The Future of Crime Prevention: Police Case Analysis Using Machine Learning

Project ID: 2023-224

Final (Draft) Report

Dharsan. R – IT20003982

Krishanthini. M – IT19980928

Traveena. C – IT20001452

Anubama. L – IT20068196

BSc (Hons) in Information Technology

Specializing in Software Engineering

Department of Computer Science & Software Engineering

Sri Lankan Institute of Information Technology

Sri Lanka

September 2023

The Future of Crime Prevention: Police Case Analysis Using Machine Learning

Project ID: 2023-224

Dharsan. R – IT20003982

Krishanthini. M – IT19980928

Traveena. C – IT20001452

Anubama. L – IT20068196

BSc (Hons) in Information Technology

Specializing in Software Engineering

Department of Computer Science & Software Engineering

Sri Lankan Institute of Information Technology

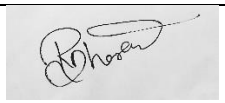
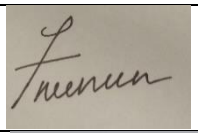
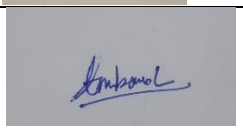
Sri Lanka

September 2023

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
Dharsan. R	IT20003982	
Krishanthini. M	IT19980928	
Traveena. C	IT20001452	
Anubama. L	IT20068196	

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

Signature of the supervisor

(Ms. Hansika Mahaadikara) Date

Signature of co-supervisor

(Ms. Sanjeevi Chandrasiri) Date

ACKNOWLEDGEMENT

We extend our heartfelt gratitude to our module coordinator, Mr. Jayantha Amararachchi, for his unwavering support, motivation, and valuable insights that propelled our project forward with enthusiasm. We would also like to express our appreciation to Ms. Hansika Mahaadikara and our co-supervisor, Ms. Sanjeevi Chandrasiri, for their invaluable guidance and support from the project's inception to its completion. Their wealth of ideas greatly enriched our project's development, and their patience and unwavering commitment helped us overcome challenges. Additionally, we acknowledge the contributions of our lecturers, assistant lecturers, instructors, fellow group members, and the dedicated academic and non-academic staff at SLIIT, who provided unwavering support and assistance throughout the project. Finally, our heartfelt thanks go out to our beloved families and friends, who stood by us as pillars of strength and provided invaluable moral support during challenging moments in the project's journey.

ABSTRACT

This research thesis explores the future of crime prevention through innovative machine learning applications, encompassing four interconnected components: The first component leverages historical crime data to predict and analyze various crime attributes, including types, demographics, age groups, vehicles involved, and stolen items. It empowers law enforcement with actionable insights, enabling proactive prevention, effective resource allocation, and long-term crime rate forecasts. Next, within specific divisions, advanced statistical models forecast future accident percentages and identify causation factors. This component not only predicts accidents but also formulates tailored prevention strategies, leveraging historical data to promote safety and reduce incidents. Additionally, another part focuses on using machine learning to analyze police case documents, predicting potential crimes based on historical data patterns. This predictive model assists law enforcement in resource allocation, optimizing crime prevention efforts, and enhancing investigation efficiency. And lastly, a specialized system concentrates on reducing and preventing crimes against women. By clustering crimes based on location, type, and year, it identifies patterns and trends. Utilizing historical data, this component forecasts the likelihood of crimes in specific areas and times, ultimately contributing to the enhancement of women's safety. Collectively, these components offer a comprehensive approach to data-driven crime analysis and prevention, underpinned by historical data, advanced analytics, and machine learning. By transforming law enforcement practices, improving efficiency, and contributing to safer communities, this research showcases the potential of technology in shaping the future of crime prevention and public safety. These innovative approaches provide valuable tools to enhance law enforcement efforts and create safer environments.

Table of Content

LIST OF FIGURES

Figure Page

LIST OF TABLES

Table **Page**

LIST OF ABBREVIATIONS

Abbreviations	Description
ML	Machine Learning
ARIMA	Auto Regressive Integrated Moving Average
NLP	Natural Language Processing
IO	Input/Output
PDF	Portable Document Format
CSV	Comma-Separated Values
SDLC	Software Development Life Cycle
GUI	Graphical User Interface
NLTK	Natural Language Toolkit
SVC	Support Vector Classifier
DIG	Deputy Inspector General
UAT	User Acceptance Testing
API	Application Programming Interface
HTTP	Hypertext Transfer Protocol
TF-PDF	Term Frequency, Proportional Document Frequency

1. INTRODUCTION

1.1 Background Study and Literature

1.1.1 Introduction

Law enforcement agencies in Sri Lanka grapple with formidable challenges in effectively addressing accidents and crimes. Historically, traditional approaches to case analysis and investigation have proven to be labour-intensive, error-prone, and lacking predictive capabilities. These conventional methods often lead to delayed justice, while the escalating volume of data overwhelms human analytical capacities. Moreover, the absence of proactive predictive tools hinders law enforcement's ability to anticipate and prevent accidents and crimes. The pressing need for a more advanced, data-driven solution to enhance public safety and optimize law enforcement efforts becomes evident in the face of these challenges.

In response to these issues, this research project, titled "The Future of Crime Prevention: Police Case Analysis Using Machine Learning," endeavours to revolutionize how law enforcement agencies tackle the multifaceted issues of crime and accidents. This integrated system comprises four main components, each with its distinct objectives.

Accidents are unpredictable events that can have devastating consequences, making it paramount to understand their patterns and causes. The "Accident Case Analysis" component aims to empower law enforcement agencies by predicting accident percentages, fatal percentages, and accident causes for the next three years within specified divisions. This component serves as a solution to the labor-intensive and time-consuming nature of traditional accident case analysis. By leveraging machine learning techniques, it forecasts future accident trends, identifies causes, and enables in-depth analyses. It empowers law enforcement with actionable insights to proactively address safety concerns and allocate resources effectively, even predicting peak accident occurrence times.

Crime analysis has always been pivotal for law enforcement, primarily relying on statistical methodologies. This research recognizes the potential of integrating advanced machine learning techniques to enable proactive crime prediction and prevention. The core objective of our research is to create a robust system that empowers users to gain insights into crime dynamics by providing essential inputs: the year, month, and location of interest. Through predictive capabilities, it identifies the highest occurring crimes, demographics of affected individuals,

vehicle details, and information regarding stolen objects. This system offers predictions of crime rates for the next five years and generates graphical representations of crime statistics, harnessing the power of machine learning to address the pressing need for accurate and proactive crime analysis.

Additionally, the field of crime analysis has evolved significantly, shifting from manual data entry and analysis to technology-driven solutions. The case document classification using text analysis system is proposed to eliminate manual case searches, expedite investigations, reduce errors, and enhance decision-making by extracting crucial information from unstructured text data. This solution promises substantial benefits, including reduced case backlog and improved efficiency and accuracy in the criminal justice system.

Furthermore, the economic crisis in Sri Lanka has led to a drastic increase in criminal cases, with a specific focus on crimes against women. These crimes have a severe impact on women's well-being, often going unreported due to stigma, fear, and mistrust in the legal system. The "Crimes Against Women Analysis" component proposes a solution by clustering crimes against women based on characteristics such as location, type, and year. This component aims to develop a crime forecasting model, enabling law enforcement agencies to take preventive measures and ensure women's safety.

By integrating machine learning algorithms into police case analysis, this research project aims to overcome the challenges faced by traditional methods. It provides law enforcement agencies with advanced analytical tools, predictive capabilities, and efficient systems to allocate resources effectively, prevent accidents and crimes, and ultimately enhance public safety in Sri Lanka. In the subsequent sections of this report, we will delve into each component's methodology, data analysis, and findings, shedding light on their potential to transform law enforcement and public safety.

1.1.2 Background survey

Numerous studies have extensively explored approaches in police accident case analysis, crime analysis, text classification, and crimes against women. Li, Wu, and Peng [1] developed a traffic accident prediction model using spiking neural networks (SNNs) and convolutional neural networks (CNNs). This research addresses the critical need for accurate post-impact prediction in traffic accident management, utilizing SNNs to capture spatial and temporal features effectively, potentially improving prediction accuracy over traditional methods. V. Prasannakumar, H. Vijith, R. Charutha, and N. Geetha [2] conducted research on road accident analysis using geo-information technology to identify accident hotspots and employed spatial statistics, including Moran's I and Getis-Ord Gi* statistics, within a GIS-based framework to assess spatial clustering, aiding traffic management and safety decisions. It aimed to assess the spatial and temporal patterns of accidents in the Thiruvananthapuram city area, providing insights into accident hotspots and cold spots. This research contributed to understanding the distribution and variations of road accidents, which can inform traffic management strategies and accident reduction efforts, emphasizing the significance of geospatial and temporal data analysis in accident research and prevention.

In the study conducted by Manzoor et al. [3], the focus was on predicting the severity of road accidents, a critical concern given the numerous factors involved. The authors employed an ensemble model called RFCNN and compared its performance with various base learner models, including tree-based ensemble models (RF, AC, ETC, GBM) and an ensemble of regression algorithms (Voting classifier, LR+SGD). Notably, Random Forest (RF) identified 20 significant features from the dataset, and these features were used for the experiments. The results demonstrated that RFCNN outperformed other models in terms of accuracy, achieving an impressive accuracy rate of 99.1%, while also excelling in precision, recall, and F-score. The study emphasized the importance of identifying significant features and highlighted the potential for improving accident severity prediction while reducing data collection costs. Senanayake and Joshi [4] developed the Road Accident Pattern Miner (RAP miner), an innovative system using a hybrid learning algorithm and the Self-Expiring Association (SEA) algorithm for association rule mining. This system achieved an impressive accuracy rate of 92.75%, surpassing other algorithms' maximum accuracy of 92.25%. Integrating SEA with Case-Based Reasoning (CBR) significantly enhanced the RAP miner's performance, reducing processing time by a remarkable 67%. This advancement holds promise for real-time accident prediction based on location and severity, offering significant contributions to road safety. Yang, Han, and Chen’s study [5], the prediction

of traffic accident severity was achieved using a Random Forest algorithm. Their model exhibited a remarkable accuracy rate of 80%, outperforming other machine learning models. Key accident characteristics, such as location, collision pattern, road information, and speed limits, were employed to enhance prediction accuracy, offering valuable insights for traffic safety management and accident prevention.

A noteworthy study by Neil Shah, Nandish Bhagat, and Manan Shah [6] has spotlighted the potential of machine learning and computer vision in crime forecasting. Their research emphasizes the significance of harnessing data streams such as facial recognition, number plate recognition, augmented and mixed realities, location determination, and object identification to predict and preempt criminal activities. The proposal to employ motive as a critical criterion for assessing the nature of a crime and comprehensive categorization for efficient analysis signifies the depth of their approach. Steven Walczak's work [7] has made strides in the application of neural network models for predicting specific crime types based on temporal and spatial information. His findings reveal that neural network models can predict crime types with remarkable accuracy, offering valuable insights for police decision-making in crime prevention strategies.

Karabo Jenga, Cagatay Catal, and Gorkem Kar [8] have explored the versatility of machine learning and data mining in crime prediction and prevention. Their research underscores the adaptability of these technologies in addressing the complex dynamics of criminal activities. Furthermore, the work of Suhong Kim, Param Joshi, Paraminder Singh Kalsi, and Pooya Taheri [9] highlights the significance of decision trees in crime analysis. Using a dataset of over 560,000 records to predict crimes in Vancouver, their study achieved notable accuracy rates ranging from 39% to 44%, depending on crime categories and timing. This research underscores the utility of decision trees in enhancing crime prediction accuracy. The literature on crime analysis using machine learning is diverse, covering domains like e-government, serial crime patterns, theft, cyber-crimes, social crimes, and text mining.

The background survey on crime analysis and classification using machine learning techniques is extensive and diverse, with a focus on various aspects of crime detection, prediction, and prevention. The studies mentioned in the references provide valuable insights into the use of machine learning algorithms for crime analysis, classification, and prediction, covering different domains, such as e-government, serial criminal patterns, theft crimes, cyber-crimes, social crimes, and text mining for crime analysis. Chih-Hao Ku et al. [10] propose a decision support system for automated crime report analysis and classification in e-government. They leverage machine

learning techniques to automatically classify crime reports into predefined categories, which can assist law enforcement agencies in efficiently analyzing and managing crime data.Dahbur and Muscarello [11] present a classification system for serial criminal patterns using artificial intelligence and law. They propose a rule-based expert system that uses decision trees and statistical techniques to identify patterns in serial crimes, which can aid in profiling and predicting serial criminal behavior. Ghankutkar et al. [12] propose a machine learning model for analyzing crime news. They utilize natural language processing techniques and machine learning algorithms to classify crime news articles into categories such as robbery, murder, and fraud, to aid in crime analysis and prediction. Qi [13] proposes a text classification approach for theft crimes based on the TF-IDF (Term Frequency-Inverse Document Frequency) technique and the XGBoost machine learning model. The proposed method effectively classifies theft crimes into different categories, such as pickpocketing, burglary, and shoplifting, based on text data extracted from crime reports. Alruily et al. [14] focus on crime type document classification from an Arabic corpus. They propose a machine learning-based approach that uses features such as keywords, text statistics, and machine learning classifiers to automatically classify Arabic crime documents into different crime types, such as theft, assault, and fraud, which can assist in crime analysis in Arabic-speaking regions.

In the context of mitigating crimes against women and children, Rokonuzzaman Reza [15] introduced an innovative approach leveraging machine learning techniques. Their system analyzed diverse data sources, including social media and news articles, to detect patterns of oppression. Impressively, this system achieved an exceptional accuracy rate of 99%. It not only identifies such incidents but also offers real-time monitoring and intervention capabilities, ensuring swift responses to address these issues promptly, thus contributing significantly to creating a safer environment for women and children. Additionally, Adderley [16] conducted a comprehensive examination of crime predictive systems with a focus on data analysis and machine learning algorithms. Their research aimed to forecast criminal activities, taking into account the effectiveness, ethical considerations, and potential biases within these systems. While specific accuracy rates may vary, their work underscores the importance of considering these critical factors in crime prediction models.

In a different approach, Islam et al. [17] introduced the "Joy 109" app, which empowers users to send distress messages containing GPS locations, audio recordings, and photos to relevant authorities. This innovative tool enhances response times and situational awareness, although a specific accuracy rate is not mentioned. Furthermore, Kiani et al. [18] proposed a crime prediction

analysis approach involving clustering algorithms. This method identifies patterns in crime data and forecasts criminal activities within specific areas. While accuracy rates are not explicitly mentioned, the use of clustering algorithms highlights a data-driven strategy for crime prediction. Lastly, Karmakar et al. [19] introduced the "SafeBand" wearable device, complemented by mobile apps for victim assistance and police support. This wearable allows for location tracking and emergency messaging, providing an added layer of security. Specific accuracy rates for this device are not provided in the context. These studies underscore the remarkable potential of machine learning, data analysis, and innovative technologies in the fields of traffic accident prediction, crime analysis, document classification and crimes against women.

1.2 Research Gap

In the realm of traffic accident prediction and analysis, there is a substantial research gap, particularly when it comes to combining the prediction of accident percentages for future years, causative factors, and in-depth accident case analysis into a unified system. While existing studies have made significant progress in predicting accident locations and causes, the holistic approach proposed here remains unexplored. The novelty lies in the integration of these critical facets, providing invaluable insights for long-term traffic safety planning, targeted prevention strategies, and a more comprehensive understanding of accident dynamics. This unified system represents a significant research gap, as no such comprehensive approach currently exists in traffic safety management. Similarly, in the field of crime case analysis, the absence of an all-encompassing and user-friendly system that fully utilizes the Decision Tree model for predicting future crime patterns stands out as a conspicuous research gap. Existing research often takes a fragmented approach, focusing on singular aspects of crime analysis, leaving a void in the development of a comprehensive crime analysis system. This research aims to bridge that gap by offering a platform that not only predicts crime patterns but also provides insights into demographics, crime types, patterns involving stolen objects, and long-term crime rate predictions. This holistic perspective on crime patterns is currently lacking, making the proposed system a significant contribution to the field.

Additionally, within the domain of automated text analysis systems for decision support in criminal investigations, there is a notable research gap, especially in the Sri Lankan context. Existing studies often draw from international research, which may not directly apply to the unique challenges faced by the Sri Lankan law enforcement and court system. This knowledge gap impedes the development of a tailored automated text analysis system customized for Sri Lanka, hindering effective decision-making in criminal investigations. Addressing this gap through empirical research on Sri Lanka's specific requirements is essential. Moreover, in the context of preventing violence against women and children, existing studies primarily focus on victim support, analysis, and prediction, with limited attention to preventive measures. The absence of a standardized method for clustering crimes against women compounds the research gap, making it difficult to compare results across studies. The proposed system, encompassing crime prediction, location-based precautions, and data analysis, seeks to fill this gap by offering a comprehensive approach to predicting and preventing crimes against women, providing valuable support for government and law enforcement agencies.

Overall, the research gap lies in the absence of a unified system that combines these diverse components, spanning accident case analysis, crime analysis, case document classification, and crimes against women analysis. The proposed system aims to address this gap comprehensively, offering a novel and integrated approach that has the potential to revolutionize safety management and criminal justice.

1.3 Research Problem

The research problem at hand encompasses multiple domains, each with its unique challenges and gaps. In the field of traffic accident prediction and analysis, there is a pressing issue: the absence of a unified system combining accident percentage prediction for future years, causative factors identification, and in-depth accident case analysis. While existing research has made progress in predicting accident locations and causes, it lacks a holistic approach. Similarly, within the domain of crime case analysis, there is an urgent problem: the lack of an all-encompassing and user-friendly system that fully leverages the Decision Tree model for predicting future crime patterns. Existing research often focuses on fragmented aspects of crime analysis, leaving a void in the development of a comprehensive crime analysis system. Furthermore, in the realm of automated text analysis systems for decision support in criminal investigations, the problem is evident, particularly in the Sri Lankan context, where empirical research addressing the country's specific requirements is lacking. Lastly, the prevention of violence against women and children faces a challenge: the absence of a standardized method for clustering crimes against them, hindering comparisons across studies. Existing research primarily focuses on victim support, analysis, and prediction, with limited emphasis on preventive measures. In essence, the overarching research problem is the need for integrated systems in these diverse domains, spanning accident case analysis, crime analysis, case document classification, and crimes against women analysis, to address critical gaps and revolutionize safety management and criminal justice comprehensively.

1.4 Research Objectives

1.4.1 Main Objective

Our main objective is to develop a comprehensive machine learning system that can prevent and solve crimes by analyzing and predicting accident locations, criminal cases, clustering crimes against women and classifying police case documents. The system aims to enhance public safety by identifying patterns and trends in criminal activity, automating document classification, and providing interactive visualizations to forecast potential future criminal activities.

1.4.2 Specific Objectives

a. A system to predict accident percentage for specified locations, predict accident percentage for causes, and analyze accidents and provide prevention strategies.

b. Analyze criminal cases to predict future crime rates by identifying patterns according to the nature of the crime.

c. Develop an automated text analysis system that classifies crime case files into relevant categories

d. Clustering crimes against women and crime forecasting prediction to take preventive steps

2. METHODOLOGY

2.1 Introduction

In this section, we present the methodological approach employed to address the functions and components of the proposed system. Our research follows a structured methodology rooted in the software lifecycle model, ensuring a systematic and well-organized approach to system implementation. Extensive prior research in the same domain has provided valuable insights and knowledge, which we harness to fulfil the primary and secondary objectives of this study. The wealth of information and findings from previous studies serves as a foundational resource, guiding us in the development and execution of the proposed system.

2.2 System Overview

This section provides an overview of the proposed integrated web application, which has been developed based on the findings from the literature review and extensive research in the field. The objective was to select appropriate technologies, software solutions, and tools for the implementation phase. The resulting system comprises following four components, each designed to address specific aspects of data analysis, prediction, and classification.

- Accident Case Analysis
- Crime Case Analysis
- Case Document Classification
- Crimes against Women Analysis.

Together, these components form a cohesive and comprehensive platform intended for use by police departments and investigators.

The proposed integrated web application combines the power of data analysis, prediction, and classification to provide a comprehensive solution for law enforcement agencies and investigators. By harnessing diverse algorithms and analytical techniques, this system empowers users to gain deeper insights into accident patterns, crime trends, legal documents, and crimes against women. These insights enable more informed decision-making and proactive measures, contributing to improved public safety and law enforcement effectiveness.

In the following sections, we will delve into each component in greater detail, elucidating their functionalities and benefits.

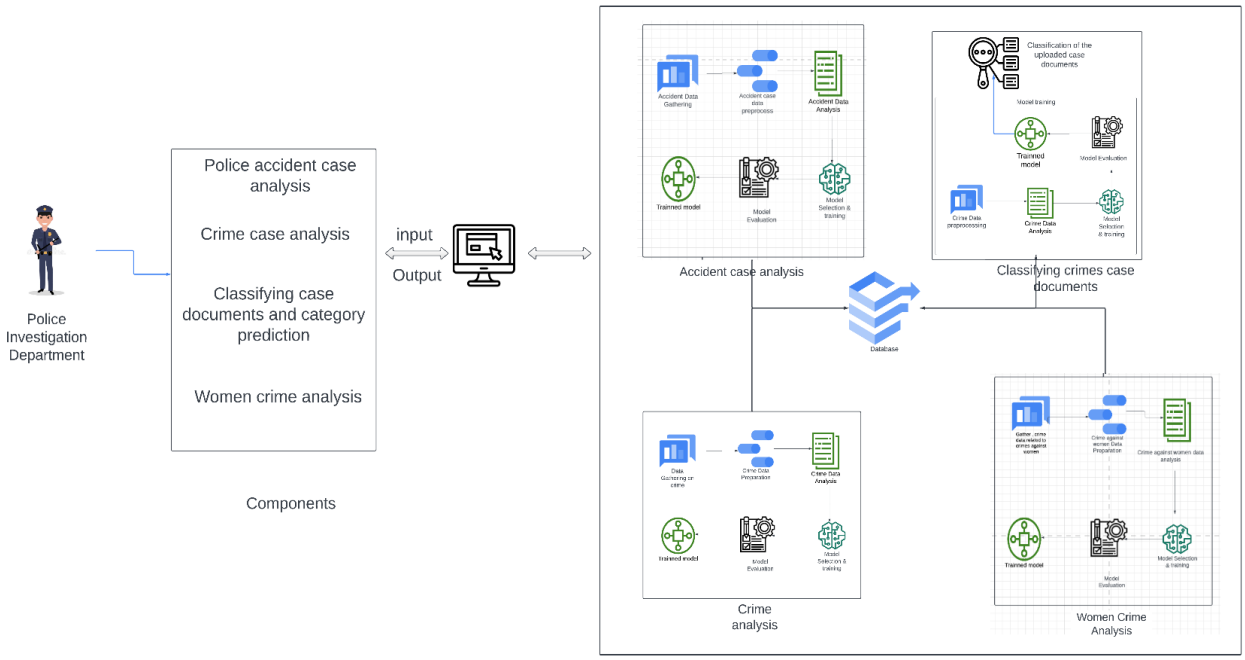


Figure 2.2.1: System high-level architecture diagram

2.3 Component Overview

The integrated system comprises four main components, each serving a purpose, police accident case analysis, crime analysis, case document classification using text analysis, and crimes against women analysis. Each of these components is geared towards predictive and analytical functions, contributing to various aspects of the system's capabilities.

