of all the crimes in in a year at a specific Location.

The model training process illustrated in this Django view function is crucial for predicting future crime rates within a specific area. When a user submits a request with the desired future year and area code, the function first loads the dataset containing historical crime data. It then proceeds with data preprocessing, including handling missing values, encoding categorical variables (like victim sex), and extracting temporal information from the date of occurrence. Next, the logic iterates through different crime types present in the dataset, focusing on the selected area code. For each crime type, it filters the data, groups it by year and month, and calculates total crime counts. This forms the historical context for that specific crime category within the chosen area. The model, a Decision Tree Regressor, is trained using this historical data, allowing it to recognize patterns and relationships. Subsequently, the trained model is utilized to predict future crime rates for the given area and year. These predictions are collected for all unique crime types within the chosen area. Finally, the results, including the crime type and its corresponding predicted rate, are presented through a user-friendly webpage. This model training and prediction process empowers users to gain insights into expected crime rates, enhancing their understanding of potential trends within their selected area.

2.3 Diagrams

2.3.1 Work Breakdown Structure

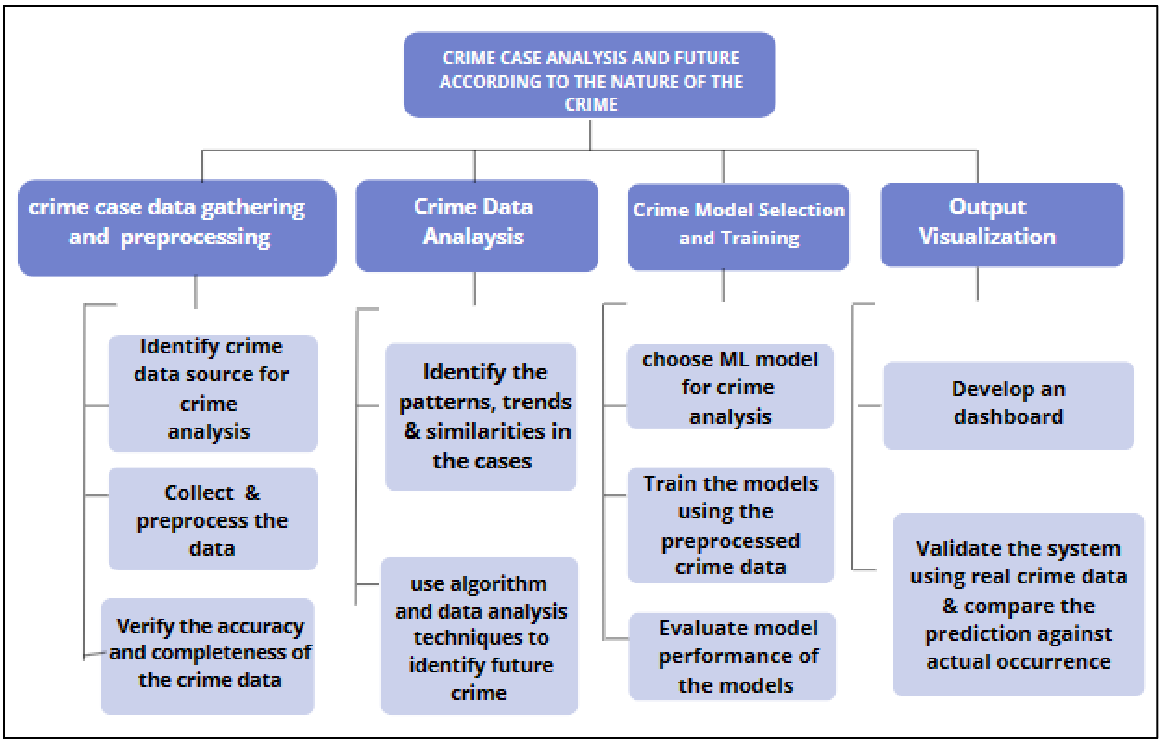


Figure 2.3.1: Work Breakdown Structure of the system

2.3.2 Flowchart

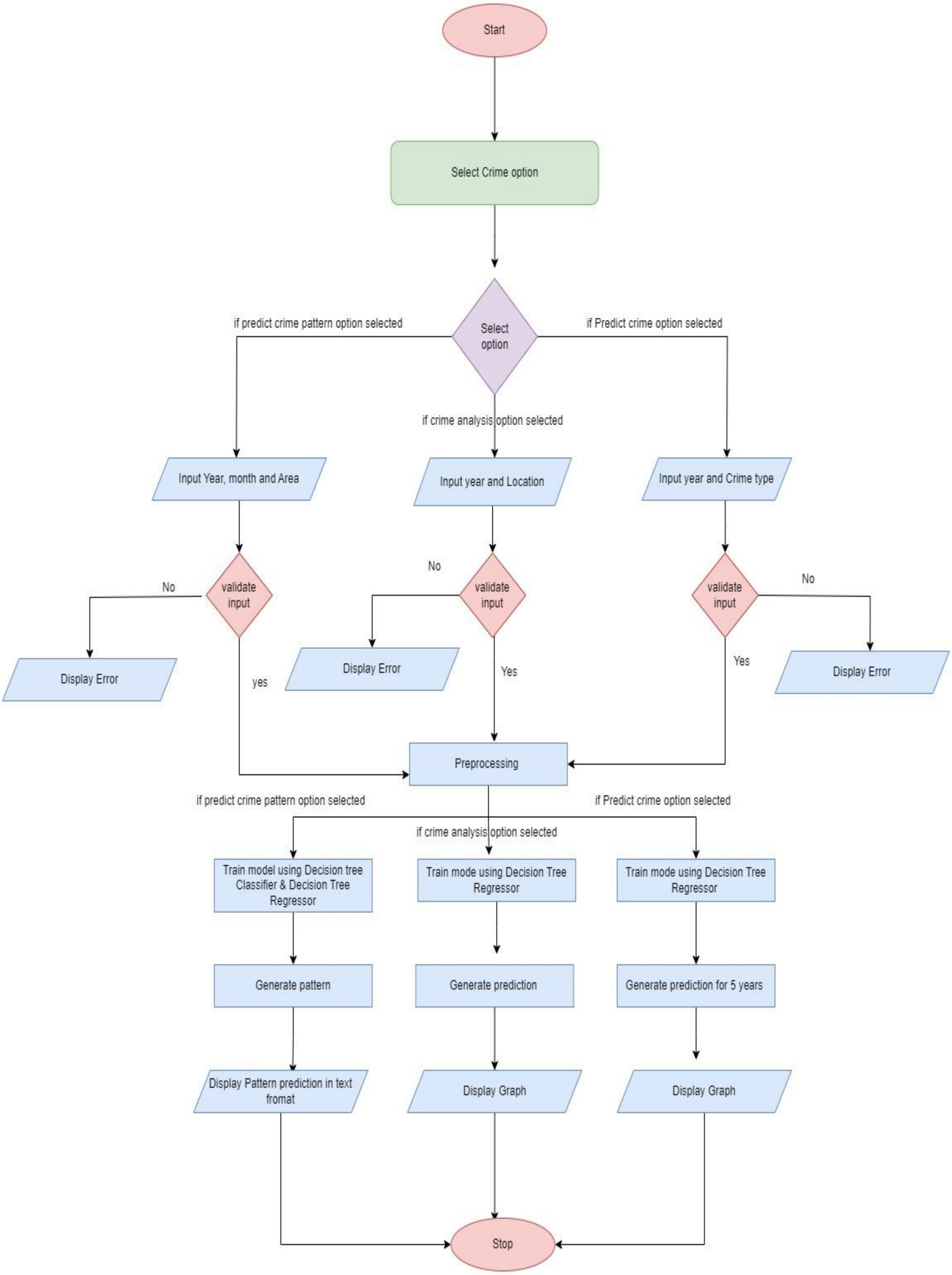


Figure 2.3.2: Flow Chart of the system

2.4 Development Process

The adoption of the waterfall model for the system development approach is a deliberate choice driven by several key considerations. Firstly, the linear nature of the waterfall model aligns seamlessly with the well-defined needs and specifications of the project. It offers a structured and systematic approach that progresses in a straightforward manner, making it particularly suitable when project requirements remain stable over time. The selection of this methodology followed a comprehensive evaluation of the system's requirements, which were clearly articulated and documented at the outset. This clear and uncomplicated approach allowed for the organization of tasks into schedulable phases, each with specific objectives that could be accomplished within predefined timeframes. Importantly, the development phases in the waterfall model do not overlap, promoting clarity and focus on each stage of the project. In accordance with the guidelines established in this section, a systematic investigation of the identified problem area was conducted, methodically characterizing the functionalities that would focus on the initial problem statement. Expectations for the crime analysis solution were well-defined, with the anticipation of achieving a robust and effective system that addresses the identified problem comprehensively. Crucially, the waterfall model facilitated effective time management throughout the one-year research period. The clear delineation of project phases and tasks allowed for the efficient allocation of resources and efforts, ensuring that the research progressed in a structured and meaningful manner. As illustrated in Figure 2.4.1, the requirements were methodically divided into functional premises, providing a comprehensive and organized framework for the development and research process.

