

**THE FUTURE OF CRIME PREVENTION:
POLICE CASE ANALYSIS USING
MACHINE LEARNING
(Analyze and Classify Similar Case Documents
and Predict Category)**

Traveena Chandrasegar

(IT20001452)

Bachelor of Science (Hons) in Information Technology
Specializing in Software Engineering

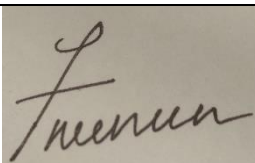
Department of Information Technology
Sri Lanka Institute of Information Technology
Sri Lanka

September 2023

Declaration of The Candidate & Supervisor

I declare that this is my own work and this dissertation 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 dissertation, 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
Traveena.C	IT20001452	

The above candidate has carried out research for the bachelor's degree Dissertation under my supervision.

Signature of the supervisor:

Date:

(Ms. Hansika Mahaadikara)

Signature of co-supervisor:

Date:

(Ms. Sanjeevi Chandrasiri)

Abstract

The proposed research focuses on leveraging machine learning techniques to analyze police case documents and predict their category. By analyzing historical crime data, the research aims to identify patterns and trends that can be used to develop predictive models. These models can then be used by law enforcement agencies to prioritize resources and prevent crimes from occurring. This study proposes a solution to support criminal investigations by providing a technological analysis to justify the guilt of an accused criminal. The current process of manually searching historical judgments databases for similar cases is time-consuming and prone to errors and biases. The study suggests an automated text analysis system that can help agencies quickly and consistently discover important patterns in crime occurrences. The proposed system would eliminate the need for individuals to manually search for relevant and repetitive keywords in past cases. Instead, the automated system would analyze the text data from historical judgments databases and identify patterns and similarities and classify them into a group according to its similarity in the crime occurrences. This would enable investigators to make more informed decisions during the interrogation and crime pattern detection processes. The system would also incorporate natural language processing techniques to extract relevant information from unstructured text data. The proposed system can be used by police and investigation teams and can be accomplished by using machine learning techniques. The system is particularly useful in aiding investigations of crimes against women and accidents, as it can help police investigators to easily detect connections between incidents and gain valuable insights that can be used to improve public safety. Overall, this study seeks to improve the efficiency of police investigations and aid in crime prevention efforts using advanced analytical techniques.

Keywords— Crime Investigation, Document Classification, Unstructured text data analysis, Semantic Analysis, Machine Learning, Text Analytics

ACKNOWLEDGEMENT

The work described in this research paper was carried out as the 4th year research project for the subject Comprehensive Design Analysis Project. The completed final project is the result of combining all the hard work of the group members and the encouragement, support and guidance given by many others. Therefore, it is our duty to express our gratitude to all who gave us the support to complete this major task.

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

I also wish to thank all our colleagues and friends for all their help, support, interest and valuable advice. Finally, I would like to thank all others whose names are not listed particularly but have given their support in many ways and encouraged us to make this a success.

TABLE OF CONTENT

Declaration of The Candidate & Supervisor	i
Abstract	ii
ACKNOWLEDGEMENT	iii
TABLE OF CONTENT	iv
LIST OF FIGURES	vi
LIST OF TABLES	vi
LIST OF ABBREVIATIONS	vii
LIST OF APPENDICES	vii
1. INTRODUCTION	1
1.1. Background Study and Literature	2
1.1.1. Background Study.....	2
1.1.2. Literature Review	3
1.2. Research Gap	6
1.3. Research Problem	8
1.4. Research Objectives	10
1.4.1. Main Objective	10
1.4.2. Sub Objectives	10
1.4.3. Business Objectives	10
2. METHODOLOGY	11
2.1. Introduction	11
2.1.1. System Overview	11
2.2. Commercialization aspect of the product	20
2.3. Testing and Implementation	20
2.3.1. Testing	20

2.3.2. Implementation	27
3. RESULTS AND DISCUSSION	31
3.1. Results	31
3.2. Research Findings	33
3.3. Discussion	35
4. FUTURE SCOPE	37
5. CONCLUSION	38
REFERENCES	39
APPENDICES	41
Appendix A: User Interface Designs	41

LIST OF FIGURES

Figure 1.1: Case Statistic Graph – Sri Lanka.....	8
Figure 1.2 Graph of Recorded Cases vs Pending Cases.....	9
Figure 2.1: Component System diagram.....	13
Figure 2.2: High level architecture diagram.....	13
Figure 2.3: Project Code Management for the system.....	14
Figure 2.5: Work Breakdown Structure diagram.....	15
Figure 2.6: Import libraries.....	27
Figure 2.7: PDF Text Data Extraction and Pre-processing.....	28
Figure 2.8: Text Data Transformation, Model Training, and Saving for Crime Case Categorization.....	30

LIST OF TABLES

Table 1.1: Comparison of existing solutions.....	7
Table 2.1: Ensure the form is not submitted with missing data Manual Test Case 01	23
Table 2.2 Functionality Add data Page Manual Test Case 02.....	24
Table 2.3: Ensure success message is displayed after re-training Manual Test Case 03	25
Table 2.4: Display Category Prediction Manual Test Case 04.....	26
Table 3.1 Accuracy measures for crime pattern analysis.....	32

LIST OF ABBREVIATIONS

Abbreviations	Description
ML	Machine Learning
NLP	Natural Language Processing
NLTK	Natural Language Toolkit
CSV	Comma-Separated Values
PDF	Portable Document Format
SDLC	Software Development Life Cycle
GUI	Graphical User Interface
IO	Input/Output
TF-IDF	Term Frequency– Inverse Document Frequency

LIST OF APPENDICES

Appendix	Description	Page
Appendix A	User Interface Designs	Error! Bookmark not defined.

1. INTRODUCTION

In this section of the final report, a comprehensive overview is presented of the research context, the existing body of literature, the gaps that are aimed to be addressed, the research challenges that have been identified, and the specific objectives that are intended to be achieved in this study.

The global concern for security is considered paramount by every nation. Various measures are taken by governments, including the enactment of legislation, to enhance the safety of their citizens. Likewise, a pivotal role is played by the outcomes of court judgments in criminal cases in shaping the safety regulations that are enforced within a country. Delays in rendering judgments in such cases are seen as a significant drawback to the nation's legal system. Consequently, the primary focus is placed on the expediting of the judicial processes.

A significant issue in Sri Lanka's judicial system is grappled with a mounting backlog of pending criminal cases. This issue is largely attributed to the traditional, manual approach to examining historical cases in search of commonalities and recurring patterns, which significantly contributes to the yearly surge in pending criminal cases.

The manual examination of thousands of pages of text-based court documents, the quest for interconnections, and the recording of depositions is considered an arduous task that demands a considerable amount of time and effort from legal professionals, including scholars in the field. As a result, the adoption of a technology-driven solution that employs text-based analytics on unstructured data is advocated by our proposal. This approach offers a more efficient alternative to the labor-intensive manual analysis.

Analyze and Classify Similar Case Documents and Predict Category system is designed to provide real-time analysis of historical case summaries, thereby facilitating informed judgments and expediting the legal process.

1.1. Background Study and Literature

1.1.1. Background Study

In crime analysis, the challenge of uncovering hidden patterns and similarities between crime incidents to ascertain the background of crimes is a significant issue confronted by criminal prosecutors and crime analysts. Various types of crime incidents exist, including abduction, housebreaking, rape, sexual abuse, robbery, theft, grievous hurt, extortions, cheating, and more.