The police accident case analysis component within the integrated system serves as a crucial tool for enhancing road safety planning and optimizing resource allocation. This component comprises multiple stages, each contributing to a comprehensive understanding of accident predictions, patterns and their underlying causes. Initially, the autoregressive integrated moving average (ARIMA) model is employed to predict future accident percentages by analyzing historical data, considering variables such as division name and year. This predictive analysis enables proactive measures for improving road safety. Subsequently, the K-means clustering technique is applied to categorize accidents based on factors such as light conditions, weather, drunken driving, and traffic. This clustering process identifies patterns and similarities within the dataset, allowing the system to categorize accidents according to their root causes. The ARIMA model is then utilized to forecast future accident trends within each cluster, utilizing statistical methods to enhance the accuracy of predictions. Furthermore, the component employs various statistical analysis techniques to uncover meaningful patterns, trends, and relationships within accident cases. In this analysis, we aim to understand the distribution and trends in accident data based on various temporal factors such as day, month, day of the week, and hour of the day. Leveraging this information, the system can derive appropriate prevention strategies tailored to address specific accident causes. This comprehensive approach empowers authorities to implement proactive measures and improve road safety. Users of this component can input historical and current accident data, and the system dynamically predicts accident percentages and fatal percentages for future years, incorporating updated datasets. The outcomes are presented through percentages and graphical representations, enabling law enforcement agencies to make informed decisions, formulate prevention strategies, and take actions aimed at preventing accidents, ultimately contributing to enhanced public safety.

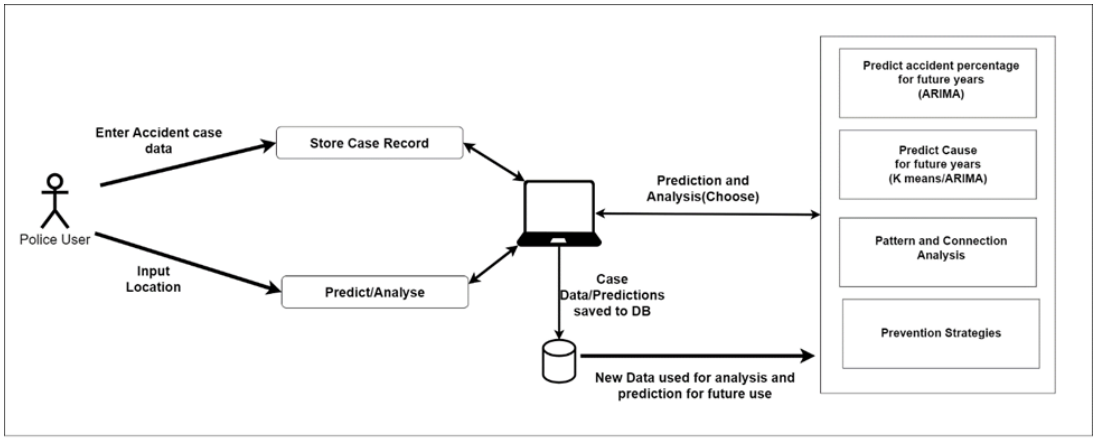


Figure 2.3.1: - High Level Architectural Diagram of police case analysis

The cornerstone of the crime analysis system lies in the gathering and pre-processing 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 pre-processing, 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.

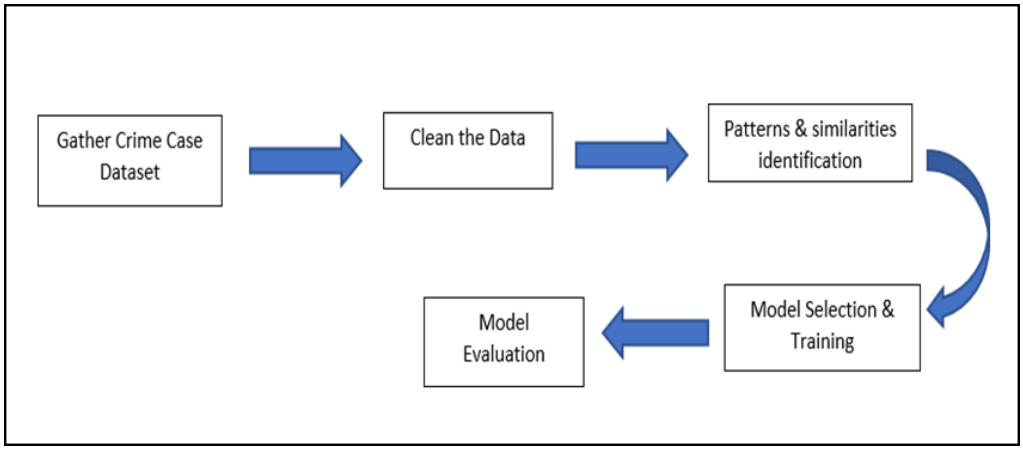


Figure 2.3.2: - High level architecture diagram of Analysing and grouping commonalities among criminal cases and predicting the future crimes in terms of the pattern of the crime.

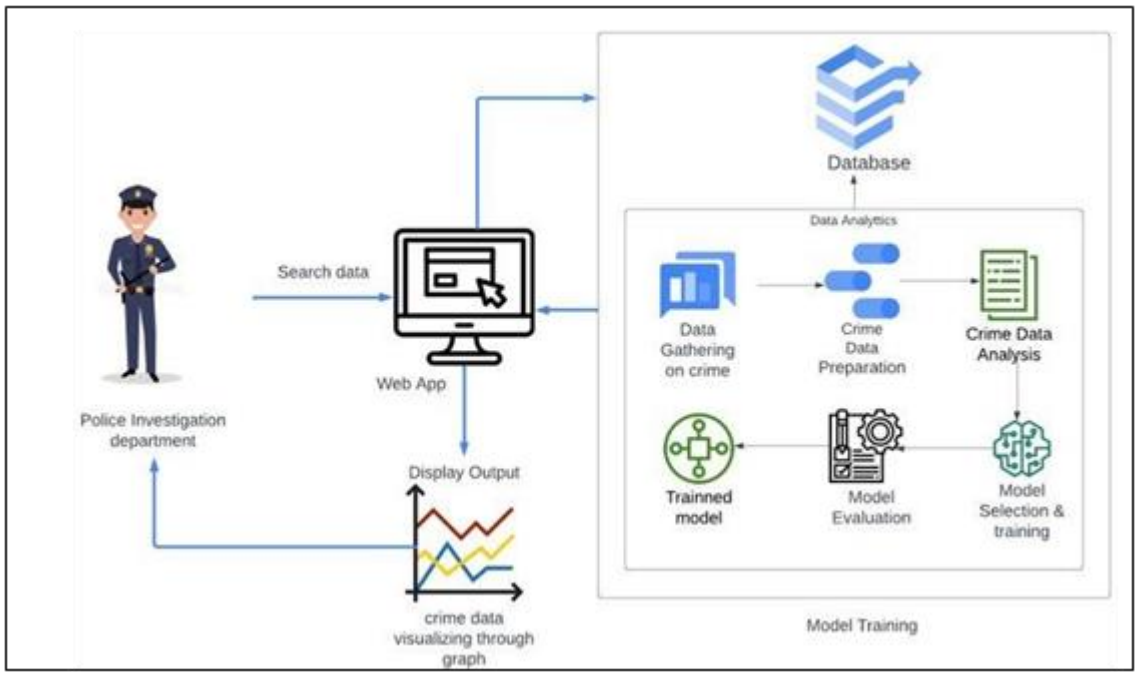


Figure 2.3.2: - - Component overview for Analysing and grouping commonalities among criminal cases and predicting the future crimes in terms of the pattern of the crime.

The case document classification component to develop an automated text analysis system for Sri Lankan law enforcement agencies. It will enhance the quality and accuracy of criminal investigations by replacing labour-intensive manual processes. The process begins with meticulous text extraction from various sources, ensuring precise information capture. Extracted data undergoes manual review, with each document carefully assigned relevant labels based on its content. This labelling process is vital, requiring a deep understanding of document context. It creates a structured training dataset with rows representing individual documents and columns indicating labels, each with a binary value (True or False) indicating label presence. To address multi-label text classification, the research evaluates three algorithms: OneVsRestClassifier with Naive Bayes, LinearSVC, and Logistic Regression. These are crucial for complex legal documents, enabling simultaneous classification of multiple labels. Text data pre-processing techniques like removing punctuation, stop words, and stemming/lemmatization standardize the data. The dataset is split into training and testing sets for unbiased algorithm evaluations. Metrics tailored for multi-label classification, including accuracy, precision, recall, and F1-score, assess algorithm effectiveness in predicting label presence.

The research aims to identify the most suitable algorithm for multi-label text classification in PDF case documents, considering factors like prediction accuracy, handling multiple labels, and computational efficiency. Implementation promises streamlined investigations, rapid identification of similar past cases, improved reliability, and efficiency in document categorization. Efficient label extraction provides investigators with swift access to information, enhancing decision-making and expediting investigations. By eliminating manual reading and labelling, it conserves time and resources, allowing more focus on analysis and interpretation. The system's accuracy and consistency contribute to a robust knowledge base, empowering the investigative team with valuable insights, supporting legal research advancements, and ultimately improving public safety.

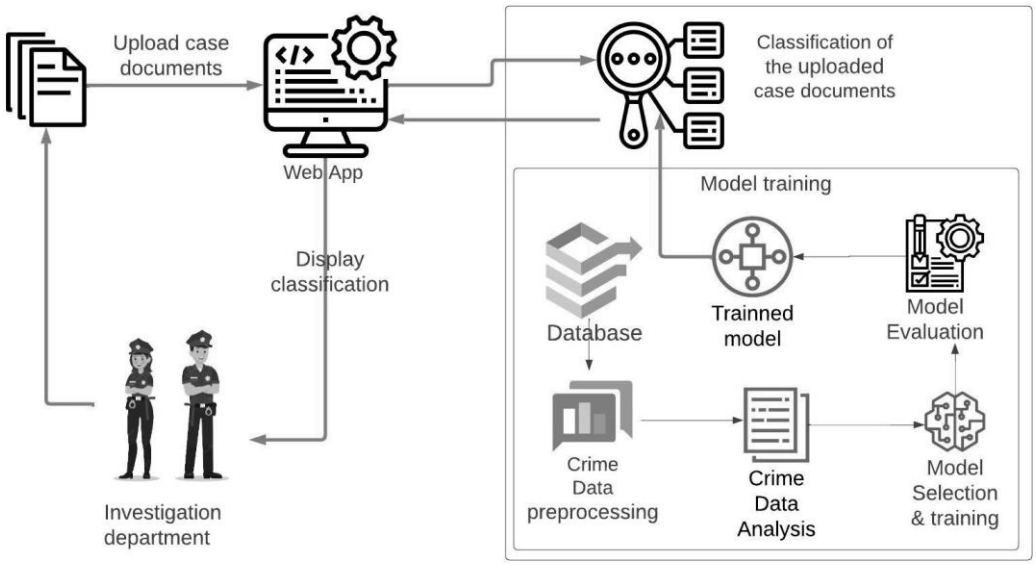


Figure 2.3.3: - Component overview for Analyze and Classify Similar Case Documents and Predict Category.

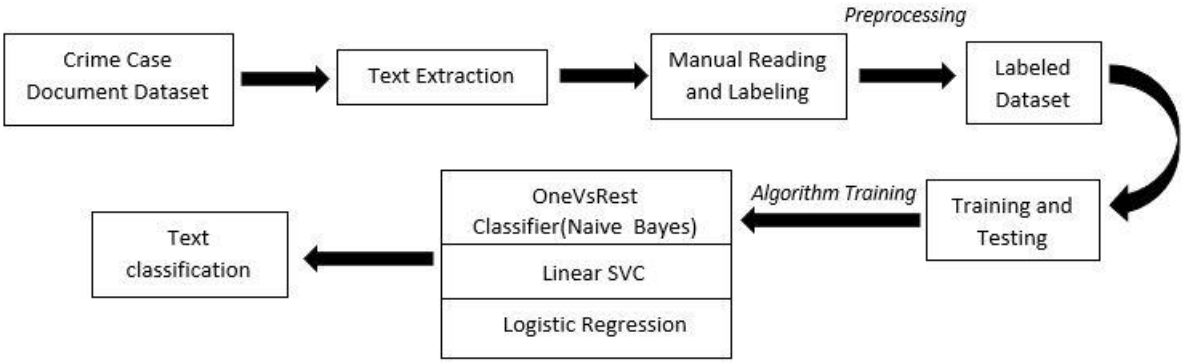


Figure 2.3.4: - High level architecture diagram of Analyze and Classify Similar Case Documents and Predict Category.

Meanwhile, the crimes against women component adopts a comprehensive methodology for the analysis and visualization of women's crime data. The objective of this study is to develop a predictive model to analyse and visualize crimes against women with a focus on predicting future crime rates in various states and federations of Sri Lanka. The methodology included several main steps. It begins with the collection of crime statistics from authoritative sources, followed by thorough data pre-processing to ensure data quality. A Random Forest

Regressor model is employed for predicting future crime counts based on user-defined parameters like state, year, and crime type. Predicted results are then interpreted into categorical crime rate levels, aiding in understanding anticipated crime trends. Random Forest Regressor model was chosen because of its ability to handle complex, non-linear relationships between predictors and crime rates. This model has been carefully trained using historical data to capture patterns and trends. To evaluate its predictive performance, user-friendly interfaces were created that accept user inputs, predict future crime rates based on a selected crime category, and interpret the results into different categories such as "low crime area" or "very high crime area". Data visualization techniques, including bar and pie charts, are utilized to represent crime patterns visually. In addition, they were used to graphically depict crime trends. Finally, the study was extended to provide crime prevention strategies tailored to specific criminal groups. The Random Forest Regression was chosen for its robustness in using multi-attribute datasets, its ability to perform non-linear modelling and its ability to provide reliable predictions of future crime rates, making it a suitable choice for our analytical purposes.

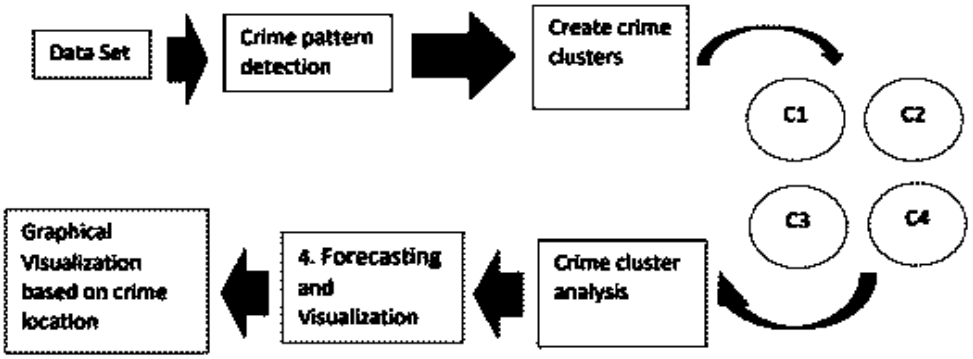


Figure 2.3.5: - High level architecture diagram of clustering crimes against women and crime forecasting

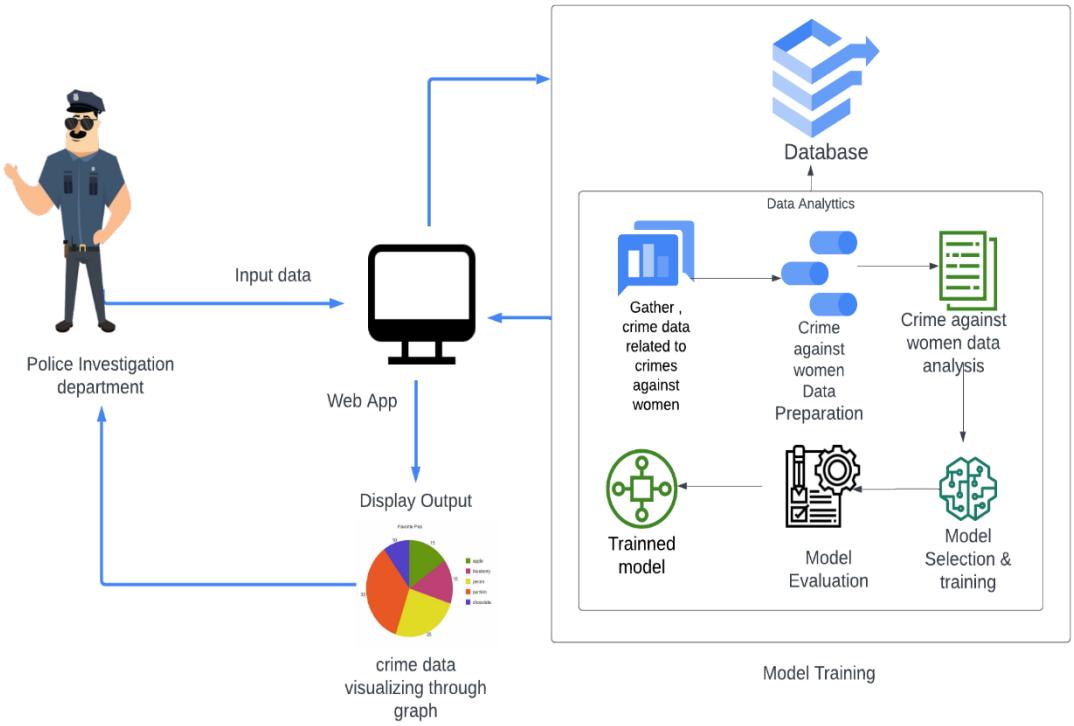


Figure 2.3.6: - Component overview for clustering crimes against women and future crime forecasting prediction.

The proposed integrated web application combines the power of data analysis, prediction, and classification to provide a comprehensive solution for police departments and investigators. By harnessing diverse algorithms and analytical techniques, this system empowers users to gain deeper insights into accident predictions, crime analysis, documents classification, and crimes against women.