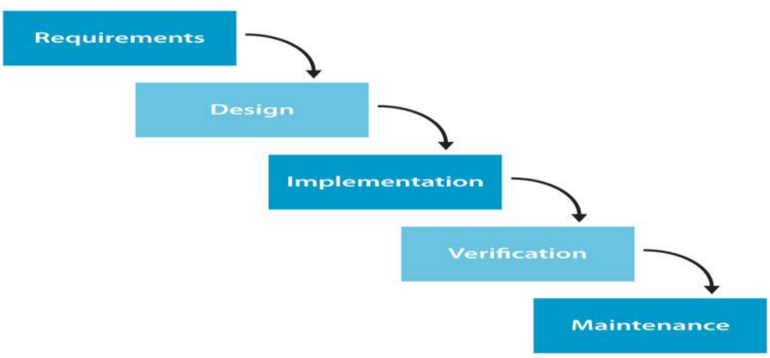


Figure 2.4.1: Development process of the system

Project Management

The core aim of software project management is to provide a team of developers with the necessary resources and methods to work efficiently, ultimately achieving the project's goals within the designated timeframe.

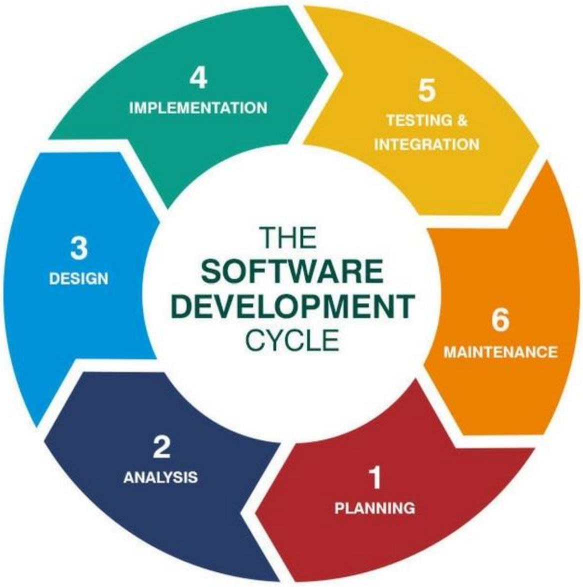


Figure 2.4.2: Software Project Management Process

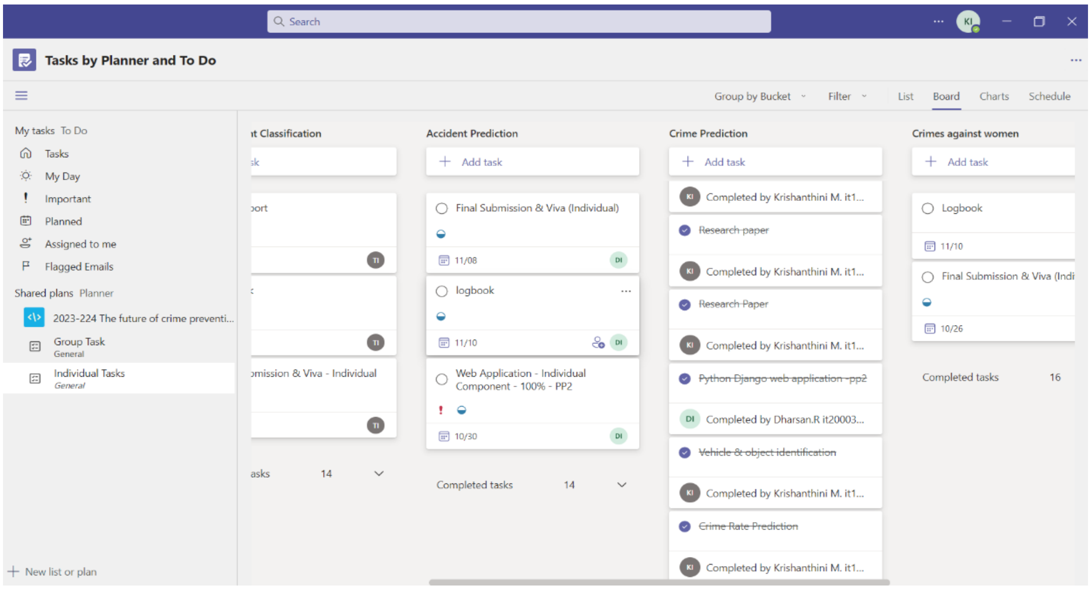


Figure 2.4.3: Software Project Management for the system

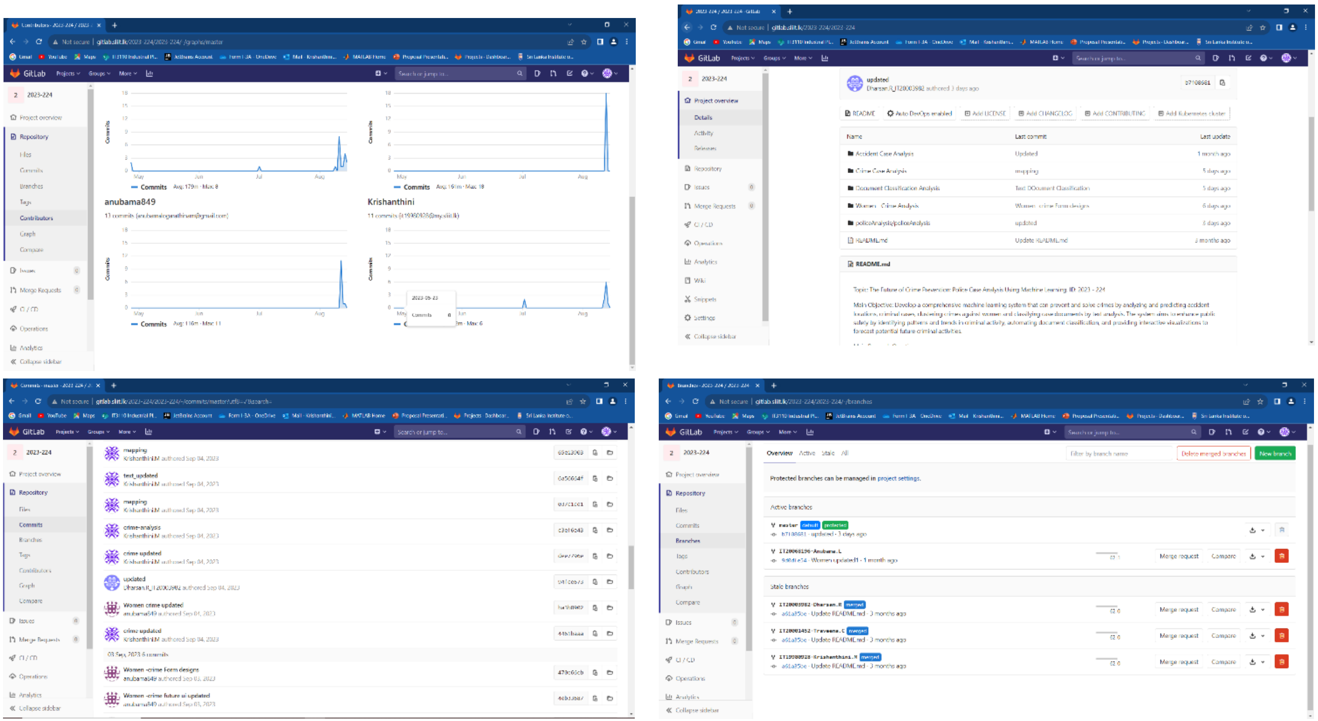


***Figure 2.4.4**: Project* Code Management for the system

2.5 Requirements Gathering

The process of gathering requirements for the crime analysis component is fundamentally centred on the collection of data from law enforcement agencies, including police and legal authorities. This critical phase relies heavily on precise documentation, which serves as the cornerstone of the project. By meticulously documenting requirements, we establish a clear and robust foundation for the entire endeavour. A key aspect of this process is the prioritization of requirements. This step involves distinguishing between essential functionalities that are indispensable to the core objectives and desirable enhancements that, while valuable, may not be crucial for initial implementation. Prioritization enables the judicious allocation of resources and efforts, ensuring that the component's development remains focused and aligned with user expectations. In adopting this approach, we can confidently assert that the resulting component will not only meet but exceed user expectations. It will significantly enhance crime analysis capabilities, empowering law enforcement agencies with valuable insights for proactive crime prevention and investigation. Ultimately, this meticulous requirement gathering process contributes to the broader goal of enhancing public safety and security.

