CRIME DATA ANALYSIS AND PERPETRATOR IDENTITY PREDICTION

Submitted in partial fulfillment of the project report for the award of the degree of
Bachelor of Computer Science with Data Analytics

Submitted by
MANIARASAN J
22127028

Under the Guidance of
Mr. R. Janarthanan MCA., M.Phil., (Ph.D).,
Assistant Professor
Department of Computer Science with Cyber Security

BACHELOR OF COMPUTER SCIENCE
WITH DATA ANALYTICS

SRI RAMAKRISHNA COLLEGE OF ARTS & SCIENCE
Formerly SNR Sons College
Reaccredited with ‘A+’ grade by NAAC
Ranked 56th by NIRF 2024
NAVA INDIA, COIMBATORE – 641 006
MARCH - 2025

SRI RAMAKRISHANA COLLEGE OF ARTS & SCIENCE

BONAFIDE CERTIFICATE

Certified that this project report **“CRIME DATA ANALYSIS AND PREDICTION OF PERPETRATOR IDENTITY”** is the bonafide work of **“MANIARASAN J (22127028)”** who carried out the project work under my supervision.

SIGNATURE OF HOD

Dr. V. VIJAYKUMAR MCA., M.Phil., Ph.D.,
Controller of Examination & Head of Department,
Department of CS with Data Analytics,
Sri Ramakrishna College of Arts & Science.

SIGNATURE OF GUIDE

Mr. R. JANARTHANAN MCA., M.Phil., (Ph.D).,
Assistant Professor,
Department of CS with Cyber Security,
Sri Ramakrishna College of Arts & Science.

Submitted for the viva-voce examination held on 20.03.2025

EXTERNAL SIGNATURE

INTERNAL SIGNATURE

DECLARATION

I hereby declare that this project work entitled **"Crime Data Analytics and Prediction of Perpetrator Identity"** is submitted to Sri Ramakrishna College of Arts & Science, An Autonomous Institution, Affiliated to Bharathiar University, Coimbatore, is a record of original work done by me under the guidance of Mr. R. Janarthanan and that this project work has not formed the basis for the award of any degree / diploma / associateship/ fellowship or similar to any candidate in any university.

Place: Coimbatore

Date: Signature of the Student

Countersigned by

Mr. R. Janarthanan MCA., M.Phil., (Ph.D).,
Assistant Professor
Department of Computer Science with Cyber Security

ACKNOWLEDGEMENT

First and fore most I would like to thank my beloved Father and Mother, **Jayaraman R and Dhanalakshmi J** for their love and support for completing my studies and this project Work.

I would like to extend my heartiest thanks to **Managing Trustee, SNR Sons Charitable Trust, Thiru. R Sundar,** with a deep sense of gratitude and respect for providing me an opportunity to study in this institution.

I wish to record my deep sense of gratitude and sincere thanks to **Dr. B. L. Shivakumar, M.Sc., M.Phil., Ph.D., Principal & Secretary** of Sri Ramakrishna College of Arts & Science, Coimbatore, for his inspiration, which make me complete this mini project successfully**.**

I would like to express my sincere thanks to **Dr. V. Vijayakumar, MCA, M.Phil., Ph.D., Controller of Examination & Head of the Department of Computer Science with Data Analytics,** who gave me an opportunity to undertake such a great challenging and innovative work.

I extend my gratitude to them for their guidance, encouragement, understanding and insightful support in the development process. I would like to thank my Class Tutor**, Dr. M. Praneesh, M.Sc., M.Phil., PGDNLP, Ph.D., Assistant Professor, Department of Computer Science with Data Analytics,** for his continuous support throughout my course of period and also for motivating me to think bigger and be unique from everyone.

I would like to express my special thanks of gratitude to my guide**, Mr. R. Janarthanan, MCA., M.Phil., Ph.D., Assistant Professor, Department of Computer Science with Cyber Security,** who gave me the opportunity to do this wonderful research work which also helped me in doing lots of research and I came to know many things that I am thankful to him and even my friends.

Maniarasan J

CRIME DATA ANALYSIS AND PREDICTION OF PERPETRATOR IDENTITY

ABSTRACT

The "Crime Data Analysis and Prediction of Perpetrator Identity" project addresses the growing need for advanced technological solutions in crime prevention and investigation. Traditional methods of crime analysis often rely on manual processes that are time-intensive and lack predictive capabilities, limiting their effectiveness in identifying perpetrators. This project leverages the power of data analytics and machine learning to analyze historical crime data and predict critical attributes of perpetrators, such as age, gender, and their relationship to the victim.

The project is structured into two primary phases: Crime Data Analysis and Perpetrator Identity Prediction. In the first phase, historical crime data is preprocessed and analyzed to identify significant patterns, trends, and high-risk areas. These insights are visualized through an interactive dashboard, enabling law enforcement agencies to make informed, data-driven decisions. The second phase focuses on predictive modeling, employing advanced machine learning algorithms such as Random Forest Regressor for age prediction, Multi-Layer Perceptron (MLP) for gender classification, and XGBoost (Extreme Gradient Boosting) for relationship prediction. These models, trained on the Kaggle Homicide Reports Dataset (1980–2014), ensure robust and reliable results. The system offers an innovative and scalable approach to streamlining crime investigations and improving public safety. By integrating predictive analytics and intuitive visualizations, the project enables law enforcement agencies to reduce investigation time, allocate resources effectively, and focus on high-priority cases. Future enhancements, such as real-time data integration, advanced AI models for behavioral analysis, and surveillance system connectivity, hold the potential to further extend the system's capabilities. This project demonstrates the transformative power of technology in modernizing crime prevention and investigation strategies, contributing significantly to the goal of creating safer communities.

TABLE OF CONTENTS

CHAPTER 1
INTRODUCTION

1.1. AN OVERVIEW

Crime is one of the most pressing challenges faced by societies globally, posing threats to public safety, economic stability, and social order. The need for efficient and data-driven methods to prevent and investigate crimes has become increasingly urgent. The "Crime Data Analysis and Prediction of Perpetrator Identity" project addresses this challenge by leveraging historical crime data and advanced machine learning techniques to assist law enforcement agencies in analyzing trends and predicting key perpetrator attributes.

This project is designed to transform traditional, manual crime investigation processes into a modernized, automated system capable of handling large datasets with precision. By analyzing crime records, such as the type of crime, weapon used, victim characteristics, and geographical information, the system uncovers patterns and trends that aid in identifying high-risk areas and recurring crime types. Furthermore, predictive models are deployed to forecast critical attributes of perpetrators, including age, gender, and their relationship with victims, thereby streamlining suspect identification and investigation processes.

The project integrates a two-phase approach. In the Crime Data Analysis phase, historical datasets are cleaned, processed, and visualized to extract actionable insights. In the Prediction of Perpetrator Identity phase, advanced machine learning models such as Random Forest Regressor, Multi-Layer Perceptron (MLP), and XGBoost are employed to predict perpetrator characteristics with high accuracy. These predictions are complemented by an intuitive and interactive dashboard that provides law enforcement and policymakers with easy access to visualizations and predictive results.

By combining the power of data analytics and predictive modeling, this project represents a significant leap forward in modernizing crime investigation techniques. It enables faster decision-making, reduces manual effort, and enhances the accuracy of investigations, making it a valuable tool for law enforcement agencies worldwide.

1.2. OBJECTIVES OF THE PROJECT

The **"Crime Data Analysis and Prediction of Perpetrator Identity"** project is designed to enhance the efficiency and accuracy of crime investigations by utilizing data analytics and machine learning techniques.

The primary objectives of the project are to analyze crime data by preprocessing, cleaning, and examining large datasets of historical crime records to uncover meaningful patterns, trends, and high-risk areas that aid in understanding crime dynamics. It aims to predict perpetrator characteristics by developing machine learning models to determine key attributes such as age, gender, and their relationship with victims, assisting law enforcement in narrowing down suspect profiles and improving investigative efficiency to uncover meaningful patterns, trends, and high-risk areas that aid in understanding the dynamics of crimes. Advanced machine learning algorithms, including Random Forest Regressor for predicting perpetrator age, Multi-Layer Perceptron (MLP) for gender classification, and XGBoost (Extreme Gradient Boosting) for relationship prediction, will be leveraged to ensure accuracy and robustness through rigorous model training and validation to ensure the accuracy and robustness of predictions through rigorous model training and validation. The project also focuses on visualizing crime data by designing an intuitive and interactive dashboard for crime trends, geospatial distributions, and predictive results, enabling law enforcement and policymakers to explore data seamlessly for actionable insights. Furthermore, it enhances crime prevention by providing insights into crime-prone areas and recurring patterns, allowing proactive measures and efficient resource allocation based on data-driven analysis. By integrating predictive analytics into law enforcement workflows, the project streamlines investigations, reduces manual effort, and accelerates case resolutions while enhancing decision-making through accurate and timely predictions. To design a flexible system capable of handling larger datasets and integrating additional features or datasets in the future. Lastly, it establishes a scalable and extensible framework capable of handling larger datasets and integrating additional features, enabling future extensions such as real-time data analysis, advanced AI models for behavioral predictions, and integration with surveillance systems. By achieving these objectives, the project seeks to modernize crime investigation methods, empower law enforcement agencies with data-driven tools, and contribute to creating safer communities.

1.3. ORGANIZATON PROJECT

The **"Crime Data Analysis and Prediction of Perpetrator Identity"** project is meticulously structured into well-defined phases to ensure a systematic and efficient approach to development, testing, and deployment. Each phase focuses on specific objectives, delivering a comprehensive solution for analyzing crime data and predicting key attributes of perpetrators.

Below is a detailed explanation of the project's organization:

Phase 1: Requirement Analysis

This initial phase lays the foundation for the project by gathering and analyzing the requirements from stakeholders, including law enforcement agencies and policymakers. Key activities in this phase include:

- Identifying the features and functionalities to be implemented, such as data analysis, predictive modeling, and visualization.
- Understanding the dataset structure and determining the attributes required for analysis and prediction (e.g., crime type, weapon used, victim details, and geographical data).
- Selecting the appropriate tools, technologies, and machine learning algorithms that align with the project's objectives.

The deliverable from this phase is a comprehensive requirement specification document that defines the scope and goals of the project.

Phase 2: Data Collection and Preprocessing

In this phase, the focus is on acquiring and preparing the data for analysis and modeling. Key activities include:

- **Data Collection**: Gathering historical crime data from reliable sources, such as the Kaggle Homicide Reports Dataset (1980–2014), which contains detailed information on crime incidents, including victim and perpetrator attributes.
- **Data Cleaning**: Removing inconsistencies, handling missing values, and correcting anomalies in the dataset. For example, missing perpetrator ages are replaced with the median, and categorical values like weapon types are standardized.
- **Feature Encoding**: Converting categorical variables (e.g., crime type, weapon, gender) into numeric formats using encoding techniques to ensure compatibility with machine learning models.
- **Normalization**: Scaling numerical features to improve the performance of machine learning algorithms.

The deliverable is a preprocessed dataset ready for analysis and model training.

Phase 3: Crime Data Analysis

This phase involves conducting exploratory data analysis (EDA) to extract meaningful insights from the dataset. Key activities include:

- **Trend Analysis**: Identifying crime trends over time (e.g., seasonal patterns, annual increases or decreases in specific crime types).
- **Geospatial Analysis**: Mapping crime hotspots using geographic data to visualize high-risk areas.
- **Correlation Analysis**: Examining relationships between variables (e.g., the correlation between weapon type and crime type or between victim and perpetrator attributes).
- **Visualization**: Creating intuitive charts, graphs, and heatmaps using tools like Matplotlib, Seaborn, and Plotly to make the findings accessible and actionable.