2.3.1 Work Breakdown Structure (WBS)

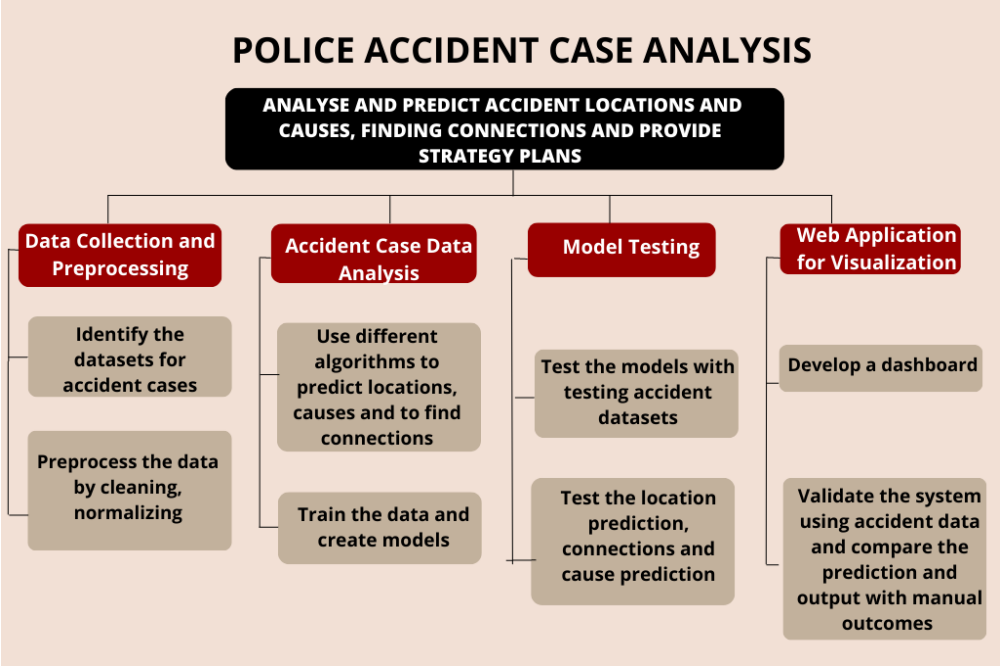


Figure 2.3.1.1: - WBS of accident case analysis

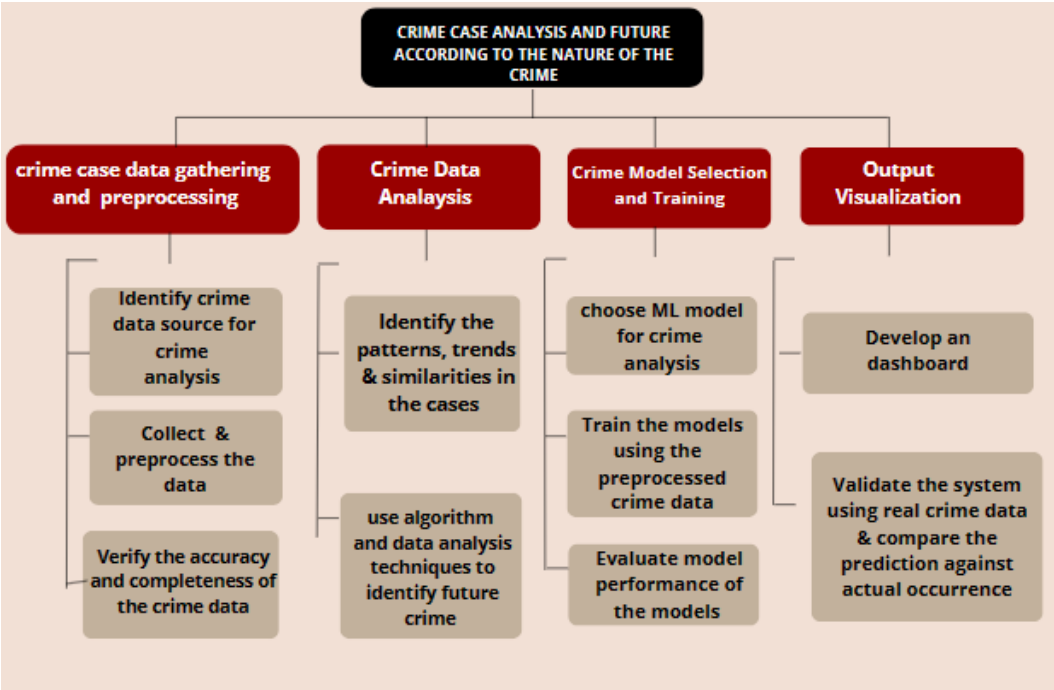


Figure 2.3.1.2: - WBS of crime case analysis

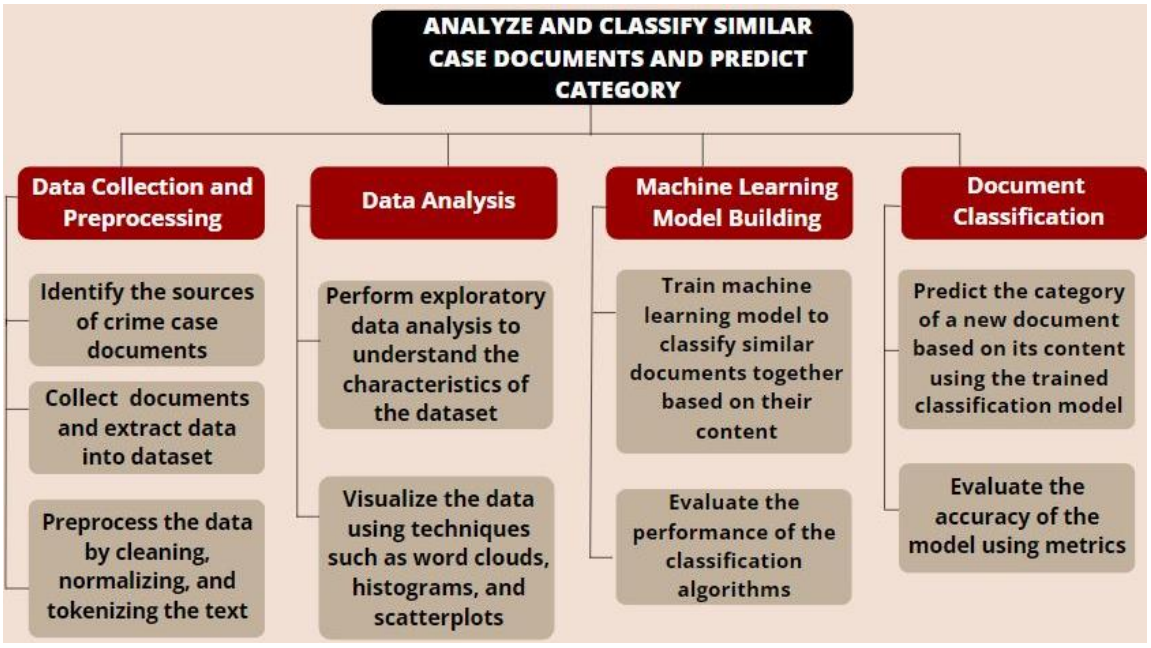


Figure 2.3.1.3: - WBS of document classification analysis

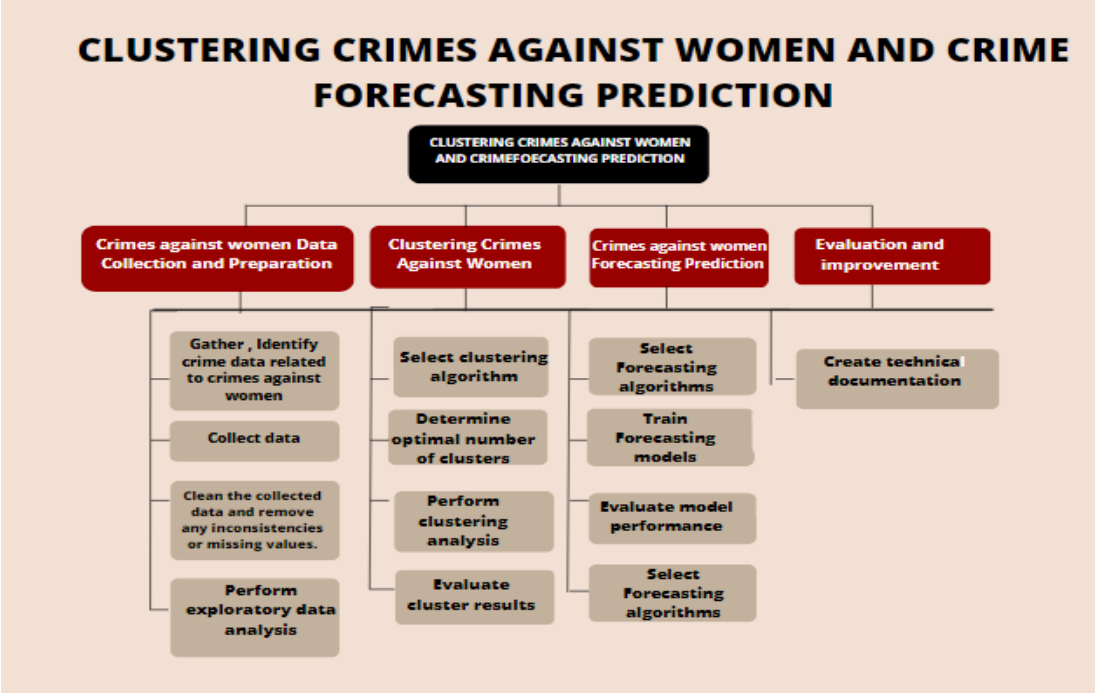


Figure 2.3.1.4: - WBS of crimes against women analysis

2.4 Development Process

The development process chosen for our system follows a Waterfall model. This model is selected because it offers a linear and systematic approach, making it well-suited for our project's needs and clearly defined specifications. The Waterfall model involves distinct stages, each with tasks that can be scheduled and completed within specific timeframes. These stages and the associated strategy proceed sequentially without overlapping. Following the guidelines in this section, we will address and investigate the issues raised, provide a figurative characterization of the system's functions aimed at resolving these issues, examine the expected outcomes, and emphasize the importance of effective time management, which has been crucial throughout our year-long research endeavour. The system's requirements have been systematically divided into functional phases, as illustrated in Figure 2.5.1, which include Requirement Analysis, Design, Development, Testing, and Maintenance.

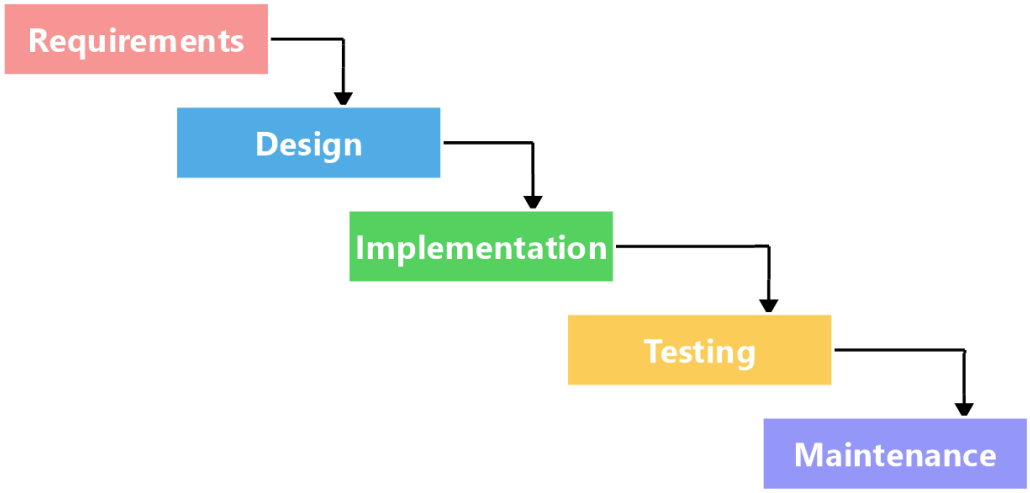


***Figure 2.5.1**:* Development process of the system

2.4.1 Project Management

Project Management in the context of our Police Case Analysis project involves a distinctive approach due to the unique software development life cycle. Unlike traditional project management, software development entails multiple iterations, encompassing testing, updates, and continuous user feedback. To align with the dynamic nature of our project and adapt to evolving requirements from law enforcement agencies and stakeholders, we have embraced an agile methodology. Within our project teams, we have used a collaborative tool such as Microsoft teams for scheduling meetings and formulating project plans. These teams are responsible for both group and individual tasks, ensuring that every aspect of the project is systematically addressed and executed. The visual representations below provide insights into our structured approach to project management and task allocation

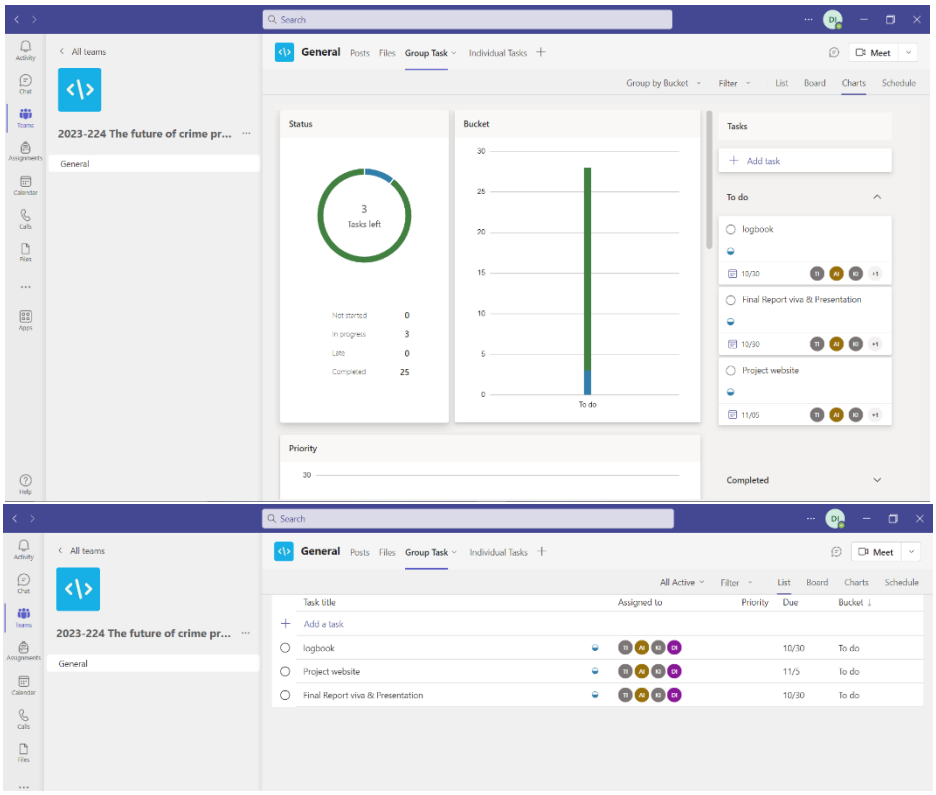


Figure 2.4.1.1: Project management of the system through MS teams

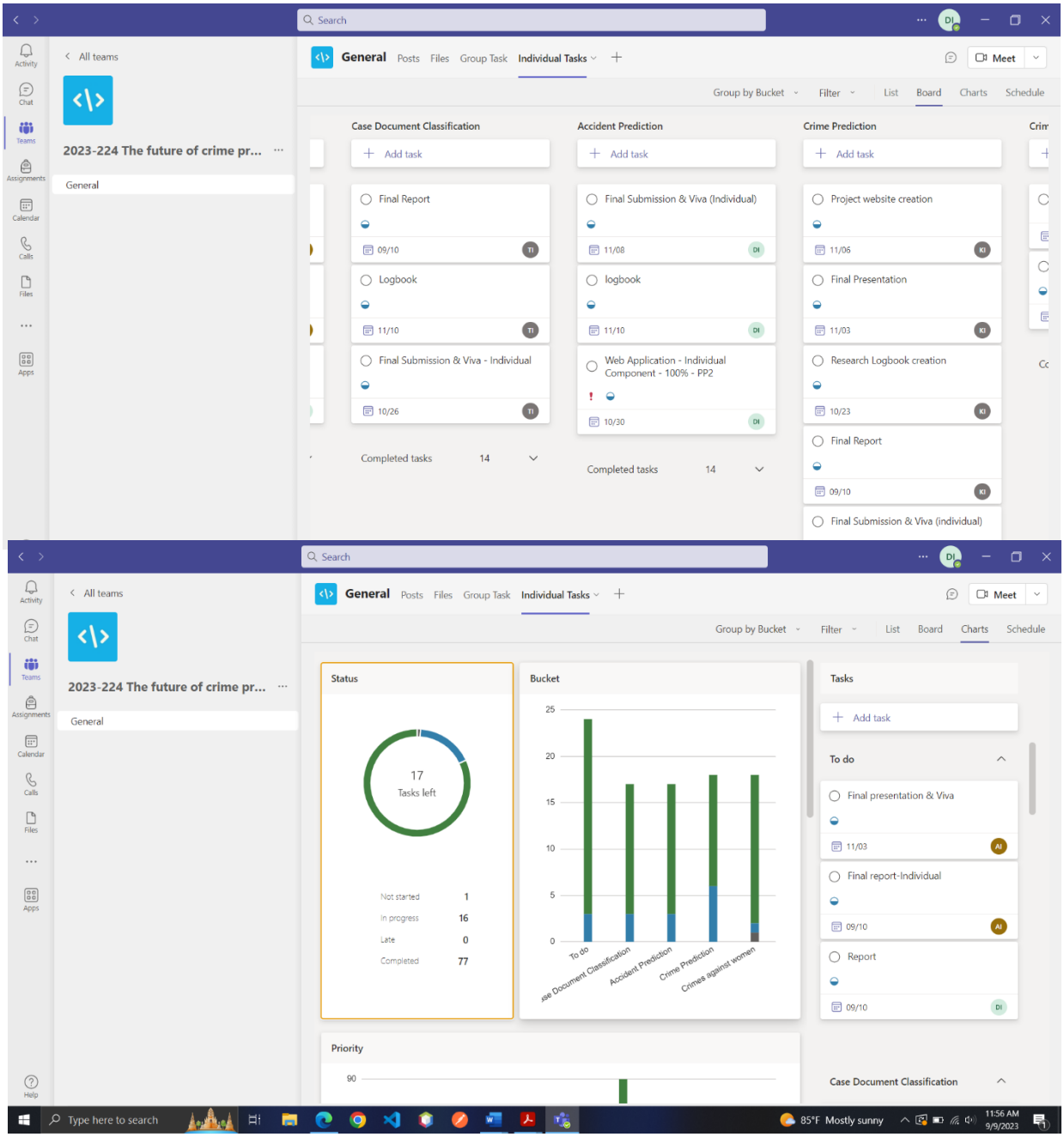


Figure 2.4.1.2: Project management of the system through MS teams (Individual)

2.4.1.1 Project Code Management

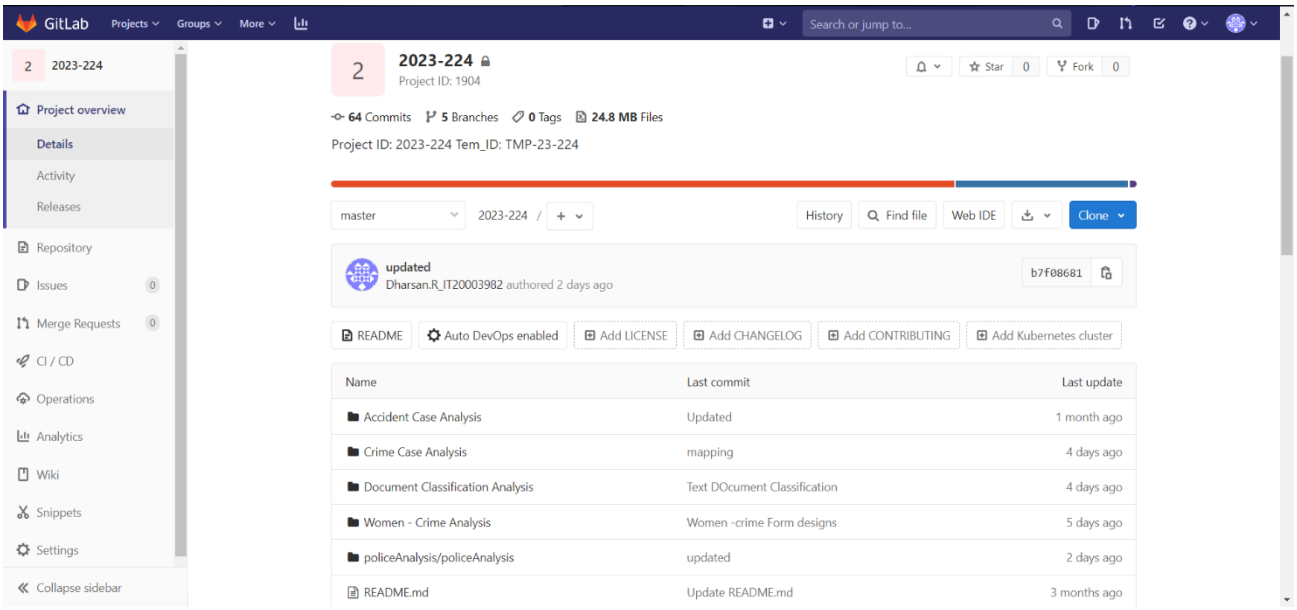


Figure 2.4.1.1.1: Code management in gitlab

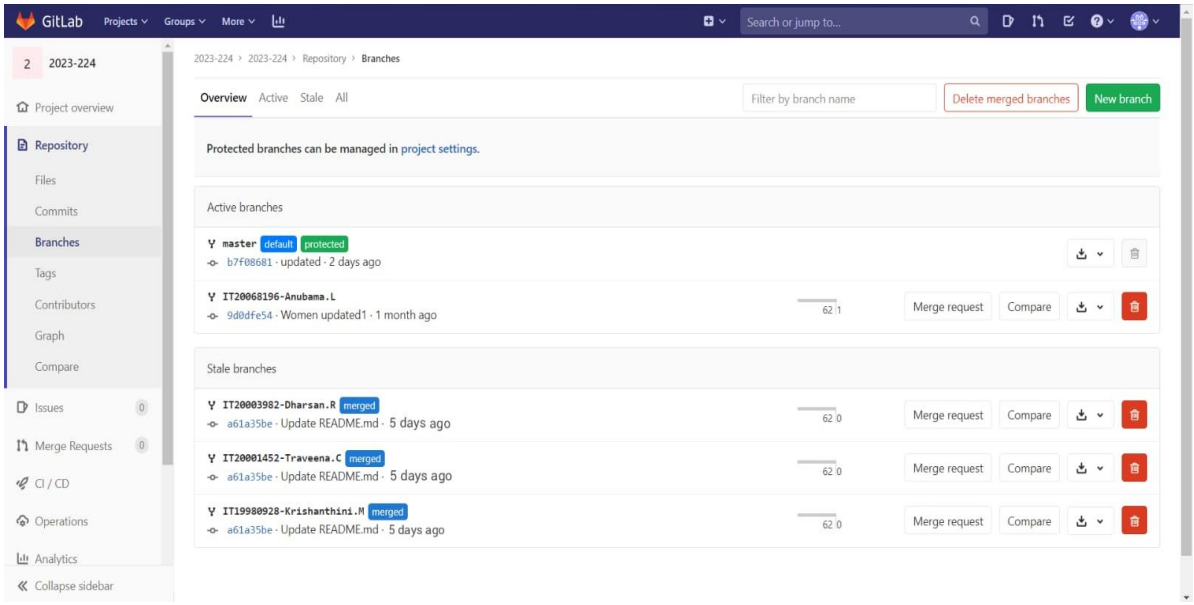


Figure 2.4.1.1.2: Merge branches

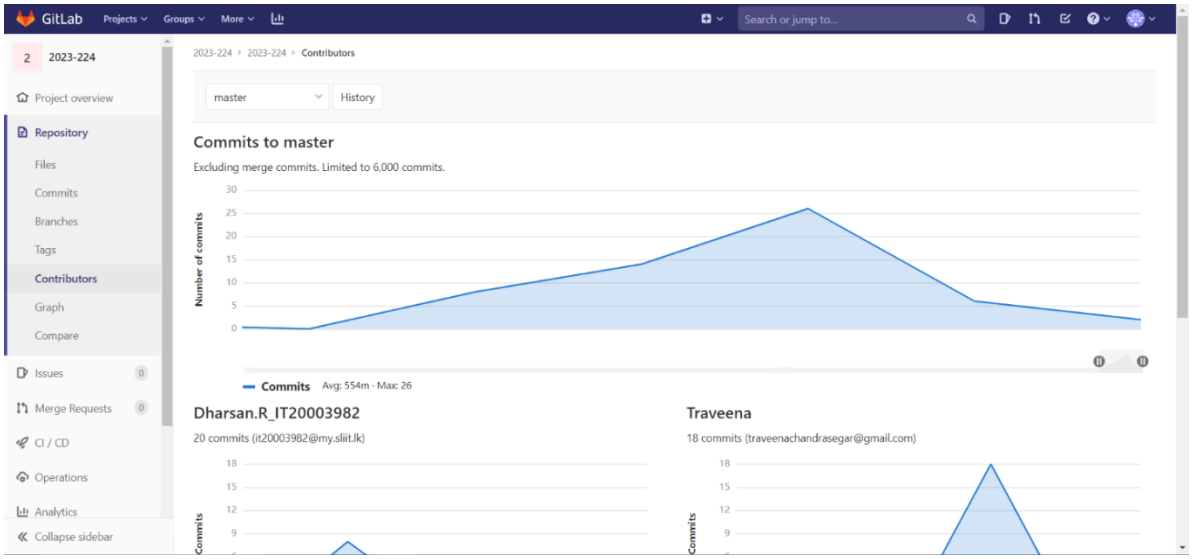


Figure 2.4.1.1.3: Overall commits

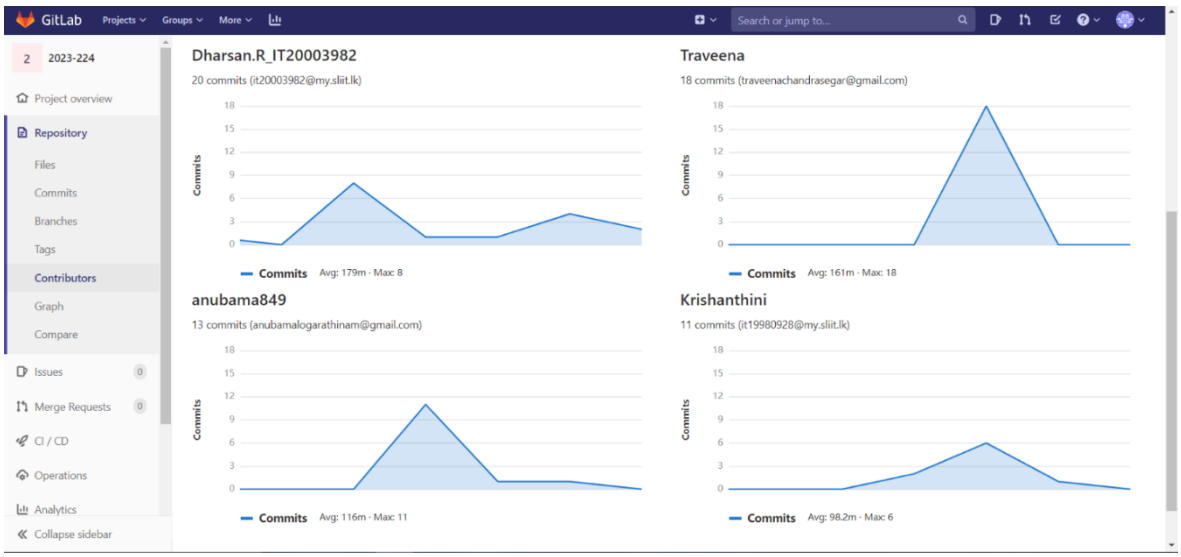


Figure 2.4.1.1.4: Individual commits

2.5 Requirement Gathering

The primary aim of this system is to serve the requirements of the police force, without any commercial objectives. In our pursuit of gathering the necessary data for each system component, we engaged in collaborative efforts with key stakeholders. We initiated discussions with the Deputy Inspector General (DIG) in Kurunegala, resulting in the acquisition of essential datasets. Additionally, we sought the expertise of legal professionals and senior lecturers from the Faculty of Law in Colombo, who graciously contributed valuable data. The datasets we meticulously collected encompassed accident and crime-related information, as well as case documents. This comprehensive dataset compilation exceeded five thousand records for each component. Importantly, the data we gathered strictly pertains to case details and does not involve any personally identifiable information, ensuring the utmost privacy and security.

This extensive dataset forms the cornerstone of our machine learning algorithms, enabling us to derive accurate predictions and conduct in-depth analyses. Users of our system will have the opportunity to visualize these outcomes through graphical representations. It's worth noting that the process of dataset acquisition presented significant challenges, necessitating a year-long effort and close collaboration with law enforcement authorities, which ultimately culminated in the successful completion of our project.

Our requirements encompass both functional and non-functional aspects, which we will elaborate on in the sections below.

2.5.1 Functional Requirements:

- Functional requirements for accident case analysis include predicting accident percentages for future years and divisions, forecasting accident percentages and causes for future years, conducting accident pattern analysis to identify high-risk days, hours, and months, and offering prevention strategies based on analysis results.
- Functional requirements for crime analysis entail an intuitive user interface for querying crime data based on user-provided parameters. The system must implement a Decision Tree machine learning model to predict prevalent crime

patterns, affected demographics, vehicles involved, and stolen objects. Additionally, predictive models for estimating crime rates over the next five years and visualization tools for generating graphical representations of location-specific crimes are essential features.