Functional Requirements:

- The system must provide an intuitive user interface allowing users to input parameters, including the year, month, and location, for querying crime data.
- Implement a Decision Tree machine learning model that can predict pattern with the highest occurring crimes, age and gender that will be affected, vehicle involved, and object stolen based on user-provided parameters (year, month, location).
- Create predictive models capable of estimating crime rates for the next five years, given a specific year and type of crime.
- Develop visualization tools to generate graphical representations of all crimes in a specific location for a specified year, enhancing data interpretation and user insights.

Non-functional Requirements:

Usability: The system should have an intuitive and user-friendly interface to ensure ease of use for law enforcement personnel and analysts.

Security: Ensure robust security measures to safeguard sensitive crime data, including user access controls and data encryption.

Availability: Maintain a high level of system availability, minimizing downtime to ensure continuous access to crime analysis capabilities.

Scalability: Design the system to accommodate increasing volumes of crime data and user demands without performance degradation.

Accuracy- The Decision Tree model deliver a high degree of accuracy in predicting crime patterns and demographic impacts, enhancing the system's reliability and value to users.

2.5.1 Resources Used

2.5.1.1 Software Boundaries

Backend - Python Language

The primary purpose of utilizing Python and Django as the backend for our system is to leverage their robust and versatile capabilities in web development. Python, renowned for its readability and versatility, serves as the backbone for the system's logic and functionality. Its extensive libraries and frameworks facilitate rapid development and scalability.

Frontend – HTML, CSS&JAVASCRIPT

Utilizing HTML, CSS, and JavaScript for the frontend enables the creation of dynamic and interactive web interfaces. HTML structures content, CSS styles it for visual appeal, and JavaScript adds interactivity and responsiveness to enhance the user experience.

Visual Studio Code Editor

With the integration of the Microsoft Python plugin, employing Python within Visual Studio Code becomes a seamless and efficient experience. This enhancement ensures that VS Code serves as a proficient Python editor, offering consistent performance across various operating systems and Python interpreter setups.

Libraries

1. **Pandas:** Pandas is a popular Python library used for data manipulation and analysis. It provides data structures like Data Frames and Series that allow you to efficiently work with structured data. Pandas simplifies tasks such as data cleaning, transformation, aggregation, and visualization. It's widely used in data science and data analysis projects for its flexibility and ease of use.

2. **Scikit-Learn (sklearn):** Scikit-Learn is a machine learning library for Python. It offers a wide range of tools for tasks like classification, regression, clustering, dimensionality reduction, and more. For this component sklearn.trees and sklearn.impute is majorly used.

3. **SciPy:** SciPy is an open-source library built on top of NumPy, designed for scientific and technical computing. It provides a vast array of mathematical functions, optimization algorithms, signal processing tools, and statistical functions. SciPy is particularly useful for scientific research, engineering, and data analysis where complex mathematical operations are required.

4. **io (Input/Output):** The "io" module in Python is used for handling input and output operations. It includes classes and functions for reading and writing data to different types of streams, such as files, strings, or memory buffers. The "io" module is particularly useful for tasks like reading from or writing to files, sockets, or other data streams.

5. **NumPy:** NumPy, short for "Numerical Python," is a fundamental library for numerical computing in Python. It provides support for large, multi-dimensional arrays and matrices, along with a wide range of high-level mathematical functions to operate on

these arrays. NumPy is the foundation for many other libraries, including Pandas and SciPy, and is essential for tasks like scientific computing, data analysis, and machine learning.

Framework – Django

Django is a framework for building web applications in Python. It provides a structured way to create websites and web applications, making development faster and more organized. With Django, you can handle tasks like handling user authentication, managing databases, and handling web requests efficiently. It follows the "batteries-included" philosophy, which means it includes many built-in features and tools to simplify common web development tasks, saving developers time and effort. Whether you're building a simple website or a complex web application, Django helps you do it more easily and with fewer errors, making it a popular choice for web development.

Web application

A web application, often referred to as a web app, is a type of software that operates within a web browser. Instead of being installed on your device, it is accessed over the internet through a web browser interface. These applications are hosted on remote servers, not on your local computer or device. Web services are a category of web applications, and it's important to note that while many websites have web applications embedded within them, not all websites do. In essence, web applications provide functionality and services online, making them accessible from any device with internet access and a compatible web browser.

2.6 Commercialization of the Product

The Crime Analysis Component, primarily driven by its social impact potential, represents a remarkable opportunity to improve public safety and enhance community well-being. While the primary objective is not solely profit-driven, the component's positive influence on society can be harnessed to create sustainable models that support its widespread adoption.

By targeting governmental agencies, community organizations, and non-profit entities, the Crime Analysis Component can be offered through partnerships and grant-funded initiatives. These organizations often lack the resources to invest in advanced technology, making them ideal recipients of this socially oriented tool. Collaboration with philanthropic foundations, government grants, and social impact investors can provide the necessary financial backing for deployment in underserved communities.

Additionally, engaging in advocacy and awareness campaigns can foster a broader understanding of the component's potential benefits. Its ability to predict and mitigate crime can lead to safer neighbourhoods, reduced social inequalities, and stronger community cohesion. Thus, its value extends beyond monetary gains, offering a chance to create meaningful social change. By taking this approach, we aim to harness the component's potential to serve society's greater good while ensuring its sustainable development and adoption.

2.7 TESTING & IMPLEMENTATION

2.7.1 Testing

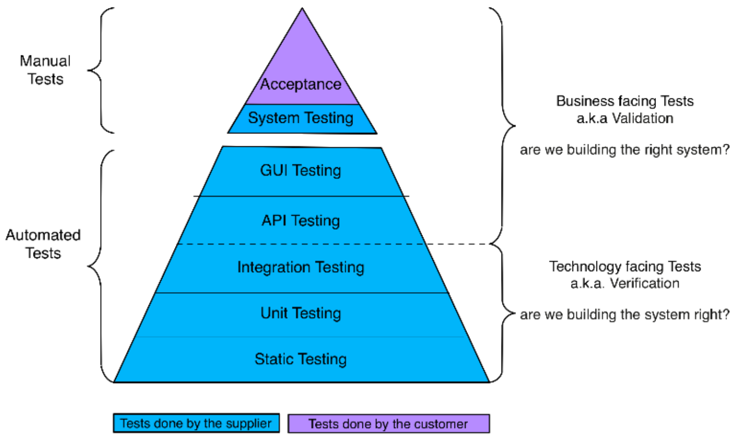


Figure 2.7.1: Levels of testing in software testing

The diagram presented in Figure 2.7.1 illustrates a prominent approach to ensure the quality of the software before its final release. This method involves systematically testing the application at each stage of development and addressing any identified issues or bugs. Through this rigorous testing process, applications undergo refinement, resulting in a higher level of quality and a superior final product. Software testing plays a crucial role in the development lifecycle by identifying defects and errors that may have inadvertently occurred during the earlier phases of development. Contrary to popular misconceptions, testing is not just a formality but an essential step to ensure the effective performance and reliability of a product or computer program. It is indispensable at every stage of the Software Development Life Cycle (SDLC) and encompasses various testing categories to comprehensively evaluate the software's functionality and correctness.

Static Testing:

In the context of your crime analysis component, static testing involves reviewing and analysing the source code, design documents, and other project artifacts without executing the program. It aims to identify issues such as code inconsistencies, syntax errors, and potential vulnerabilities at an early stage to ensure code quality.

Unit Testing:

Unit testing in this component involves testing individual units or functions of the code to ensure they work as expected. For example, you would test specific functions responsible for data preprocessing, decision tree model training, or data visualization to confirm they produce correct results.

Integration Testing:

Integration testing focuses on verifying that different components or module of crime analysis system work together seamlessly. It ensures that the various parts of the application, such as the user input interface, decision tree model, and data visualization components, interact correctly and produce the intended outcomes when combined.