Sri Lanka faces a rising backlog of criminal cases, partly due to the labor-intensive process of manually searching past cases for similarities. And this backlog of crime cases poses significant challenge to the Sri Lankan legal system where it leads to delay in delivering justice, which can be detrimental to both the victim and the accused. The manual process of go through past cases is inefficient and consumes comparatively high amount of time and effort for the legal professionals.

To address these challenges, an automated text analysis system is proposed. this system will leverage technology to analyze large volume of unstructured text data form past cases. Such a system would eliminate the need for manual case searches, ensuring more consistent, reliable results. And instead of depending on the human effort, it uses algorithms and techniques to identify similarities, patterns and relevant information within the documents. This would expedite investigations, reduce errors and biases, and enhance decision-making for both investigators and the court system.

This research assists in identify similarity between past crime case judgement documents and classify them into labeled category accordingly. Furthermore, predicting the category of a newly given document by comparing the existing document and trained model. The prime goal of this study is to support the police department and the investigation teams of the legal sector in fastening the process of analyzing the crime documents to find the similarities in legal documents during investigating a crime.

This research is part of the application “Analyze and Classify Similar Case Documents and Predict Category” promises substantial benefits for the Sri Lankan criminal justice

system, including reduced case backlog, enhanced efficiency and accuracy, and quicker case resolution and also offers a modern and efficient approach to tackling the backlog of criminal cases. Ultimately, it would bolster trust in the system, ensuring justice for victims and their families.

1.1.2. Literature Review

The analysis of crime-related documents has evolved significantly in recent years, with researchers employing various techniques and technologies to extract valuable insights from both structured and unstructured data sources. This literature review aims to provide a comprehensive overview of recent advancements in the field, drawing upon key research papers that have contributed to the understanding of criminal activities and the development of decision support systems for law enforcement agencies.

Structured Data Analysis

Some studies in the domain of crime document analysis have primarily focused on structured data sources. Dahbur and Muscarello's research [2,12] present a classification system that utilizes artificial intelligence and decision trees to identify patterns in serial crimes. By employing rule-based expert systems and statistical techniques, their work contributes to the profiling and prediction of serial criminal behavior, thereby assisting law enforcement agencies in effectively combating serial offenders.

Unstructured Data Analysis

while others have employed a combination of both structured and unstructured data in their analyses [13, 15]. Among the studies that center on the analysis of unstructured data, one research approach involves utilizing cosine similarity to compute the similarity coefficient between indictments and judgments [1,11].

A substantial portion of research in this field has embraced unstructured data analysis, combining natural language processing and machine learning techniques for comprehensive insights. Ghankutkar et al. [3] leveraged these approaches to categorize crime news articles into types such as robbery, murder, and fraud. Their work

demonstrates the utility of natural language processing and machine learning in automating the classification of crime news, providing invaluable support for crime analysis and prediction.

Qi's research [4] introduces a text classification approach for theft crimes using the TF-IDF technique and the XGBoost machine learning model. This method effectively categorizes theft crimes into subtypes based on textual data extracted from crime reports, showcasing the potential of machine learning for fine-grained crime classification.

The need for diverse language support in crime analysis is addressed by Alruily et al. [5], who propose a machine learning-based approach for classifying Arabic crime documents into various crime types. By considering features such as keywords, text statistics, and machine learning classifiers, their work enhances crime analysis in Arabic-speaking regions, emphasizing the importance of linguistic diversity in crime document analysis.

Cyber-Crime and Social Crime Analysis

Ch et al. [6] focus on the analysis of cyber-crime offenses, employing decision trees and random forests to classify offenses into categories such as hacking, identity theft, and phishing. Their work highlights the significance of machine learning in addressing contemporary forms of criminal activity in the digital sphere.

Abbass et al.'s framework [7] extends crime analysis to social media platforms, predicting social crimes through Twitter tweets. Their text mining approach, driven by natural language processing and machine learning, classifies tweets into categories like hate speech, harassment, and discrimination, supporting social crime prediction and prevention in the age of online communication.

Comprehensive Crime Analysis

Kim et al.'s research [8] offers a holistic perspective on crime analysis, combining data mining techniques with machine learning algorithms. By exploring spatiotemporal patterns, social network analysis, and crime hotspots, their approach provides

invaluable insights for crime prevention and law enforcement strategies, emphasizing the multidimensional nature of crime-related data.

Kaur et al.'s work [9] advances crime analysis through text mining techniques. Their application of natural language processing and machine learning algorithms to crime-related text data enables the extraction of information, pattern identification, and classification of crimes into categories such as property crimes, violent crimes, and white-collar crimes, underscoring the versatility of these techniques in diverse crime contexts.

Machine Learning for Crime Data Analysis

McClendon and Meghanathan [10] advocate for the use of machine learning algorithms in crime data analysis. Their approach, incorporating decision trees and support vector machines, explores various aspects of crime data, including crime types, locations, and time periods. By identifying patterns and providing insights for crime analysis and prediction, their work contributes to the growing body of literature that underscores the efficacy of machine learning in understanding and combating criminal activities.

Crime Data Analysis and Link Analysis Techniques

Sergei Ananyan's study [13] aimed to uncover fresh patterns and connections among incident types, locations, timings, weaponry, narcotics involvement, and descriptive particulars, relying on historical data. This was accomplished using the Poly Analyst Link Analysis engine, along with a visual map illustrating correlations between the identified terms and specific values within structured attributes. The research provided a notably thorough and impartial overview of incidents by examining both the structured and textual components of the database.

In summary, the body of literature pertaining to the application of machine learning techniques in crime analysis and classification encompasses a wide array of domains. These encompass e-government, serial criminal behavior patterns, theft crimes, cybercrimes, social crimes, and the utilization of text mining for crime analysis. These studies collectively underscore the potential of machine learning in automating the

analysis and categorization of crime data, thereby offering invaluable support to law enforcement agencies in detecting, predicting, and preventing crimes. The variety of techniques employed, such as natural language processing, text mining, decision trees, random forests, support vector machines, and XGBoost, underscores the versatility of machine learning in crime analysis.

Nevertheless, it's essential to acknowledge certain limitations present in the existing literature. These include reliance on limited data sources, the prevalence of language-specific approaches, the absence of interpretability in some machine learning models, and the potential biases within crime data. To overcome these limitations and enhance the accuracy, robustness, and interpretability of machine learning methods for crime analysis and classification, additional research is imperative.

In conclusion, recent research in the field of crime-related document analysis has witnessed significant advancements, driven by the integration of machine learning, natural language processing, and data mining techniques. These studies have expanded the analytical capabilities of law enforcement agencies and researchers, offering tools and methodologies for the effective classification, prediction, and understanding of criminal activities. As the digital landscape continues to evolve, it is imperative to further explore and refine these techniques to address emerging challenges in crime analysis. This literature review sets the stage for our own research endeavor, which seeks to implement semantic-based analysis and ranking of court case judgment summary documents, aiming to provide a more efficient and accurate solution for authorities and scholars in the field of crime-related document analysis.

1.2. Research Gap

While technology has played a role in criminal investigations globally, there exists a notable research gap concerning automated text analysis systems to aid decision-making within criminal investigations, especially within the context of Sri Lanka. Many of the existing studies on these systems have been conducted in other countries and may not be directly relevant to Sri Lanka's unique circumstances. Furthermore, there is a noticeable absence of empirical research concerning the specific challenges encountered by law enforcement agencies and the court system in Sri Lanka when

dealing with criminal cases. This dearth of research makes it challenging to pinpoint the exact prerequisites and limitations that must be taken into account when designing an automated text analysis system tailored for Sri Lanka. Thus, there is a pressing need for further investigation into the distinctive challenges and requisites of the Sri Lankan criminal justice system as it relates to handling criminal cases. Such research could serve as a valuable foundation for developing an automated text analysis system specifically attuned to the Sri Lankan context, one that can effectively facilitate decision-making in the realm of criminal investigations.