- The system should collect crime case documents, pre-process text data, and employ NLP and machine learning techniques to identify patterns and relationships in the documents. It must use topic modelling to uncover trends and employ supervised machine learning and text classification for categorization and prediction of new cases, ensuring adaptability for improved accuracy.
- The system must detect patterns and trends in crimes against women, employing clustering based on factors such as location, crime type, and time. It should provide insights and recommendations for prevention and forecast future crimes using historical data.

2.5.2 Non-functional Requirements:

- Performance
- Reliability
- Security
- Usability
- Scalability

2.6 Resources Used

2.6.1.1 Software Boundaries

Backend - Python Language

Backend - Python Language: Python serves as the backbone of the application. It's used to write the server-side logic for handling HTTP requests and responses. In this case, Python is used for processing user inputs, making predictions based on machine learning models, and handling data manipulation tasks. Libraries like Pandas, NumPy, Scikit-Learn, and Django are employed to manage data, perform calculations, and implement machine learning models.

Visual Studio Code Editor

Visual Studio Code (VS Code) is the integrated development environment (IDE) where Python scripts are written and developed. It provides features like code autocompletion, debugging tools, and extensions for Python, making it a powerful tool for writing and managing Python code efficiently.

Frontend – HTML, CSS, JavaScript

HTML, CSS, JavaScript: The frontend of the web application is built using HTML, CSS, and JavaScript. HTML is used for creating the structure and content of web pages. CSS is used for styling the user interface, making it visually appealing. JavaScript is employed for adding interactivity to the web application, such as handling user inputs and triggering requests to the backend.

Framework – Django

Within the realm of software boundaries, we have harnessed Django, a high-level Python web framework renowned for its ability to expedite the development of secure and robust websites. Crafted by seasoned developers, Django alleviates many of the complexities associated with web development, enabling us to focus on building our application without the need to reinvent the wheel. It's important to note that all four components seamlessly operate within the Django framework, facilitating cohesion and consistency across the system.

Web Application

The web application acts as the user interface through which users interact with the backend component. Users can input data, select options, and trigger predictions through web forms. The web application communicates with the Python backend by sending HTTP requests with user inputs and receives responses containing prediction results.

In summary, Python and VS Code handle the backend logic and machine learning, while HTML, CSS, and JavaScript are used for creating an interactive and visually pleasing frontend. Together, these technologies enable the development of a user-friendly web application for crime prediction and prevention.

Libraries Used

1. **Pandas** - Pandas is a versatile library that provides data structures like DataFrames and Series. It's essential for data manipulation tasks such as data cleaning, transformation, and analysis. It simplifies working with structured data and supports various data sources.

2. **Matplotlib and Seaborn -** Matplotlib is a widely-used library for creating static, animated, or interactive visualizations in Python. Seaborn is built on top of Matplotlib and specializes in creating informative and attractive statistical graphics. Together, they facilitate data exploration and presentation.

3. **Scikit-Learn (sklearn) -** Scikit-Learn is a comprehensive machine learning library offering tools for classification, regression, clustering, and more. The imported classifiers, such as LinearSVC and MultinomialNB, are used for building machine learning models. It also includes modules for feature extraction, selection, and model evaluation.

4. **Django** - Django is a robust web framework that simplifies web application development. It follows the Model-View-Controller (MVC) architectural pattern and provides built-in tools for handling HTTP requests, managing databases, and rendering views. Django is particularly well-suited for building data-driven web applications.

5. **Numpy -** Numpy is the fundamental library for numerical computing in Python. It offers efficient array operations and mathematical functions, making it indispensable for scientific and mathematical applications.

6. **PyPDF2** - PyPDF2 is used to extract text and perform operations on PDF documents. It's useful for parsing and extracting data from PDF files, which can be valuable in various applications, including data analysis and document processing.

7. **NLTK (Natural Language Toolkit) -** NLTK is a comprehensive library for natural language processing (NLP) tasks. It includes tokenizers, stemmers, and access to linguistic resources, making it a valuable resource for text analysis and processing**.**

8. **Neattext -** Neattext simplifies text pre-processing and cleaning tasks. It provides functions to remove unwanted characters, normalize text, and perform various cleaning operations, enhancing the quality of text data for analysis.

9. **String -** The `string` library provides constants and utilities for working with strings, including character sets like ASCII letters and punctuation. It's handy for text manipulation tasks.

10. **Tempfile and Os -** Tempfile and Os are essential for managing temporary files and handling file paths. They ensure efficient and safe file operations within the application.

11. **Urllib and Base64 -** Urllib is used for making HTTP requests and handling URLs. Base64 encodes and decodes binary data, which can be valuable for encoding and decoding data for transmission or storage.
12. **IO -** The `io` module provides classes and functions for handling input and output operations. It's often used for working with streams and data buffers.

13. **Pickel -** The `pickle` library is used for serializing (pickling) and deserializing (unpickling) Python objects. It's valuable for storing and retrieving complex data structures.

14. **Base64 (import base64) -** Base64 is used to encode binary image data into a format that can be embedded directly into HTML and displayed as images in the web application.

15. **CSV (import csv) -** The CSV module is used for reading and writing CSV (Comma-Separated Values) files. It's used to add new data entries to the dataset.

16. **Random (import random) -** The Random module is used for generating random values. In this project, it's used to generate random colours for data visualization.

17. **Plotly Express (import plotly.express as px) -** Plotly Express is a library for interactive data visualization. It's used for creating interactive charts and graphs, although it's not used extensively in this code.

18. **statsmodels.tsa. arima.model.ARIMA:** This is a part of the stats models library and is specifically used for fitting ARIMA (AutoRegressive Integrated Moving Average) models. ARIMA is employed for time series forecasting and is crucial for predicting future accident percentages.

19. **sklearn.cluster.KMeans:** This module is part of scikit-learn, a machine learning library in Python. It's used for K-means clustering, which groups data points into clusters based on similarity.

2.7 Commercialization aspects of the product

The proposed system is intended for a service purpose, with the goal of leveraging machine learning techniques to address important societal issues related to public safety and crime prevention. By providing accurate and reliable analysis, prediction, and classification of accident and crime-related data, the system has the potential to help law enforcement agencies, policy makers, and other stakeholders make informed decisions, allocate resources effectively, and take proactive measures to prevent and reduce crime. Furthermore, the system's ability to identify patterns and commonalities across cases can facilitate the development of targeted intervention strategies and support efforts to improve public safety in a more holistic and comprehensive manner.

2.8 Testing and Implementation

2.8.1 Testing

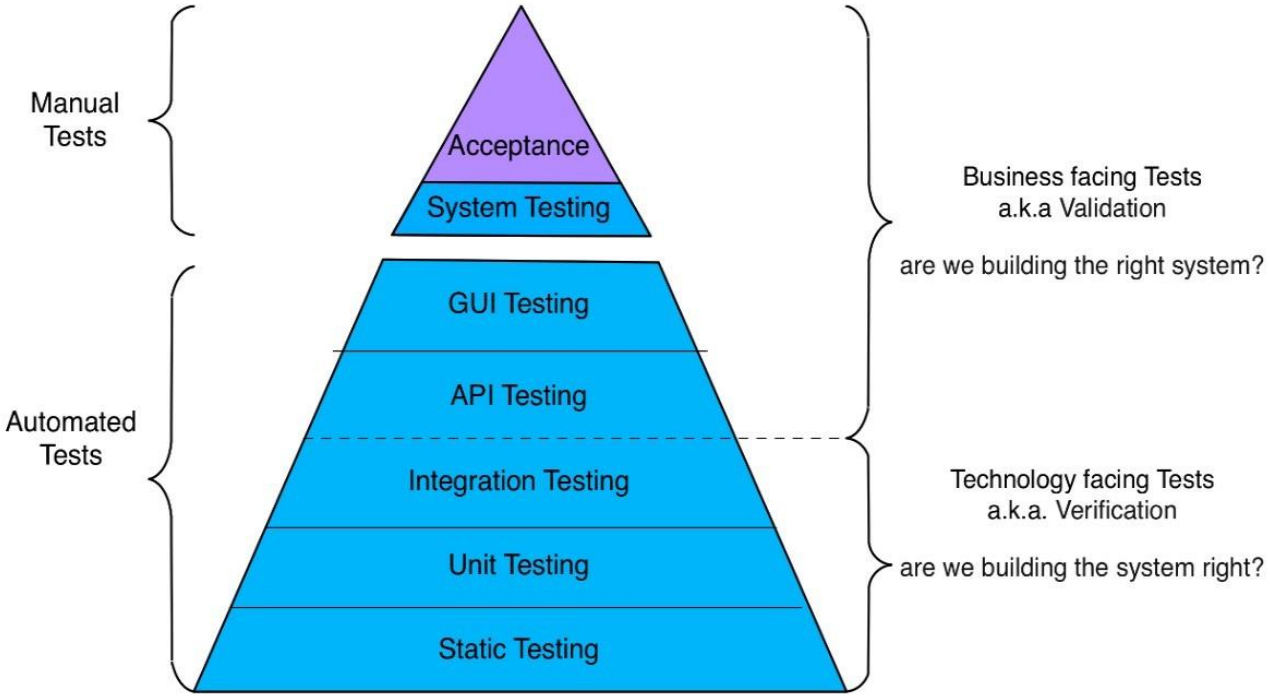


Figure 2.8.1.1: Testing Phase.

The diagram above illustrates a systematic approach to software testing at each developmental stage, encompassing applications and defect resolution. This comprehensive testing methodology consistently enhances the quality of applications, ensuring they are top-notch deliverables. Testing is a fundamental component of the software development life cycle, serving to unearth and rectify issues that may have silently accumulated during various development phases. Furthermore, it is imperative to validate the quality of the product or software application. Regardless of the specific component in our SDLC, which includes Accident Case Analysis, Crime Analysis, Case Document Classification, and Crimes Against Women Analysis, each stage must undergo thorough testing to guarantee its functionality and reliability.

1. **Unit Testing**

Unit testing is a crucial phase in our project's development, ensuring the functionality and accuracy of each component in isolation. Starting with the Accident Case Analysis component, unit tests meticulously verify data pre-processing, ARIMA model predictions, and the effectiveness of visualization tools, guaranteeing precise accident percentage and pattern forecasts. Moving to the Crime Analysis component, unit tests assess the accuracy of data processing, Decision Tree model predictions, and visualization tools, validating precise crime pattern and prediction outcomes. In the Case Document Classification component, the testing process rigorously evaluates the system's ability to accurately categorize case documents based on various factors like content, metadata, and context. Lastly, in the Crimes Against Women Analysis component, unit tests ensure that clustering algorithms accurately group similar cases and predictive models provide precise forecasts using historical data. These tests also validate the correctness of data processing and transformation functions, critical for accurate analysis in all components.

2. **Module Testing**

In module testing, each of the four key components of our system, namely Accident Case Analysis, Crime Analysis, Case Document Classification, and Crimes Against Women Analysis, undergoes individual evaluation. This phase verifies that each component functions correctly in isolation, addressing specific functionalities and predictions. By examining and validating the accuracy of these individual modules, we ensure their reliability and effectiveness within the larger system Integration Testing.

3. **Integration Testing**

Integration testing is the subsequent phase, where the interactions between these components are rigorously tested. It assesses how well these modules work together, ensuring that data flows seamlessly between them and that they collectively produce accurate and coherent outcomes. This step helps identify and resolve any integration-related issues, guaranteeing the smooth operation of the integrated web application.

4. **System Testing**

System testing encompasses a comprehensive evaluation of the entire integrated web application. It verifies that the system functions as a cohesive whole, accurately predicting accident percentages, crime patterns, and document classifications. This phase validates the system's ability to provide valuable insights and user-friendly visualizations. System testing is essential to ensure the reliability and robustness of the entire platform.

5. **User Acceptance Testing**

User Acceptance Testing (UAT) is a critical phase where the system is put to the test by actual users, such as police departments and investigators. During UAT, users interact with the system, input parameters, and assess its performance, usability, and the accuracy of predictions. Feedback and observations from users play a pivotal role in refining and fine-tuning the system to meet their needs effectively.

6. **Maintenance**

The maintenance phase represents an ongoing commitment to keeping the system up to date and ensuring its long-term reliability. This involves addressing any issues that arise during real-world usage, updating datasets, and incorporating user feedback for continuous

improvement. Maintenance ensures that the system remains a valuable and dependable tool for law enforcement agencies and investigators over time.

Our integrated system, consisting of the Accident Case Analysis, Crime Analysis, Case Document Classification, and Crimes Against Women Analysis components, has successfully passed through all testing phases without encountering errors. During the testing process, we divided the entire system into its constituent components, allowing for a realistic and thorough examination of each part. This approach ensured that our system was rigorously tested, guaranteeing its reliability and accuracy in delivering valuable insights and predictions to law enforcement agencies and investigators.

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

Test Case No	Test Case 01
Description	Predict Accident Percentage for specified division name and future year
Test Steps	1. Login to the system. 2. Navigate to accident 3. Navigate to accident prediction section 4. Type or select the division name 5. Select the future year 6. Click predict button
Test Data	String
Expected Result	Successfully predicting the accident percentage and fatal percentage for future years
Actual Result	Pass
User Role	Police

Table 2.8.1.1 Predict Accident Percentage Test Case 01

Test Case No	Test Case 02
Description	Predict Accident and fatal percentage for future years
Test Steps	1. Login to the system. 2. Navigate to accident 3. Navigate to accident prediction section 4. Enter or select division name 5. Click analysis button
Test Data	String
Expected Result	Predicted percentages for accident and fatal for future years displayed and graph also displayed.
Actual Result	Pass
User Role	Police

Table 2.8.1.2 Predict accident and fatal percentage for future years - Test Case 02

Test Case No	Test Case 03
Description	Predict cause of the accident and the percentage for each cause
Test Steps	1. Login to the system. 2. Navigate to accident 3. Navigate to cause prediction section 4. Enter or select future year 5. Click submit
Test Data	String

Expected Result	Predicted percentage for each cause should be displayed and the graph for those values should be displayed
Actual Result	Pass
User Role	Police

Table 2.8.1.3 Cause Prediction - Test Case 03

Test Case No	Test Case 04
Description	Provide prevention strategies
Test Steps	1. Login to the system. 2. Navigate to accident 3. Navigate to cause prediction section 4. Click prevention strategy button 5. Select cause 6. Click submit
Test Data	String
Expected Result	All prevention strategies displayed
Actual Result	Pass
User Role	Police

Table 2.8.1.4 Prevention Strategies Test Case 04

Test Case No	Test Case 05
Description	Select the type of analysis in the pattern analysis section

Test Steps	1. Login to the system. 2. Navigate to accident. 3. Navigate to pattern analysis section. 4. Select one option from (Day, Hour, Month).
Test Data	String
Expected Result	Predicted and statistical graph for the selected option and the most accident occurring day/hour/month should be displayed
Actual Result	Pass
User Role	Police

Table 2.8.1.5 Pattern Analysis Test Case 05

Test Case No	Test Case 06
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 2.8.1.6 Ensure the form is not submitted with missing data - Test Case 06

Test Case No	Test Case 07
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 2.8.1.7 Form Submission with Invalid Data - Test Case 07

Test Case No	Test Case 08
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 2.8.1.8 Display of Prediction Results - Test Case 08

Test Case No	Test Case 09
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 2.8.1.9 Functionality of Prevention Strategies Test Case 09

Test Case No	Test Case 10
Description	Verify whether error message displayed when user does not select any option in a checkbox field of a form.
Test Steps	5. Login to the system.

	6. Navigate to the "Crime Classifier" option. 7. Click on the 'ADD DATA' option. 8. Not select any option in the 'Categories' feild 1. Submit the form by clicking on the 'Upload' button.
Test Data	String
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	Investigation Team.

Table 2.8.1.10 Ensure the form is not submitted with missing data Manual Test Case 10

Test Case No	Test Case 11
Description	Verify whether message displayed after re-train model is successful
Test Steps	1. Login to the system. 2. Navigate to the "Crime Classifier" option. 3. Click on the 'ADD DATA' option.

	4. Submit the form by clicking on the 'Upload' button. 5. Click on the 're-train Model' button.
Test Data	String
Expected Result	The form should be submitted. Alert message should be displayed and accepted after the re-training of model
Actual Result	Pass
User Role	Investigation Team.

Table 2.8.1.11 Ensure success message is displayed after re-training Manual Test Case 11

Test Case No	Test Case 12
Description	Verify whether after a successful form submission, the prediction results are displayed correctly.
Test Steps	1. Login to the system. 2. Navigate to the 'Crime Classifier' option. 3. Select Pdf document that need to predict category. 4. Click 'Upload' button. 5. Click on the 'Predict Category' button.
Test Data	String, Integer

Expected Result	The predicted category for the uploaded document should be displayed.
Actual Result	Pass
User Role	Investigation Team.

Table 2.8.1.12 Display Category Prediction - Test Case 12

Test Case No	Test Case 13
Description	Verify that you are redirected to the correct page, with data that is newly added to the through 'ADD DATA' option.
Test Steps	1. Login to the system. 2. Navigate to the "Crime Classifier" option. 3. Click on the 'ADD DATA' option. 4. Submit the form by clicking on the 'Upload' button.
Test Data	String
Expected Result	Should redirect to the 'Preview Data' page. Newly added data should be there as the last record of the data table
Actual Result	Pass
User Role	Investigation Team.

Table 2.8.1.13 Functionality Add data Page - Test Case 13

Test Case No	Test Case 14
Description	Verify whether empty validation message is fired for each empty mandatory field.
Test Steps	1. Login to the system. 2. Navigate to the ‘Women-crime’ option. 3. Click on the ‘Add data’ option. 4. Submit the form by clicking on the ‘Upload’ 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 2.8.1.14 Ensure the form is not submitted with missing data Manual Test Case 14

Test Case No	Test Case 15
Description	Verify whether error message is fired when user enters invalid data in one or more fields (e.g., text in the "Year" field).
Test Steps	1. Login to the system. 2. Navigate to the ‘Women-crime’ option.

	3. Click on the ‘Add data’ option. 4. Enter text in ‘Year’ field or crime fields(e.g, Rape, Kidnapping and Abduction etc) 5. Submit the form by clicking on the ‘Upload’ 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 2.8.1.15 Form Submission with Invalid Data Manual Test Case 15

Test Case No	Test Case 16
Description	Verify whether after a successful form submission, the prediction results are displayed correctly.
Test Steps	1. Login to the system. 2. Navigate to the ‘Women-crime’ option. 3. Click on the ‘Highest Crime Prediction’ option. 4. Select ‘Division’, ‘Year’ options from the dropdown. 5. Click on the ‘Predict’ button.

Test Data	String, Integer
Expected Result	The predicted counts for each crime type should be shown. Highest predicted crime and its count should be displayed.
Actual Result	Pass
User Role	Police team.

Table 2.8.1.16 Display of Prediction Results – Test Case 16

Test Case No	Test Case 17
Description	Verify that you are redirected to the correct page, possibly with information related to preventing the highest predicted crime.
Test Steps	1. Login to the system. 2. Navigate to the 'Women-crime' option. 3. Click on the 'Highest Crime Prediction' option. 4. Select 'Division', 'Year' options from the dropdown. 5. Click on the 'Predict' button. 6. Predicted counts for each crime type are shown
Test Data	String
Expected Result	'Prevention Strategies' option should be displayed.

	User should be redirected to the correct page, possibly with information related to preventing the highest predicted crime.
Actual Result	Pass
User Role	Police team.

Table 2.8.1.17 Functionality of Prevention Strategies Link Manual Test Case 17

2.8.2 Implementation

In the implementation phase, we leveraged the wealth of data collected during the project's research and development stages. The extensive datasets gathered for each component, including accident and crime-related data, case documents, and demographic information, served as the foundation for building our integrated web application. Utilizing the Django framework with Python, we seamlessly translated our research findings into a functional and user-friendly system. This implementation not only enables us to predict accident percentages for future years and analyze crime patterns but also offers robust data classification and analysis capabilities. By harnessing machine learning algorithms, statistical models, and data visualization tools, we have created a powerful platform that empowers law enforcement agencies and investigators with the insights needed to enhance safety planning, crime prevention, and decision-making. Our implementation process ensures that our system is well-equipped to address the complex challenges in the fields of accident case analysis, crime analysis, case document classification, and crimes against women analysis, ultimately contributing to improved public safety and law enforcement effectiveness.

Code

Accident case analysis implementation

```
from django.shortcuts import render, redirect
from .accident_Prediction_RP_FINAL import perform_prediction
from .forms import YearForm,ClusterNameForm
import matplotlib.pyplot as plt
from io import BytesIO
import base64
import pandas as pd
import matplotlib
import io
from sklearn.cluster import KMeans
from statsmodels.tsa.arima.model import ARIMA
from django.http import HttpResponse, HttpResponseNotFound
import csv
```

Figure 2.8.2.1: Import libraries for accident prediction

The code snippets imported in our component play a pivotal role in enabling the functionality of our integrated web application. We utilize Django for web framework support, allowing seamless user interactions. The 'accident_Prediction_RP_FINAL' module integrates predictive analysis using the ARIMA model to forecast accident percentages and fatal percentages for future years and divisions. We employ forms, such as 'YearForm' and 'ClusterNameForm,' to gather user inputs for customized queries. Data visualization is facilitated by 'matplotlib' to generate graphical representations for user insights. Data manipulation and analysis are performed with 'pandas,' while 'io' aids in data handling. The 'KMeans' module from 'scikit-learn' is used for K-means clustering, and 'statsmodels' is employed for time series analysis using ARIMA. The code ensures a seamless user experience and accurate predictions, enhancing the overall functionality of our system.