API Testing:

API testing ensures that these interfaces function correctly. It checks if the APIs provide the expected responses and handle various inputs appropriately.

GUI Testing:

This component includes a graphical user interface (GUI) for user interaction, GUI testing assesses the functionality and usability of the interface. It validates that users can input data, receive meaningful results, and navigate the application intuitively.

System Testing:

System testing assesses the entire crime analysis system, considering all integrated components, user interactions, and external dependencies. It verifies that the complete system functions as intended, including features like crime pattern prediction, data visualization, and user input processing.

Acceptance Testing:

Acceptance testing ensures that the crime analysis component meets the requirements and expectations of end-users. It involves real-world scenarios and user-based tests to confirm that the system fulfils its intended purpose effectively, including accurate crime predictions and user-friendly interactions.

Maintenance

The maintenance phase, marking the final stage of this Software Development Life Cycle (SDLC) model, encompasses several crucial functions. These functions include the ongoing management of software updates, addressing any issues, and implementing fixes to ensure the application's continued performance and reliability. Throughout the development process, the created component has undergone rigorous testing across various phases, consistently demonstrating error-free behaviour. To conduct testing in a realistic and comprehensive manner, it's advisable to divide the entire system into distinct segments, allowing for thorough evaluation and validation of each component's functionality. In essence, the maintenance phase serves as the safeguard for the application's long-term sustainability, ensuring it remains up-to-date, functional, and free of defects, thus delivering a seamless and reliable user experience.

Test cases that are done for each testing method is shown below.

Table 2.7.1: Test Case 01

Test Case No	Test Case 01
Description	Verify whether empty validation message is fired for each empty mandatory field.
Test Steps	1. Login to the system. 2. Navigate to the 'crime' option. 3. Click on the 'Predict Crime Pattern' option. 4. Submit the form by clicking on the 'Predict' button without providing values to the fields.
Test Data	String, Integer
Expected Result	The form should not be submitted with missing data. Validation messages should get fired for each empty mandatory field
Actual Result	Pass
User Role	Police team.

Table 2.7.2: Test Case 02

Test Case No	Test Case 02
Description	Verify whether error message is fired when user enters invalid year is given as input (e.g., year that is not in the range).
Test Steps	1. Login to the system. 2. Navigate to the 'Crime' option. 3. Click on the 'Predict Crime Pattern' option. 4. Enter '2050' in the year field. 5. Submit the form by clicking on the 'Predict' button.
Test Data	Integer
Expected Result	Verify that appropriate error messages are displayed. Ensure the form is not submitted with invalid data.
Actual Result	Pass
User Role	Police team.

Table 2.7.3: Test Case 03

Test Case No	Test Case 03
Description	Verify whether correct form of data is submitted
Test Steps	1. Login to the system. 2. Navigate to the ‘Crime’ option. 3. Click on the ‘Predict Crime Rate” 4. Input text for date 5. Click on the ‘Predict’ button.
Test Data	String, Integer
Expected Result	The system should pop up error message. The system should not predict any outcomes.
.Actual Result	Pass
User Role	Police team.

Table 2.7.4: Test Case 04

Test Case No	Test Case 04
Description	Verify that the pattern of the crime is displayed to specific input given
Test Steps	1. Login to the system. 2. Navigate to the 'Crime' option. 3. Click on the 'Predict Crime Pattern. 4. Select 'Area', 'Month' options from the dropdown. 5. Given the desired Year. 6. Click on the 'Predict' button. 7. Predicted pattern for each crime is displayed.
Test Data	String
Expected Result	The system should Display the area, month and year selected with the other information.
Actual Result	Pass
User Role	Police team.

Table 2.7.5: Test Case 05

Test Case No	Test Case 05
Description	Verify that the rate prediction is displayed for five years from the given year.
Test Steps	1. Login to the system. 2. Navigate to the 'Crime' option. 3. Click on the 'Predict Crime Rate. 4. Select 'Crime' option from the dropdown. 5. Input the desired Year. 6. Click on the 'Predict' button. 7. Predicted from the start year inputted
Test Data	String
Expected Result	The system should Display a graph from the inputted year.
Actual Result	Pass
User Role	Police team.

2.7.2 Implementation

Predicting the outcome of a product requires a structured approach, as illustrated in the figure. Developing an effective web application that can engage users and encourage them to share it within their network is a desirable goal. However, building such an application and training the underlying model can be a complex task. To achieve success in creating a web application, it's essential to navigate through the following stages:

1.Specification:

The first step involves clearly defining the project's goals and requirements. It's crucial to have a detailed understanding of what the web application is expected to achieve and the specific features it should include.

2.Design and Implementation:

Once the specifications are in place, the design and implementation phase come into play. During this stage, the actual development of the web application takes shape. This includes designing the user interface, coding the functionality, and ensuring everything aligns with the initial specifications.

3.Validation:

After the application is developed, thorough validation is necessary. This step involves testing the web application extensively to identify and address any issues, bugs, or inconsistencies. It ensures that the application functions correctly and provides a positive user experience.

4.Evolution:

The process doesn't end with the initial launch. To keep the web application relevant and effective, it must evolve over time. This involves continuous monitoring, gathering user feedback, and making improvements or enhancements to adapt to changing user needs and technological advancements.

To train the datasets using Decision tree algorithm, must work through these steps.

- Collect all the datasets from relevant law enforcement agents for example police.
- Store that clearly with the details in csv file.
- Train the datasets.
- Then predict the pattern in the crimes.

Code

Install the necessary libraries:

```
TERMINAL                                                        powershell

-a----          9/5/2023   4:19 PM          6529 Lables.csv
-a----          9/3/2023   2:20 PM           692 manage.py
-a----          9/5/2023   4:19 PM        260513 modified_file2.csv
-a----          9/5/2023   4:19 PM        181753 women_crime.csv


PS C:\Users\acer\Desktop\New folder\policeAnalysis> pip install numpy
>> pip install pandas
>> pip install matplotlib
>> pip install scipy
>> pip install scikit-learn
>> pip install Django
>>
```

Figure 2.7.2.1: Import libraries for predict crime pattern.

Code

Install the necessary libraries :