The deliverable is a detailed analysis report supported by visualizations to aid decision-making.

Phase 4: Model Development and Training

The goal of this phase is to build machine learning models for predicting perpetrator characteristics. Key activities include:

- **Feature Selection**: Identifying and selecting the most relevant features from the dataset for training the models.
- **Algorithm Selection**: Implementing advanced machine learning algorithms:
  - **Random Forest Regressor**: For predicting the age of perpetrators based on attributes like crime type, weapon, and victim age.
  - **Multi-Layer Perceptron (MLP)**: For classifying the gender of perpetrators based on input data.
  - **XGBoost**: For predicting the relationship between the victim and the perpetrator (e.g., family, acquaintance, stranger).
- **Model Training**: Training the selected models using the preprocessed dataset and tuning hyperparameters to optimize accuracy.
- **Model Validation**: Evaluating the models using metrics such as accuracy, precision, recall, and mean squared error (for regression models).

The deliverable is a set of trained and validated machine learning models ready for integration.

Phase 5: System Integration and Dashboard Development

This phase focuses on integrating the trained models into a unified system and developing an interactive user interface. Key activities include:

- **System Integration**: Building a backend system that connects the machine learning models to the front-end application.
- **Dashboard Development**: Creating an interactive dashboard using tools like Flask or Dash to enable users to:
    - Input crime details and receive predictions for perpetrator characteristics.
    - Visualize crime data trends, geospatial mappings, and model predictions.

The deliverable is a fully functional system with a user-friendly dashboard.

Phase 6: Testing and Validation

This phase ensures the system is reliable, accurate, and user-ready. Key activities include:

- **Unit Testing**: Verifying the functionality of individual components, such as the preprocessing pipeline, prediction models, and visualization tools.
- **Integration Testing**: Ensuring seamless communication between the backend models and the dashboard interface.

The deliverable is a thoroughly tested system that meets the project's requirements.

Phase 7: Deployment and Maintenance

The final phase involves deploying the system for real-world use and planning for ongoing maintenance. Key activities include:

- **System Deployment**: Hosting the system on a cloud server or local environment for access by law enforcement agencies.
- **User Training**: Providing training sessions and documentation to help users understand and operate the system effectively.
- **Maintenance and Updates**: Establishing a maintenance plan to address bugs, incorporate new datasets, and enhance model performance over time.

The deliverable is a deployed system that is operational and scalable for future enhancements.

By following this structured approach, the project ensures each phase builds upon the previous one, delivering a robust, reliable, and user-friendly solution for crime data analysis and perpetrator prediction.

1.4. SCOPE OF THE SYSTEM

The **"Crime Data Analysis and Prediction of Perpetrator Identity"** system is designed to enhance the effectiveness of crime investigation and prevention by leveraging data analytics and machine learning techniques. It aims to bridge the gap between traditional manual processes and modern technology-driven solutions by providing advanced tools for analyzing crime data and predicting perpetrator attributes. Below is an elaboration on the system's scope:

i. Crime Data Analysis

The system focuses on analyzing historical crime data to derive valuable insights that aid in understanding crime patterns and dynamics. Key functionalities include:

- **Crime Trends Analysis**: The system identifies recurring patterns and temporal trends, such as specific months, seasons, or years with higher crime rates. This helps law enforcement agencies anticipate and prepare for potential surges in criminal activities.
- **Crime Distribution Analysis**: The system categorizes crimes based on attributes such as crime type, weapon used, victim characteristics, and locations, providing a holistic view of the nature and distribution of crimes.

ii. Prediction of Perpetrator Characteristics

The system's predictive capabilities form a core part of its functionality, helping law enforcement narrow down potential suspects based on key attributes:

- **Age Prediction**: Using a Random Forest Regressor, the system predicts the age of perpetrators based on crime details, such as weapon used, victim age, and crime type.
- **Gender Classification**: A Multi-Layer Perceptron (MLP) model classifies the perpetrator's gender with high accuracy, using features derived from the dataset.
- **Relationship Prediction**: Leveraging the XGBoost algorithm, the system predicts the relationship between the perpetrator and the victim (e.g., family, acquaintance, or stranger), offering deeper insights into crime dynamics.

These predictive models streamline investigations by reducing the time and effort required to identify suspects, enabling law enforcement to focus their efforts on the most relevant leads.

iii. Interactive Visualizations

To make the insights actionable and accessible, the system incorporates an interactive dashboard, providing:

- **Crime Trends Visualization**: Line charts, bar graphs, and pie charts represent crime trends and distributions over time.
- **Geospatial Crime Mapping**: Heatmaps display crime-prone areas, highlighting regions with higher incidences of specific crime types.
- **Predictive Model Outputs**: The dashboard showcases predicted perpetrator attributes (e.g., age, gender, and relationship), allowing investigators to quickly assess the most likely characteristics of suspects.

This visualization capability makes complex data easy to interpret, empowering both technical and non-technical users to explore and utilize insights effectively.

iv. Support for Law Enforcement and Policymakers

The system serves as a vital tool for both law enforcement agencies and policymakers:

- **For Law Enforcement Agencies**:
  - Accelerates investigations by providing predictive insights into potential suspects.
  - Assists in identifying high-risk areas, enabling primitive action to prevent crimes.
  - Reduces dependency on manual processes, saving time and resources.
- **For Policymakers**:
  - Provides data-driven insights to formulate crime prevention policies.
  - Aids in resource planning and allocation, ensuring law enforcement efforts are targeted toward critical areas.
  - Offers insights into crime trends that can inform community safety initiatives and legislative actions.

v. Scalability and Extensibility

The system is designed to be scalable and flexible, allowing for future enhancements and expanded use cases:

- **Integration with Larger Datasets**: The system can process and analyze larger datasets, enabling its application in broader geographical regions or across multiple jurisdictions.
- **Advanced AI Models**: The system can evolve to include deep learning models for behavioral analysis, psychological profiling, and more complex crime predictions.

vi. Limitations and Boundaries

While the system offers significant advancements, it has certain limitations:

- **Dependence on Data Quality**: The accuracy of predictions depends on the completeness and quality of the input data. Missing or inconsistent data can affect the system's performance.
- **Ethical and Privacy Considerations**: Ensuring the system complies with data privacy regulations and addresses potential biases in predictions is crucial for its ethical implementation.
- **Geographical Boundaries**: Initially, the system is designed for datasets from specific regions (e.g., U.S. Homicide Reports). Expanding its scope to other regions may require additional preprocessing and customization.

Key Deliverables

The scope of the system ensures the following deliverables:

- **Comprehensive Crime Analysis**: Detailed insights into crime trends, patterns, and geographical distributions.
- **Predictive Models**: Reliable predictions of perpetrator attributes (age, gender, and relationship) to aid investigations.
- **Interactive Dashboard**: A user-friendly interface for visualizing and interpreting crime data and predictions.
- **Scalability**: A robust framework capable of evolving to address larger datasets, real-time data, and future technological advancements.

By addressing these aspects, the **"Crime Data Analysis and Prediction of Perpetrator Identity"** system significantly enhances the ability of law enforcement agencies and policymakers to analyze crime data, prevent criminal activities, and resolve cases more efficiently. Its scalable and extensible design ensures that it remains relevant and adaptable to future challenges in crime prevention and public safety.

CHAPTER 2
SYSTEM ANALYSIS

2.1. EXISTING SYSTEM

The current approaches to crime investigation rely on traditional methods, which are often manual, time-consuming, and prone to errors. While these methods have served as the foundation of law enforcement for decades, they have significant limitations in handling the increasing complexity and scale of modern-day crimes. Below is an in-depth analysis of the existing system and its shortcomings:

i. Manual Data Analysis

One of the most significant drawbacks of the current system is its dependency on manual processes for recording, analyzing, and interpreting crime data.

- **Time-Consuming Processes:** Analyzing historical crime records or cross-referencing them with ongoing investigations is a labor-intensive process, often causing delays in case resolution.
- **Limited Scalability:** As crime data grows in volume, manual processes become increasingly inefficient, unable to scale or process the data effectively.

ii. Limited Analytical Capabilities

The existing system provides only basic functionalities for analyzing crime data, which severely limits its usefulness in deriving actionable insights.

- **Basic Data Handling:** Most existing systems use simple databases or files to store crime data, without incorporating advanced tools for in-depth analysis.
- **Minimal Insights:** These systems fail to uncover trends, correlations, or patterns in the data, restricting law enforcement agencies to reactive rather than proactive crime prevention strategies.

iii. Inefficiency in Identifying Perpetrators

The current methods for identifying perpetrators rely heavily on physical evidence, eyewitness accounts, and manual investigation, which are both time-consuming and error-prone.

- **Slow Investigative Processes:** Investigations are often delayed due to the lack of automated tools to process and analyze evidence or suspect profiles.

- **No Automated Predictions:** Without predictive models, law enforcement agencies struggle to identify attributes such as the perpetrator’s age, gender, or relationship with the victim, resulting in limited leads and prolonged investigations.

iv. Minimal Use of Visualization Tools

The absence of sophisticated visualization tools in the existing system limits its ability to convey actionable insights to users effectively.

- **Limited Access to Visual Insights:** Current systems rely heavily on textual data and static reports, which fail to provide a clear picture of crime trends, patterns, or hotspots.
- **Reduced Decision-Making Efficiency:** Without intuitive visual representations such as heatmaps, graphs, or charts, it becomes challenging for law enforcement officers to prioritize resources or devise effective crime prevention strategies.

Limitations of the Existing System

The existing system has several key limitations that make it inadequate for addressing modern crime investigation challenges:

- **Lack of Automation:** Manual processes dominate the workflow, leading to inefficiencies and delays.
- **Inability to Scale:** The current system cannot handle the growing volume of crime data or the complexity of modern-day crimes.
- **Absence of Predictive Analytics:** There are no tools for anticipating criminal behavior or predicting likely perpetrator attributes.
- **No Real-Time Capabilities:** The system lacks the ability to process and analyze live data, which is crucial for timely interventions.

Impact of Existing System Shortcomings

- Investigations take longer to complete, which can result in delayed justice for victims.
- Law enforcement agencies are unable to take proactive measures to prevent crimes or allocate resources effectively.
- Important patterns and trends in crime data remain undiscovered, leading to missed opportunities for intervention.

2.2. PROPOSED SYSTEM

The **"Crime Data Analysis and Prediction of Perpetrator Identity"** system is designed to address the limitations of the existing methods by leveraging advanced data analytics, machine learning, and visualization tools. It introduces automation and predictive capabilities to streamline crime investigations, enabling law enforcement agencies to analyze large datasets, uncover patterns, and make data-driven decisions effectively. Below is a detailed explanation of the proposed system and its key features:

i. Automation of Crime Data Analysis

The proposed system automates the process of analyzing crime data, eliminating the inefficiencies of manual methods.

- **Data Preprocessing**: The system cleans, normalizes, and encodes historical crime data to ensure consistency and accuracy.
- **Trend and Pattern Identification**: It identifies recurring crime trends, high-risk areas, and correlations between crime attributes, such as weapon type, victim age, and perpetrator details.
- **Visualization**: Advanced visualization tools like heatmaps, bar charts, and line graphs provide a clear and intuitive understanding of crime data, making insights easily interpretable.

ii. Predictive Analytics for Perpetrator Characteristics

The system incorporates machine learning algorithms to predict critical attributes of perpetrators based on crime details.