The following Table 1.1 depicts a summary comparison of features in the proposed system against existing systems and approaches.

RESEARCH	TEXT BASED ANALYTICS	LABELING CLASSIFIED CATEGORIES	POLICE RELATED	LOWER COST	AVAILABLE FOR SRI LANKAN DATA
AUTOMATED CRIME REPORT ANALYSIS & CLASSIFICATION FOR E-GOVERNMENT [1]	✓	✓	✗	✓	✗
CLASSIFICATION SYSTEM FOR SERIAL CRIMINAL PATTERNS [2]	✗	✗	✓	✓	✗
MODELLING MACHINE LEARNING FOR ANALYSING CRIME NEWS [3]	✓	✗	✗	✓	✗
THE TEXT CLASSIFICATION OF THEFT CRIME BASED ON TF-IDF AND XGBOOST MODEL [4]	✗	✓	✓	✓	✗
ANALYZE AND CLASSIFY SIMILAR CASE DOCUMENTS AND PREDICT CATEGORY	✓	✓	✓	✓	✓

Table 1.1: Comparison of existing solutions

1.3. Research Problem

The research problem in Sri Lanka is that the current procedure for criminal case handling in court involves manual analysis of past criminal cases from historical judgments databases. This process is time-consuming, prone to errors, and biases, and can lead to delays in justice and wrongful convictions. There is a need for an effective crime prevention method to improve public safety. While police departments have access to vast amounts of data related to past crimes, it can be difficult to analyze and make use of this data in a way that is effective for preventing future crimes.

In Sri Lanka, the annual average number of pending criminal cases is increasing due to several reasons. One of the reasons is the manual process of searching for past cases by the court to identify similar facets or patterns that exist in the new cases.

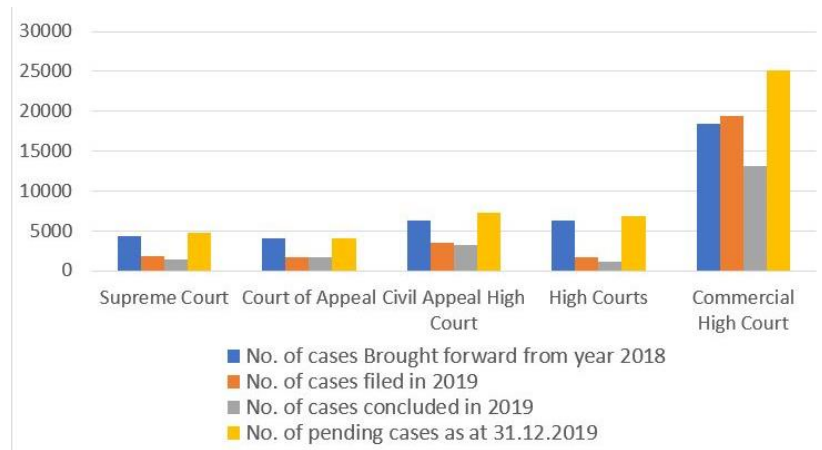


Figure 1.1: Case Statistic Graph – Sri Lanka

2019(<https://www.moj.gov.lk/images/pdf/Statistics/Case-Statistics-2019.pdf>)

This process is time-consuming, prone to errors and biases, and adds to the already overloaded workload of the court system. To address these issues, there is a need for a more efficient and effective criminal investigation process. The use of modern technologies such as automated text analysis systems can significantly improve the investigation process by identifying important patterns and similarities in crime occurrences.

One of the most unjust scenarios within criminal investigations involves issuing incorrect judgments in legal matters. Based on crime statistics reports released by the Sri Lankan Police, there has been an average of 45,962 serious crimes recorded annually from 2010 to 2018. These statistics reveal a concerning trend: a significant number of these crimes remain unresolved, reflecting unfavorably on the efficacy of our country's investigation system. Over the same period, there has been an average of 31,798 pending investigations, which accounts for a substantial 70.31% of the average number of recorded serious crimes with pending cases.

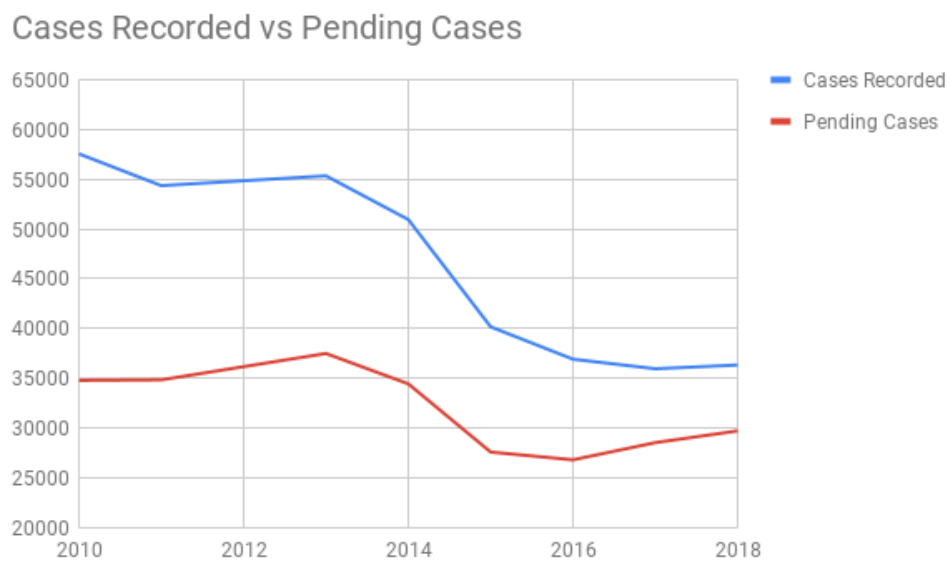


Figure 1.2 Graph of Recorded Cases vs Pending Cases

There is a need for an automated text analysis system that can quickly and consistently discover important patterns in crime occurrences and assist the judicial system in making better-informed decisions. However, the existing research in this area has largely focused on developed countries and their legal systems, and there is a lack of research on developing countries like Sri Lanka.

A primary factor contributing to this shortfall is the reliance on manual procedures within the judicial system. These processes involve sifting through historical case records to identify comparable elements and trends that can bolster the arguments in

newly filed cases. Consequently, this results in the inefficient utilization of time and energy for numerous individuals, including criminal prosecutors, crime analysts, and legal scholars.

1.4. Research Objectives

1.4.1. Main Objective

The main objective of the idea is to develop an automated text analysis system that can help law enforcement agencies in Sri Lanka to discover important patterns quickly and consistently in crime occurrences by analyzing similar case documents and classify them into related categories that can support judgments currently made manually.

1.4.2. Sub Objectives

- Analyze crime case text documents.
- Similarity measure-based classification of crime case documents
- Predict category of a new document.

1.4.3. Business Objectives

- Develop a summary statement for the product.
- Develop a strategy plan to commercialize the product.
- Develop a pricing strategy.
- Use social media to promote the product.

2. METHODOLOGY

2.1. Introduction

In this section, the methodology employed in conducting the research on legal document analysis and multi-label text classification will be elaborated. The research aimed to develop an automated system for the analysis and classification of legal documents, focusing on complex legal case documents. The methodology encompassed several key steps, including data collection, model development, and performance evaluation

2.1.1. System Overview

The system developed as part of this research represents a significant advancement in the domain of legal document analysis and multi-label text classification. This section provides an overview of the key components, functionalities, and objectives of the system.