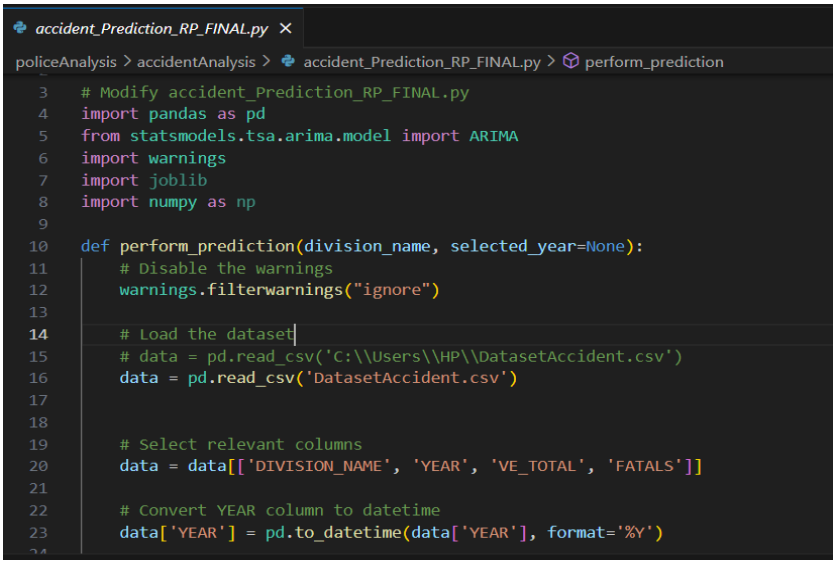


Figure 2.8.2.2: Read dataset and use necessary columns

The provided code excerpt is a Python script named 'accident_Prediction_RP_FINAL.py,' which is essential for the accident case analysis component of our system. This script performs predictive analysis using the ARIMA (AutoRegressive Integrated Moving Average) model to forecast accident percentages and fatal percentages for specific divisions and years. It first disables warnings for smoother execution. The script loads the accident dataset, selects relevant columns containing information about division names, years, total accidents (VE_TOTAL), and fatalities (FATALS). It then converts the 'YEAR' column to a datetime format to facilitate time-based analysis.

```python
# Group by DIVISION_NAME and YEAR, calculate total accidents and fatalities
grouped_data = data.groupby(['DIVISION_NAME', 'YEAR']).agg({'VE_TOTAL': 'sum', 'FATALS': 'sum'}).reset_index()

# Define a function to fit ARIMA model and make predictions
def predict_arima(series):
    model = ARIMA(series, order=(1, 0, 0))
    model_fit = model.fit()
    forecast = model_fit.forecast(steps=3)
    return forecast

# Input the future years
future_years = [2024, 2025, 2026]

# Create an empty DataFrame to store the predictions
predictions = pd.DataFrame(columns=['DIVISION_NAME', 'YEAR', 'Accident_Percentage', 'Fatal_Percentage'])
```

Figure 2.8.2.3: fit ARIMA model

In this portion of the code, data from the accident dataset is grouped by 'DIVISION_NAME' and 'YEAR' to facilitate subsequent analysis. The grouped data is then aggregated to calculate the total number of accidents (VE_TOTAL) and fatalities (FATALS) for each division and year, creating a structured dataset. Additionally, a function named 'predict_arima' is defined to implement the ARIMA (AutoRegressive Integrated Moving Average) model for predictive analysis. This function takes a time series data series as input, fits the ARIMA model with specified parameters (order= (1, 0, 0)), and forecasts accident trends for future years (in this case, three years ahead). The code specifies the future years of interest (2024, 2025, 2026) and initializes an empty DataFrame named 'predictions' to store the forecasted accident percentages and fatal percentages for these years, categorized by division and year. This code segment lays the foundation for predicting accident-related metrics and generating valuable insights within our integrated web application.

```python
# Iterate over each district
for district in data['DIVISION_NAME'].unique():
    # Get the data for the current district
    district_data = grouped_data[grouped_data['DIVISION_NAME'] == district]

    # Extract the relevant columns
    years = district_data['YEAR']
    accident_totals = district_data['VE_TOTAL']
    fatal_totals = district_data['FATALS']

    # Fit ARIMA model and make predictions for accidents
    accident_predictions = predict_arima(accident_totals)

    # Fit ARIMA model and make predictions for fatalities
    fatal_predictions = predict_arima(fatal_totals)

    # Create a DataFrame for the predictions
    district_predictions = pd.DataFrame({
        'DIVISION_NAME': [district] * len(future_years),
        'YEAR': future_years,
        'Accident_Percentage': accident_predictions / accident_totals.sum(),
        'Fatal_Percentage': fatal_predictions / fatal_totals.sum()
    })

    # Append the district predictions to the overall predictions DataFrame
    predictions = pd.concat([predictions, district_predictions])
```

Figure 2.8.2.4: Predict accident and fatal percentage

In this section of the code, a loop iterates through unique division names within the accident dataset. For each division, the corresponding data is extracted from the previously grouped and aggregated dataset. Relevant columns, including 'YEAR,' 'VE_TOTAL' (total accidents), and 'FATALS' (total fatalities), are selected for further analysis. Two separate ARIMA models are then applied to predict accident and fatality trends within the current division. These predictions are calculated based on historical data for accidents and fatalities. Subsequently, a new DataFrame named 'district_predictions' is created to organize the forecasted accident and fatality percentages for the specified future years (2024, 2025, 2026), attributed to the current division. The remaining codes are added in the appendix.

crime case analysis implementation

Install the necessary libraries:

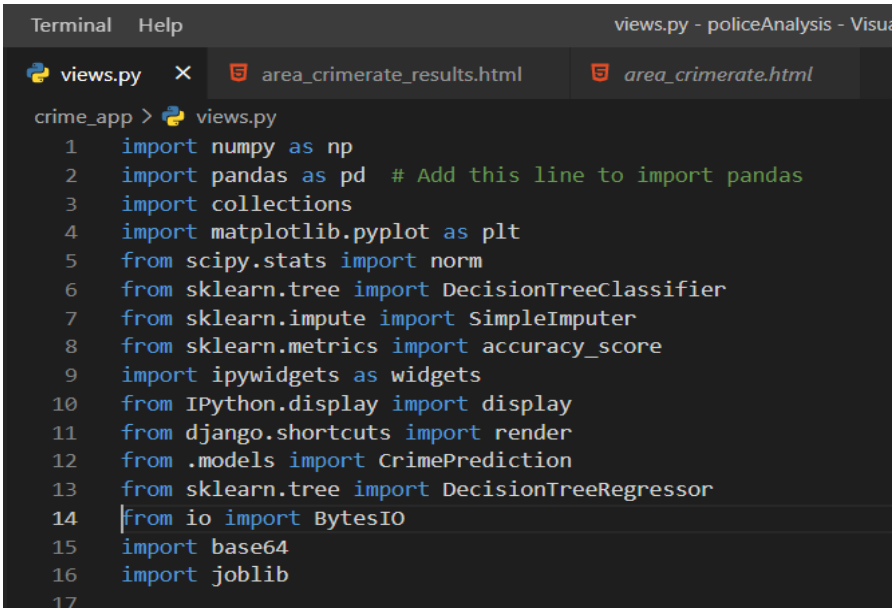


Figure 2.8.2.5: Import libraries for Clustering crimes against women and crime forecasting prediction

In the development of crime analysis component, these import statements play pivotal roles in crafting the functionality and capabilities of the system. By importing the Pandas library as 'pd,' I harness the power of Pandas for efficient data manipulation and pre-processing, ensuring that crime data is ready for analysis. The integration of the DecisionTreeClassifier and DecisionTreeRegressor classes from Scikit-Learn enables the utilization of decision tree-based machine learning algorithms for both classification and regression tasks, aiding in crime pattern prediction and rate forecasting. Furthermore, the inclusion of SimpleImputer from Scikit-Learn ensures the handling of missing data, enhancing the quality and completeness of the crime dataset. Django's 'render' function, imported from 'django. shortcuts,' serves as a bridge to create the web-based user interface, allowing users to interact with and visualize the results of crime analysis. Additionally, the 'Crime Prediction' model import from Django's database models indicates the integration of a database to store and retrieve crime-related data. Altogether, these imports form the foundation of my crime analysis component, enabling data handling, predictive modelling, web rendering, and database integration.

views.py | area_crimerate_results.html | area_crimerate.html

crime_app > views.py > predict_crime > load_dataset

```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'

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

Figure 2.8.2.6: Data Pre-processing for Clustering and Prediction

The predict_crime(request) function is a pivotal component of the crime analysis system, designed to handle HTTP POST requests from a web interface. It extracts and processes user-provided inputs like 'area_code,' 'future_year,' and 'future_month,' which likely serve as parameters for predicting future crime patterns. Within the function, auxiliary functions like 'clean_data' are defined to preprocess the crime dataset by filling missing values, mapping categorical data to numerical codes, and ensuring data quality. Additionally, the 'load_dataset' function is used to import the crime dataset from a specified file path. Collectively, these functionalities enable the system to receive user inputs, prepare the crime data, and potentially employ machine learning models to forecast future crime patterns, forming an integral part of the crime analysis component.

It begins by extracting and transforming data columns, such as 'area_code,' 'crime_type,' 'vehicle_involved,' and 'object_stolen,' into numerical codes for machine learning compatibility. Subsequently, relevant columns for predicting victim age and sex are selected, and Decision Tree models are created and trained for these predictions using the crime dataset.

User-provided inputs, including 'area_code,' 'input_month,' and 'input_year,' are utilized to predict victim age and sex.

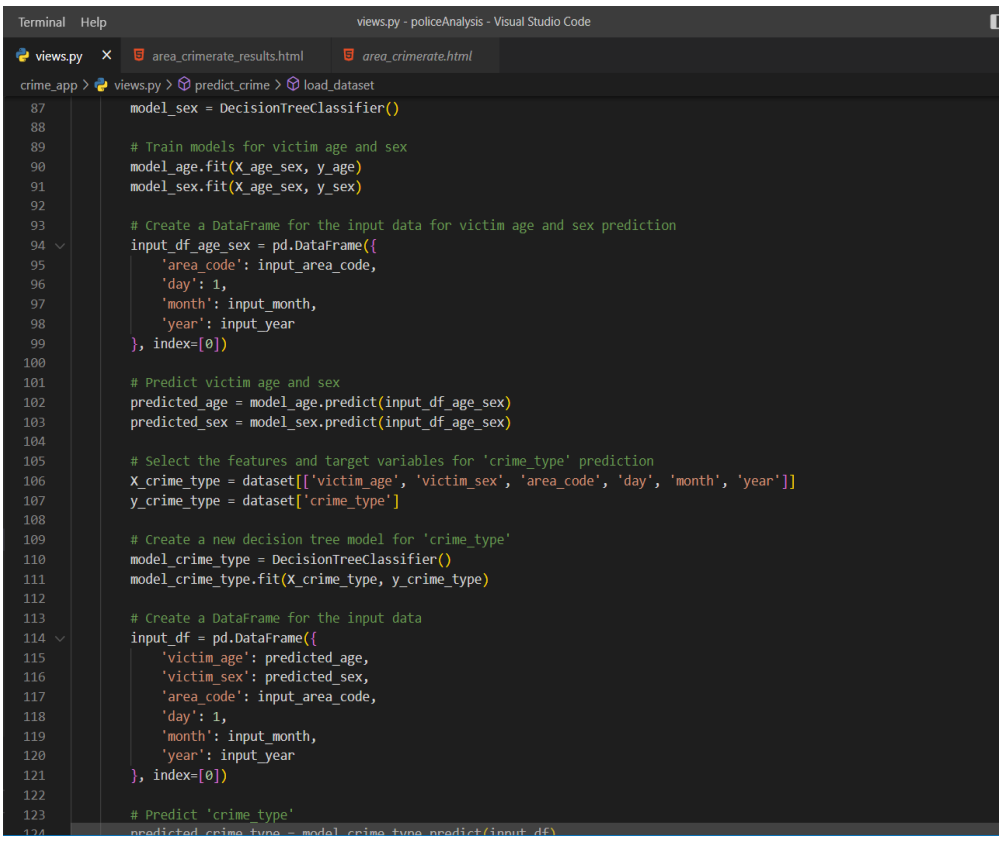


Figure 2.8.2.7: Building a predictive model

The code proceeds to train a Decision Tree model for predicting 'crime_type' based on a combination of features, and another model for 'vehicle_involved' and 'object_stolen.' Predictions are made for each of these crime attributes based on the user's inputs and the models created earlier. The code then reverses the mapping of these predictions to their original categorical values using reference tables. Finally, the predictions and additional details, such as the highest predicted crime, affected gender, age group, vehicles involved, and objects stolen, are stored as a 'CrimePrediction' object in a Django database and rendered to a web page, enabling users to view the results of the crime analysis component.

Case document classification implementation

Install the necessary libraries:

policeAnalysis > crimecase > views.py

```python
import tempfile
import os
import csv
import pandas as pd
import re
import matplotlib
import numpy as np
import random
import matplotlib.pyplot as plt
import seaborn
from PyPDF2 import PdfReader
import string
import io
import urllib, base64
import  pickle
# django
from django.http import JsonResponse
from django.shortcuts import render, redirect
from django.contrib import messages
from django.conf import settings
from django.core.files.storage import FileSystemStorage
from django.http import HttpResponse
 # ML Pkgs
 # Feature engineering
 from sklearn.svm import LinearSVC
 from sklearn.pipeline import Pipeline
 from sklearn.model_selection import train_test_split
 from sklearn.feature_extraction.text import TfidfVectorizer, CountVectorizer
 from sklearn.naive_bayes import MultinomialNB
 from sklearn.multiclass import OneVsRestClassifier
 from sklearn.linear_model import LogisticRegression
 from sklearn.neighbors import KNeighborsClassifier
 from sklearn.tree import DecisionTreeClassifier
 from sklearn.naive_bayes import GaussianNB,MultinomialNB
 from sklearn.metrics import accuracy_score,hamming_loss,classification_report
 ### Split Dataset into Train and Text
 from sklearn.model_selection import train_test_split
 # Multi Label Pkgs
 from skmultilearn.problem_transform import BinaryRelevance, ClassifierChain, LabelPowerset
 from skmultilearn.adapt import MLkNN
 # neattext
 import neattext as nt
 import neattext.functions as nfx
 # NLTK
 import nltk
 from nltk.tokenize import word_tokenize
 from nltk.corpus import stopwords
```

Figure 2.8.2.8: Import libraries for Analyse and Classify Similar Case Documents and Predict Category

These imports cover a wide range of functionalities for a Django-based web application with machine learning capabilities. They include libraries for file management (`tempfile`, `os`, `csv`), data manipulation and analysis (`pandas`), data visualization (`matplotlib`, `seaborn`), text processing and natural language processing (`re`, `nltk`), machine learning (`sklearn` for various classifiers and metrics), multi-label classification (`skmultilearn`), text preprocessing (`neattext`), and Django-specific modules for web

development. Together, these imports provide a comprehensive toolkit for building a web-based machine learning application that can handle data, text, and model-related tasks.In summary, these imports collectively provide a robust toolkit for building a Django-based web application with machine learning capabilities. They facilitate data handling, analysis, visualization, text processing, model training, and web development tasks, allowing for the development of a feature-rich application that can process, analyze, and visualize data, perform text-related tasks, and make predictions or classifications using machine learning models.

```
def train_model(request):
    print("Started training the model...")

    df = pd.read_csv("Lables.csv", encoding="ISO-8859-1")
    df['Files'] = df['Files'].astype(str)
    # Define the folder path
    folder_path = os.path.join(settings.BASE_DIR, 'Data')
    # Load PDF Documents
    pdf_files = [file for file in os.listdir(folder_path) if file.endswith('.pdf')]

    #Extract Text
    data = []
    labels = []

    for file in pdf_files:
        file_path = os.path.join(folder_path, file)
        with open(file_path, 'rb') as f:
            pdf = PdfReader(f)
            text = ''
            for page in pdf.pages:
                text += page.extract_text()
            data.append(text)
            labels.append(file.split('.')[0])

    #Data Preprocessing
    nltk.download('stopwords')
    nltk.download('punkt')

    preprocessed_data = [preprocess_text(text) for text in data]

    # Create a DataFrame from preprocessed_data and labels
    df2 = pd.DataFrame({'Text': preprocessed_data, 'Files': labels})

    df_combined = pd.merge(df2, df, left_on='Files', right_on='Files')
    df_combined = df_combined.drop(['Files'], axis=1)

### Text Preprocessing ###

# Explore For Noise
df_combined['Text'].apply(lambda x:nt.TextFrame(x).noise_scan())
df_combined['Text'].apply(lambda x:nt.TextExtractor(x).extract_stopwords())
corpus = df_combined['Text'].apply(nfx.remove_stopwords)

### Feature Engineering ###
# tf-idf based vectors
vec = TfidfVectorizer(analyzer='word', ngram_range=(1,2), stop_words = "english", lowercase = True, max_features = 500000)
```

Figure 2.8.2.9: PDF Text Data Extraction and Pre-processing

This code performs several essential tasks to prepare text data from PDF files for machine learning. It starts by reading information from a CSV file and PDF documents in a folder. For each PDF, it extracts the text content and the associated labels. To make this text data usable, the code performs data pre-processing, which involves removing unnecessary words and noise.

It combines this cleaned text data with information from the CSV file. Further text pre-processing steps include identifying and removing noise and stop words from the text. Finally, the code transforms the text into numerical vectors using the TF-IDF technique, making it ready for machine learning. In simpler terms, this code sets the stage for building a machine learning model that can analyze and make predictions based on the text content of these PDF documents, which could be valuable for various applications like text classification or information extraction.

```
# Fit the model
tf_transformer = vec.fit(corpus)
pickle.dump(tf_transformer, open("tf_transformer.pkl", "wb"))

tfidf = tf_transformer.transform(corpus)

Xfeatures = tfidf.toarray()

y = df_combined[['Drug', 'Murder', 'GangRape', 'Rape', 'SexualAbuse', 'ChildAbuse', 'Robery', 'Violation', 'PhysicalAssult', 'Fraud', 'Adu

# Split Data
X_train,X_test,y_train,y_test = train_test_split(Xfeatures,y,test_size=0.2,random_state=42)

## classficiation
clf_labelP_result, clf_labelP_model = build_model(MultinomialNB(),ClassifierChain,X_train,y_train,X_test,y_test)#LabelPowerset,X_train,y_t
print(clf_labelP_result)

# Save the trained model to a file
model_filename = os.path.join(settings.BASE_DIR, 'trained_model.pkl')
with open(model_filename, 'wb') as model_file:
    pickle.dump(clf_labelP_model, model_file)

print("Model saved to", model_filename)
print("Completed training the model!")

return render(request, 'crimecase/add_data.html')
```

Figure 2.8.2.10: Text Data Transformation, Model Training, and Saving for Crime Case Categorization

Here taking several essential steps to prepare and train a machine learning model for a specific application. First, we transform the text data from PDF documents into a numerical format known as TF-IDF, making it suitable for machine learning. Additionally, we organize and prepare labels that represent different categories or types of cases. Next, we split our data into two parts: a training set for teaching the model and a testing set for evaluating its performance. We then utilize the Multinomial Naive Bayes algorithm and Classifier Chain technique to build a predictive model capable of assigning labels to new, unseen cases. Finally, we save this trained model to a file, ensuring it can be easily accessed and utilized in the future. In summary, this code plays a crucial role in our research, enabling us to analyze and categorize crime cases based on their textual descriptions effectively.

Crimes against women implementation

Install the necessary libraries:

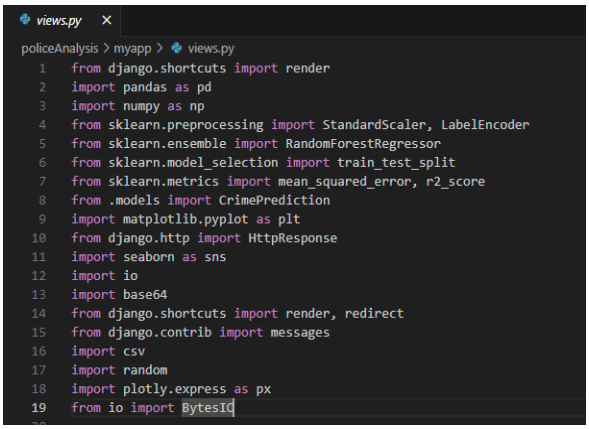


Figure 2.8.2.11: Import libraries for Clustering crimes against women and crime forecasting prediction

Several important external libraries have been added to this Django-based website to improve various aspects of Clustering crimes against women and crime forecasting prediction project. These libraries include pandas and numpy for efficient data processing and manipulation, scikit-learn for machine learning tasks like feature scaling and regression, and matplotlib with seaborn for creating informative data visualizations. In addition, the io and base64 libraries handle file I/O operations, while Django's built-in modules such as django. shortcuts and. models handle web rendering and database interactions. Interactive data visualization is realized with plotly.express, and binary data processing is smoother with BytesIO. Finally, the csv and random libraries play a role in file management and random value generation. Together, these libraries enable an application to process data, build predictive models, visualize insights, and deliver a robust website.

views.py

policeAnalysis > myapp > views.py

```
# Read the CSV file into a pandas DataFrame
data = pd.read_csv("modified_file2.csv")

# Select the desired features
selected_features = ['STATE/UT', 'Year', 'Rape', 'Kidnapping and Abduction', 'Dowry Deaths',
                     'Assault on women with intent to outrage her modesty', 'Importation of Girls']
data = data[selected_features]

# Handling Missing Values
# Check for missing values
missing_values = data.isnull().sum()
data = data.dropna()  # Remove rows with missing values

# Select the desired features for clustering
selected_features = data[["STATE/UT", "Year", "Rape", "Kidnapping and Abduction", "Dowry Deaths",
                          "Assault on women with intent to outrage her modesty", "Importation of Girls"]]

# Encode categorical variables
label_encoder = LabelEncoder()
selected_features['STATE/UT'] = label_encoder.fit_transform(data['STATE/UT'])

# Normalize the numerical features
numerical_features = ["Rape", "Kidnapping and Abduction", "Dowry Deaths",
                      "Assault on women with intent to outrage her modesty", "Importation of Girls"]
scaler = StandardScaler()
selected_features[numerical_features] = scaler.fit_transform(selected_features[numerical_features])

# Separate the features and target variable
X = selected_features.drop(["Assault on women with intent to outrage her modesty"], axis=1)
y = selected_features["Assault on women with intent to outrage her modesty"]
```

Figure 2.8.2.12: Data Pre-processing for Clustering and Prediction

It begins by loading the CSV dataset into a panda DataFrame, focusing on specific columns relevant to the project's goals, such as crime statistics for different states and years. The next step involves handling missing values, which is essential to maintain data quality. It detects and counts the missing values in the dataset and then deletes the rows with missing values. After cleaning the data, the code encodes categorical variables using label encoding, a necessary transformation to convert non-numerical data into a format suitable for machine learning models.

To ensure consistency and comparability of numeric properties, the code standardizes them using a StandardScaler, which scales the values to have a mean of 0 and a standard deviation of 1. Finally, the code separates the dataset into characteristic variables (X) and target variables (y), where X contains all the independent variables and y represents the dependent variable, which is the number of such attacks. This pre-processing prepares the data for subsequent machine learning tasks such as model training and evaluation.