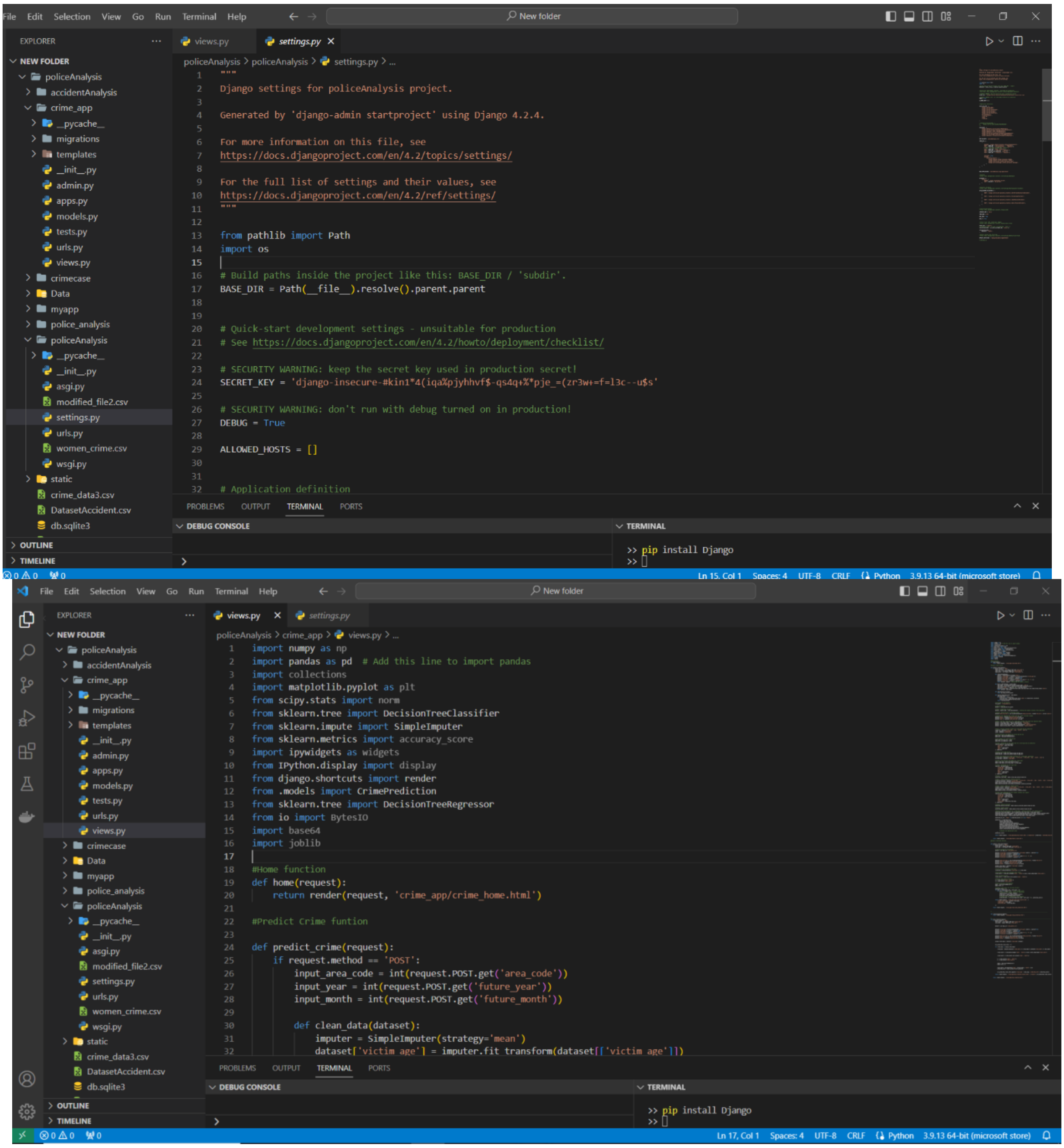


Figure 2.7.2.3: Import libraries for predict crime pattern.

3 RESULTS & DISCUSSION

Within this section that we bridge the gap between raw data and actionable knowledge, unveiling the answers to our research questions, and shedding light on the insights gleaned during the investigative process. Through the presentation of results and their subsequent discussion, we aim to contribute to the existing body of knowledge, address research objectives, and stimulate further inquiry into the subject matter.

3.1 Results

Figure 3.1.1: Results of predict crime pattern.

The figure presented above illustrates the outcomes of forecasting future crime patterns within our system. Users are prompted to provide input parameters, including the year, area, and month, via the "Predict Crime Pattern" page. Once these inputs are entered, our system employs predictive modelling to estimate several key factors. Specifically, it predicts the type of crime that is most likely to occur, the gender demographic that is most likely to be affected, the probable age group of those affected, and whether any vehicles are likely to be involved objects might be stolen. This predictive capability equips users with valuable insights for informed decision-making and crime prevention.

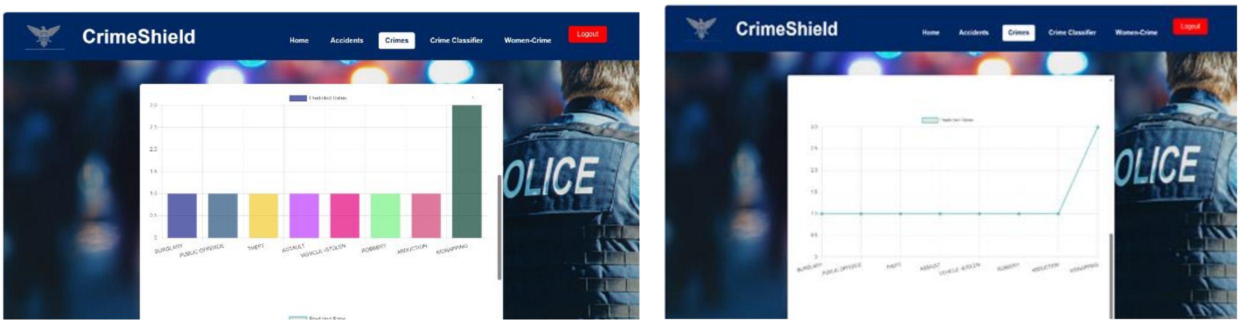


Figure 3.1.2: Results of predict crime rate for a year in a specific Location.

Figure 3.1.2 offers a graphical depiction of a crucial sub-objective within the system's framework. This sub-objective focuses on user interaction, where individuals are invited to input a particular year and location of interest. Subsequently, the system employs its underlying algorithms to assess and rate various crime types based on the provided criteria. This feature carries significant practical value, as it enables users to gain insights into the depth and gravity of criminal activities within a specific area during a given timeframe. These crime ratings can aid law enforcement agencies, local authorities, and communities in prioritizing their crime prevention efforts and adopting appropriate safety measures tailored to the prevailing circumstances. In essence, Figure 2.7 highlights a user-friendly and informative aspect of the system, contributing to its utility in real-world crime analysis and management.

Figure 3.1.3: Results of predict crime rate for next five years.

The depicted image provides a visual representation of one of the system's key functionalities, predicting crime rates for the upcoming five years. In this scenario, users are encouraged to specify a particular crime type and initiate the prediction process by selecting a starting year of interest. Once the user-provided data is inputted into the system, it proceeds to leverage its predictive models to estimate the crime rates for the subsequent five years, commencing from the specified initial year. This feature serves as a valuable tool for both law enforcement agencies and the public, facilitating proactive planning and resource allocation in response to anticipated changes in crime rates. By offering insights into the future landscape of crime, this aspect of the system enhances its practicality in assisting stakeholders in making informed decisions related to crime prevention and safety measures.

3.2 Research Finding

The research findings of the crime analysis component reveal several significant insights into the patterns and dynamics of criminal activities. Through the application of Decision Tree algorithms, the system effectively predicts various crime attributes, including the highest occurring crimes, affected gender demographics, likely age groups, vehicles involved, and objects stolen. These predictions are informed by historical crime data, enabling law enforcement agencies and analysts to proactively address crime-related challenges. Additionally, the system's ability to forecast crime rates for the next five years offers a valuable tool for long-term planning and resource allocation. This forward-looking approach enhances law enforcement agencies' preparedness in addressing emerging crime trends and allocating resources more efficiently. Furthermore, the component's capability to categorize and predict all crime types within a specific area and year empowers users to gain a comprehensive understanding of local crime landscapes. This comprehensive perspective aids in targeted crime prevention strategies and resource allocation. Overall, the research findings underscore the effectiveness of machine learning models, specifically Decision Trees, in crime analysis, offering actionable insights and enhancing the efficiency of crime prevention and investigation efforts.

3.3 Discussion

The crime analysis component discussed here demonstrates the valuable application of machine learning, particularly Decision Tree algorithms, in addressing crime-related challenges. By leveraging historical crime data, the system can predict various crime attributes, including the most frequent types of crimes, the demographics most affected, and even specifics like vehicles involved and objects stolen. This predictive capability offers law enforcement agencies and analysts a powerful tool for proactive crime prevention and resource allocation. Moreover, the system's capacity to forecast crime rates for the next five years provides a forward-looking perspective, aiding long-term planning and preparedness. It allows for a more informed allocation of resources and strategic decision-making.

4 FUTURE SCOPE

The following can be done to extend our work –

Future work in enhancing the crime analysis system involves expanding its scope to encompass a broader spectrum of criminal activities. Currently, the system primarily focuses on crimes like burglary, robbery, theft, vehicle theft, public offenses, assault, kidnapping, and abduction. To make the system more comprehensive, it should include additional crime categories, such as cybercrimes, white-collar crimes, and drug-related offenses, which are increasingly prevalent in modern society. Additionally, the system's dataset is currently limited to specific areas. Expanding the dataset to cover a more extensive range of regions and demographics would enhance the system's applicability and accuracy. This would enable a more comprehensive analysis of crime patterns on a broader scale. Furthermore, future iterations of the system should explore advanced machine learning methods, such as deep learning and ensemble techniques, to improve prediction accuracy and provide more sophisticated insights into crime trends. These advanced methods can handle complex relationships within the data and extract valuable patterns that may go unnoticed with traditional algorithms. By addressing these future prospects, the crime analysis system can evolve into a more powerful and versatile tool for law enforcement agencies and policymakers, enabling more effective crime prevention and intervention strategies.