- **Age Prediction**: A Random Forest Regressor is used to predict the likely age of the perpetrator based on features such as weapon type, victim characteristics, and crime location.
- **Gender Classification**: A Multi-Layer Perceptron (MLP) model predicts the perpetrator's gender, ensuring high accuracy and reliability.
- **Relationship Prediction**: The XGBoost algorithm predicts the relationship between the victim and perpetrator, categorizing it as family, acquaintance, or stranger.

These predictive models enable law enforcement agencies to narrow down suspect profiles, improving investigation speed and accuracy.

iii. Interactive Dashboard

The proposed system features an interactive and user-friendly dashboard that allows users to:

- **Input Crime Details**: Users can provide details such as crime type, weapon used, victim demographics, and location to receive predictions.
- **Visualize Insights**: The dashboard provides geospatial maps, charts, and graphs to visualize crime data trends and prediction results.

The dashboard ensures that both technical and non-technical users can easily interact with the system and derive actionable insights.

iv. Data Integration and Scalability

The system is designed to consolidate data from multiple sources and handle large volumes of information efficiently.

- **Unified Data Platform**: The system integrates crime reports, surveillance data, and demographic information into a centralized database, ensuring seamless analysis.
- **Scalability**: It is built to process larger datasets, support real-time data integration, and accommodate additional features such as behavioral profiling and advanced AI models in the future.

v. Proactive Crime Prevention

The proposed system helps law enforcement agencies shift from reactive to proactive crime prevention strategies by:

- **Identifying High-Risk Areas**: By analyzing historical crime data, the system highlights regions with higher crime rates, enabling targeted patrolling and resource allocation.
- **Anticipating Crime Trends**: The system provides insights into seasonal and geographical crime patterns, helping law enforcement anticipate and prevent future incidents.

vi. Real-Time Decision Support

The system is designed to support real-time decision-making by:

- **Providing Immediate Predictions**: Based on user inputs, the system instantly predicts perpetrator attributes, enabling faster investigations.
- **Updating Trends Dynamically**: With real-time data integration, the system can continuously update crime trends and predictions, ensuring up-to-date insights.

vii. Addressing Ethical and Privacy Concerns

The proposed system incorporates measures to address ethical and privacy considerations:

- **Data Security**: Ensures all sensitive data is stored and processed securely to prevent unauthorized access.
- **Bias Mitigation**: Employs techniques to eliminate biases in machine learning models, ensuring fair and unbiased predictions.

Key Features of the Proposed System

The proposed system is designed with the following features:

- **Automated Crime Data Analysis**: Faster and more accurate analysis of historical crime data.
- **Predictive Modeling**: Reliable predictions of perpetrator attributes, such as age, gender, and relationship to the victim.
- **Interactive Dashboard**: A user-friendly interface for data exploration and visualization.
- **Proactive Decision-Making**: Enables law enforcement agencies to anticipate and prevent crimes effectively.

Advantages of the Proposed System

- **Enhanced Accuracy**: The use of advanced machine learning models ensures more accurate predictions compared to manual methods.
- **Faster Investigations**: Automation and predictive capabilities significantly reduce the time required for investigations.
- **Actionable Insights**: Data visualization and hotspot analysis provide actionable insights for resource allocation and crime prevention.

- **Future-Ready**: The system is designed to evolve with technological advancements, including**.**

2.3. HARDWARE SPECIFICATIONS

The **"Crime Data Analysis and Prediction of Perpetrator Identity"** system requires specific hardware components to ensure efficient data processing, model training, and system performance.

i. Development Environment

For building and training machine learning models, the development environment requires the following hardware:

- **Processor**: Intel Core i3, with a minimum clock speed of 2.6 GHz for handling high computational loads during model training.
- **RAM**: 16 GB or higher to enable smooth handling of large datasets and efficient execution of algorithms.
- **Storage**:
  - **Primary Storage (SSD)**: Minimum 512 GB SSD for faster data access and model storage.
  - **Secondary Storage (HDD)**: 1 TB HDD or more for storing raw datasets, logs, and backup files.
- **Operating System**: Windows 10 (64-bit) or Linux-based distributions like Ubuntu 20.04 for flexibility and compatibility with required tools and libraries.

ii. Deployment Environment

For deploying the system and providing real-time predictions, the deployment environment requires the following:

- **Processor**: Intel Core i5 (8th Generation) or equivalent, capable of handling multi-threaded operations.
- **RAM**: 8 GB or higher for smooth operation of the web application and dashboard.
- **Storage**:
  - **Primary Storage (SSD)**: Minimum 256 GB SSD for storing processed datasets and deployed models.
  - **Cloud Storage**: Integration with cloud platforms for scalability

- **Network Connectivity**: A high-speed internet connection (minimum 50 Mbps) for accessing real-time data and enabling remote collaboration.

iii. User Environment

End users, such as law enforcement officers and policymakers, require the following hardware to access and interact with the system:

- **Device**: Desktop, laptop with modern browser support (e.g., Chrome, Firefox).
- **Processor**: Intel Core i3 or higher.
- **RAM**: 4 GB or higher to run web-based dashboards and applications.
- **Storage**: Minimum 128 GB for temporary caching and storing local data reports if required.
- **Display**: Full HD (1920x1080 resolution) or higher for clear visualization of data and geospatial maps.

2.4. SOFTWARE SPECIFICATIONS

- **Programming** : Python, HTML, CSS, and JavaScript
- **Library Analysis** : NumPy, Pandas, Scikit-learn, XGBoost, Joblib, Folium
- **Data Visualization** : Matplotlib, Seaborn, Plotly
- **Model Deployment** : Flask
- **Database** : MySQL
- **Integrated Development Environment (IDE)** : PyCharm or VS Code for Python development, debugging, and code editing.

CHAPTER 3
DESIGN AND DEVELOPMENT

3.1. DESIGN PROCESS

The design process for the **"Crime Data Analysis and Prediction of Perpetrator Identity"** system ensures that the system's components are structured to handle data efficiently, process inputs accurately, and deliver meaningful outputs. Below are the detailed subcomponents of the design process:

3.1.1. DATABASE DESIGN

The database design focuses on organizing the data in a structured manner for efficient storage, retrieval, and processing.

Key Objectives:

- To store raw crime data and processed datasets in a relational or document-based format.
- To maintain tables/entities for crime records, victim details, perpetrator attributes, and relationships between them.
- To ensure scalability and support for complex queries during analysis and prediction.

Database Schema:

The database consists of the following main entities:

i. **Crime Details**:
   - Attributes: Crime ID, Crime Type, Location, Date, Weapon Used, Crime Solved (Yes/No).

ii. **Victim Details**:
   - Attributes: Victim ID, Victim Gender, Victim Age, Victim Ethnicity, Victim Count.

iii. **Perpetrator Details**:
   - Attributes: Perpetrator ID, Perpetrator Gender, Perpetrator Age, Perpetrator Relationship.

iv. **Geographical Data**:

  - Attributes: State, City, Zip Code, Region.

v. **Storage Technology**:

  - **Relational Database**: MySQL or PostgreSQL for structured and relational data.
  - **NoSQL Database**: MongoDB for semi-structured data and scalability.

3.1.2. INPUT DESIGN

The input design defines how data is collected and entered into the system to ensure accuracy and consistency.

Key Objectives:

- To enable users to input crime details, victim attributes, and other relevant data through an interactive interface.
- To validate input data to ensure completeness and accuracy.
- To process both historical crime datasets (e.g., Kaggle) and real-time user-provided inputs.

Input Sources:

i. **Crime Dataset**: Historical crime data from platforms like Kaggle (in CSV format).

ii. **User Inputs**: Data entered through the web interface, including:

  - Crime type (dropdown menu).
  - Weapon used (dropdown menu).
  - Victim attributes such as age, gender, and count (numerical inputs).
  - Crime location (state, city).

Validation:

- Drop-down menus for categorical data to prevent invalid inputs.
- Numerical range validation for fields like age and victim count.
- Mandatory fields to ensure all critical details are provided.

Input Interface:

- User-friendly forms on a web-based dashboard using HTML, CSS, and JavaScript frameworks like Bootstrap.

3.1.3. OUTPUT DESIGN

The output design focuses on how the system presents results and predictions to users in an intuitive and actionable format.

Key Objectives:

- To display the results of data analysis and model predictions, such as perpetrator age, gender, and relationship.
- To visualize trends, patterns, and crime hotspots through graphs and interactive maps.
- To ensure outputs are easy to interpret for both technical and non-technical users.

Outputs Provided by the System:

i. **Predictions**:
    - Predicted perpetrator age using Random Forest Regressor.
    - Predicted perpetrator gender using MLP (Multi-Layer Perceptron).
    - Predicted victim-perpetrator relationship using XGBoost.

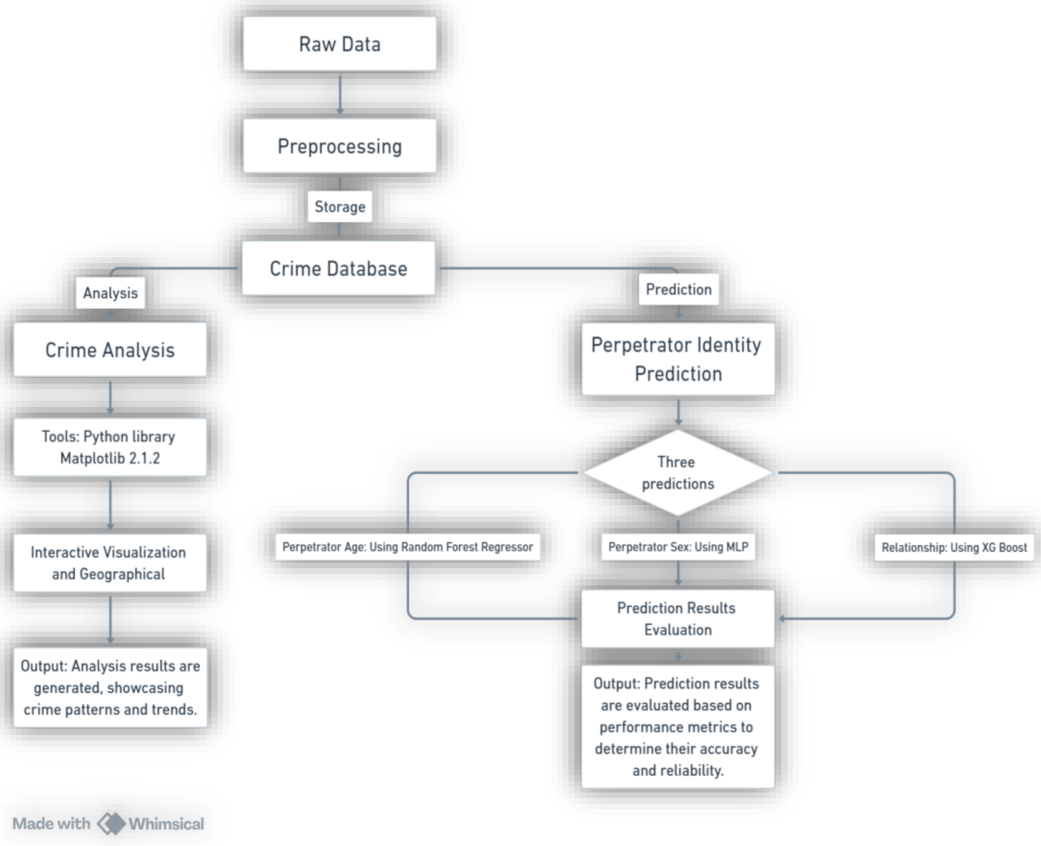
ii. **Crime Analysis Insights**:
    - Crime trends over time displayed using line graphs.
    - Crime distribution by location using bar charts and heatmaps.
    - Hotspot maps highlighting high-crime areas using geospatial tools.