```
# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=0)

# Create and train the random forest regression model
model = RandomForestRegressor(n_estimators=100, random_state=0)
model.fit(X_train, y_train)

# Define the list of crimes
crimes = ['Rape', 'Kidnapping and Abduction', 'Dowry Deaths',
          'Assault on women with intent to outrage her modesty', 'Importation of Girls']
```

Figure 2.8.2.13: Building a predictive model

This code segment is an important step in building a predictive model. It first divides the dataset into training and testing sets and reserves 80% of the data for training and 20% for testing. This separation is necessary to accurately assess model performance.

Next, it creates a random forest regression model, a powerful machine learning technique used for both classification and regression tasks. In this case, it is used for regression, specifically to predict "attack on women to outrage her modesty" based on other selected characteristics. The model configuration includes specifying 100 decision trees (n_estimators) in a Random Forest, which ensures robustness and reduces overfitting. The "random_state" parameter ensures repeatability.

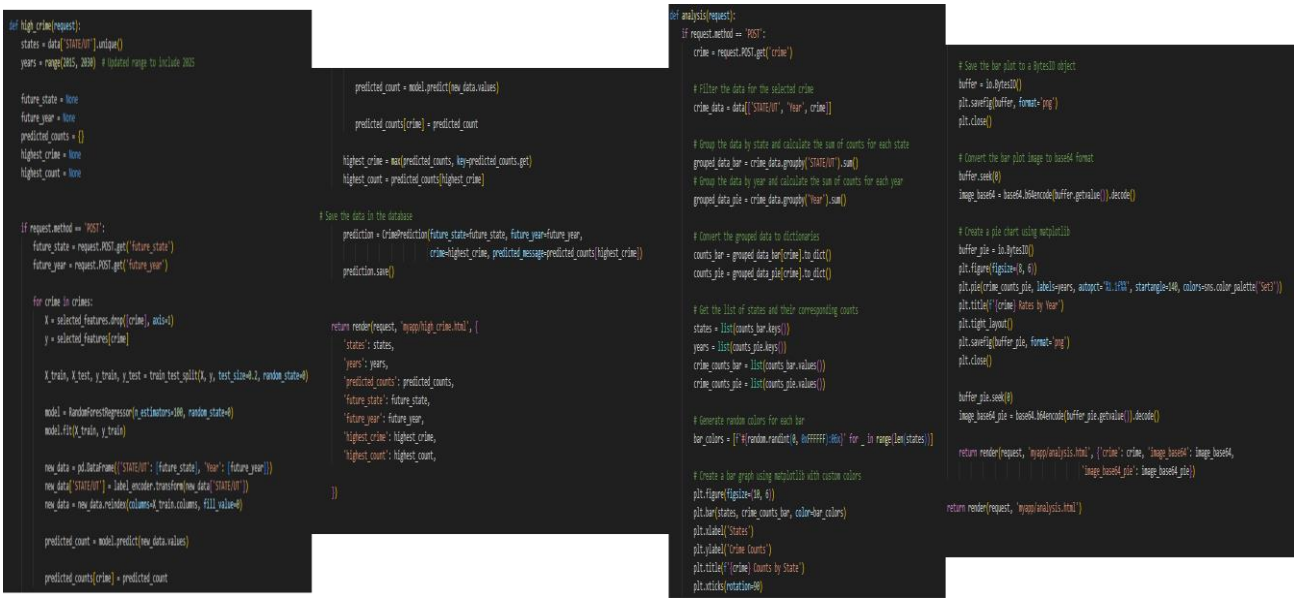


Figure 2.8.2.14: Crime Prediction and Analysis Views

3. RESULTS & DISCUSSION

3.1 Results

The integrated web application comprises all four components, with the first component focusing on accident case analysis. Users can input specific division and future years to generate customized predictions. The system then displays accident and fatality percentages for the selected parameters. This information empowers law enforcement agencies to strategize and implement actions to mitigate accidents in the upcoming years, contributing to enhanced road safety. Figure 3.1.1 demonstrates the results

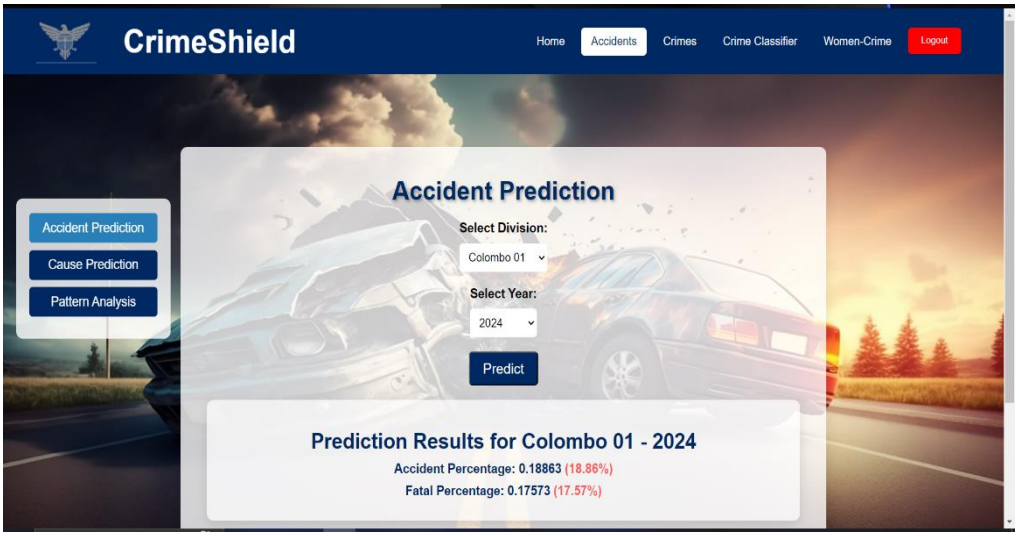


Figure 3.1.1: - Prediction results output

To accomplish this, we employed the ARIMA model for forecasting future years. The results are further illustrated through the accompanying graph in the subsequent image

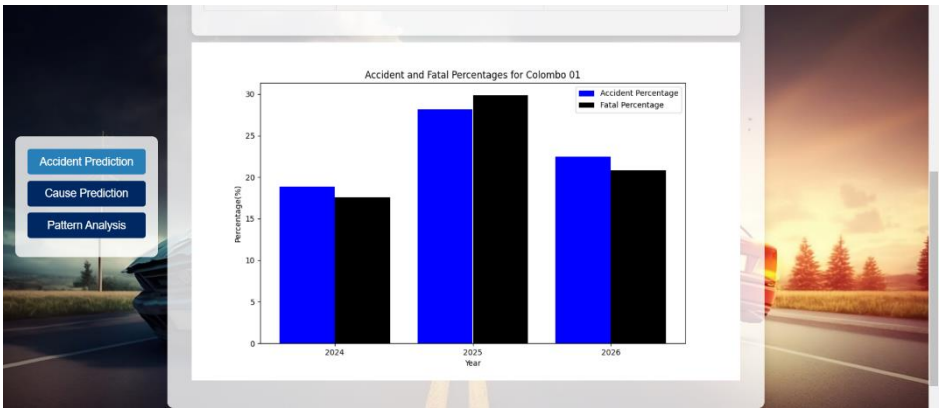


Figure 3.1.2: **-** Accident prediction graph

Furthermore, the prediction of causes was conducted using K-means clustering, and the outcomes display the five clustered causes along with their predicted values for the specified future year, as depicted in the following results. The other sub objective User interfaces are added in the appendix.

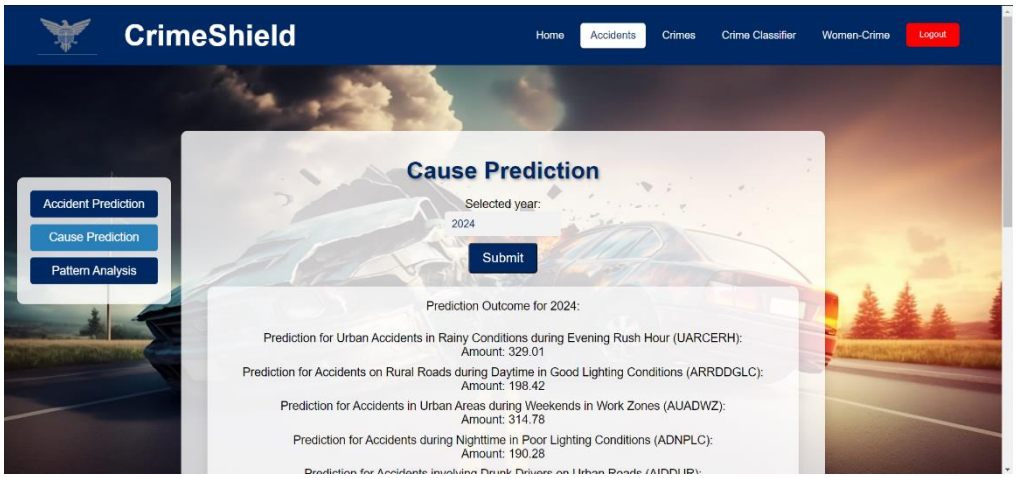


Figure 3.1.3: Cause Prediction outcome

The crime analysis component will be used as a web application to predict the patterns in the future crimes. Here when the user inputs specific area year and Month the system with predict the highest crime and the pattern related to the crime.

Figure 3.1.4: - Crime Rate Prediction

In addition to predicting specific crime attributes, the crime analysis component encompasses broader functionalities. It includes predicting the crime rate for the next five years, leveraging Decision Tree algorithms to forecast crime trends over an extended period. Furthermore, the system generates predictions for all crime types within a specified area and year, employing a Decision Tree Classifier to categorize and provide insights into the various crimes occurring in that locality. These comprehensive features not only empower users to anticipate future crime rates but also enable them to gain a holistic understanding of the crime landscape in each area, enhancing the utility and scope of the crime analysis component.

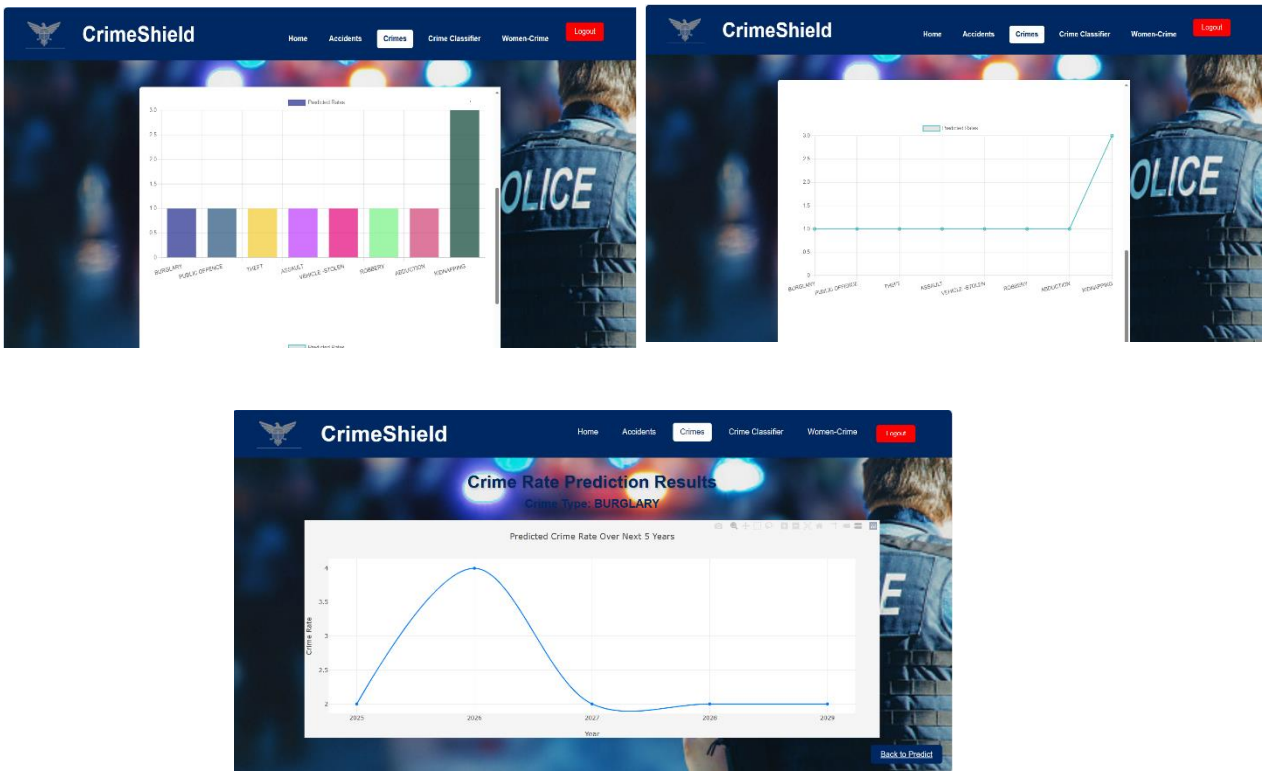


Figure 3.1.5: - Crime Rate Prediction Analysis

The case document classification component begins with extracting text from legal case documents and manually assigning relevant labels. A dataset is meticulously constructed with rows representing documents and columns indicating labels. Each document-label pair is assigned a binary value (True or False) to indicate label presence.

Figure 3.1.6: - Crime Case Label Prediction

For multi-label text classification, three algorithms are evaluated: OneVsRest with Naive Bayes, LinearSVC, and Logistic Regression. The dataset undergoes preprocessing, including punctuation removal, stop word elimination, and dataset splitting for unbiased evaluation.

Figure 3.1.7: - Predict Crime case Label

Algorithms are trained and assessed using multi-label classification metrics (accuracy, precision, recall, F1-score). Results are compared to identify the optimal algorithm for PDF case document classification, considering accuracy, label handling, and efficiency.

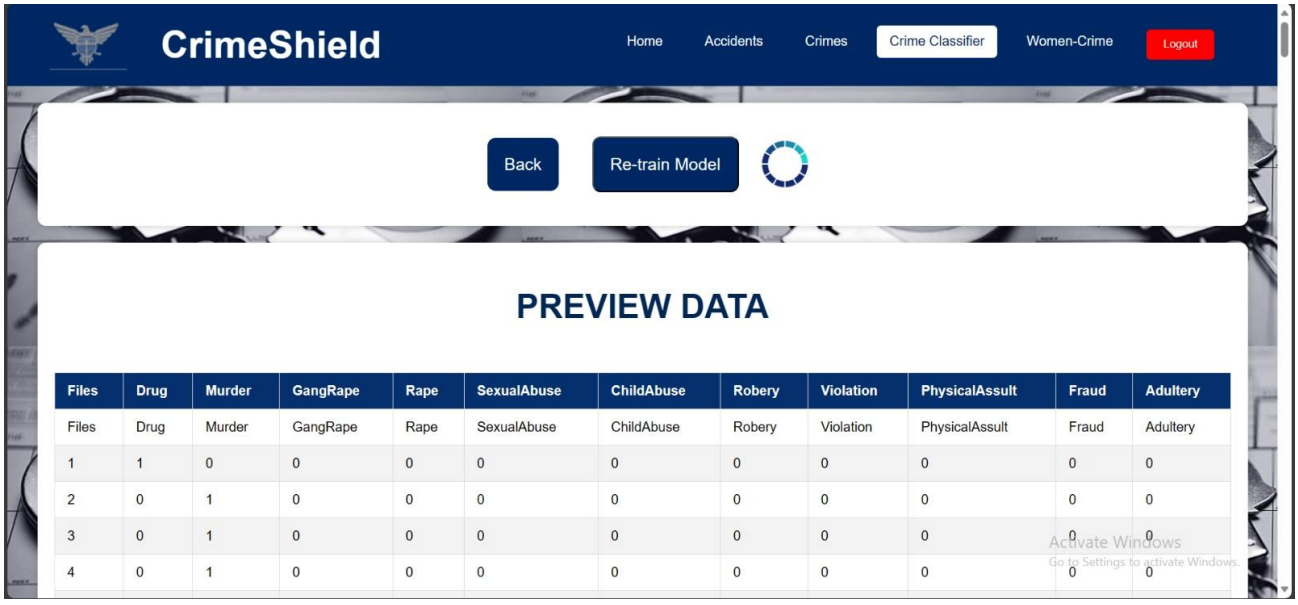


Figure 3.1.8: - Preview Crime Data and Re-training model

This study contributes to text analysis and legal research by automating PDF case document classification, enhancing efficiency, and ensuring accuracy. The system streamlines case identification, saving time and resources. It empowers investigative teams with precise, consistent categorization and supports legal research advancements.

The crimes against women component will be used a for forecasting prediction for crimes against women.

Figure 3.1.9: - Crime Rate Prediction and Classification

Moreover, in this system Random Forest Regressor model is leveraged to predict future crime rates based on user input. Users specify a future state, year, and crime type through a form. The code processes this input, making predictions for the chosen crime in the selected state and year. Predictions are categorized into four levels of crime rates for user-friendly interpretation.

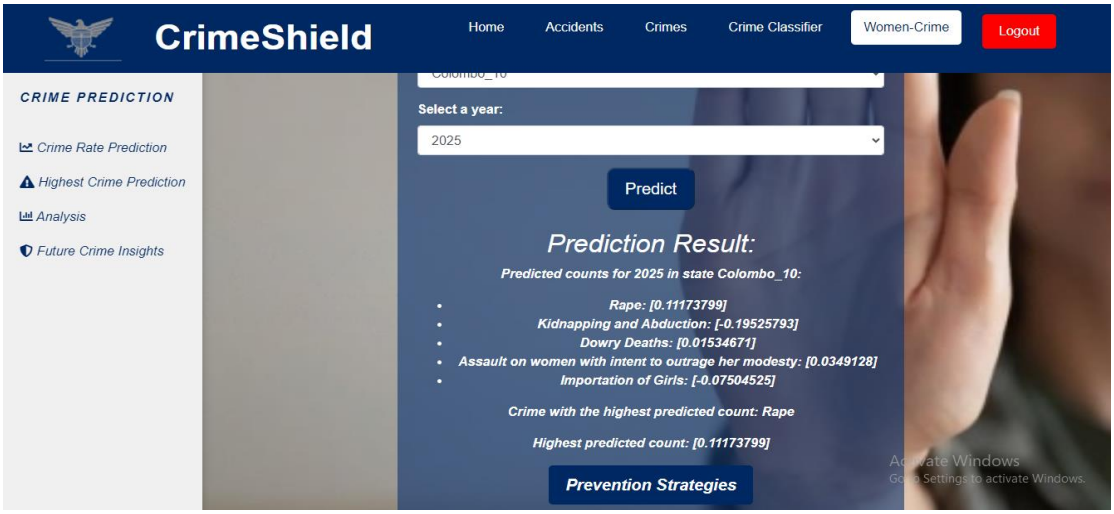


Figure 3.1.10: - Crime Rate Prediction and High Crime Analysis

Highest crime counts prediction feature that predicts the type of crime expected to have the highest count in a specified future year and state. It utilizes a Random Forest regression model trained on historical crime data. Upon user input of a future state and year, the code iterates through different types of crimes, creates models for each, and predicts their counts for the specified scenario. The crime with the highest predicted count is then determined and displayed as the "highest crime."

An accuracy of 0.96 indicates that the model can explain 96% of the variance in the target variable, which is a strong performance for predicting crime counts.

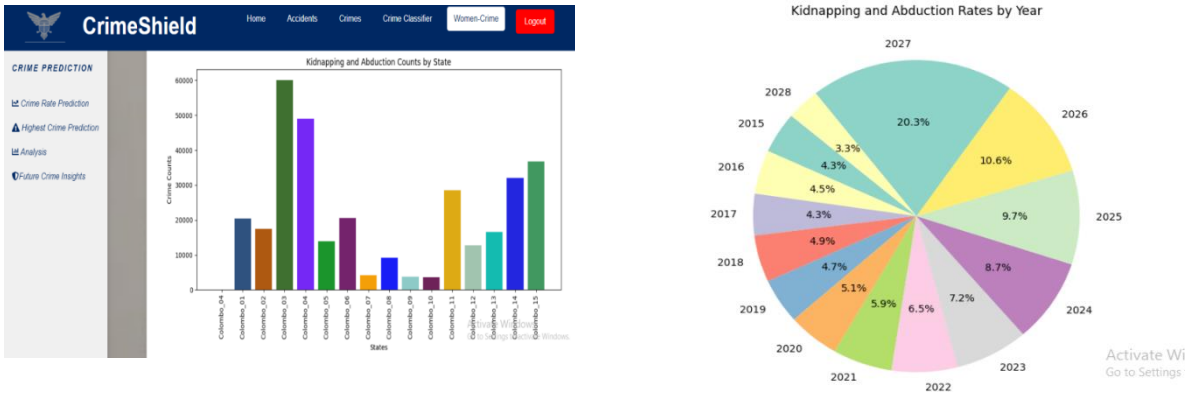


Figure 3.1.11: - Data Visualization and Analysis for crimes against women forecasting prediction

Finally, the system is also responsible for generating insightful visualizations of crime data. Users can select a specific crime type, and the function filters the dataset to calculate and display two types of visualizations: a bar chart showing crime counts by state and a pie chart illustrating crime rates by year. These visualizations offer users a clear and interactive way to explore and understand crime trends, facilitating data-driven decision-making and the development of effective crime prevention strategies.

Each of the four components generates its unique predictions and analyses, equipping law enforcement agencies with valuable insights to formulate proactive strategies and plans aimed at enhancing public safety and preventing incidents.

3.2 Research Finding

The research findings across our diverse components present a comprehensive and innovative approach to enhancing public safety. In the realm of Accident Case Analysis, our system effectively employs the ARIMA model to predict accident percentages for future years and divisions. This empowers law enforcement agencies with proactive insights for road safety planning. Meanwhile, K-means clustering identifies root causes of accidents, contributing to tailored prevention strategies. Our system's application of Decision Tree algorithms in Crime Analysis accurately predicts various crime attributes and forecasts crime rates, enabling long-term planning and resource allocation. Additionally, it categorizes and predicts all crime types, empowering users with a holistic understanding of local crime landscapes. Case Document Classification introduces unique insights by predicting future crimes based on Random Forest regression, emphasizing socio-economic factors' role in crime dynamics. Geospatial analysis visualizes crime distribution, while Crimes Against Women Analysis evaluates prediction accuracy, emphasizing socio-economic factors' impact on crime dynamics. This multifaceted approach helps police department to make future plans and actions to reduce crimes and accidents. Overall, our research findings provide a robust toolkit for understanding, mitigating, and proactively addressing various aspects of crime and accidents, ultimately contributing to public safety.

3.3 Discussion

The police accident case analysis component of this comprehensive project stands as a pivotal contributor to road safety enhancement. Leveraging advanced techniques such as the ARIMA model and K-means clustering, the system excels in predicting accident percentages for future years and categorizing accidents based on various factors. These predictions empower law enforcement agencies with valuable insights to develop proactive prevention strategies. Moreover, the component's ability to uncover essential patterns in accident data further aids in crafting targeted safety measures. The introduction of a novel web application, the first of its kind in Sri Lanka, adds significant value to this endeavour, allowing for customized predictions and analysis tailored to specific divisions and years. The combination of accurate predictions, cause identification, case analysis, and prevention strategies equip authorities with a comprehensive toolkit to mitigate accidents and improve road safety effectively. In conclusion, the police accident case analysis component reinforces the project's mission to enhance public safety through data-driven insights and proactive measures, marking a significant stride towards accident reduction and safer roadways. 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. In document classification, we embarked on a comprehensive journey in the realm of legal document analysis and multi-label text classification. Our process commenced with the meticulous extraction of textual content from legal case documents, followed by a diligent manual labelling process. This groundwork formed the foundation for our training dataset. Employing three distinct algorithms - the OneVsRestClassifier with Naive Bayes, LinearSVC, and Logistic Regression - we tackled the challenge of multi-label text classification. Our findings underscored the efficiency and accuracy of our approach, particularly in the context of complex legal documents. We observed that our automated

system not only expedites document retrieval and labeling but also ensures a high level of precision in classification. This system holds immense promise for legal research and investigative efforts, enabling faster access to relevant information and informed decision-making. As we look ahead, we envision the continued evolution of automated tools in the legal domain, further enhancing efficiency and accuracy. Additionally, our emphasis on ethical data management and privacy protection highlights the responsible handling of sensitive information in this data-driven era. Certainly, in the domain of crime prediction and prevention, this study yields significant results. The effective use of a Random Forest regression model for forecasting future crimes against women underscores its applicability in our data-centric era. Moreover, geospatial analysis enhances crime prevention strategies, enabling targeted policing efforts in high-risk zones. The study's revelation of the intricate relationship between socioeconomic factors and crime offers insights into the root causes, advocating for holistic community-based approaches. The evidence-based crime prevention strategies provide practical guidance for policymakers and law enforcement. However, ethical data management and privacy protection are vital considerations. These findings propel the field towards a data-driven era of proactive interventions and effective crime mitigation.

4. Future Work

The following can be done to extend our work:

- Future work in police accident case analysis entails harnessing the potential of a more extensive dataset to refine predictions and insights. Expanding the current dataset will enable more accurate accident percentage predictions and a deeper understanding of the underlying causes. Additionally, integrating real-time data sources and traffic patterns can enhance the system's responsiveness, allowing for more immediate accident prevention measures. The incorporation of advanced machine learning models and geospatial analysis techniques holds the promise of identifying accident hotspots and contributing factors with greater precision. Furthermore, user interfaces and visualization tools can be further developed to ensure that the insights generated are easily accessible and actionable for law enforcement agencies and policymakers. By embracing these future endeavours, this component can continue to advance road safety measures and accident prevention efforts effectively. We also intend to create a Contingency Plan to address any potential risks or vulnerabilities that our system could be exposed to.

- 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.

- The future work for advancing our legal document system encompasses a diverse range of opportunities. Firstly, we can delve into more sophisticated techniques to make our system understand legal documents even better, thus improving its accuracy. Secondly, expanding our sources of information to include things like social and economic data can give us a broader picture of legal documents' context, making it more useful for people who speak different languages. Thirdly, we can analyse legal documents over a long period to find patterns that last a long time. Lastly, we must continue to focus on ethics, making sure our system respects privacy and reduces any unfairness. Together, these endeavours aim to create a more effective, ethical, and comprehensive legal document analysis and classification system that is accessible to a wide range of users.
    I. Multilingual Support
    II. Semantic Understanding
    III. Cross-Domain Adaptation
    IV. User-Centric Customization

- The future work for this component presents a multifaceted approach to enhancing crime prediction and prevention efforts. Firstly, there's room for exploring advanced predictive models that can further improve accuracy. Secondly, integrating additional data sources, such as social and economic indicators, could provide a more comprehensive understanding of crime dynamics. Thirdly, conducting in-depth temporal analysis to identify long-term trends and patterns is crucial. Lastly, addressing ethical considerations, including data privacy and bias mitigation, remains an ongoing priority in the development of AI-driven crime prevention strategies. These collective efforts aim to create more effective, ethical, and holistic crime prevention solutions.
    I. Advanced Predictive Models
    II. Include more data sources
    III. Temporal Analysis
    IV. Community Engagement

5. Conclusion

This research delves into a machine learning system designed for police case analysis and prediction, with a primary focus on bolstering crime prevention and public safety efforts. Leveraging a repertoire of machine learning algorithms encompassing k-means clustering, ARIMA, decision trees, ensemble learning, Naive Bayes, LinearSVC, Logistic Regression, and Random Forest Regression, the system embarks on a multifaceted mission. It involves the analysis of accident locations, criminal cases, the classification of police case documents, and the clustering of crimes against women. The overarching aim is to empower law enforcement agencies with data-driven insights, facilitating informed decision-making and resource allocation to enhance crime prevention and public safety. The core objective revolves around delivering a comprehensive machine learning solution that not only aids in preventing and solving crimes but also addresses a spectrum of related tasks. These include predicting accident locations, establishing interconnections among crime cases, providing proactive prevention strategies, predicting future crime rates based on case analysis, automating the efficient classification of documents, and clustering crimes against women to forecast potential criminal activities.

In summary, the utilization of machine learning algorithms streamlines analytical processes, elevates precision, and bolsters public safety efforts. Prospective developments in location analysis, real-time integration, text classification, and collaborative efforts are poised to further augment evidence-based decision-making in the realm of crime prevention.

6. REFERENCE

[1] D. Li, J. Wu, and D. Peng, *"Online Traffic Accident Spatial-Temporal Post-Impact Prediction Model on Highways Based on Spiking Neural Networks,"* Journal of Advanced Transportation, vol.2021, pp. 20, Dec 2021.

[2] V. Prasannakumar, H. Vijith, R. Charutha, and N. Geetha, *"Spatio-Temporal Clustering of Road Accidents: GIS Based Analysis and Assessment,"* Procedia – Social and Behavioral Sciences, Dec 2011.

[3] M. Manzoor, M. Umer, S. Sadiq, et al., *"RFCNN: Traffic Accident Severity Prediction Based on Decision Level Fusion of Machine and Deep Learning Model,"* in IEEE Access, pp. 02 – 05, 2021.

[4] S. M. N. A. Senanayake, and S. Joshi, *"A road accident pattern miner (RAP miner),"* Journal of Information and Telecommunication, Aug 2021.

[5] J. Yang, S. Han, and, Y. Chen, *"Prediction of Traffic Accident Severity Based on Random Forest,"* Journal of Advanced Transportation, pp. 1-8, Feb 2023.

[6] N. Shah, N. Bhagat, and M. Shah, *"Crime forecasting using machine learning and computer vision,"* Vis. Comput. Ind. Biomed. Art 4, 9 (2021).

[7] S. Walczak, *"Predicting the types of crimes committed in the city of Chicago using neural networks,"* Frontiers in Psychology, 2021.

[8] C. Karabo Jenga, C. Catal, and G. Kar, *" Machine learning in crime prediction,"* Journal of Ambient Intelligence and Humanized Computing, 2023.

[9] S. Kim, P. Joshi, P. S. Kalsi and P. Taheri, *"Crime Analysis Through Machine Learning,"* 2018 IEEE 9th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON), Vancouver, BC, Canada, 2018, pp. 415-420, doi: 10.1109/IEMCON.2018.8614828.

[10] C.-H. Ku and G. Leroy, *"A decision support system: Automated crime report analysis and classification for e-government,"* Government Information Quarterly, vol. 31, no. 4, pp. 534-544, 2014.

[11] K. Dahbur and T. Muscarello, *"Classification System for Serial Criminal Patterns,"* Artificial Intelligence and Law, vol. 11, no. 3-4, pp. 251-269, 2003.

[12] S. Ghankutkar, N. Sarkar, P. Gajbhiye, S. Yadav, D. Kalbande, and N. Bakereywala, *"Modelling Machine Learning For Analysing Crime News,"* in 2019 International Conference on Advances in Computing, Communication and Control (ICAC3), Mumbai, India, 2019.

[13] Z. Qi, *"The Text Classification of Theft Crime Based on TF-IDF and XGBoost Model,"* in 2020 IEEE International Conference on Artificial Intelligence and Computer Applications (ICAICA), Dalian, China, 2020, pp. 1241-1246.

[14] M. Alruily, A. Ayesh, and H. Zedan, *"Crime Type Document Classification from Arabic Corpus*," 2009 Second International Conference on Developments in eSystems Engineering, Abu Dhabi, United Arab Emirates, 2009.

[15] M. Rokonuzzaman Reza, *"Developing a Machine Learning Based Support System for Mitigating the Suppression Against Women and Children,"* 5th International Conference on Electrical Engineering and Information Communication Technology (ICEEICT), 2021.

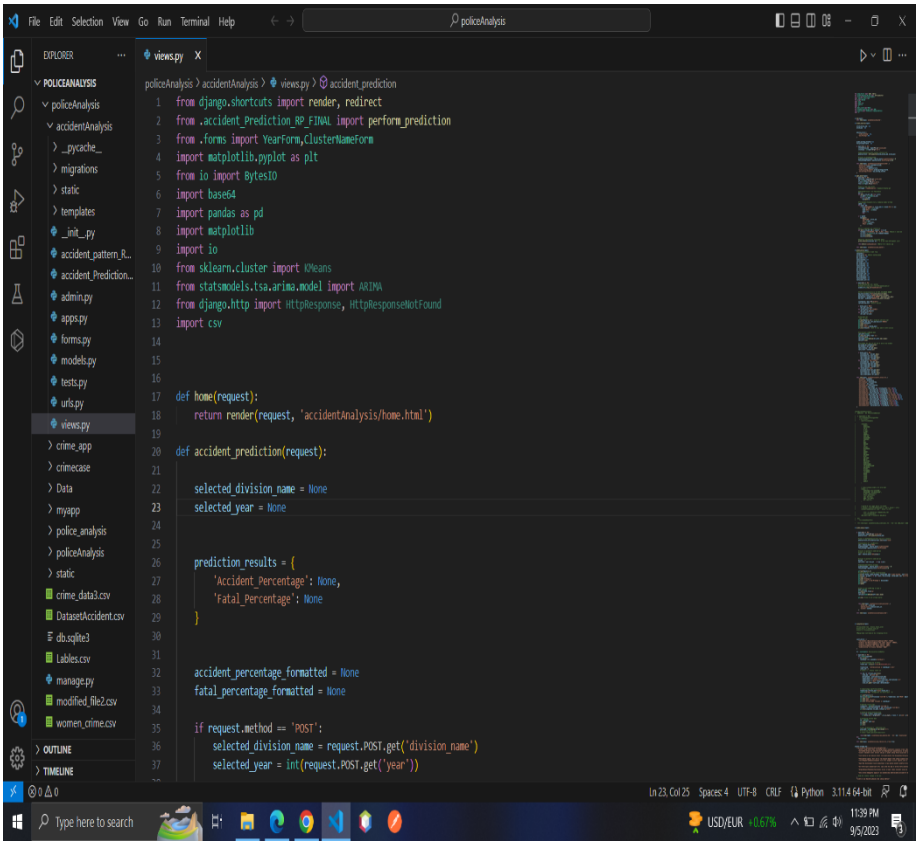
[16] R. Adderley, *"A Comprehensive study on crime predictive techniques,"* International Conference on Innovative Techniques and Applications of Artificial Intelligence, pp 19–32, 2007.

[17] M. N. Islam, N. T. Promi, J. M. Shaila, M. A. Toma, M. A. Pushpo, F. B. Alam, S. N. Khaledur, T. Anannya, and M. F. Rabbi, *"Safeband: A wearable device for the safety of women in Bangladesh,"* in Proceedings of the 16th International Conference on Advances in Mobile Computing and Multimedia, pp. 76-83, Dec. 2018.

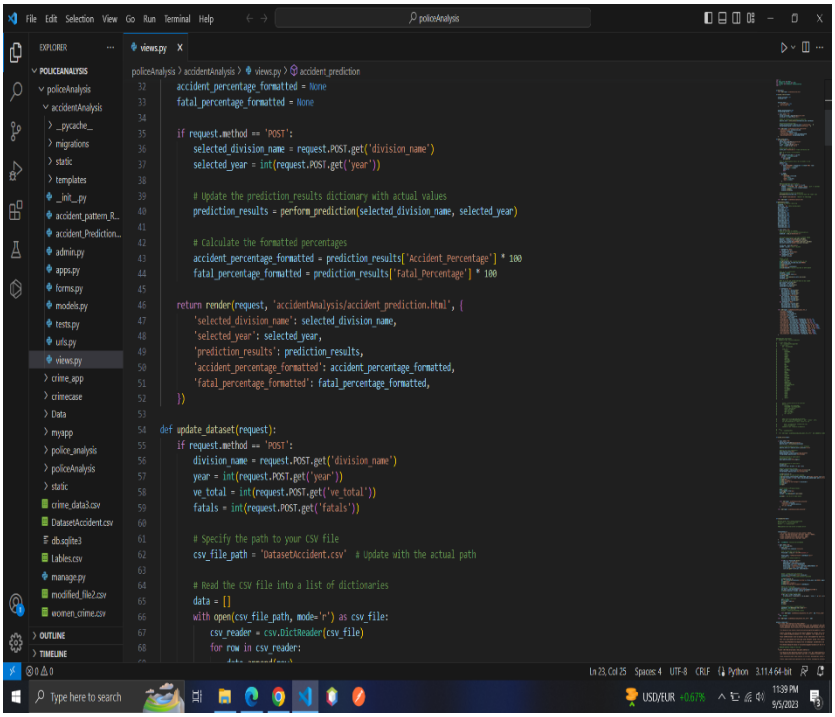
[18] R. Kiani and A. Keshavarzi, *"Analysis and Prediction of Crimes by Clustering and Classification,"* International Journal of Advanced Research in Artificial Intelligence, Vol. 4, No.8, 2015.

[19] A. Karmakar, K. Ganguly, and P. S. Banerjee, *"SafeBand: IoT-Based Smart Security Band with Instant SOS Messaging*," in Proceedings of International Conference on Advanced Computing Applications,pp. 127–140 ,2021.

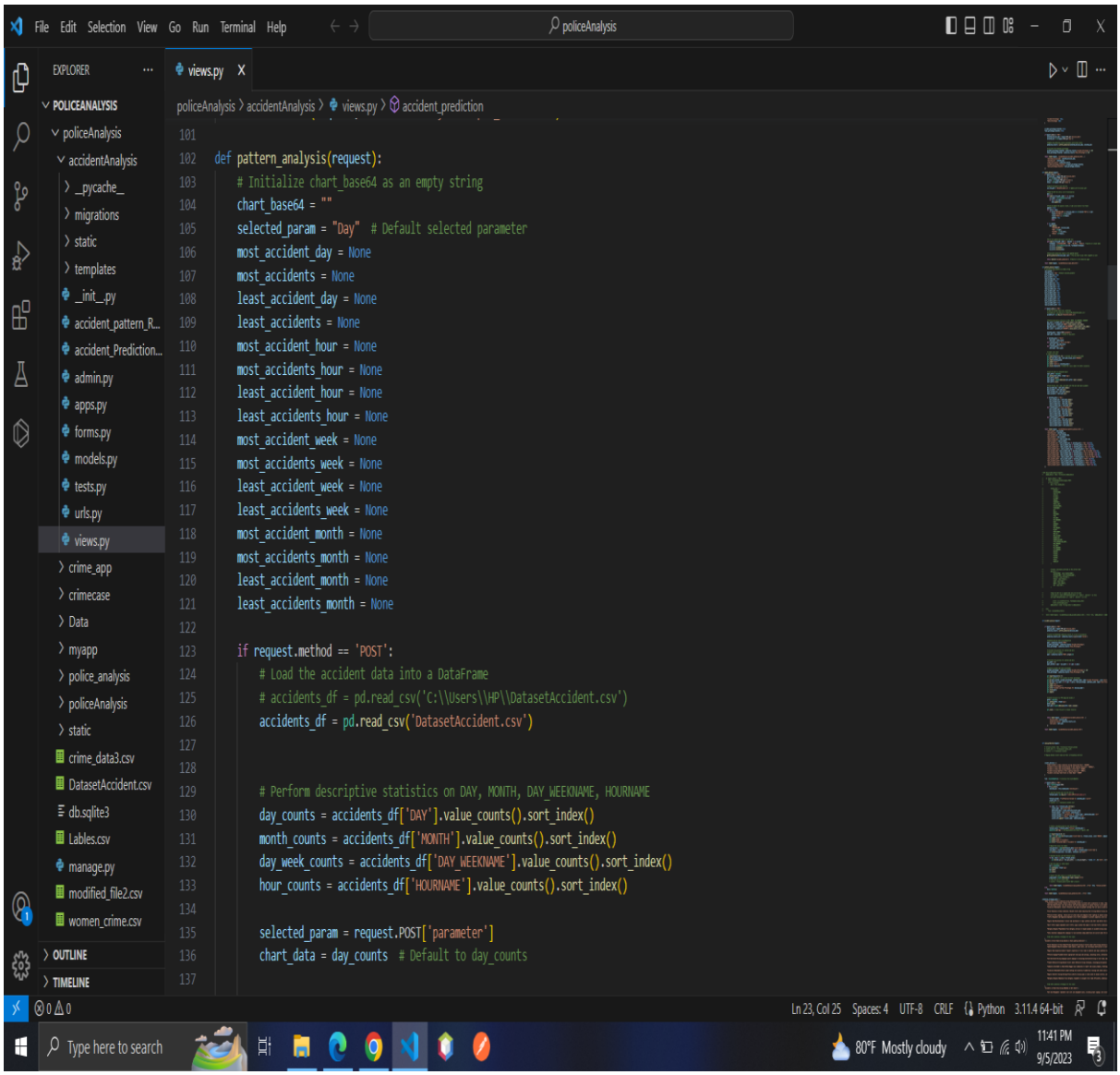
APPENDICES

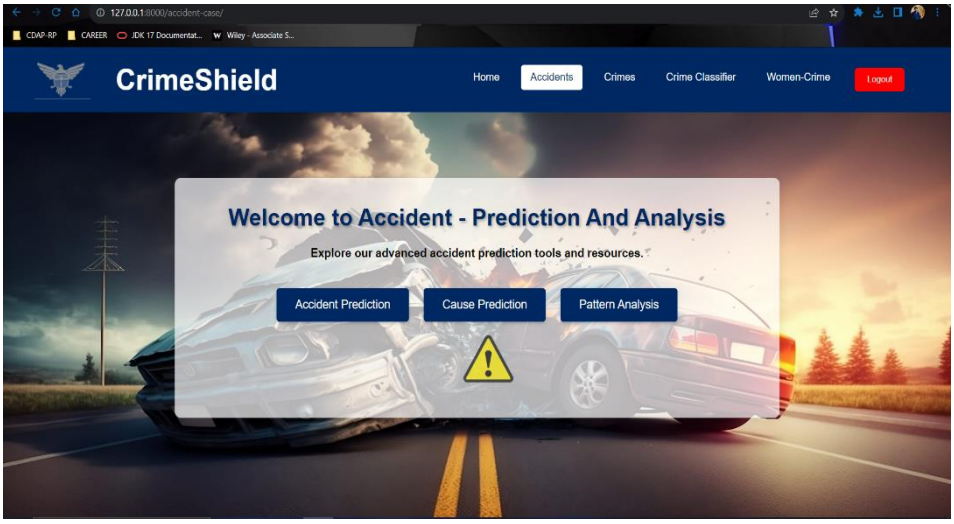


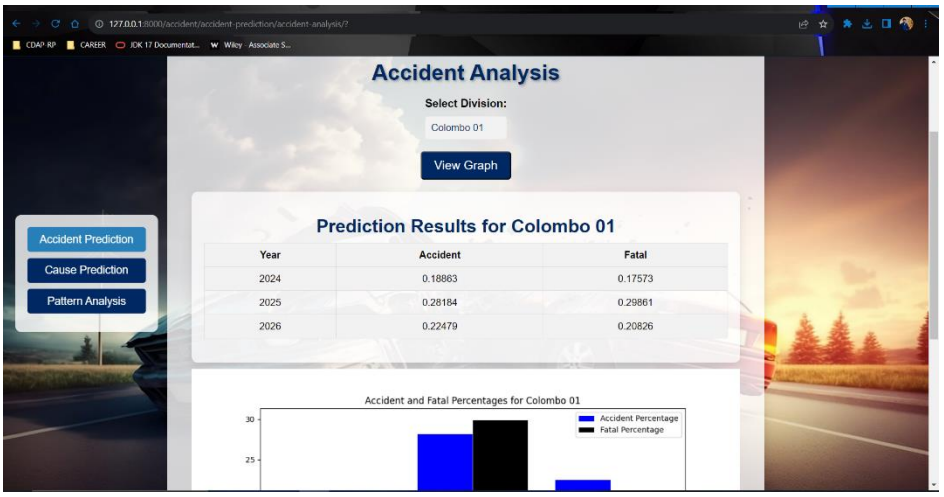
```python
from django.shortcuts import render, redirect
from .accident_Prediction_RP_FINAL import perform_prediction
from .forms import YearForm,ClusterNameForm
import matplotlib.pyplot as plt
from io import BytesIO
import base64
import pandas as pd
import matplotlib
import io
from sklearn.cluster import KMeans
from statsmodels.tsa.arima.model import ARIMA
from django.http import HttpResponse, HttpResponseNotFound
import csv