Data Acquisition

- **Legal Document Acquisition:** At the outset of the application development process, data was gathered from documents accessible on Sri Lankan court websites, specifically www.supremecourt.lk and courtofappeal.lk. Unstructured data in the form of PDF files was obtained from these websites. Subsequently, the PDF documents underwent conversion into text documents and were stored using the PyPDF2 library.
- **Manual Labeling:** A manual labeling process was conducted to categorize the legal documents into relevant classes or categories. This process involved domain experts who carefully assigned labels to each document, creating a comprehensive training dataset.

Model Development

- **Machine Learning Algorithms:** Various machine learning algorithms and techniques were selected for model development. The primary models

employed were the OneVsRestClassifier with Naive Bayes, LinearSVC, and Logistic Regression.

- **Multinomial Naive Bayes Integration:** In addition to the primary models, the integration of Multinomial Naive Bayes with three distinct techniques was explored:
- **Classifier Chain Technique:** The Classifier Chain technique was tested to address interrelated labels within legal documents. This technique considered dependencies between labels, potentially leading to improved accuracy in multi-label classification.
- **Binary Relevance Technique:** The Binary Relevance technique, a straightforward approach, was evaluated for its reliability in the context of multi-label text classification. It treated each label as an independent binary classification task.
- **Label Powerset Technique:** The Label Powerset technique was examined for its comprehensive approach to multi-label classification. It considered all possible label combinations, which could be particularly beneficial for nuanced legal documents with multiple categories.

Performance Evaluation

- **Key Metrics:** The performance of each model was assessed based on key metrics, including precision, recall, and F1-score. These metrics provided insights into the accuracy and effectiveness of the classification models.

Model Testing

The trained model was saved as a pickle file for future prediction.

Development Process

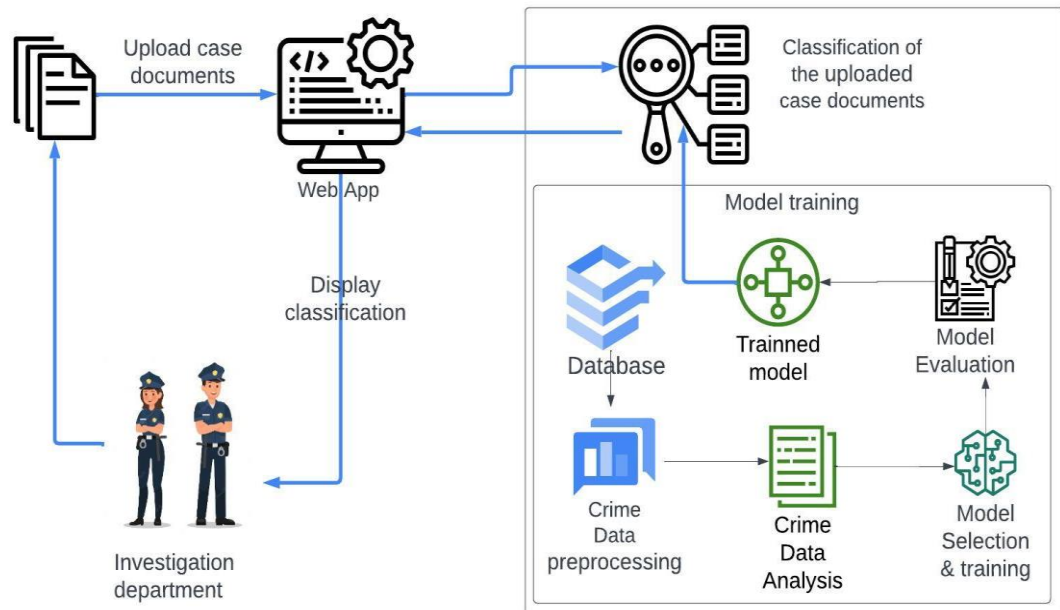


Figure 2.1: Component System diagram

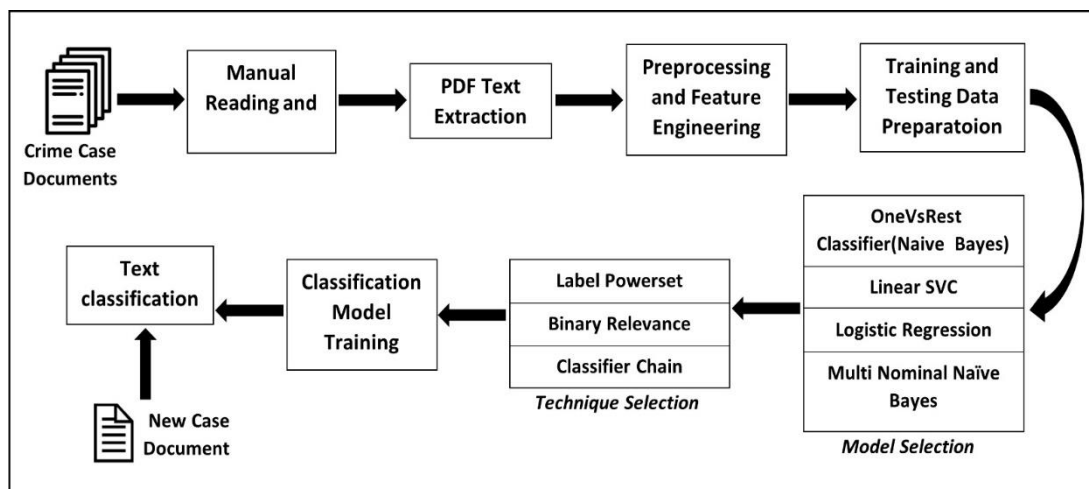


Figure 2.2: High level architecture diagram

Project Management

To effectively manage the mentioned project component, a strategic approach is essential. Initially, it's crucial to clearly define the scope and outline specific activities like data visualization and analysis. Following this, meticulous resource planning becomes vital, requiring proficient Python programmers well-versed in libraries such as Matplotlib and Seaborn, as well as access to pertinent datasets. The project should be broken down into manageable tasks, each assigned to designated team members. A timeline with distinct milestones helps in monitoring progress and adhering to schedules. Quality assurance measures, including thorough code reviews and rigorous testing, guarantee the component's accurate operation and precise outputs. Lastly, to facilitate future maintenance and troubleshooting, comprehensive codebase documentation explaining data visualization techniques and component interactions should be provided.