Output Interface:

- **Dashboard Design**:
    - Developed using HTML, CSS, and JavaScript libraries such as Plotly and Folium.
    - Interactive charts for exploring data trends and patterns.
    - Dynamic maps for visualizing geospatial data.

Output Formats:

- **Web Dashboard**: Interactive visualizations and prediction results.
- **Downloadable Reports**: CSV or PDF formats summarizing crime trends and predictions.


Raw Data
Preprocessing
Storage
Crime Database
Analysis
Prediction
Crime Analysis
Perpetrator Identity Prediction
Tools: Python library Matplotlib 2.1.2
Three predictions
Interactive Visualization and Geographical
Perpetrator Age: Using Random Forest Regressor
Perpetrator Sex: Using MLP
Relationship: Using XG Boost
Prediction Results Evaluation
Output: Analysis results are generated, showcasing crime patterns and trends.
Output: Prediction results are evaluated based on performance metrics to determine their accuracy and reliability.
Made with Whimsical

Conclusion

The **Database Design**, **Input Design**, and **Output Design** are integral parts of the design process, ensuring efficient data management, accurate data collection, and meaningful presentation of results. Together, these components enable the system to deliver actionable insights and predictions that aid law enforcement in improving crime prevention and investigation efforts.

CHAPTER 4
TESTING AND IMPLEMENTATION

4.1. System Testing

System testing is a critical phase in the development of the **"Crime Data Analysis and Prediction of Perpetrator Identity"** system. It ensures that all components of the system function as intended and meet the specified requirements. The primary goal is to validate the system's accuracy, reliability, and usability before deployment.

Types of Testing Performed

i. **Unit Testing**

- **Objective**: Verify the correctness of individual components, such as machine learning models, preprocessing scripts, and API endpoints.
- **Testing Areas**:
  - Data preprocessing: Ensure handling of missing values, encoding of categorical variables, and normalization.
  - Machine learning models: Validate that models (Random Forest, MLP, XGBoost) produce predictions within expected ranges.
  - API endpoints: Test input parsing, response formats, and error handling.
- **Tools Used**:
  - Pytest for testing Python modules.
  - Mock data for simulating various input scenarios.

ii. **Integration Testing**

- **Objective**: Validate that different components of the system work together seamlessly.
- **Testing Areas**:
  - Integration between the database and machine learning models: Ensure the models can retrieve preprocessed data and store results correctly.
  - Real-time input handling: Validate that user inputs from the dashboard are processed and predictions are displayed correctly.
- **Tools Used**:
  - Postman for testing API calls.
  - Selenium for frontend-backend integration testing.

iii. Functional Testing

- **Objective**: Ensure that all functionalities meet the specified requirements.
- **Testing Areas**:
  - Data input validation: Check that all required fields are completed and invalid inputs are handled appropriately.
  - Prediction accuracy: Validate that the system provides correct and meaningful predictions for age, gender, and relationship.
  - Visualization: Test the accuracy and responsiveness of visualizations, such as crime trends and hotspot maps.
- **Tools Used**: Manual testing with predefined test cases.

iv. Performance Testing

- **Objective**: Assess the system's performance under varying loads and ensure it meets response time requirements.
- **Testing Areas**:
  - Model performance: Measure the time taken by machine learning models to process data and provide predictions.
  - Dashboard performance: Ensure the frontend can handle large datasets and display visualizations without delays.
- **Tools Used**:
  - JMeter for API performance testing.
  - Profiling tools like cProfile for identifying bottlenecks in Python code.

v. User Acceptance Testing (UAT)

- **Objective**: Validate the system's usability and functionality from the end-user perspective.
- **Testing Areas**:
  - Ease of use: Ensure the dashboard interface is intuitive and accessible.
  - Predictions: Verify that the outputs are accurate and presented in an understandable format.
  - Feedback handling: Collect user feedback to identify areas for improvement.
- **Methodology**: Involve law enforcement officers and other stakeholders in testing the system and gathering their feedback.

Outcome of Quality Assurance

- The system meets all functional and non-functional requirements, ensuring accuracy, reliability, and usability.
- A robust framework for crime data analysis and prediction, with high user satisfaction.
- A scalable and maintainable system ready for future enhancements.

By implementing a comprehensive QA strategy, the **"Crime Data Analysis and Prediction of Perpetrator Identity"** system achieves high standards of performance, reliability, and user experience, ensuring its success in addressing modern crime investigation challenges.

4.3. System Implementation

The implementation of the **"Crime Data Analysis and Prediction of Perpetrator Identity"** system involves the integration of various components, ensuring the system operates seamlessly from data input to prediction and visualization. This phase focuses on deploying the system in a production environment, ensuring usability, scalability, and reliability.

Steps in System Implementation

i. Preparation for Deployment

- **Objective**: Prepare the environment for deploying the system and its components.
- **Activities**:
  - Install required software, libraries, and frameworks such as Python, Flask, scikit-learn, XGBoost, and Bootstrap.
  - Set up the database (MySQL or MongoDB) for storing and retrieving processed crime data.

ii. Integration of System Components

- **Objective**: Ensure all modules and components of the system work together cohesively.
- **Activities**:
  - **Database Integration**: Connect the backend to the database for storing and fetching crime data.
  - **Model Integration**: Load the trained machine learning models (e.g., Random Forest, MLP, XGBoost) using Joblib for real-time predictions.
  - **Frontend-Backend Integration**: Link the web-based dashboard to the backend through API endpoints.

iii. Deployment

- **Objective**: Deploy the system on a production environment to make it accessible to users.
- **Activities**:
    - **Local Deployment**: Test the system locally to ensure all components function as intended.
    - **Containerization**: Use Docker to package the application and its dependencies into containers, ensuring portability and consistency.
    - **Web Server Configuration**: Set up NGINX or Gunicorn for hosting the Flask application and managing user requests.
    - **Cloud Deployment** (Optional): Deploy the system on cloud platforms like AWS, Azure, or Google Cloud for scalability and remote access.

iv. User Training and Documentation

- **Objective**: Ensure end-users can effectively use the system and understand its features.
- **Activities**:
    - Conduct training sessions for law enforcement personnel and other stakeholders.
    - Provide a user manual and system documentation detailing system functionalities, inputs, and outputs.

v. Testing in the Live Environment

- **Objective**: Validate the system's performance, reliability, and accuracy in the production environment.
- **Activities**:
    - Conduct performance tests to evaluate response times and system efficiency.
    - Verify the accuracy of predictions with real-world data inputs.
    - Test the user interface for accessibility and usability.

vi. Monitoring and Maintenance

- **Objective**: Ensure the system remains operational and addresses any issues that arise post-deployment.

- **Activities**:
    - Monitor the system using tools like Prometheus and Grafana for tracking performance and uptime.
    - Regularly update the system to incorporate new features or datasets.

By following a structured implementation process, the system is successfully deployed to achieve its goals of modernizing crime analysis and assisting in accurate perpetrator identification.

4.4. System Maintenance

System maintenance is a crucial phase in the lifecycle of the **"Crime Data Analysis and Prediction of Perpetrator Identity"** system. It ensures the system operates efficiently, remains up-to-date, and adapts to changing requirements or advancements in technology. Proper maintenance enhances system reliability, scalability, and user satisfaction over time.

Objectives of System Maintenance

i. **Ensure System Reliability**:
- Address issues such as bugs, errors, or crashes to maintain uninterrupted operation.

ii. **Adapt to Changing Requirements**:
- Modify or upgrade the system to accommodate new datasets, features, or user needs.

iii. **Enhance Performance**:
- Optimize the system to improve response times, scalability, and efficiency under varying loads.

iv. **Ensure Security**:
- Regularly update security protocols to protect sensitive crime data.

Conclusion

Effective system maintenance ensures the **"Crime Data Analysis and Prediction of Perpetrator Identity"** system remains operational, secure, and scalable over time. By addressing bugs, incorporating updates, and optimizing performance, the system. A structured maintenance strategy is essential for maintaining long-term reliability and achieving the system’s objectives of enhancing crime analysis and investigation.

CHAPTER 5
CONCLUSION

The **"Crime Data Analysis and Prediction of Perpetrator Identity"** system marks a significant step forward in modernizing crime analysis and investigation. By leveraging advanced data analytics and machine learning, the system addresses the limitations of traditional methods, offering predictive insights and actionable data for law enforcement agencies. The potential for future development is vast, from real-time integrations and advanced models to global scalability and community involvement. With continued innovation and refinement, this system can evolve into a comprehensive platform that not only aids investigations but also contributes to proactive crime prevention and public safety enhancement. Through its current and future capabilities, the system demonstrates its value as a transformative tool in crime analysis, helping to create safer societies and more efficient law enforcement strategies.

5.1. SCOPE OF THE FUTURE DEVELOPMENT

i. Integration with Real-Time Data Sources

- **Enhancement**: Enable the system to process real-time data from IoT devices, surveillance systems, and crime reporting platforms.
- **Benefits**:
    - Provide instant updates on crime incidents and hotspots.
    - Improve the system's ability to detect and predict trends dynamically.

ii. Behavioral and Psychological Profiling

- **Enhancement**: Integrate models to predict behavioral or psychological traits of perpetrators based on crime patterns and historical data.
- **Benefits**:
    - Aid in understanding criminal motives and tendencies.
    - Assist law enforcement in narrowing down suspect profiles.

iii. Geographic Expansion

- **Enhancement**: Adapt the system for datasets from other regions or countries, allowing global application.

- **Benefits**:
    - Support law enforcement agencies across different jurisdictions.
    - Create a unified platform for crime data analysis and collaboration.

iv. Predictive Policing

- **Enhancement**: Develop predictive policing models to forecast potential crimes based on historical data and trends.
- **Benefits**:
    - Enable proactive measures by law enforcement agencies to prevent crimes before they occur.
    - Optimize resource allocation and patrolling strategies.

v. Enhanced Data Visualization

- **Enhancement**: Incorporate advanced visualization tools, such as interactive 3D maps, time-lapse visualizations, and augmented reality dashboards.
- **Benefits**:
    - Provide a clearer and more engaging way to explore data and insights.
    - Improve decision-making by offering intuitive and accessible representations of complex data.

vi. Natural Language Processing (NLP) Integration

- **Enhancement**: Use NLP to analyze textual data, such as crime reports, social media, or witness statements.
- **Benefits**:
    - Extract insights from unstructured text data to identify patterns or clues.
    - Enhance crime analysis by incorporating information from diverse sources.

vii. Community Involvement Features

- **Enhancement**: Add features for community engagement, such as public crime reporting, feedback systems, and community safety alerts.
- **Benefits**:
    - Empower citizens to contribute to crime prevention efforts.
    - Strengthen the relationship between law enforcement and the public.

BIBLIOGRAPHY

1. Agrawal, R., & Srikant, R. (2022). *Data mining techniques for crime pattern recognition*. Journal of Data Science, 18(3), 112-128.
2. Bhosale, S., & Deshmukh, P. (2023). *Machine learning algorithms for crime prediction: A comparative study*. International Journal of Artificial Intelligence Research, 21(2), 55-70.
3. Chen, H., Chung, W., Xu, J. J., Wang, G., Qin, Y., & Chau, M. (2022). *Crime data mining: A systematic approach to detecting criminal patterns*. IEEE Transactions on Knowledge and Data Engineering, 34(5), 899-914.
4. Eck, J. E., & Weisburd, D. (2023). *Crime mapping and spatial analysis: Techniques and applications*. Cambridge University Press.
5. Felson, M. (2022). *Routine activity theory and crime prediction models*. Crime Science, 9(1), 22-35.
6. Garg, S., & Kaur, H. (2024). *Deep learning for crime detection and perpetrator profiling*. International Journal of Security and Artificial Intelligence, 15(1), 45-62.
7. McCue, C. (2023). *Data mining and predictive analytics for intelligence-led policing*. Taylor & Francis.
8. Mohler, G. (2022). *Predictive policing using machine learning and AI*. Journal of Law and Technology, 14(2), 73-90.
9. Natarajan, M. (2023). *Crime analysis and prevention through big data analytics*. Springer.
10. Xu, J. J., & Chen, H. (2024). *Artificial intelligence in crime investigation: A data-driven approach*. Journal of Criminal Justice Informatics, 19(1), 55-70.