5 CONCLUSIONS

In conclusion, the crime analysis component presented here demonstrates the potential of machine learning, particularly Decision Tree algorithms, in crime prediction and analysis. While the system currently focuses on specific crime categories and limited geographical areas, it serves as a valuable foundation for future enhancements. Expanding the scope to include a wider range of crime types and extending the dataset to cover diverse regions would improve its applicability. Furthermore, exploring advanced machine learning methods could enhance prediction accuracy and offer more sophisticated insights into crime patterns. Despite its current limitations, this system represents a promising step toward data-driven crime analysis, empowering law enforcement agencies with valuable tools for proactive crime prevention and resource allocation. Future developments hold the potential to make significant strides in addressing complex crime-related challenges and furthering the effectiveness of law enforcement efforts.

6 REFERENCE

[1] N. Shah, N. Bhagat, and M. Shah, "*Crime forecasting using machine learning and computer vision*," Journal of Ambient Intelligence and Humanized Computing, vol. 11, no. 9, pp. 3697-3705, 2020.

[2] S. Walczak, "*Predicting the types of crimes committed in the city of Chicago using neural networks*," Neural Computing and Applications, vol. 31, no. 8, pp. 3795-3802, 2019.

[3] C. Karabo Jenga, C. Catal, and G. Kar, "*Crime forecasting: A machine learning and computer vision approach to crime prediction and prevention*," Journal of Ambient Intelligence and Humanized Computing, vol. 14, no. 10, pp. 4329-4345, 2023.

[4] S. Kim, P. Joshi, P. S. Kalsi, and P. Taheri, "*Crime prediction using decision trees*" in Proceedings of the 2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM), pp. 1285-1292, 2015. DOI: 10.1109/ASONAM.2015.7339281.

[5] McClendon L, Meghanathan N (2015) Using machine learning algorithms to analyze crime data. Mach Lear Appl Int J 2(1):1–12. https://doi.org/10.5121/mlaij.2015.2101

[6] Gorr W, Harries R (2003) Introduction to crime forecasting. Int J Forecast 19(4):551–555. https://doi.org/10.1016/S0169-2070(03)00089-X

[7] Tabedzki C, Thirumalaiswamy A, van Vliet P (2018) Yo home to Bel-Air: predicting crime on the streets of Philadelphia. In: University of Pennsylvania, CIS 520: machine learning

[8] Prithi S, Aravindan S, Anusuya E, Kumar AM (2020) GUI based prediction of crime rate using machine learning approach. Int J Comput Sci Mob Comput 9(3):221–229

[9] Bharati A, Sarvanaguru RAK (2018) Crime prediction and analysis using machine learning. Int Res J Eng Technol 5(9):1037–1042

[10] Kang HW, Kang HB (2017) Prediction of crime occurrence from multi-modal data using deep learning. PLoS One 12(4):e0176244. https://doi.org/10.1371/journal.pone.0176244

[11] Bandekar SR, Vijayalakshmi C (2020) Design and analysis of machine learning algorithms for the reduction of crime rates in India. Procedia Comput Sci 172:122–127. https://doi.org/10.1016/j.procs.2020.05.018

[12] Hossain S, Abtahee A, Kashem I, Hoque M, Sarker IH (2020) Crime prediction using spatio-temporal data. arXiv preprint arXiv:2003.09322. https://doi.org/10.1007/978-981-15-6648-6_22

[13] Stalidis P, Semertzidis T, Daras P (2018) Examining deep learning architectures for crime classification and prediction. arXiv preprint arXiv: 1812.00602. p. 1–13

[14] Ahishakiye E, Taremwa D, Omulo EO, Niyonzima I (2017) Crime prediction using decision tree (J48) classification algorithm. Int J Comput Inf Technol 6(3):188–195

[15] Sun CC, Yao CL, Li X, Lee K (2014) Detecting crime types using classification algorithms. J Digit Inf Manag 12(8):321–327. https://doi.org/10.14400/JDC.2014.12.8.321

[16] Shojaee S, Mustapha A, Sidi F, Jabar MA (2013) A study on classification learning algorithms to predict crime status. Int J Digital Content Technol Appl 7(9):361–369

[17] Obuandike GN, Isah A, Alhasan J (2015) Analytical study of some selected classification algorithms in WEKA using real crime data. Int J Adv Res Artif Intell 4(12):44–48. https://doi.org/10.14569/IJARAI.2015.041207

[18] Wibowo AH, Oesman TI (2020) The comparative analysis on the accuracy of k-NN, naive Bayes, and decision tree algorithms in predicting crimes and criminal actions in Sleman regency. J Phys Conf Ser 1450:012076. https://doi.org/10.1088/1742-6596/1450/1/012076

[19] Gorr W, Olligschlaeger AM, Thompson Y (2000) Assessment of crime forecasting accuracy for deployment of police. Int J Forecast 2000:743–754

[20] Alves LGA, Ribeiro HV, Rodrigues FA (2018) Crime prediction through urban metrics and statistical learning. Phys A Stat Mech Appl 505:435–443. https://doi.org/10.1016/j.physa.2018.03.084

[21] Stec A, Klabjan D (2018) Forecasting crime with deep learning. arXiv preprint arXiv:1806.01486. p. 1–20

[22] Zhuang Y, Almeida M, Morabito M, Ding W (2017) Crime hot spot forecasting: a recurrent model with spatial and temporal information. In: Paper presented at the IEEE international conference on big knowledge. IEEE, Hefei 9-10 August 2017. https://doi.org/10.1109/ICBK.2017.3