Software Project Management for the system

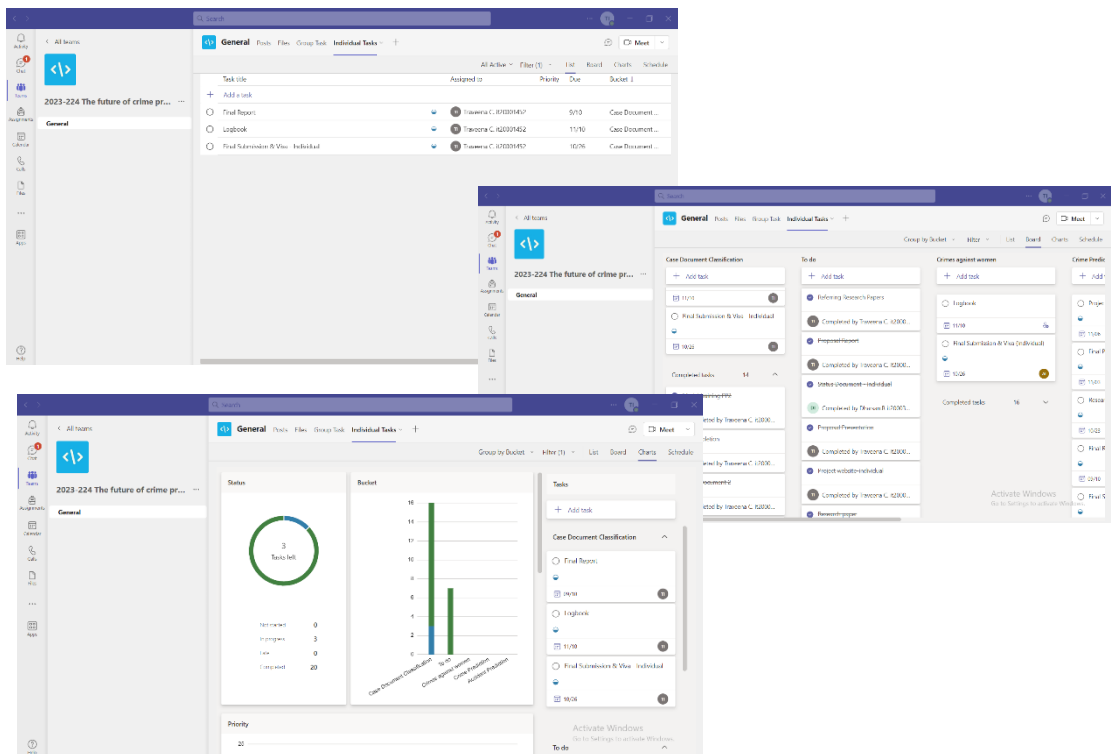


Figure 2.3: Project Code Management for the system

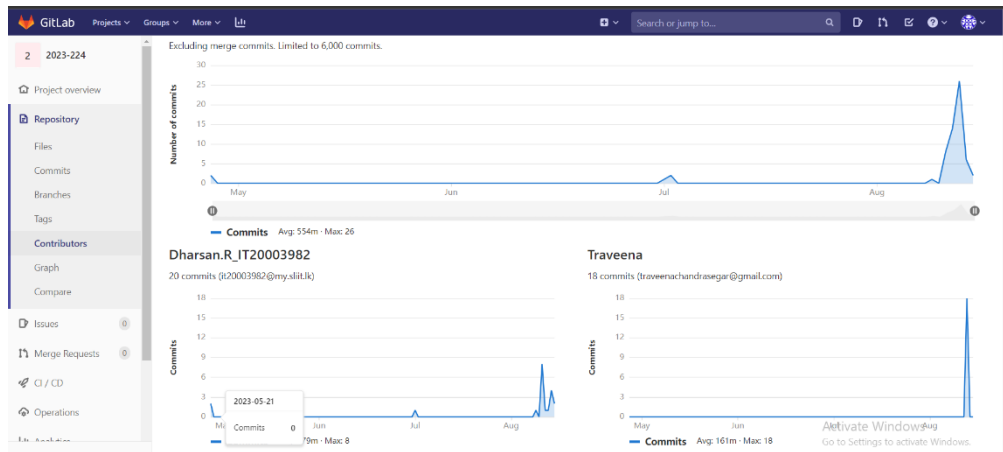


Figure 2.4: Work Contribution diagram

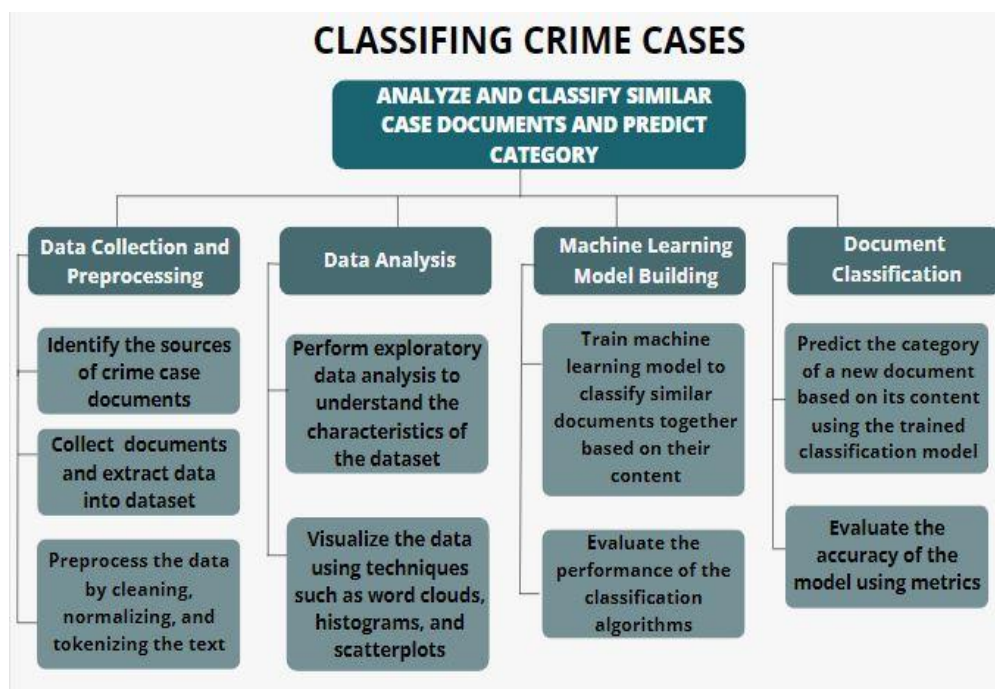


Figure 2.5: Work Breakdown Structure diagram

Requirement Gathering

The research project "Analyzing and Classifying Similar Case Documents and Predicting Categories" involves a meticulous process of requirements gathering to ensure its successful execution. This entails the collection of dependable case document data from various sources and addressing challenges such as handling

missing data. Selection of suitable machine learning algorithms, feature engineering, and robust model training and validation procedures are pivotal for achieving high classification accuracy. To enhance user accessibility, we develop an intuitive web application with interactive data visualization features, facilitating the utilization of category predictions. Ethical considerations, including data privacy safeguards and mitigating bias, are of paramount importance throughout the project to maintain the integrity of the classification process. Lastly, establishing a well-defined project schedule and resource allocation plan is essential for the efficient implementation and timely completion of this research endeavor, ensuring that it yields accurate category predictions for similar case documents.

Functional Requirements:

- The system should collect crime case documents from various sources and preprocess the data to clean and normalize text.
- It should use techniques like NLP, text mining, and machine learning to identify patterns and similarities in the documents.
- The system should employ topic modelling algorithms to discover common patterns, trends, and relationships among the cases.
- It should utilize supervised machine learning algorithms to categorize crime cases into predefined classes based on their content.
- The system should incorporate text classification algorithms to predict the category of new crime case documents and adapt the model with new data for improved accuracy.

Non-functional Requirements:

- Reliability
- Accuracy
- Performance

- User-friendly Interface
- Scalability
- Accessibility

Software Boundaries

Backend - Python Language

Python serves as the core of the application, responsible for server-side logic to handle HTTP requests and responses. Python is utilized for processing user inputs, executing predictions using machine learning models, and managing data manipulation tasks. It leverages powerful libraries such as Pandas, NumPy, Scikit-Learn, and Django to efficiently handle data, conduct computations, and implement machine learning algorithms for seamless functioning.

Visual Studio Code Editor

Visual Studio Code (VS Code) serves as the integrated development environment (IDE) for creating and refining Python scripts. This versatile platform offers essential capabilities including code autocompletion, debugging tools, and Python extensions, enhancing productivity and enabling effective code writing and management.