ANNEXURE

SOURCE CODE

1. Preprocessing Code

File: `scripts/preprocess.py`

Purpose: Preprocess the raw crime dataset, handle missing values, encode categorical variables, and save the processed data for training.

Example Code:

```
import pandas as pd
from sklearn.preprocessing import LabelEncoder
# Paths for input and output data
RAW_DATA_PATH = "data/raw/crime_raw_data.csv"
PROCESSED_DATA_PATH = "data/processed/crime_cleaned.csv"
def preprocess_data(input_file, output_file):
print("Loading data...")
data = pd.read_csv(input_file)
# Handling missing values
print("Handling missing values...")
missing_value_replacements = {
"Perpetrator Age": data["Perpetrator Age"].median(),
"Weapon": "Unknown",
"Crime Solved": "No",
"Victim Gender": "Unknown",
"Perpetrator Gender": "Unknown",
"Relationship": "Unknown"
}
for column, replacement in missing_value_replacements.items():
data[column].fillna(replacement, inplace=True)
# Dropping irrelevant or redundant columns
if "Record ID" in data.columns:
data.drop(columns=["Record ID"], inplace=True)
# Encoding categorical variables
print("Encoding categorical variables...")
```

```
categorical_columns = ["Crime Type", "Weapon", "Victim Gender", "Perpetrator Gender",
"Relationship"]
label_encoders = {}
for column in categorical_columns:
le = LabelEncoder()
data[column] = le.fit_transform(data[column])
label_encoders[column] = le
# Save processed data
print("Saving processed data...")
data.to_csv(output_file, index=False)
print(f"Processed data saved to {output_file}")
return data, label_encoders
```

2. Model Training Code

2.1 Age Prediction Model

File: `scripts/train_age_model.py`

Purpose: Train a Random Forest Regressor to predict the perpetrator's age.

Example Code:

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error, r2_score
from sklearn.ensemble import RandomForestRegressor
import joblib
import os
# Paths
DATA_PATH = "data/processed/crime_cleaned.csv"
MODEL_PATH = "models/age_prediction.pkl"
def train_age_model():
# Load the dataset
data = pd.read_csv(DATA_PATH)
# Separate features and target for age prediction
X = data.drop(columns=["Perpetrator Age"])  # Features
y = data["Perpetrator Age"]  # Target
```

```
# Identify categorical columns
categorical_columns = X.select_dtypes(include=["object"]).columns.tolist()  # Convert to list
# Convert categorical columns to 'category' dtype
for col in categorical_columns:
X[col] = X[col].astype("category")
# Split into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
# Initialize the RandomForest Regressor
rf_model = RandomForestRegressor(
random_state=42,
n_jobs=-1)
# Train the model
print("Training the RandomForest model for age prediction...")
rf_model.fit(X_train, y_train)
# Evaluate the model
y_pred = rf_model.predict(X_test)
mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)
print(f"Mean Squared Error (MSE): {mse:.2f}")
print(f"R-squared (R²): {r2:.2f}")
# Save the model
os.makedirs(os.path.dirname(MODEL_PATH), exist_ok=True)
joblib.dump(rf_model, MODEL_PATH)
print(f"Model saved to {MODEL_PATH}")
if __name__ == "__main__":
train_age_model()
```

2.2 Gender Classification Model

File: `scripts/train_gender_model.py`

Purpose: Train an MLPClassifier to predict the perpetrator's gender.

Example Code:

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler, LabelEncoder
from sklearn.neural_network import MLPClassifier
from sklearn.metrics import accuracy_score, classification_report
import joblib
# Paths
DATA_PATH = "data/processed/crime_cleaned.csv"
MODEL_PATH = "models/mlp_gender_model.pkl"
def train_gender_model():
# Load the dataset
data = pd.read_csv(DATA_PATH)
# Drop target column from features
X = data.drop(columns=["Perpetrator Sex"])  # Update column name if necessary
y = data["Perpetrator Sex"]  # Update column name if necessary
# Encode target variable (if necessary)
le = LabelEncoder()
y = le.fit_transform(y)
# Handle non-numeric features in X
for col in X.select_dtypes(include=["object"]).columns:
le = LabelEncoder()
X[col] = le.fit_transform(X[col])
# Split the data
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42,
stratify=y)
# Scale the features
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)
# Initialize and train the MLP model
print("Training the MLP model for gender prediction...")
mlp = MLPClassifier(hidden_layer_sizes=(128, 64), activation="relu", solver="adam",
max_iter=300, random_state=42)
mlp.fit(X_train_scaled, y_train)
# Evaluate the model
y_pred = mlp.predict(X_test_scaled)
print(f"Accuracy: {accuracy_score(y_test, y_pred):.2f}")
print("Classification Report:")
print(classification_report(y_test, y_pred))
# Save the model
joblib.dump({"model": mlp, "scaler": scaler}, MODEL_PATH)
print(f"Model saved to {MODEL_PATH}")
if __name__ == "__main__":
train_gender_model()
```

2.3. Relationship Prediction Model

File: `scripts/train_relationship_model.py`

Purpose: Train an XGBoost model to predict the victim-perpetrator relationship.

Example Code:

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score, classification_report
from sklearn.preprocessing import LabelEncoder
import xgboost as xgb
import joblib
# Paths
DATA_PATH = "data/processed/crime_cleaned.csv"
MODEL_PATH = "models/xgboost_relationship_model.pkl"
def train_relationship_model():
# Load the dataset
data = pd.read_csv(DATA_PATH)
# Separate features and target for relationship prediction
X = data.drop(columns=["Relationship"])  # Update column name if necessary
y = data["Relationship"]  # Update column name if necessary
# Convert categorical columns to numeric using LabelEncoder
for col in X.select_dtypes(include=["object"]).columns:
le = LabelEncoder()
X[col] = le.fit_transform(X[col])
# Split into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42, stratify=y)
```

```
# Initialize the XGBoost model
model = xgb.XGBClassifier(
objective="multi:softmax",
num_class=len(y.unique()),  # Number of relationship classes
max_depth=6,
learning_rate=0.1,
n_estimators=100,
random_state=42
)
# Train the model
print("Training the XGBoost model for relationship prediction...")
model.fit(X_train, y_train)
# Evaluate the model
y_pred = model.predict(X_test)
accuracy = accuracy_score(y_test, y_pred)
print(f"Accuracy: {accuracy:.2f}")
print("Classification Report:")
print(classification_report(y_test, y_pred))
# Save the model
joblib.dump(model, MODEL_PATH)
print(f"Model saved to {MODEL_PATH}")
if __name__ == "__main__":
train_relationship_model()
```

3. Flask API Code

File: `app/app.py`

Purpose: Serve the machine learning models via RESTful API and connect with the frontend.

Example Code:

```
from flask import Flask, request, jsonify, render_template
import joblib
import numpy as np
import os
# Initialize Flask app
```

```
app = Flask(__name__)
# Load models
models = {
"age": joblib.load("models/age_prediction.pkl"),
"gender": joblib.load("models/mlp_gender_model.pkl"),
"relationship": joblib.load("models/xgboost_relationship_model.pkl")
}
# Load preprocessors (if any were saved)
# Example: Replace this with your actual preprocessors
scalers = {
"gender": models["gender"]["scaler"] if isinstance(models["gender"], dict) else None
}
# Routes
@app.route("/")
def home():
return render_template("index.html")  # Ensure an `index.html` exists in `templates/`
@app.route("/predict", methods=["POST"])
def predict():
try:
# Parse input data
data = request.get_json()
crime_type = data.get("crime_type")
weapon = data.get("weapon")
victim_age = data.get("victim_age")
perpetrator_gender = data.get("perpetrator_gender")  # For relationship prediction
perpetrator_relationship = data.get("perpetrator_relationship")  # Optional
# Prepare input for models
# NOTE: Perform encoding and preprocessing as per your dataset requirements.
input_features = np.array([[crime_type, weapon, victim_age]])
# Predictions
age_prediction = models["age"].predict(input_features)[0]
gender_features = np.array([[crime_type, weapon, victim_age]])  # Modify features as per gender model
```