def home(request):
    return render(request, 'accidentAnalysis/home.html')

def accident_prediction(request):

    selected_division_name = None
    selected_year = None


    prediction_results = {
        'Accident_Percentage': None,
        'Fatal_Percentage': None
    }


    accident_percentage_formatted = None
    fatal_percentage_formatted = None

    if request.method == 'POST':
        selected_division_name = request.POST.get('division_name')
        selected_year = int(request.POST.get('year'))
```

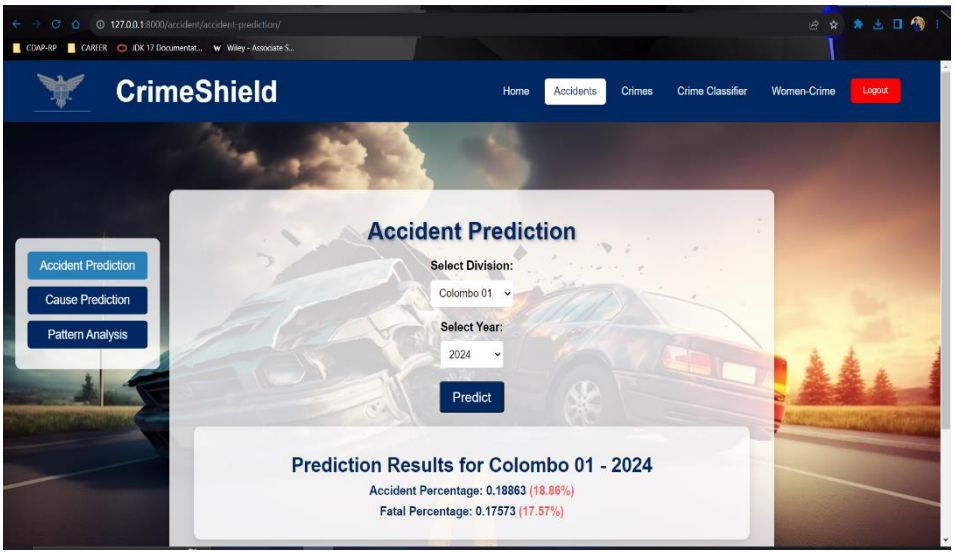


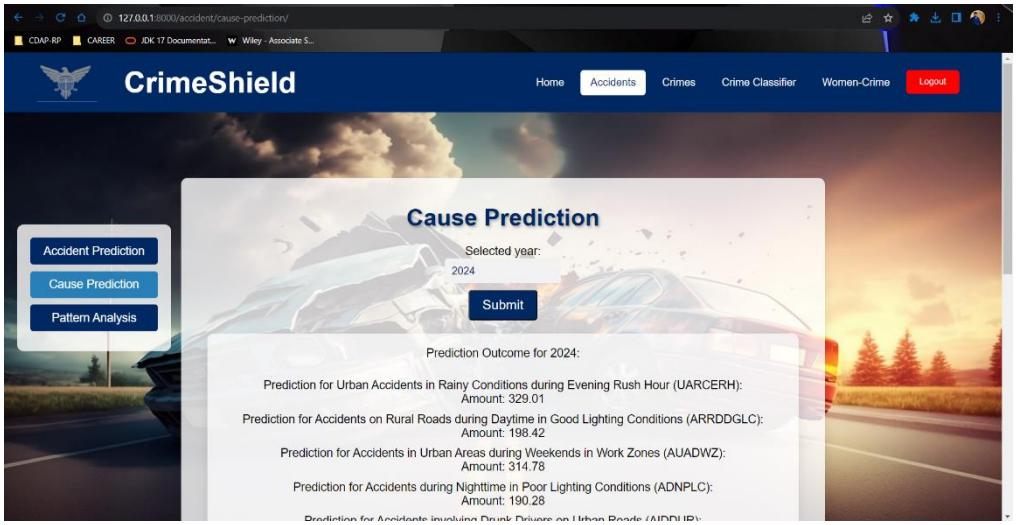
```python
    accident_percentage_formatted = None
    fatal_percentage_formatted = None

    if request.method == 'POST':
        selected_division_name = request.POST.get('division_name')
        selected_year = int(request.POST.get('year'))

        # Update the prediction results dictionary with actual values
        prediction_results = perform_prediction(selected_division_name, selected_year)

        # Calculate the formatted percentages
        accident_percentage_formatted = prediction_results['Accident_Percentage'] * 100
        fatal_percentage_formatted = prediction_results['Fatal_Percentage'] * 100

    return render(request, 'accidentAnalysis/accident_prediction.html', {
        'selected_division_name': selected_division_name,
        'selected_year': selected_year,
        'prediction_results': prediction_results,
        'accident_percentage_formatted': accident_percentage_formatted,
        'fatal_percentage_formatted': fatal_percentage_formatted,
    })

def update_dataset(request):
    if request.method == 'POST':
        division_name = request.POST.get('division_name')
        year = int(request.POST.get('year'))
        ve_total = int(request.POST.get('ve_total'))
        fatals = int(request.POST.get('fatals'))