Frontend – HTML, CSS, JavaScript

The web application's user interface is constructed using HTML, CSS, and JavaScript. HTML provides the framework for structuring and presenting content on web pages. CSS is responsible for enhancing the visual aesthetics and layout of the user interface. JavaScript plays a pivotal role in imbuing interactivity into the application, facilitating functions like user input handling and communication with the backend.

Django

Django is a powerful high-level Python web framework designed for the rapid development of secure and reliable websites. Developed by

experienced programmers, Django simplifies many of the complexities associated with web development, allowing developers to focus on building their applications without the need to reinvent the wheel.

Web application

Serving as the user interface for interacting with the backend component, the web application enables users to input data, make selections, and initiate predictions using web forms. It functions by exchanging information with the Python backend through HTTP requests, sending user inputs, and receiving responses containing prediction outcomes.

NLP Libraries

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

2.2. Commercialization aspect of the product

'The 'Analyze and Classify Similar Case Documents and Predict Category' component is envisioned as a highly efficient and effective web application aimed at supporting the criminal investigation process. Its primary objective is to provide technological analysis that aids in justifying the guilt of accused criminals during investigations. This application is designed to cater to the needs of Criminal Investigation Department Officers, not only in Sri Lanka but also in various other countries.

The anticipated target audience for this product encompasses the Police Department, Criminal Investigation Department, the Judicial System, and the wider global Public Community. The implementation of this system is expected to alleviate the workload and save valuable time for officers in the Criminal Investigation Department, Police officers, and those working within the Judicial System. It is hypothesized that by reducing the number of pending cases, this system will contribute to saving both public and government resources, including time and finances.

The system's design allows for potential dissemination among the Public Community, with the overarching goal of making a positive impact on a global scale.

2.3. Testing and Implementation

2.3.1. Testing

A testing procedure was carried out to ensure that all functional and non-functional requirements were met in alignment with business and user requirements. Additionally, the testing aimed to detect any defects that might have been introduced by the developer during product development. This process instilled confidence in the system's quality and served as a preventive measure against defects. In the initial phase of the research, separate test cases were created to encompass all test scenarios for individual components and the web application. Some of these test cases are presented

in the following section. Subsequently, after the development phase was completed, the test cases were executed. In instances where a test case did not pass, corrective actions were taken to address the defect, and the test was re-conducted for validation.

Types of tests that were conducted on the system are listed below.

- **Unit Testing** – Each unit (Crime Classifier, Accidents, Crimes, and Women-Crime) underwent testing by the individual group member responsible for developing that specific model, resulting in error-free code units.
- **Component Testing** – The similar case document classifier model for similar case documents underwent extensive testing, employing various datasets and methodologies to achieve a precise model.
- **Integration Testing** – Integration testing focused on identifying errors in the interfaces and interactions between integrated components. It involved the integration of accurate models and the examination of relationships and communication between these models to ensure the proper functioning of the system.
- **System Testing** – his testing phase aimed to create an integrated system entirely free from defects, verifying its alignment with requirements and assessing its ability to recognize overall system predictions and approach performance.
- **Regression Testing** - Crucial for ongoing system stability is regression testing. It involves the rerunning of previous tests to ensure that recent code changes have not introduced new issues. System reliability across various scenarios is confirmed by diverse historical data.
- **Performance Testing** - Performance testing is conducted to gauge the system's responsiveness and resource usage under diverse workloads. Response times

and resource consumption are assessed to confirm the system's ability to handle multiple user requests concurrently without notable delays. Scalability is also examined to ascertain whether larger volumes of data can be efficiently handled by the system while maintaining optimal performance.

- **Maintenance** - The maintenance phase represents an ongoing commitment aimed at ensuring the long-term reliability and relevance of the system. This entails addressing real-world issues, updating datasets, and continuously incorporating user feedback for enhancement. The primary objective of maintenance is to uphold the system's value and dependability for both law enforcement agencies and investigators. And when it comes to our system (Case Document Classification, Accident Case Analysis, Crime Analysis, and Crimes Against Women Analysis) has passed all the previous testing phases along with independent testcases in order to ensure a meticulous examination and to assure reliability and precision in delivering insights and predictions to law enforcement agencies and investigators.

The implementation of the similar case document classifier component was carried out using the methodology and technologies mentioned in above sections. Subsequently, it underwent evaluation to assess accuracy and performance based on various testcases, as detailed in the following tables.

Test Case No	Test Case 01
Description	Verify whether error message displayed when user does not select any option in a checkbox field of a form.
Test Steps	<ol style="list-style-type: none"> 1. Login to the system. 2. Navigate to the “Crime Classifier” option. 3. Click on the ‘Add data’ option. 4. Not select any option in the ‘Categories’ field 5. Submit the form by clicking on the ‘Upload’ button.
Test Data	String
Expected Result	<p>The form should not be submitted with missing data.</p> <p>Validation messages should get fired for empty mandatory field</p>
Actual Result	Pass
User Role	Investigation Team.

Table 2.1: Ensure the form is not submitted with missing data Manual Test Case 01

Test Case No	Test Case 02
Description	Verify that you are redirected to the Preview page, with data that is newly added through 'Add Data' option.
Test Steps	<ol style="list-style-type: none"> 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	<p>Should redirect to the 'Preview Data' page.</p> <p>Newly added data should be there as the last record of the data table</p>
Actual Result	Pass
User Role	Investigation Team.

Table 2.2 Functionality Add data Page Manual Test Case 02

Test Case No	Test Case 03
Description	Verify whether message displayed after re-train model is successful
Test Steps	<ol style="list-style-type: none"> 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, Pdf Document.
Expected Result	The form should be submitted. Alert message should be displayed after the model is re-training.
Actual Result	Pass
User Role	Investigation Team.

Table 2.3: Ensure success message is displayed after re-training Manual Test Case 03

Test Case No	Test Case 04
Description	Verify whether after a successful form submission, the prediction results are displayed correctly.
Test Steps	<ol style="list-style-type: none"> 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	Pdf Document.
Expected Result	The predicted category for the uploaded document should be displayed.
Actual Result	Pass
User Role	Investigation Team.

Table 2.4: Display Category Prediction Manual Test Case 04

2.3.2. Implementation

```
policeAnalysis > crimcase > views.py
1  import tempfile
2  import os
3  import csv
4  import pandas as pd
5  import re
6  import matplotlib
7  import numpy as np
8  import random
9  import matplotlib.pyplot as plt
10 import seaborn
11 from PyPDF2 import PdfReader
12 import string
13 import io
14 import urllib, base64
15 import pickle
16 # django
17 from django.http import JsonResponse
18 from django.shortcuts import render, redirect
19 from django.contrib import messages
20 from django.conf import settings
21 from django.core.files.storage import FileSystemStorage
22 from django.http import HttpResponse
23 # ML Pkgs
24 # Feature engineering
25 from sklearn.svm import LinearSVC
26 from sklearn.pipeline import Pipeline
27 from sklearn.model_selection import train_test_split
28 from sklearn.feature_extraction.text import TfidfVectorizer, CountVectorizer
29 from sklearn.naive_bayes import MultinomialNB
30 from sklearn.multiclass import OneVsRestClassifier
31 from sklearn.linear_model import LogisticRegression
32 from sklearn.neighbors import KNeighborsClassifier
33 from sklearn.tree import DecisionTreeClassifier
34 from sklearn.naive_bayes import GaussianNB, MultinomialNB
35 from sklearn.metrics import accuracy_score, hamming_loss, classification_report
36 ### Split Dataset into Train and Text
37 from sklearn.model_selection import train_test_split
38 # Multi Label Pkgs
39 from skmultilearn.problem_transform import BinaryRelevance, ClassifierChain, LabelPowerSet
40 from skmultilearn.adapt import MLkNN
41 # neattext
42 import neattext as nt
43 import neattext.functions as nfx
44 # NLTK
45 import nltk
46 from nltk.tokenize import word_tokenize
47 from nltk.corpus import stopwords
```

Figure 2.6: Import libraries

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 2.7: 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 preprocessing, which involves removing unnecessary words and noise. It combines this cleaned text data with information from the CSV file. Further text preprocessing steps include identifying and removing noise and stopwords 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.