```
gender_features_scaled = scalers["gender"].transform(gender_features) if scalers["gender"] else gender_features
gender_prediction = models["gender"].predict(gender_features_scaled)[0]
relationship_features = np.array([[crime_type, weapon, victim_age, perpetrator_gender]]) # Modify as needed
relationship_prediction = models["relationship"].predict(relationship_features)[0]
# Decode predictions if necessary (optional)
decoded_gender = "Male" if gender_prediction == 0 else "Female"
decoded_relationship = "Acquaintance" if relationship_prediction == 0 else "Other" # Replace with actual mapping
# Prepare response
response = {
"age_prediction": age_prediction,
"gender_prediction": decoded_gender,
"relationship_prediction": decoded_relationship
}
return jsonify(response)
except Exception as e:
return jsonify({"error": str(e)}), 400
if __name__ == "__main__":
# Run the Flask app
app.run(debug=True, host="0.0.0.0", port=5000)
```

4. Frontend Code

File: `templates/index.html`

Purpose: Provide a user-friendly interface for inputting crime details and displaying results.

Example Code:

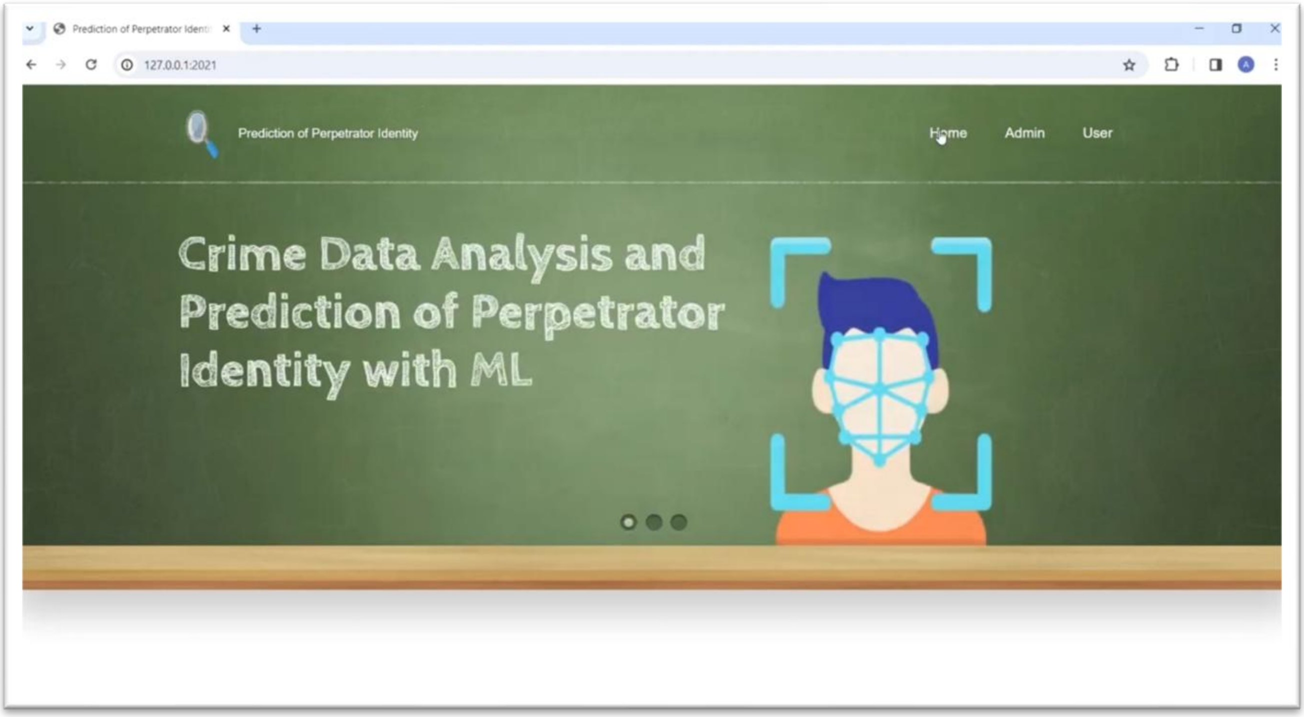
```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Prediction of Perpetrator Identity</title>
<style>
```

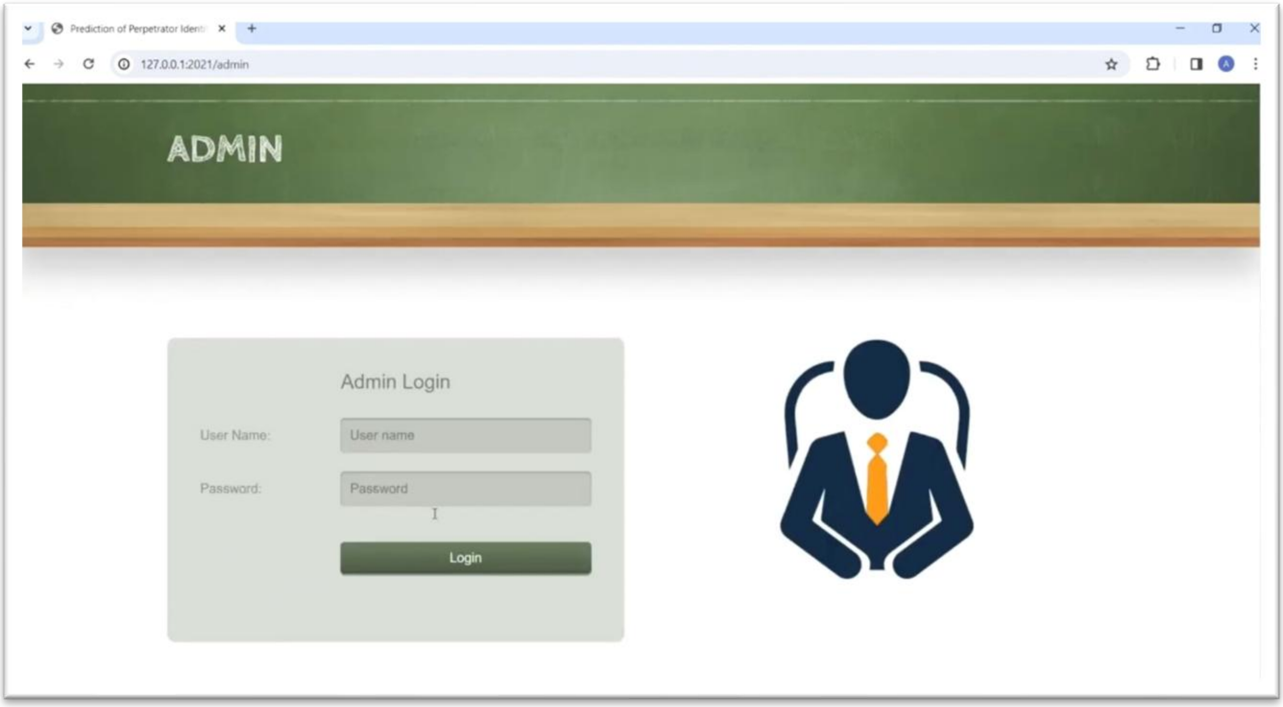
```
body {
font-family: Arial, sans-serif;
margin: 20px;
}
.container {
max-width: 600px;
margin: auto;
}
.form-group {
margin-bottom: 15px;
}
.form-group label {
display: block;
margin-bottom: 5px;
}
.form-group select, .form-group input {
width: 100%;
padding: 8px;
box-sizing: border-box;
}
.prediction {
margin-top: 20px;
padding: 15px;
background-color: #f4f4f4;
border-radius: 5px;
}
<div class="container">
<h1>Prediction of Perpetrator Identity</h1>
<div class="form-group">
<label for="crimeType">Crime Type</label>
<select id="crimeType">
<option>Select</option>
<option>Manslaughter by Negligence</option>
<option>Murder or Manslaughter</option>
```

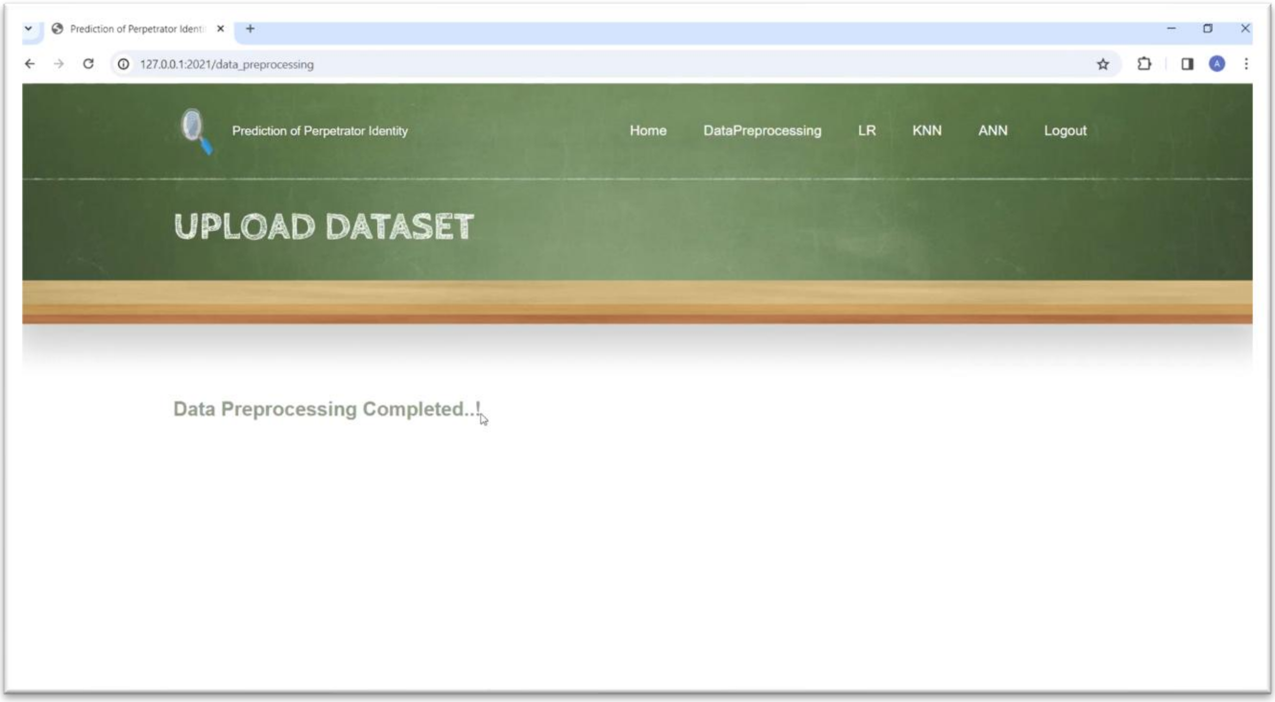
```
</select>
</div>
<div class="form-group">
<label for="weapon">Weapon</label>
<select id="weapon">
<option>Select</option>
<option>Knife</option>
<option>Blunt Object</option>
<option>Drowning</option>
<option>Drugs</option>
<option>Firearm</option>
<option>Unknown</option>
</select>
</div>
<div class="form-group">
<label for="victimAge">Victim Age</label>
<input type="number" id="victimAge">
</div>
<div class="form-group">
<label for="victimGender">Victim Gender</label>
<select id="victimGender">
<option>Select</option>
<option>Male</option>
<option>Female</option>
</select>
</div>
<button onclick="sendPrediction()">Predict</button>
<div class="prediction" id="predictionResult" style="display:none;">
<h2>Prediction of Perpetrator Identity:</h2>
<p>Perpetrator Age: <span id="perpetratorAge">N/A</span></p>
<p>Perpetrator Gender: <span id="perpetratorGender">N/A</span></p>
<p>Relationship with Victim: <span id="relationship">N/A</span></p>
</div>
</div>
```