        # Specify the path to your CSV file
        csv_file_path = 'DatasetAccident.csv'  # Update with the actual path

        # Read the CSV file into a list of dictionaries
        data = []
        with open(csv_file_path, mode='r') as csv_file:
            csv_reader = csv.DictReader(csv_file)
            for row in csv_reader:
```

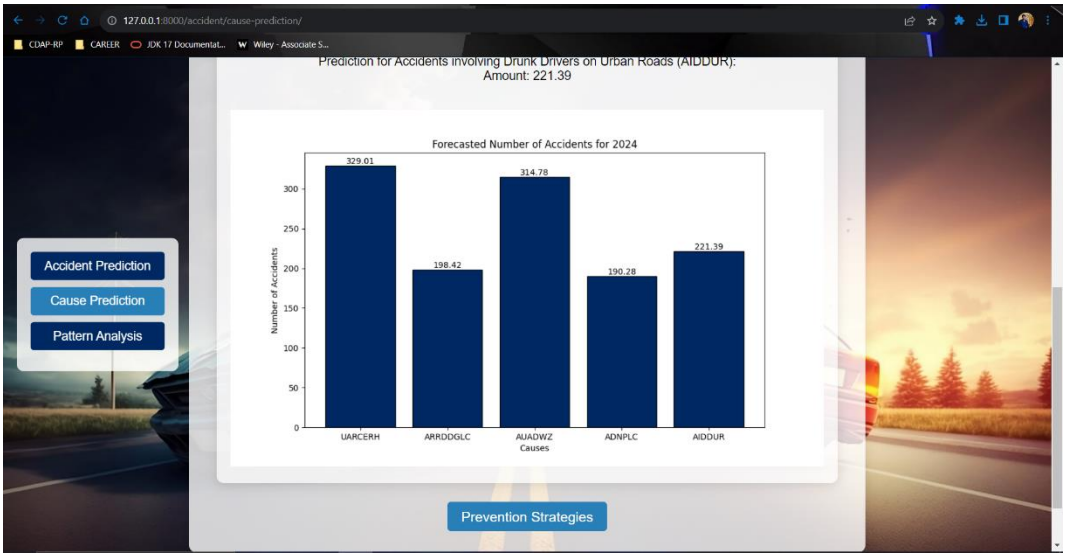

File Edit Selection View Go Run Terminal Help
policeAnalysis
EXPLORER
views.py
POLICEANALYSIS
policeAnalysis
accidentAnalysis
pycache
migrations
static
templates
init.py
accident_pattern_R...
accident_Prediction...
admin.py
apps.py
forms.py
models.py
tests.py
urls.py
views.py
crime_app
crimecase
Data
myapp
police_analysis
policeAnalysis
static
crime_data3.csv
DatasetAccident.csv
db.sqlite3
Lables.csv
manage.py
modified_file2.csv
women_crime.csv
OUTLINE
TIMELINE
policeAnalysis > accidentAnalysis > views.py > accident_prediction
def pattern_analysis(request):
# Initialize chart_base64 as an empty string
chart_base64 = ""
selected_param = "Day" # Default selected parameter
most_accident_day = None
most_accidents = None
least_accident_day = None
least_accidents = None
most_accident_hour = None
most_accidents_hour = None
least_accident_hour = None
least_accidents_hour = None
most_accident_week = None
most_accidents_week = None
least_accident_week = None
least_accidents_week = None
most_accident_month = None
most_accidents_month = None
least_accident_month = None
least_accidents_month = None
if request.method == 'POST':
# Load the accident data into a DataFrame
# accidents_df = pd.read_csv('C:\\Users\\HP\\DatasetAccident.csv')
accidents_df = pd.read_csv('DatasetAccident.csv')
# Perform descriptive statistics on DAY, MONTH, DAY_WEEKNAME, HOURNAME
day_counts = accidents_df['DAY'].value_counts().sort_index()
month_counts = accidents_df['MONTH'].value_counts().sort_index()
day_week_counts = accidents_df['DAY_WEEKNAME'].value_counts().sort_index()
hour_counts = accidents_df['HOURNAME'].value_counts().sort_index()
selected_param = request.POST['parameter']
chart_data = day_counts # Default to day_counts
Ln 23, Col 25 Spaces: 4 UTF-8 CRLF Python 3.11.4 64-bit
Type here to search
80°F Mostly cloudy
11:41 PM
9/5/2023

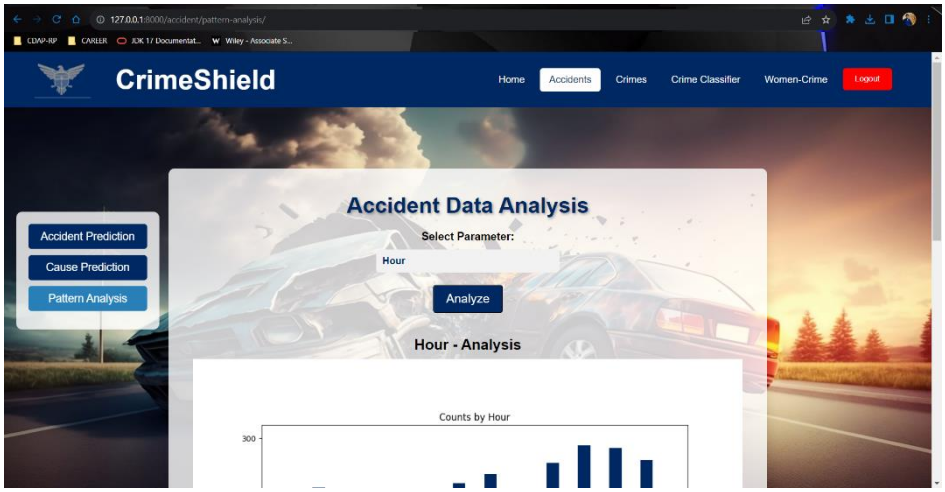

CrimeShield
Home
Accidents
Crimes
Crime Classifier
Women-Crime
Logout
Welcome to Accident - Prediction And Analysis
Explore our advanced accident prediction tools and resources.
Accident Prediction
Cause Prediction
Pattern Analysis


Accident Analysis
Select Division:
Colombo 01
View Graph
Accident Prediction
Cause Prediction
Pattern Analysis
Prediction Results for Colombo 01
Year
Accident
Fatal
2024
0.18863
0.17573
2025
0.28184
0.29861
2026
0.22479
0.20826
Accident and Fatal Percentages for Colombo 01
Accident Percentage
Fatal Percentage


CrimeShield
Home
Accidents
Crimes
Crime Classifier
Women-Crime
Logout
Accident Prediction
Cause Prediction
Pattern Analysis
Accident Prediction
Select Division:
Colombo 01
Select Year:
2024
Predict
Prediction Results for Colombo 01 - 2024
Accident Percentage: 0.18863 (18.86%)
Fatal Percentage: 0.17573 (17.57%)


CrimeShield
Home
Accidents
Crimes
Crime Classifier
Women-Crime
Logout
Accident Prediction
Cause Prediction
Pattern Analysis
Cause Prediction
Selected year:
2024
Submit
Prediction Outcome for 2024:
Prediction for Urban Accidents in Rainy Conditions during Evening Rush Hour (UARCERH):
Amount: 329.01
Prediction for Accidents on Rural Roads during Daytime in Good Lighting Conditions (ARRDDGLC):
Amount: 198.42
Prediction for Accidents in Urban Areas during Weekends in Work Zones (AUADWZ):
Amount: 314.78
Prediction for Accidents during Nighttime in Poor Lighting Conditions (ADNPLC):
Amount: 190.28


Prediction for Accidents involving Drunk Drivers on Urban Roads (AIDDUR):
Amount: 221.39
Forecasted Number of Accidents for 2024
Number of Accidents
Causes
UARCERH
ARRDDGLC
AUADWZ
ADNPLC
AIDDUR
329.01
198.42
314.78
190.28
221.39
Accident Prediction
Cause Prediction
Pattern Analysis
Prevention Strategies

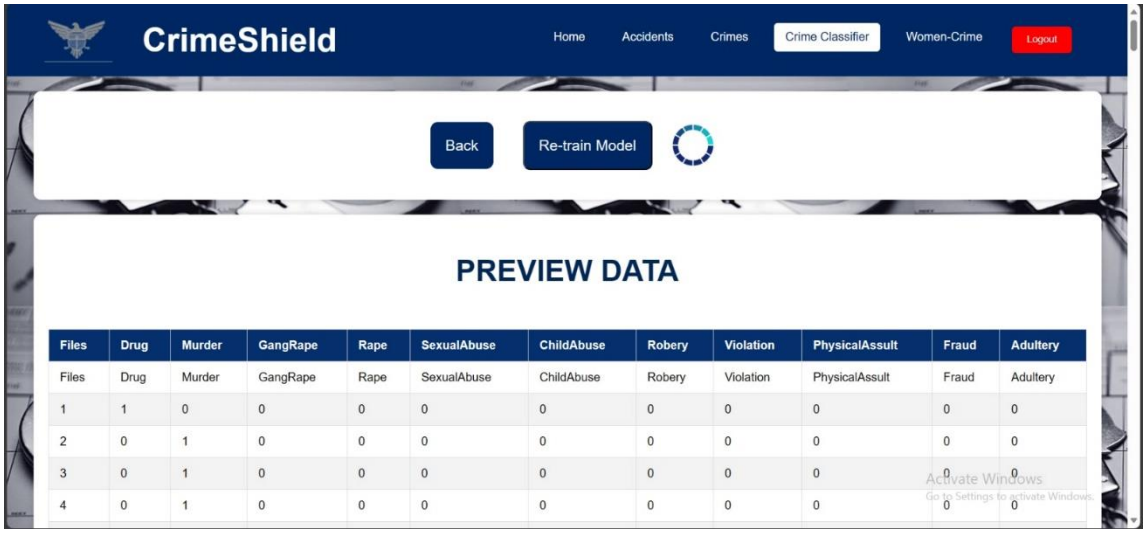

CrimeShield
Home
Accidents
Crimes
Crime Classifier
Women-Crime
Logout
Accident Prediction
Cause Prediction
Pattern Analysis
Accident Data Analysis
Select Parameter:
Hour
Analyze
Hour - Analysis
Counts by Hour

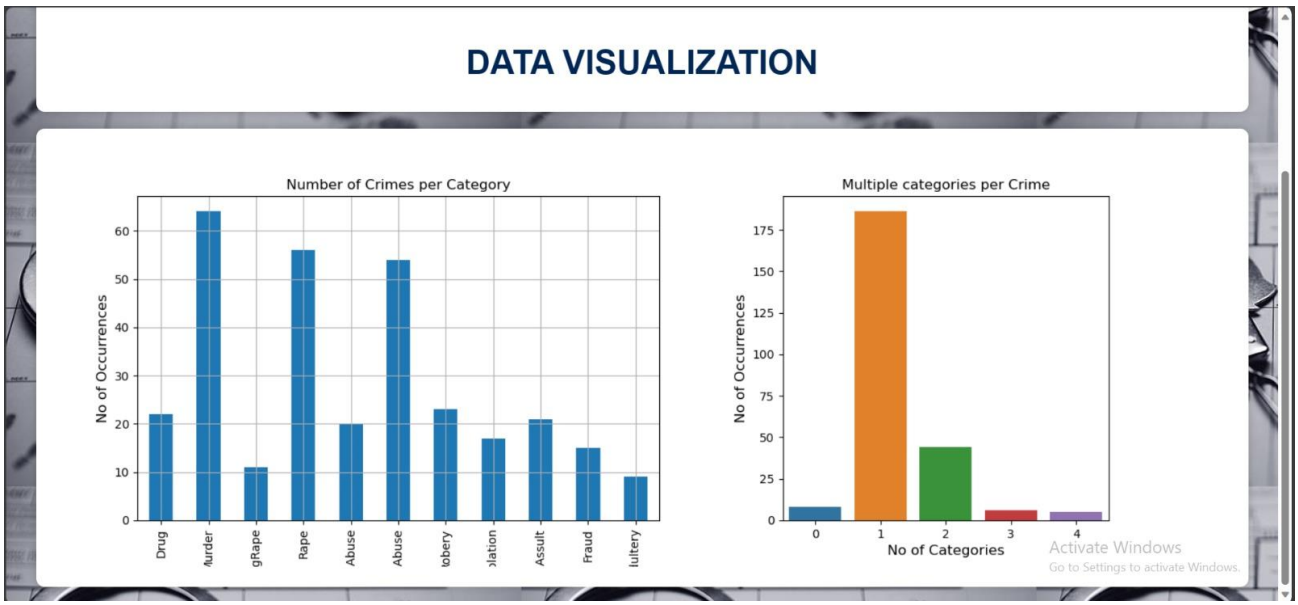

127.0.0.1:8000/accident/pattern-analysis/
CDAP-RP
CAREER
JDK 17 Documentat...
Wiley - Associate S...
Accident Prediction
Cause Prediction
Pattern Analysis
Count
150
100
50
0
Most Accident Occurring - Hour: 7:00pm-7:59pm(Accidents: 295)
Least Accident Occurring - Hour: Unknown Hours (Accidents: 27)


CrimeShield
Home
Accidents
Crimes
Crime Classifier
Women-Crime
Logout
Predict Crime Rate
For 5 Years
Crime Type:
Select Crime
Start Year:
Predict
POLICE
Back

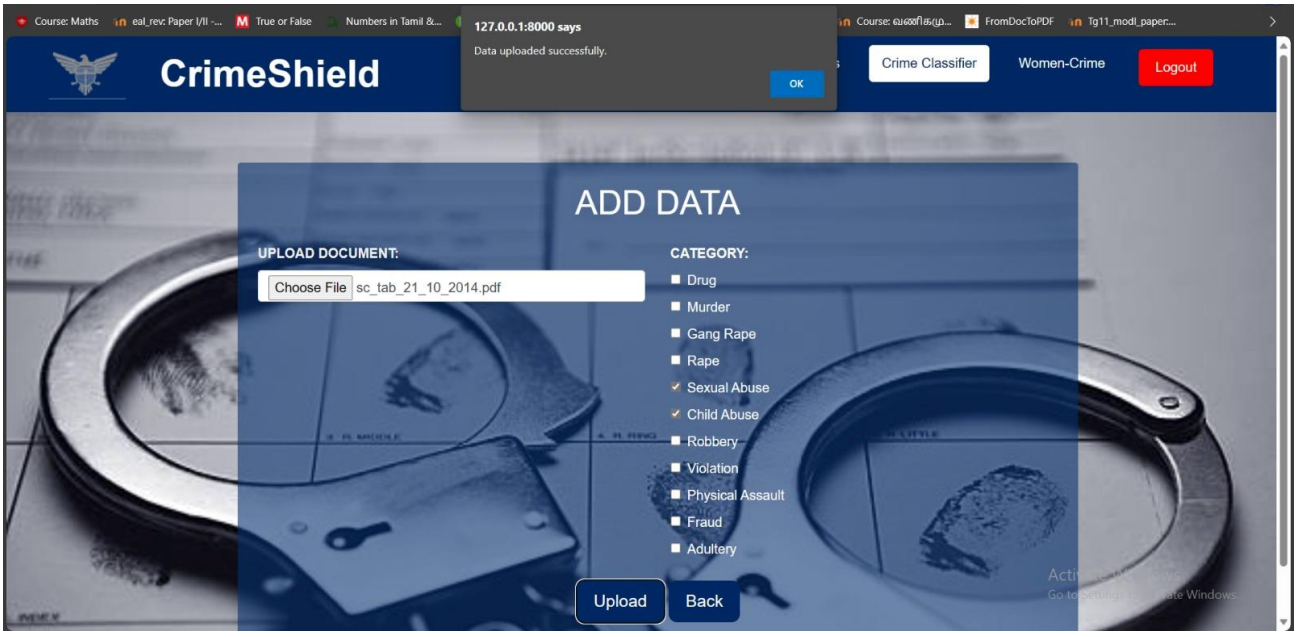

CrimeShield
Home
Accidents
Crimes
Crime Classifier
Women-Crime
Logout
Welcome to Crime Prediction
Explore our advanced crime prediction tools and resources.
Predict Crime Pattern
Predict Crime Rate
Crime Analysis
Crime Prevention Strategies
POLICE


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
Back
Re-train Model
PREVIEW DATA
Files Drug Murder GangRape Rape SexualAbuse ChildAbuse Robery Violation PhysicalAssult Fraud Adultery
1 1 0 0 0 0 0 0 0 0 0 0
2 0 1 0 0 0 0 0 0 0 0 0
3 0 1 0 0 0 0 0 0 0 0 0
4 0 1 0 0 0 0 0 0 0 0 0
Activate Windows
Go to Settings to activate Windows.


DATA VISUALIZATION
Number of Crimes per Category
No of Occurrences
0
10
20
30
40
50
60
Drug
Murder
gRape
Rape
Abuse
Abuse
Robery
Violation
Assult
Fraud
Adultery
Multiple categories per Crime
No of Occurrences
0
25
50
75
100
125
150
175
0
1
2
3
4
No of Categories
Activate Windows
Go to Settings to activate Windows.

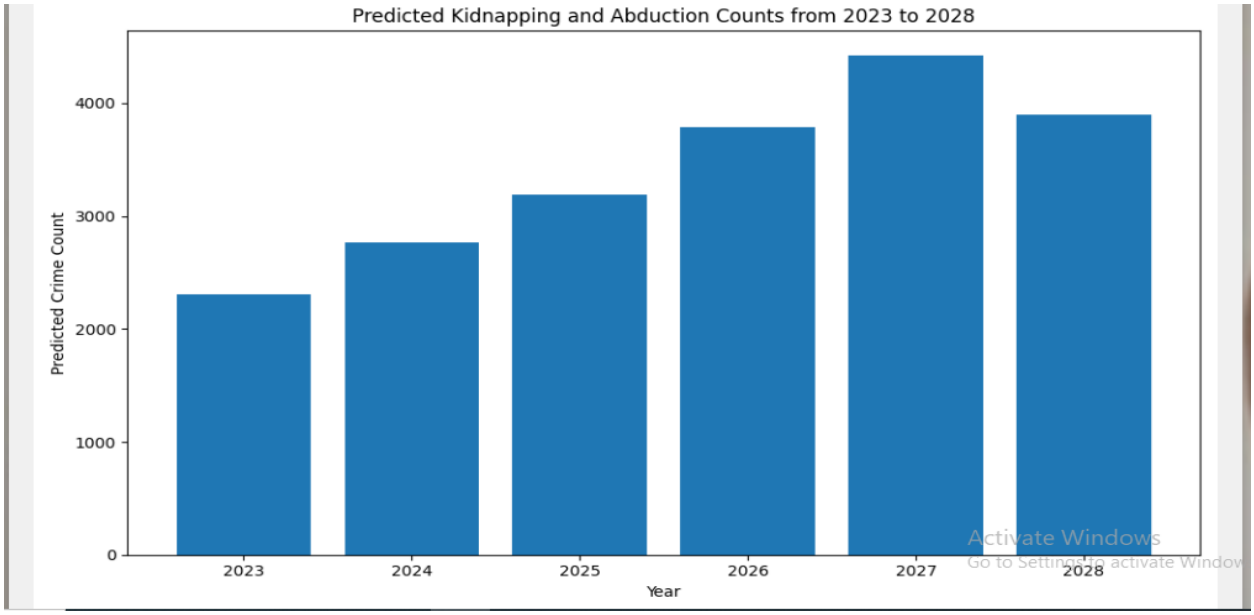

CrimeShield
Home
Accidents
Crimes
Crime Classifier
Women-Crime
Logout
CRIME CASE LABEL PREDICTION
UPLOAD DOCUMENT:
Choose File
sc_appeal_31_2016.pdf
Upload

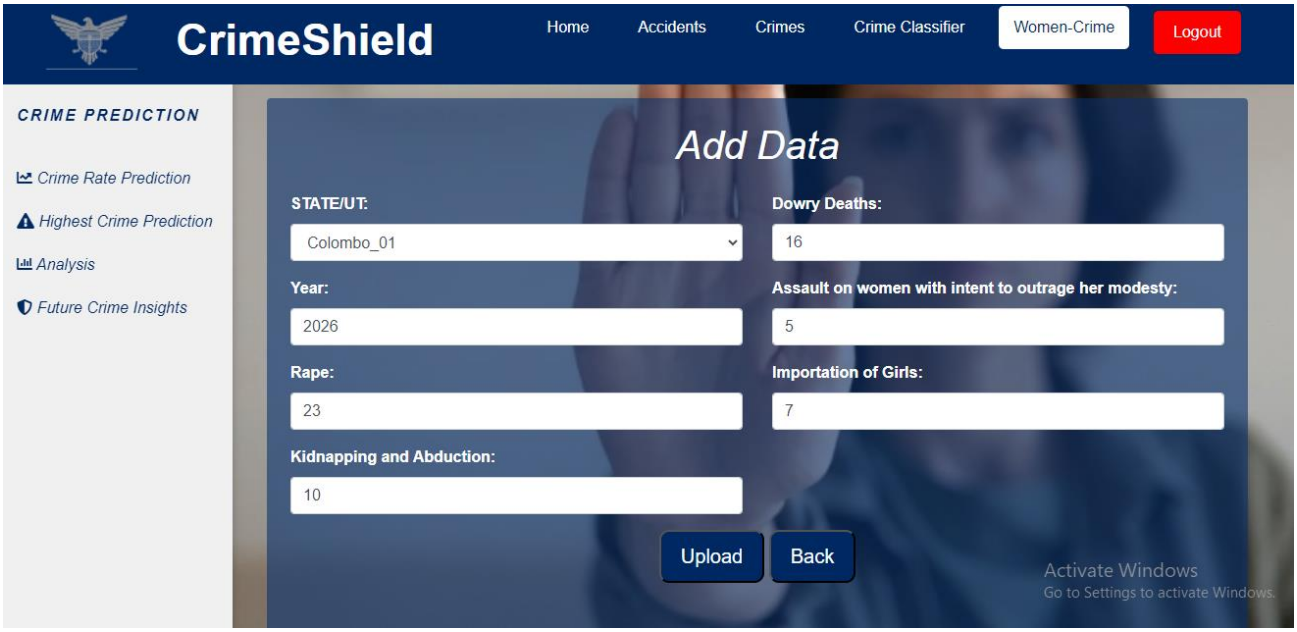

127.0.0.1:8000 says
Data uploaded successfully.
OK
CrimeShield
Crime Classifier
Women-Crime
Logout
ADD DATA
UPLOAD DOCUMENT:
Choose File
sc_tab_21_10_2014.pdf
CATEGORY:
Drug
Murder
Gang Rape
Rape
Sexual Abuse
Child Abuse
Robbery
Violation
Physical Assault
Fraud
Adultery
Upload
Back

Back

PREVIEW DATA

STATE/UT	Year	Rape	Kidnapping and Abduction	Dowry Deaths	Assault on women with intent to outrage her modesty	Importation of Girls
STATE/UT	Year	Rape	Kidnapping and Abduction	Dowry Deaths	Assault on women with intent to outrage her modesty	Importation of Girls
Colombo_01	2009	50	30	16	149	0
Colombo_01	2009	23	30	7	118	0
Colombo_01	2009	27	34	14	112	0
Colombo_01	2009	20	20	17	126	0
Colombo_01	2009	23	26	12	109	0
Colombo_01	2009	0	0	0	1	0
Colombo_01	2009	54	51	7	139	0
Colombo_01	2009	37	39	24	118	0
Colombo_01	2009	56	49	62	414	0





Highest Crime States

Year	Highest Crime State
2023	Colombo_14
2024	Colombo_03
2025	Colombo_03
2026	Colombo_04
2027	Colombo_04
2028	Colombo_01