7 APPENDICES

```python
import numpy as np
import pandas as pd  # Add this line to import pandas
import collections
import matplotlib.pyplot as plt
from scipy.stats import norm
from sklearn.tree import DecisionTreeClassifier
from sklearn.impute import SimpleImputer
from sklearn.metrics import accuracy_score
import ipywidgets as widgets
from IPython.display import display
from django.shortcuts import render
from .models import CrimePrediction
from sklearn.tree import DecisionTreeRegressor
from io import BytesIO
import base64
import joblib

#Home function
def home(request):
    return render(request, 'crime_app/crime_home.html')
```

```python
def predict_crime(request):
    if request.method == 'POST':
        input_area_code = int(request.POST.get('area_code'))
        input_year = int(request.POST.get('future_year'))
        input_month = int(request.POST.get('future_month'))

        def clean_data(dataset):
            imputer = SimpleImputer(strategy='mean')
            dataset['victim_age'] = imputer.fit_transform(dataset[['victim_age']])
            dataset['crime_type'].fillna(-1, inplace=True)
            dataset['area_code'].fillna(-1, inplace=True)
            dataset['victim_sex'] = dataset['victim_sex'].map({'M': 0, 'F': 1})
            dataset['victim_sex'].fillna(-1, inplace=True)
            return dataset, imputer  # Returning the imputer instance

        def map_to_codes(dataset, column_name):
            code_table = pd.DataFrame(columns=[column_name, 'code'])
            code_table[column_name] = dataset[column_name].drop_duplicates()
            code_table['code'] = range(101, 101 + len(code_table))
            dataset[column_name] = dataset[column_name].map(code_table.set_index(column_name)['code'])
            return dataset, code_table

        def load_dataset(file_path):
            return pd.read_csv(file_path)

        def reverse_map(predictions, code_table):
            reversed_values = []
            for prediction in predictions:
                reversed_value = code_table[code_table['code'] == prediction].iloc[0][0]
                reversed_values.append(reversed_value)
            return reversed_values

        # File path for the dataset
        file_path = 'crime_data3.csv'
```

```python
# Load the dataset
dataset = load_dataset(file_path)

# Step 1: Preprocess the data
dataset, imputer_age = clean_data(dataset)  # Getting the imputer instance from clean_data

# Convert 'date_occurred' to datetime format
dataset['date_occurred'] = pd.to_datetime(dataset['date_occurred'], format='%d.%m.%Y', errors='coerce')

# Separate the combined date column into day, month, and year columns
dataset['day'] = dataset['date_occurred'].dt.day
dataset['month'] = dataset['date_occurred'].dt.month
dataset['year'] = dataset['date_occurred'].dt.year
```

```python
# Step 2: Extract relevant columns for victim age and sex prediction
X_age_sex = dataset[['area_code', 'day', 'month', 'year']]
y_age = dataset['victim_age']
y_sex = dataset['victim_sex']

# Create decision tree models for victim age and sex
model_age = DecisionTreeRegressor()
model_sex = DecisionTreeClassifier()

# Train models for victim age and sex
model_age.fit(X_age_sex, y_age)
model_sex.fit(X_age_sex, y_sex)

# Create a DataFrame for the input data for victim age and sex prediction
input_df_age_sex = pd.DataFrame({
    'area_code': input_area_code,
    'day': 1,
    'month': input_month,
    'year': input_year
}, index=[0])

# Predict victim age and sex
predicted_age = model_age.predict(input_df_age_sex)
predicted_sex = model_sex.predict(input_df_age_sex)

# Select the features and target variables for 'crime_type' prediction
X_crime_type = dataset[['victim_age', 'victim_sex', 'area_code', 'day', 'month', 'year']]
y_crime_type = dataset['crime_type']

# Create a new decision tree model for 'crime_type'
model_crime_type = DecisionTreeClassifier()
model_crime_type.fit(X_crime_type, y_crime_type)
```

```python
    input_df = pd.DataFrame({
        'victim_age': predicted_age,
        'victim_sex': predicted_sex,
        'area_code': input_area_code,
        'day': 1,
        'month': input_month,
        'year': input_year
    }, index=[0])

    # Predict 'crime_type'
    predicted_crime_type = model_crime_type.predict(input_df)

    # Step 3: Train a model for 'vehicle_involved'
    X_vehicle_involved = dataset[['victim_age', 'victim_sex', 'area_code', 'day', 'month', 'year','crime_type']]
    y_vehicle_involved = dataset['vehicle_involved']
    model_vehicle_involved = DecisionTreeClassifier()
    model_vehicle_involved.fit(X_vehicle_involved, y_vehicle_involved)

    # Step 4: Train a model for 'object_stolen'
    X_object_stolen = dataset[['victim_age', 'victim_sex', 'area_code', 'day', 'month', 'year','crime_type']]
    y_object_stolen = dataset['object_stolen']
    model_object_stolen = DecisionTreeClassifier()
    model_object_stolen.fit(X_object_stolen, y_object_stolen)

    # Create input DataFrame for new columns (without user inputs)
    input_df_new = pd.DataFrame({
        'victim_age': predicted_age,
        'victim_sex': predicted_sex,
        'area_code': input_area_code,
        'day': 1,
        'month': input_month,
        'year': input_year,
        'crime_type': predicted_crime_type,
    }, index=[0])

    # Predict 'vehicle_involved'
    predicted_vehicle_involved = model_vehicle_involved.predict(input_df_new)
```

```
        # Predict 'object_stolen'
        predicted_object_stolen = model_object_stolen.predict(input_df_new)

        # Reverse map the predictions for 'crime_type', 'vehicle_involved', and 'object_stolen'
        predicted_crime_type_mapped = reverse_map(predicted_crime_type, crime_types_table)
        predicted_vehicle_involved_mapped = reverse_map(predicted_vehicle_involved, vehicle_involved_table)
        predicted_object_stolen_mapped = reverse_map(predicted_object_stolen, object_stolen_table)
        input_area_code_mapped = reverse_map([input_area_code], area_codes_table)

        predicted_sex_str = "Male" if predicted_sex[0] == 0 else "Female"

        prediction = CrimePrediction(
                future_year=input_year,
                future_month=input_month,
                area_code=input_area_code_mapped[0],
                highest_crime=predicted_crime_type_mapped[0],
                affected_gender=predicted_sex_str,
                age_group_affected=predicted_age[0],
                vehicles_involved=predicted_vehicle_involved_mapped[0],
                object_stolen=predicted_object_stolen_mapped[0],

                # Replace with appropriate value
            )
        prediction.save()

        return render(request, 'crime_app/prediction_result.html', {'prediction': prediction, 'selected_area_text': input_area_code})


    return render(request, 'crime_app/predict_crime.html')
```

```
def predict_crime_rate(request):
    if request.method == 'POST':
        crime_type = request.POST.get('crime_type')
        start_year = int(request.POST.get('start_year'))

        # Load and preprocess the dataset
        dataset = pd.read_csv('crime_data3.csv')

        dataset['victim_age'].fillna(round(dataset['victim_age'].mean()), inplace=True)
        dataset['crime_type'].fillna(-1, inplace=True)
        dataset['area_code'].fillna(-1, inplace=True)
        dataset['victim_sex'] = dataset['victim_sex'].map({'M': 0, 'F': 1})
        dataset['victim_sex'].fillna(-1, inplace=True)


        dataset['date_occurred'] = pd.to_datetime(dataset['date_occurred'], format='%d.%m.%Y')
        dataset['year'] = dataset['date_occurred'].dt.year
        dataset['day'] = dataset['date_occurred'].dt.day
        dataset['month'] = dataset['date_occurred'].dt.month

        unique_crime_types = dataset['crime_type'].unique()
        print(unique_crime_types)

        # Filter data for the given crime type
        crime_data = dataset[dataset['crime_type'] == crime_type]

        # Group data by year and calculate total counts
        crime_counts = crime_data.groupby(['year', 'month']).size().reset_index(name='total_count')

        # Sort the data by year
        crime_counts = crime_counts.sort_values(['year', 'month'])

        # Features and target variable
        X = crime_counts[['year','month']]
        y = crime_counts['total_count']
```

```
        # Train a Decision Tree Regressor
        model = DecisionTreeRegressor()
        model.fit(X, y)

        # Predict crime counts for the next 10 years
        predicted_counts = []
        for year in range(start_year, start_year + 5):
            for month in range(1, 13):  # Loop through months
                # Construct the feature vector for prediction
                feature_vector = [[year, month]]
                prediction = model.predict(feature_vector)
                predicted_counts.append(prediction[0])
                # Prepare data for rendering in the template
                predicted_data = list(zip(range(start_year, start_year + 5), predicted_counts))

        return render(request, 'crime_app/crime_rate_results.html', {
            'unique_crime_types': unique_crime_types,
            'crime_type': crime_type,
            'predicted_data': predicted_data,

        })

    return render(request, 'crime_app/crime_rate_prediction.html')
```

```
def crime_prevention(request):
    return render(request, 'crime_app/crime_prevention.html')


# ...................................................................................................
def area_crimerate(request):
    if request.method == 'POST':
        future_year = int(request.POST.get('future_year'))
        area_code = request.POST.get('area_code')


        dataset = pd.read_csv('crime_data3.csv')


        dataset['victim_age'].fillna(round(dataset['victim_age'].mean()), inplace=True)
        dataset['crime_type'].fillna(-1, inplace=True)
        dataset['area_code'].fillna(-1, inplace=True)
        dataset['victim_sex'] = dataset['victim_sex'].map({'M': 0, 'F': 1})
        dataset['victim_sex'].fillna(-1, inplace=True)

        dataset['date_occurred'] = pd.to_datetime(dataset['date_occurred'], format='%d.%m.%Y')
        dataset['year'] = dataset['date_occurred'].dt.year
        dataset['month'] = dataset['date_occurred'].dt.month


        unique_crime_types = dataset['crime_type'].unique()


        all_predicted_crime_rates = []

        for crime_type in unique_crime_types:

            crime_data = dataset[(dataset['crime_type'] == crime_type) & (dataset['area_code'] == area_code)]
```

```python
        crime_counts = crime_data.groupby(['year', 'month']).size().reset_index(name='total_count')


        crime_counts = crime_counts.sort_values(['year', 'month'])


        X = crime_counts[['year', 'month']]
        y = crime_counts['total_count']


        model = DecisionTreeRegressor()
        model.fit(X, y)


        future_data = pd.DataFrame({'year': [future_year], 'month': [1]})
        predicted_count = model.predict(future_data)[0]


        all_predicted_crime_rates.append({'crime_type': crime_type, 'predicted_rate': predicted_count})

    return render(request, 'crime_app/area_crimerate_results.html', {'results': all_predicted_crime_rates})


return render(request, 'crime_app/area_crimerate.html')
```

policeAnalysis > crime_app > urls.py > ...

```python
"""
URL configuration for CrimePrediction project.

The `urlpatterns` list routes URLs to views. For more information please see:
    https://docs.djangoproject.com/en/4.2/topics/http/urls/
Examples:
Function views
    1. Add an import:  from my_app import views
    2. Add a URL to urlpatterns:  path('', views.home, name='home')
Class-based views
    1. Add an import:  from other_app.views import Home
    2. Add a URL to urlpatterns:  path('', Home.as_view(), name='home')
Including another URLconf
    1. Import the include() function: from django.urls import include, path
    2. Add a URL to urlpatterns:  path('blog/', include('blog.urls'))
"""
from django.contrib import admin
from django.urls import path
from django.urls import path, include
from django.views.generic import RedirectView
from crime_app.views import predict_crime,crime_prevention,predict_crime_rate,area_crimerate
from . import views
from django.views.generic import TemplateView

urlpatterns = [
    path('admin/', admin.site.urls),
    path('', views.home, name='home'),
    path('predict_crime_rate/', views.predict_crime_rate, name='predict_crime_rate'),
    path('predict_crime/', views.predict_crime, name='predict_crime'),
    path('crime_prevention/', views.crime_prevention, name='crime_prevention'),
    path('area_crimerate/', views.area_crimerate, name='area_crimerate'),
]