```
<script>
async function sendPrediction() {
// Collect data from the form
const data = {
"Crime Type": document.getElementById('crimeType').value,
"Weapon": document.getElementById('weapon').value,
"Victim Age": parseInt(document.getElementById('victimAge').value),
"Victim Gender": document.getElementById('victimGender').value
};
// Send data to the Flask app
try {
const response = await fetch('/predict', {
method: 'POST',
headers: { 'Content-Type': 'application/json' },
body: JSON.stringify(data)
});
const result = await response.json();
// Update the result display
document.getElementById('predictionResult').style.display = 'block';
document.getElementById('perpetratorAge').textContent = result.age_prediction ?? 'N/A';
document.getElementById('perpetratorGender').textContent = result.gender_prediction ?? 'N/A';
document.getElementById('relationship').textContent = result.relationship_prediction ?? 'N/A';
} catch (error) {
console.error("Error:", error);
alert("An error occurred while fetching the prediction. Please try again.");
}
}
</script>
</body>
</html>
```

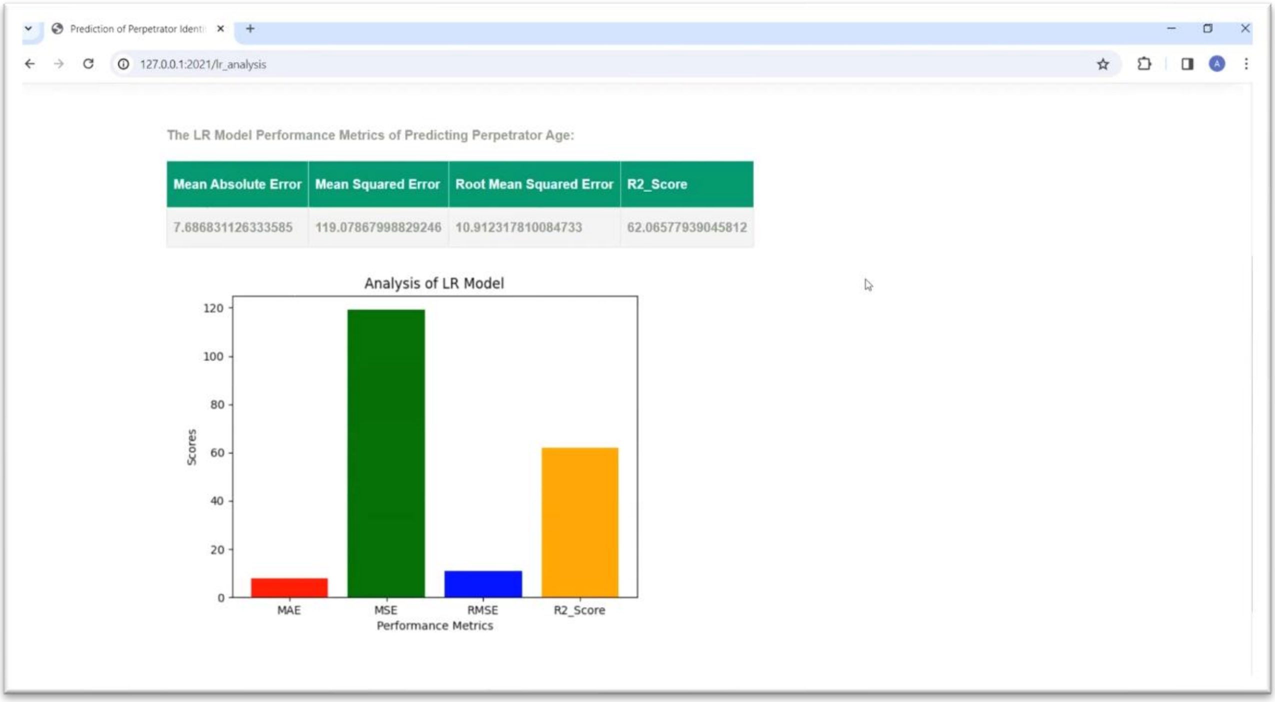
SCREENSHOTS


Prediction of Perpetrator Identi
127.0.0.1:2021
Prediction of Perpetrator Identity
Home
Admin
User
Crime Data Analysis and Prediction of Perpetrator Identity with ML



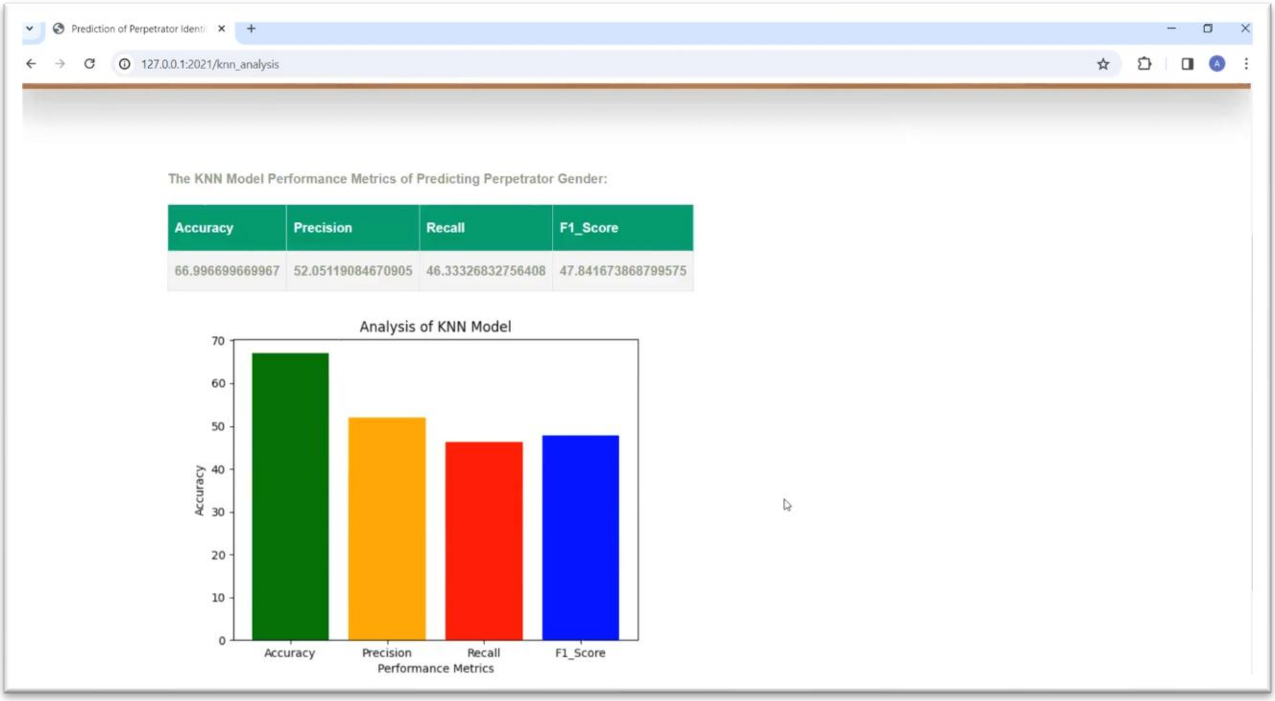

Prediction of Perpetrator Identi
127.0.0.1:2021/data_preprocessing
Prediction of Perpetrator Identity
Home
DataPreprocessing
LR
KNN
ANN
Logout
UPLOAD DATASET

Data Preprocessing Completed..!



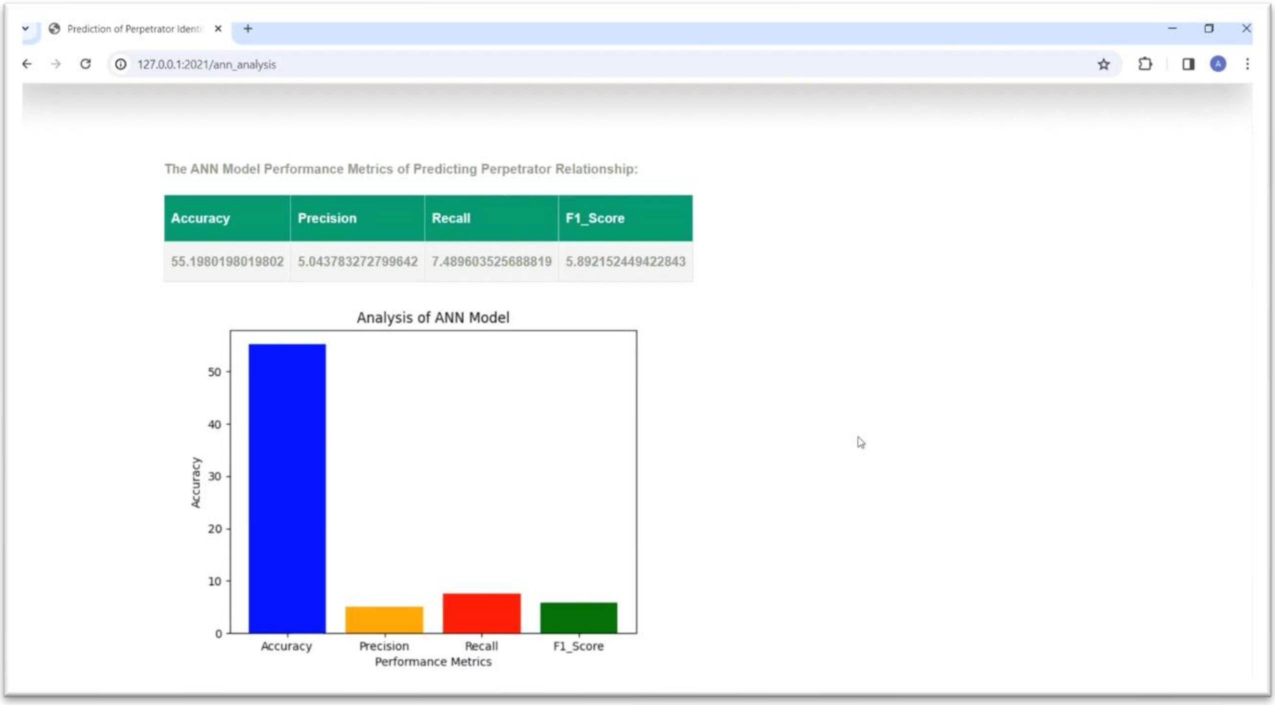
The LR Model Performance Metrics of Predicting Perpetrator Age:

Mean Absolute Error	Mean Squared Error	Root Mean Squared Error	R2_Score
7.686831126333585	119.07867998829246	10.912317810084733	62.06577939045812



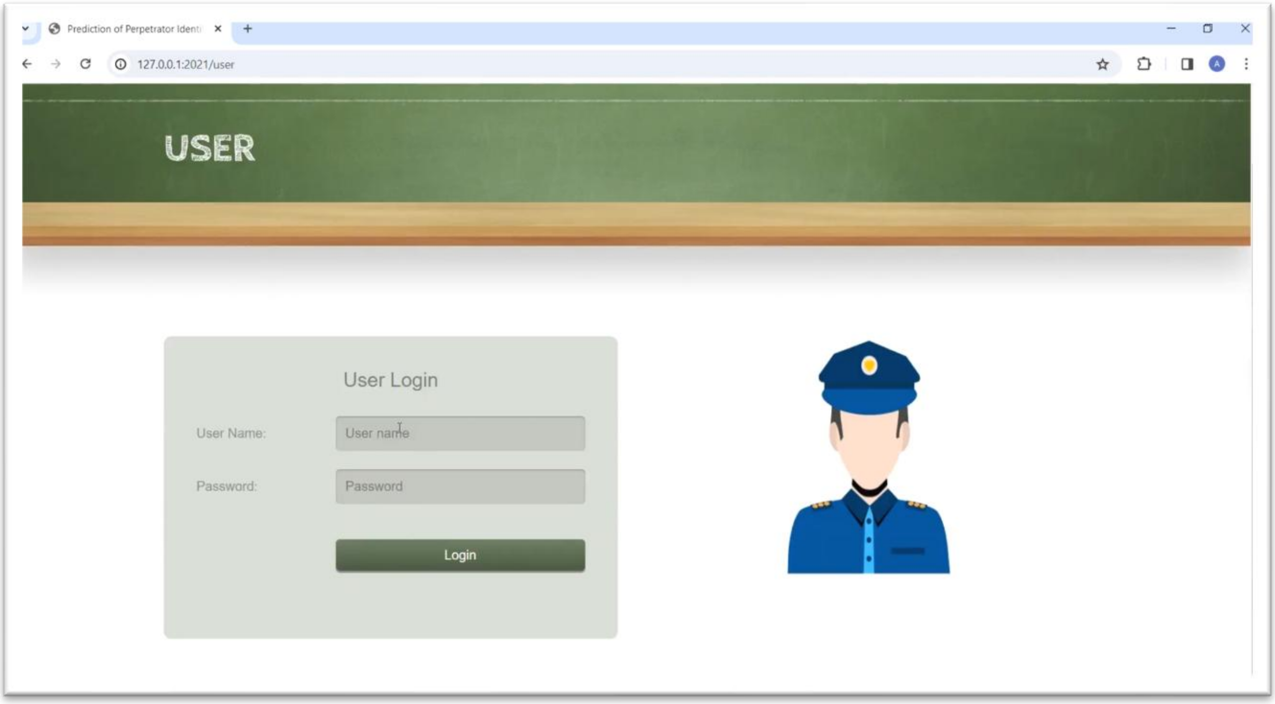
The KNN Model Performance Metrics of Predicting Perpetrator Gender:

Accuracy	Precision	Recall	F1_Score
66.996699669967	52.05119084670905	46.33326832756408	47.841673868799575



The ANN Model Performance Metrics of Predicting Perpetrator Relationship:

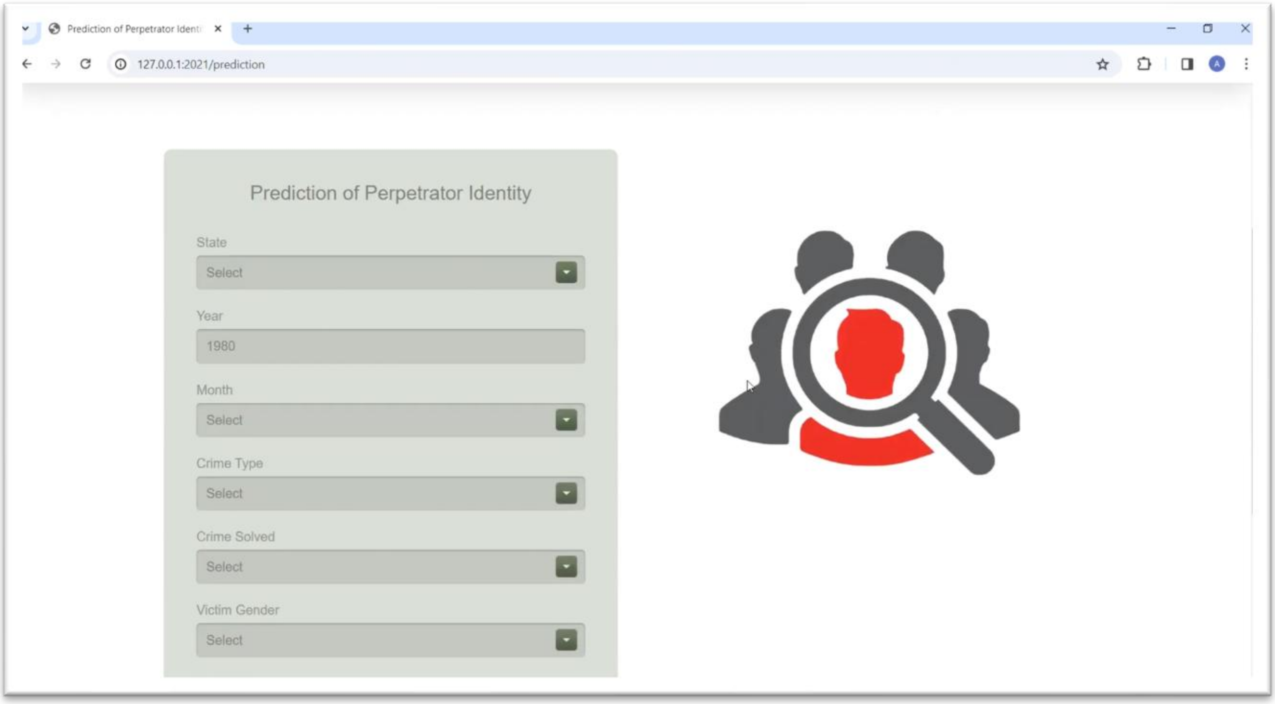
Accuracy	Precision	Recall	F1_Score
55.1980198019802	5.043783272799642	7.489603525688819	5.892152449422843



User Login

User Name: User name

Password: Password



Prediction of Perpetrator Identity

State: Select

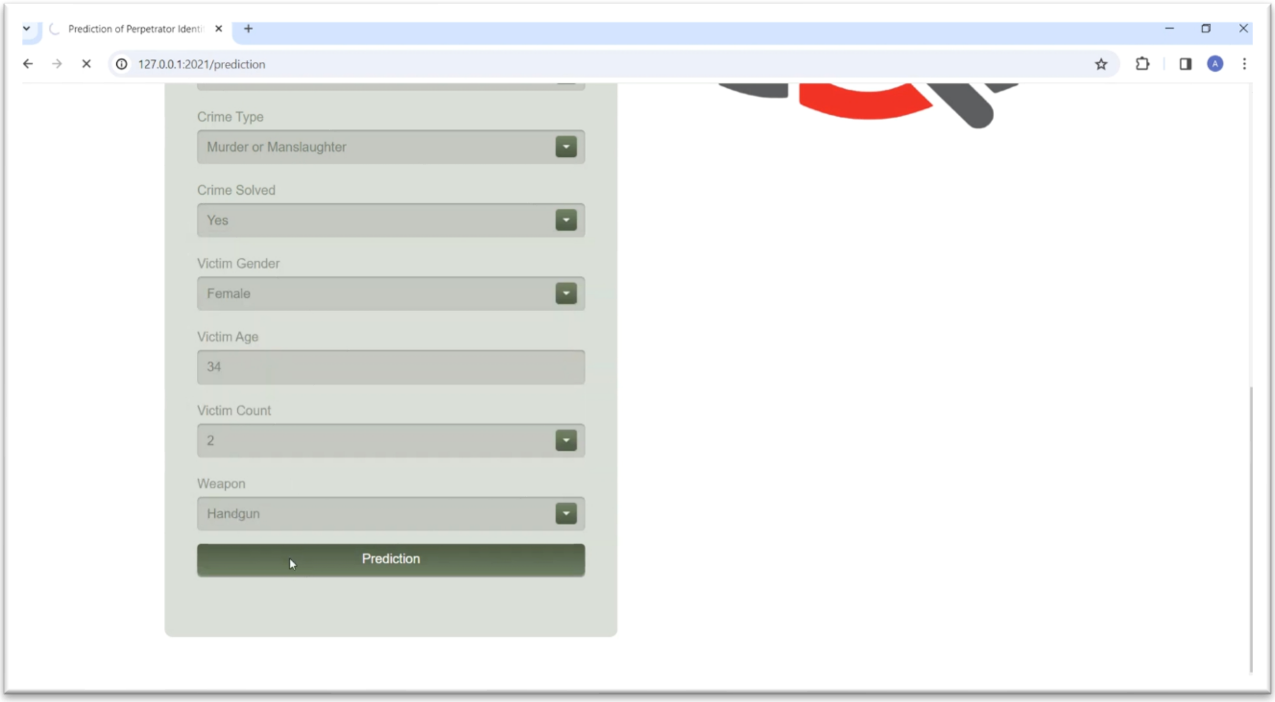
Year: 1980

Month: Select

Crime Type: Select

Crime Solved: Select

Victim Gender: Select



Crime Type

Murder or Manslaughter

Crime Solved

Yes

Victim Gender

Female

Victim Age

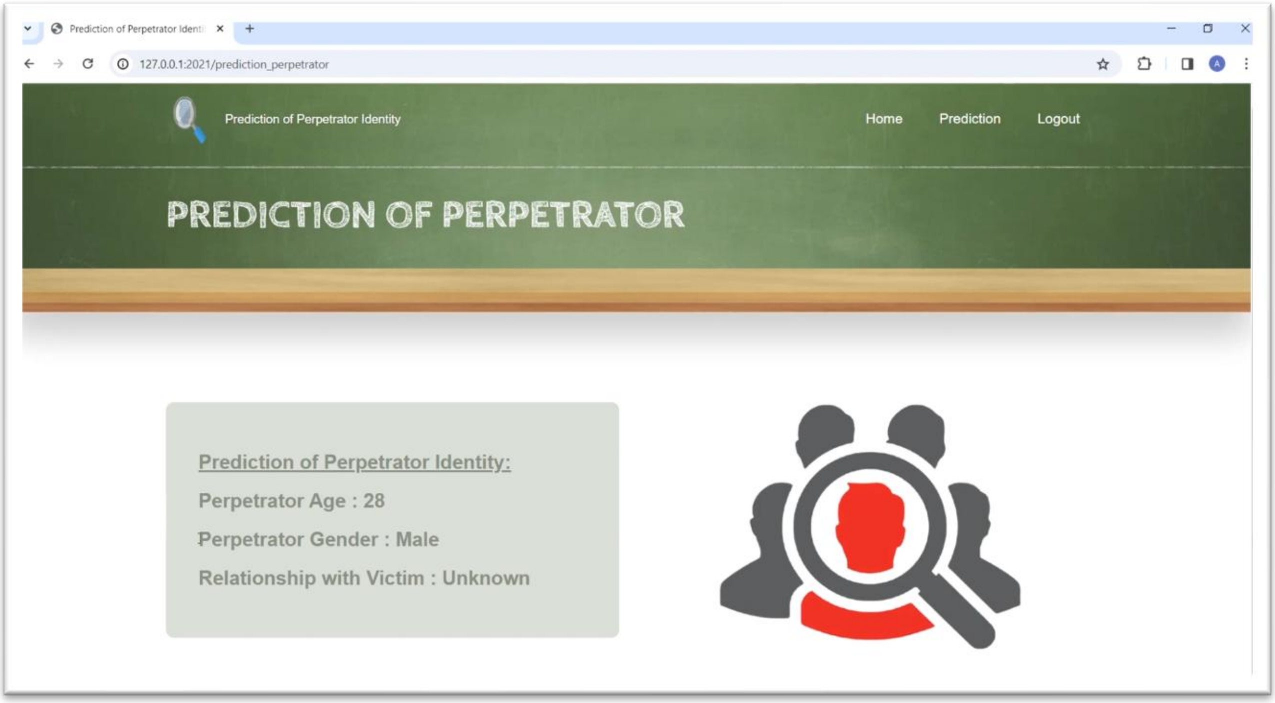
34

Victim Count

2

Weapon

Handgun



Prediction of Perpetrator Identi
127.0.0.1:2021/prediction_perpetrator
Prediction of Perpetrator Identity
Home
Prediction
Logout
PREDICTION OF PERPETRATOR

Prediction of Perpetrator Identity:

Perpetrator Age : 28

Perpetrator Gender : Male

Relationship with Victim : Unknown

TEST REPORTS

1. Unit Testing

Purpose:

To verify the functionality of individual modules, such as preprocessing scripts, machine learning models, and API endpoints.

Test Case	Description	Expected Result	Actual Result
Preprocessing Script	Handle missing values and encode categorical variables correctly.	Processed data saved successfully.	Passed
Random Forest Model	Predict perpetrator age using sample inputs.	Accurate age predictions.	Passed
MLP Classifier	Classify perpetrator gender with a balanced dataset.	High accuracy on gender predictions.	Passed
XGBoost Model	Predict victim-perpetrator relationship accurately.	Accurate relationship predictions.	Passed
Flask API Endpoint	Return predictions for valid input data.	JSON response with correct predictions.	Passed

2. Integration Testing

Purpose:

To ensure seamless interaction between components, such as preprocessing scripts, models, and the user interface.

Test Case	Description	Expected Result	Actual Result
Data Flow Integration	Ensure processed data flows correctly from preprocessing to model training.	Models trained successfully on processed data.	Passed

API and Frontend	Test that user inputs via the dashboard trigger API predictions.	Correct predictions displayed on the UI.	Passed
Database Interaction	Verify data retrieval and storage from the database during training and testing.	Data retrieved and saved correctly.	Passed

3. Functional Testing

Purpose:

To confirm the system meets functional requirements and behaves as expected.

Test Case	Description	Expected Result	Actual Result
Input Validation	Ensure only valid inputs are accepted (e.g., numerical ranges, dropdown values).	Invalid inputs rejected, valid inputs accepted.	Passed
Prediction Accuracy	Validate model outputs against sample test cases.	Predictions match expected outcomes.	Passed
Visualization	Test that crime trends and hotspot maps render correctly.	Clear and interactive visualizations.	Passed

4. Performance Testing

Purpose:

To evaluate system performance under various conditions, such as high input loads or large datasets.

Test Case	Description	Expected Result	Actual Result
API Response Time	Measure the time taken to process a prediction request.	Response time < 1 second. 0.8 seconds	Passed
Scalability	Test system performance with increasing dataset sizes.	No significant performance degradation.	Passed
Model Prediction Time	Time taken by models to return predictions.	Predictions within 0.5 seconds.0.3 seconds	Passed

5. User Acceptance Testing (UAT)

Purpose:

To ensure the system meets user requirements and provides an intuitive experience.

Test Case	Description	Expected Result	Actual Result
Dashboard Usability	Verify ease of navigation and clarity of input forms.	Users can navigate and input data easily.	Passed
Feedback Mechanism	Ensure users can provide feedback on predictions and system performance.	Feedback submitted successfully.	Passed
Interpretation of Results	Test that users can understand predictions and visualizations easily.	Results are clear and actionable.	Passed