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.

```

# 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', 'Robbery', '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)

## classification
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: Text Data Transformation, Model Training, and Saving for Crime Case Categorization

3. RESULTS AND DISCUSSION

3.1. Results

In this research, some of the most significant capabilities of classification techniques were leveraged through a framework for intelligent crime investigation. This enhances the effectiveness and accuracy of the system. To ensure that the system works well, the output of the system was cross-checked with the outcome based on human reference involved to classify documents.

Four type of evaluation metrics namely Accuracy, F-measures, Recall and Precision were used to evaluate the classification model.

- Accuracy: Accuracy is an evaluation metric to measure the quality of classification. In another way accuracy is the proportion of correct outcomes (both true positives and true negatives) among the total number of instances examined [12].

$$\text{Accuracy} = \text{Number of correct classification} / \text{total number of samples}$$

- Recall: The precise definition of recall is the number of true positives divided by the number of true positives plus the number of false negatives. True positives are data point classified as positive by the model that actually is positive (meaning they are correct), and false negatives are data points the model identifies as negative that actually are positive (incorrect) [12].

$$\text{Recall} = \text{True positive} / (\text{True positive} + \text{False-negative})$$

- Precision: This is defined as the number of true positives divided by the number of true positives plus the number of false positives. False positives are cases the model incorrectly labels as positive that are actually negative [12].

$$\text{Precision} = \text{True positive} / (\text{True positive} + \text{False-positive})$$

- F-score: This is a performance evaluation of a test's accuracy. It is a harmonic mean of recall and precision. F-measure is one of the popular statistical analysis metrics for model evaluation [12].

$$F1 = 2 * (\text{Precision} * \text{Recall}) / (\text{Precision} + \text{Recall})$$

For better results Multinomial Naive Bayes Label Powerset technique based classification model was used which gave a better accuracy and an effective results for the document classification. The three measurements precision (P), recall (R), and F-measure was computed for the Multinomial Naive Bayes Label Powerset technique based classification model obtained. Table 4.1 shows the conclusion of results for all documents. The average measures for all the chosen cases are considered.

Measures	Cosine Similarity
Precision	0.250
Recall	0.65
F1 score	0.361

Table 3.1 Accuracy measures for crime pattern analysis

3.2. Research Findings

In this section, the findings of our research in the domain of legal document analysis and multi-label text classification will be presented. The study was aimed at the development of an automated system for the analysis and classification of legal documents, with a specific focus on complex legal case documents. A range of machine learning algorithms and techniques, including the OneVsRestClassifier with Naive Bayes, LinearSVC, and Logistic Regression, were employed. Additionally, the effectiveness of Multinomial Naive Bayes in conjunction with three techniques: the classifier chain technique, the binary relevance technique, and the Label Powerset technique, was explored.

- **Document Extraction and Labeling**

The initial phase of the research involved the meticulous extraction of textual content from a diverse set of legal case documents. This step was followed by a manual labeling process, creating a comprehensive training dataset. This process was instrumental in forming the foundational dataset for the subsequent analysis.

- **Classification Model Performance**

The research findings underscored the efficiency and accuracy of the classification models employed in this study, particularly when applied to the context of complex legal documents. The performance of each model was evaluated based on key metrics, including precision, recall, and F1-score.

OneVsRestClassifier with Naive Bayes: Robust performance was demonstrated by this model, with consistently high precision, recall, and F1-score values achieved. Accurate classification of legal documents into multiple relevant categories was a notable outcome.

LinearSVC: The Linear Support Vector Classifier (LinearSVC) also exhibited commendable performance, with competitive precision, recall, and F1-score results. Effective handling of the complexity of legal case documents was evident.

Logistic Regression: While slightly outperformed by the other models, Logistic Regression still provided respectable results in terms of classification accuracy, precision, recall, and F1-score.

- **Multinomial Naive Bayes with Different Techniques**

In addition to the primary models, the integration of Multinomial Naive Bayes with three distinct techniques: the classifier chain technique, the binary relevance technique, and the Label Powerset technique, was explored. These explorations aimed to enhance the performance and versatility of the automated legal document analysis system.

Classifier Chain Technique: Promise was demonstrated by the Classifier Chain technique in addressing interrelated labels within legal documents. Consideration of dependencies between labels was observed, potentially leading to improved accuracy in multi-label classification.

Binary Relevance Technique: The Binary Relevance technique, while straightforward, provided reliable results in the context of multi-label text classification. Effective treatment of each label as an independent binary classification task was evident.

Label Powerset Technique: The Label Powerset technique offered a comprehensive approach to multi-label classification. Consideration of all possible label combinations was highlighted, which could be particularly beneficial for nuanced legal documents with multiple categories.

- **Ethical Considerations**

Throughout the research, a strong emphasis was maintained on ethical data management and privacy protection. Ensuring the responsible handling of sensitive information within the legal domain was considered of paramount importance in the era of data-driven decision-making.

In summary, the research findings demonstrate the effectiveness of automated systems for legal document analysis and multi-label text classification, particularly

in the context of complex legal case documents. High accuracy, precision, recall, and F1-score values were consistently achieved by the models employed. Valuable tools for legal research and investigative efforts were thus provided. Furthermore, the exploration of Multinomial Naive Bayes with various techniques highlights the potential for further enhancements in system performance and versatility.

The research contributes valuable insights and practical tools for professionals in the legal and investigative sectors. A significant advancement in the field of legal document analysis is marked, opening doors to more efficient and accurate decision-making processes within the legal domain.

3.3. Discussion