```

```
            <li class="nav-option">
                <form method="post" action="{% url 'user_logout' %}">
                    {% csrf_token %}
                    <button class="logout-button" type="button" id="logout-button">Logout</button>
                </form>
            </li>
        </ul>
    </div>
    <div class="container">
        <div class="content">
            <div class="box">
                <h1>Crime Pattern for the year {{ prediction.future_year }} in Month {{ prediction.future_month }} At {{ prediction.area_co
                <h3 style="text-align: left;">Selected Area:   <strong>{{ prediction.area_code }}</strong></h3>
                <h3 style="text-align: left;">Future Year:     <strong>{{ prediction.future_year }}</strong></h3>
                <h3 style="text-align: left;">Future Month:    <strong>{{ prediction.future_month }}</strong></h3>
                <h3 style="text-align: left;">Highest Crime that can occur :    <strong>{{ prediction.highest_crime }}</strong></h3>
                <h3 style="text-align: left;">Mostly Likely Affected Gender:    <strong>{{ prediction.affected_gender }}</strong></h3>
                <h3 style="text-align: left;">Mostly Likely Age Group Affected:   <strong>{{ prediction.age_group_affected }}</strong></h3>
                <h3 style="text-align: left;">Mostly Likely Vehicles Involved:   <strong>{{ prediction.vehicles_involved }}</strong></h3>
                <h3 style="text-align: left;">Mostly Likely Object Stolen:    <strong>{{ prediction.object_stolen }}</strong></h3>
                <!-- <a class="back-link" href="{% url 'predict_crime' %}">Back to Prediction</a> -->
            </div><br><br>
            <div class="back-button-container">
                <a class="back-link button" href="{% url 'predict_crime' %}">Back to Predict</a>
            </div>
        </div>
    </div>
    <div id="logout-modal" class="modal">
        <div class="modal-content">
            <p style="color:□black">Are you sure you want to logout?</p>
            <button id="confirm-logout" class="logout-button">Yes</button>
            <button id="cancel-logout" class="logout-button">Cancel</button>
        </div>
    </div>
</body>
</html>
```

```
        var trace = {
            x: years,
            y: counts,
            type: 'scatter',
            mode: 'lines+markers',
            marker: { color: '#007bff' },
            line: { shape: 'spline' },
            name: 'Crime Occurrence'
        };

        var layout = {
            title: 'Predicted Crime Rate Over Next 5 Years',
            xaxis: {
                title: 'Year',
                tickmode: 'linear',  // Set the tick mode to linear
                dtick: 1             // Set the interval between ticks to 1 (every year)
            },
            yaxis: { title: 'Crime Rate' },
            margin: { l: 50, r: 50, b: 50, t: 80 },
            plot_bgcolor: '#ffffff',
            paper_bgcolor: '#f5f5f5'
        };

        var data = [trace];
        Plotly.newPlot('chart', data, layout);
    </script>
   <a class="back-button" href="{% url 'predict_crime_rate' %}">Back to Predict</button>
    <div id="logout-modal" class="modal">
        <div class="modal-content">
            <p style="color:□black">Are you sure you want to logout?</p>
            <button id="confirm-logout" class="logout-button">Yes</button>
            <button id="cancel-logout" class="logout-button">Cancel</button>
        </div>
    </div>
</body>
```

```html
    <div class="container">
        <h1>Predict Crime Rate For 5 Years </h1>
        <form method="post" action="{% url 'predict_crime_rate'%}">
            {% csrf_token %}
            <label for="crime_type">Crime Type:</label>
            <select name="crime_type" id="crime_type" required>
                <option value="" disabled selected>Select Crime</option>
                <option value="THEFT">THEFT</option>
                <option value="BURGLARY">BURGLARY</option>
                <option value="PUBLIC OFFENCE">PUBLIC OFFENCE</option>
                <option value="ASSAULT">ASSAULT</option>
                <option value="ROBBERY">ROBBERY</option>
                <option value="ABDUCTION">ABDUCTION</option>
                <option value="KIDNAPPING">KIDNAPPING</option>
                <option value="VEHICLE -STOLEN">VEHICLE -STOLEN</option>
            </select>
            <label for="start_year">Start Year:</label>
            <input type="number" name="start_year" id="start_year" required>
            <br>
            <button type="submit" onclick="validateYear()">Predict</button>
        </form>
    </div>
    <a class="back-button" href="http://127.0.0.1:8000/crime/">Back</button>
        <div id="logout-modal" class="modal">
            <div class="modal-content">
                <p style="color: black">Are you sure you want to logout?</p>
                <button id="confirm-logout" class="logout-button">Yes</button>
                <button id="cancel-logout" class="logout-button">Cancel</button>
            </div>
        </div>
</body>
</html>
```

```html
    <center><p class="typewriter"><bold>YOU CAN NOW PREDICT THE CRIME RATE ACCORDING</br>FOR AREAS</bold></p></center>
    <div class="container">
    <h1>Crime Rate Prediction</h1>
    <form method="post" action="{% url 'area_crimerate' %}">
        {% csrf_token %}
        <label for="area_code">Select Area:</label>
            <select name="area_code" required>
            <option value="" disabled>Select Area</option>
            <option value="Modara">Modara</option>
            <option value="Kotahena">Kotahena</option>
            <option value="Variyapola">Variyapola</option>
            <option value="Slave Island">Slave Island</option>
            <option value="Keselwatta">Keselwatta</option>
            <option value="Wolfendhal Street">Wolfendhal Street</option>
            <option value="Fort">Fort</option>
            <option value="Kollupitiya">Kollupitiya</option>
            <option value="Maligawatta">Maligawatta</option>
            <option value="Kurunegala Town">Kurunegala Town</option>
            <option value="Mallawapitiya">Mallawapitiya</option>
            <option value="Dematagoda">Dematagoda</option>
            <option value="Bluemandal">Bluemandal</option>
            <option value="Colombo Harbour">Colombo Harbour</option>
            <option value="Mattakkuliya">Mattakkuliya</option>
            <option value="Pettah">Pettah</option>
            <option value="Grandpass">Grandpass</option>
            <option value="Bambalapitiya">Bambalapitiya</option>
            <option value="Wellawatta">Wellawatta</option>
            <option value="Fore shore">Fore shore</option>
            <option value="Maradana">Maradana </option>
            </select>
        <br>
        <label for="future_year">Future Year:</label>
        <input type="number" id="future_year" name="future_year" required>
        <br>
        <button type="submit" value="Predict Crime Rates" onclick="validateYear()">Predict Crime Rates</button>
    </form>
```

```
// Get references to the chart elements
var barChart = document.getElementById('barChart').getContext('2d');
var lineChart = document.getElementById('lineChart').getContext('2d');

// Create a bar chart
new Chart(barChart, {
    type: 'bar',
    data: {
        labels: crimeTypes,
        datasets: [{
            label: 'Predicted Rates',
            data: predictedRates,
            backgroundColor: barColors,
            borderColor: barColors,
            borderWidth: 1
        }]
    },
    options: {
        scales: {
            y: {
                beginAtZero: true
            }
        }
    }
});

// Create a line chart
new Chart(lineChart, {
    type: 'line',
    data: {
        labels: crimeTypes,
        datasets: [{
            label: 'Predicted Rates',
            data: predictedRates,
            fill: false,
            borderColor: 'rgba(75, 192, 192, 1)', // Adjust the color as needed
            borderWidth: 2
```


CrimeShield
Home
Accidents
Crimes
Crime Classifier
Women-Crime
Logout
Welcome to Crime Prediction
Explore our advanced crime prediction tools and resources.
Predict Crime Pattern
Predict Crime Rate
Crime Analysis
Crime Prevention Strategies


CrimeShield
Home
Accidents
Crimes
Crime Classifier
Women-Crime
Logout
Crime Prevention Strategies for Police
Burglary Prevention


CrimeShield
Home
Accidents
Crimes
Crime Classifier
Women-Crime
Logout
YOU CAN NOW PREDICT THE CRIME RATE AC
FOR AREAS
Crime Rate
Prediction
Select Area:
Modara
Future Year:
Predict Crime Rates
POLICE