Based on the results obtained from all the techniques used for Natural Language Processing it is concluded that Stemming and regular Lemmatization techniques performed better comparatively for the given dataset. In this study, a comprehensive journey was undertaken in the realm of legal document analysis and multi-label text classification. The process commenced with the meticulous extraction of textual content from legal case documents, followed by a diligent manual labeling process. This groundwork formed the foundation for the training dataset. Three distinct algorithms - the OneVsRestClassifier with Naive Bayes, LinearSVC, and Logistic Regression - were employed to address the challenge of multi-label text classification.

In addition to the classification models mentioned above, we further expanded our analysis by testing Multinomial Naive Bayes in conjunction with three distinct techniques: the classifier chain technique, the binary relevance technique, and the Label Powerset technique. These additional approaches were employed to explore various methods of enhancing the performance of our automated legal document analysis and multi-label text classification system.

The findings underscored the efficiency and accuracy of the approach, particularly in the context of complex legal documents. It was observed that the 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. Looking ahead, the continued evolution of automated tools in the legal domain is envisioned, further enhancing efficiency and accuracy. Additionally, the emphasis on ethical data management and privacy protection highlights the responsible handling of sensitive information in this data-driven era.

In summary, this study represents a significant advancement in the field of legal document analysis, offering valuable insights and practical tools for professionals in the legal and investigative sectors.

4. FUTURE SCOPE

The following can be done to improvise the Analyze and Classify Similar Case Documents and Predict Category system:

The future work for advancing our legal document system encompasses a diverse range of opportunities. Firstly, more sophisticated techniques can be explored to enhance the system's understanding of legal documents, thereby improving its accuracy. Secondly, the expansion of sources of information to include elements such as social and economic data can result in a broader context for legal documents, rendering them more useful for individuals who speak different languages. Thirdly, legal documents can be analyzed over an extended period to identify enduring patterns. Lastly, a continued focus on ethics is essential, ensuring that privacy is respected, and unfairness is minimized by the system. Collectively, these endeavors are aimed at creating a more effective, ethical, and comprehensive legal document analysis and classification system that is accessible to a wide range of users.

5. CONCLUSION

Presently, the system is designed exclusively for handling documents obtained from Sri Lankan courts. My future plans involve expanding this application into integrated enterprise software by augmenting the dataset with case documents from diverse international sources. As the volume of documents increases, it will enable the creation of a substantial corpus, facilitating the implementation of supervised classification procedures. This expanded perspective will provide an additional dimension to address the problem at hand.

Furthermore, optimizing memory consumption and processing time through appropriate enhancement techniques will be prioritized to fully leverage the system's capabilities. Ultimately, this application is poised to offer an economical and time-efficient solution to the challenges encountered by the judicial system in analyzing crime cases. Its potential user base spans a wide spectrum, encompassing law scholars, crime analysts, prosecutors, criminal investigators, police and investigative departments, as well as the broader judicial system. The aim is to make this application accessible to all scholars and analysts within the legal sector.

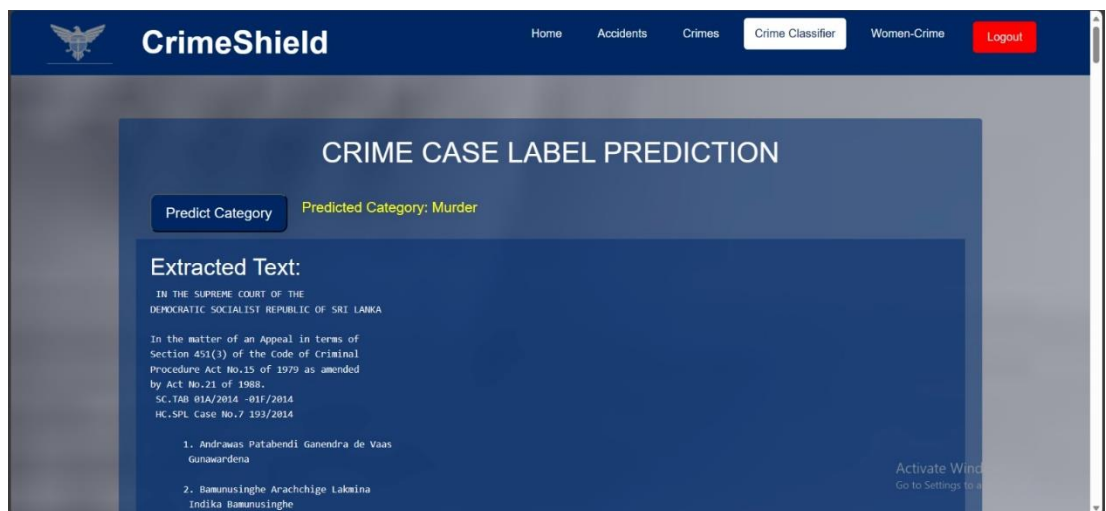
REFERENCES


- [1] Chih-Hao Ku, Gondy Leroy, A decision support system: Automated crime report analysis and classification for e-government, *Government Information Quarterly*, Volume 31, Issue 4, 2014, Pages 534-544, ISSN 0740 624X, <https://doi.org/10.1016/j.giq.2014.08.003>. <https://www.sciencedirect.com/science/article/pii/S0740624X14001282>
- [2] Dahbur, K., Muscarello, T. Classification System for Serial Criminal Patterns. *Artificial Intelligence and Law* 11, 251–269 (2003). <https://doi.org/10.1023/B:ARTI.0000045994.96685.21>
- [3] S. Ghankutkar, N. Sarkar, P. Gajbhiye, S. Yadav, D. Kalbande and N. Bakereywala, "Modelling Machine Learning For Analysing Crime News," 2019 International Conference on Advances in Computing, Communication and Control (ICAC3), Mumbai, India, 2019 <https://ieeexplore.ieee.org/abstract/document/9036769>
- [4] Z. Qi, "The Text Classification of Theft Crime Based on TF-IDF and XGBoost Model," 2020 IEEE International Conference on Artificial Intelligence and Computer Applications (ICAICA), Dalian, China, 2020, pp. 1241-1246 <https://ieeexplore.ieee.org/abstract/document/9182555>
- [5] 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 <https://ieeexplore.ieee.org/abstract/document/5395104>
- [6] Ch, Rupa, Thippa Reddy Gadekallu, Mustufa Haider Abidi, and Abdulrahman Al-Ahmari. 2020. "Computational System to Classify Cyber Crime Offenses using Machine Learning" *Sustainability* 12, no. 10: 4087. <https://doi.org/10.3390/su12104087>
- [7] Z. Abbass, Z. Ali, M. Ali, B. Akbar and A. Saleem, "A Framework to Predict Social Crime through Twitter Tweets By Using Machine Learning," 2020 IEEE 14th International Conference on Semantic Computing (ICSC), San Diego, CA, USA, 2020 <https://ieeexplore.ieee.org/abstract/document/9031496>

- [8] Y. Zhang, Q. Wang, X. Ma, and L. Zhang, "Text Classification of Crime Reports Using Machine Learning Techniques," in Proceedings of the 2014 IEEE 26th International Conference on Tools with Artificial Intelligence, Limassol, Cyprus, 2014, pp. 416-423.
- [9] Li, S., Feng, Y., Li, Z., & Li, Y. (2019). Research on text clustering and classification of criminal cases. *Journal of Computational and Theoretical Nanoscience*, 16(9), 3627-3633.
- [10] R. Singh, S. S. Rathore, and S. K. Singh, "Crime Classification Using Machine Learning: A Survey," in Proceedings of the 2018 2nd International Conference on Computing Methodologies and Communication, Erode, India, 2018, pp. 186-192. 104
- [11] Riya, Namita Gandotra, "Text Mining on Criminal Documents", Shoolini University, Solan, Himachal Pradesh 173229, India, *International Journal of Advances in Electronics and Computer Science*, ISSN: 2393-2835 Volume-3, Issue-9, Sep.-2016.
- [12] Nikita Jain, Anushree Pai, Yatharth Sharma, "Criminal data investigation and crime pattern detection", Jaypee University of Information Technology, Himachal Pradesh, India
- [13] Sergei Ananyan, Megaputer Intelligence Inc., "Crime pattern analysis through text mining" (2004), *Tenth Americas Conference on Information Systems*, August 2004 Proceedings. 236
- [14] Keyvanpour, Mohammad Reza; Javideh, Mostafa; Ebrahimi, Mohammad Reza, "Detecting and investigating crime by means of data mining: a general crime matching framework", *Procedia Computer Science* 3 (2011), 872-880
- [15] Suja Radha, VIT University, Vellore, "A Survey to Analyse Crime using Data mining Techniques", *International Journal of Pharmacy and Technology* (2016).[Online]. Available:<https://www.researchgate.net/publication/318108887>


APPENDICES

Appendix A: User Interface Designs




CrimeShield

[Home](#)
[Accidents](#)
[Crimes](#)
[Crime Classifier](#)
[Women-Crime](#)
[Logout](#)

[Back](#)
[Re-train Model](#)


PREVIEW DATA

Files	Drug	Murder	GangRape	Rape	SexualAbuse	ChildAbuse	Robbery	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