CRIME ANALYSIS

Project Report Submitted

In Partial Fulfillment of the Requirements

For the Degree Of

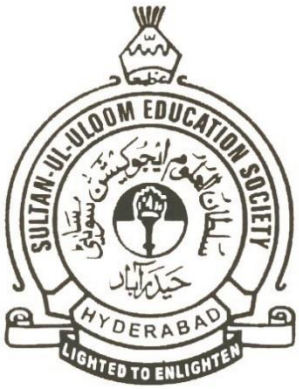
BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

Submitted By

MD MUBEEN ALI ZAKI (1604-19-733-033)
MIR AHMED ALI YASUBUDDIN (1604-19-733-038)
SYED ARFAAT (1604-19-733-051)



COMPUTER SCIENCE AND ENGINEERING DEPARTMENT
MUFFAKHAM JAH COLLEGE OF ENGINEERING & TECHNOLOGY
(Affiliated to Osmania University)
Mount Pleasant, 8-2-249, Road No. 3, Banjara Hills, Hyderabad-34

2023

MUFFAKHAM JAH
COLLEGE OF ENGINEERING &TECHNOLOGY
(Est. by Sultan-Ul-Uloom Education Society in 1980) (Affiliated to Osmania University, Hyderabad) Approved by the AICTE & Accredited by NBA

Date: 05 / 05 / 2023

CERTIFICATE

This is to certify that the project dissertation titled "CRIME ANALYSIS" being submitted by

1. MD MUBEEN ALI ZAKI (1604-19-733-033)
2. MIR AHMED ALI YASUBUDDIN (1604-19-733-038)
3. SYED ARFAAT (1604-19-733-051)

in Partial Fulfillment of the requirements for the award of the degree Of BACHELOR OF ENGINEERING IN COMPUTER SCIENCE AND ENGINEERING in MUFFAKHAM JAH COLLEGE OF ENGINEERING AND TECHNOLOGY, Hyderabad for the academic year 2022-23 is the bonafide work carried out by them. The results embodied in this report have not been submitted to any other University or Institute for the award of any degree or diploma.

Signatures:

Internal Project Guide
Ms.Manjusha Kalekuri
(Assistant Professor)

Head CSED
Dr. Syed Shabbeer Ahmad

External Examiner

8-2-249, Mount Pleasant, Road No.3, Banjara Hills, Hyderabad – 500 034
Phone: 040-23350523, 23352084, Fax: 040-2335 3428
Website: www.mjcollege.ac.in, e-mail: principal@mjcollege.ac.in

DECLARATION

This is to certify that the work reported in the major project entitled "CRIME ANALYSIS*"* is a record of the bonafide work done by us in the Department of Computer Science and Engineering, Muffakham Jah College of Engineering and Technology, Osmania University. The results embodied in this report are based on the project work done entirely by us and not copied from any other source.

1. MD MUBEEN ALI ZAKI (1604-19-733-033)
2. MIR AHMED ALI YASUBUDDIN (1604-19-733-038)
3. SYED ARFAAT (1604-19-733-051)

ACKNOWLEDGEMENT

We would like to extend our sincere gratitude to Dr. Syed Shabbeer Ahmad, Head of the Department, Computer Science and Engineering, Muffakham Jah College of Engineering and Technology, for his constant support and encouragement throughout the duration of our project, "Crime Analysis."

We would like to express our heartfelt thanks to our project guide, Ms. Manjusha Kalekuri, Assistant Professor, Department of Computer Science and Engineering, for her guidance and mentorship, which enabled us to successfully complete this project. Her insightful feedback, constructive criticism, and unwavering support were invaluable in shaping the direction of our work.

We would also like to thank our classmates, colleagues, and friends for their support and encouragement throughout the course of this project. Their feedback and suggestions helped us refine our work and improve its quality.

Finally, we would like to thank Muffakham Jah College of Engineering and Technology for providing us with the necessary resources and facilities to carry out this project. We are grateful for the opportunity to undertake this project, which has enabled us to develop our technical and research skills.

Batch-A4:

MD MUBEEN ALI ZAKI (1604-19-733-033)

MIR AHMED ALI YASUBUDDIN (1604-19-733-038)

SYED ARFAAT (1604-19-733-051)

ABSTRACT

PROBLEM STATEMENT:

To identify crime zones by performing Descriptive Analysis on the past data using K-Means and use multiple techniques to perform analysis to understand the alarming increase in crime rate, crime zones and ultimately prevent future crimes by deploying different strategies in the concerned regions.

OBJECTIVE:

To build an application that:

- Identifies Crime Hotspots (Districts) based on the past data using K-Means.
- Predicts future cluster trends for districts based on Estimated Crime Rates (Exponential Smoothing) using Random Forest Classifier.
- Predicts the Total IPC Crime for the states based on Projected Population (Geometric Growth Method) using Linear Regression.
- Forecasts the Total IPC Crime and Crime Rate of India using fbprophet.
- Build dashboards for visualization using Google Charts, Plotly and Choropleth.
- Web scrapes crime headlines and plots a heatmap to show the intensity of locations making it to the headlines frequently.

SOFTWARE USED:

- Language: Python
- IDE: Visual Studio Code
- Techniques: Machine Learning Algorithms(K-Means Clustering, Random Forest Classifier, Linear Regression and Time Series Forecasting), Visualization Dashboards, Web Scrapping model, Flask(Backend) and Frontend (HTML, CSS, JS and Tailwind CSS)

TABLE OF CONTENT

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1
INTRODUCTION

1.1 DOMAIN INTRODUCTION

Crime in India is a major concern for citizens, law enforcement agencies, and policymakers alike. The country has been grappling with various types of crime, including theft, robbery, assault, and cybercrime. The increasing crime rate not only affects the safety and security of the citizens but also has a negative impact on the economic development and social fabric of the country.

To address this issue, it is important to analyze crime data and identify crime zones, future crime zone trends, predict crime rates based on projected population and forecast crime rate. The scope of using Indian crime data for crime analysis is immense. By analyzing the data, we can understand the types of crimes that are prevalent in different regions and the factors such as population that contributes to them. This can help law enforcement agencies to prioritize their efforts and deploy resources more effectively.

Moreover, with the advancements in Machine Learning (ML) techniques, we can now use these data to identify crime-prone zones, predict future crime zone trends, predict crime rates based on population and forecast crime rate. This can help in preventing future crimes by deploying different strategies on different regions. By leveraging ML algorithms, we can also identify different regions that need the most effective crime prevention strategies that can be implemented in order to reduce crime rates.

In this project, we will explore the crime domain in India, the scope of using Indian crime data for crime analysis, and the potential of ML techniques to aid in the analysis of data so that various strategies can be adopted by the law enforcement agencies that can be deployed to prevent crime and improve the safety and security of citizens.

1.2 MACHINE LEARNING

What is Machine Learning?

In the real world, we are surrounded by humans who can learn everything from their experiences with their learning capability, and we have computers or machines which work on our instructions. But can a machine also learn from experiences or past data like a human does? So here comes the role of Machine Learning.

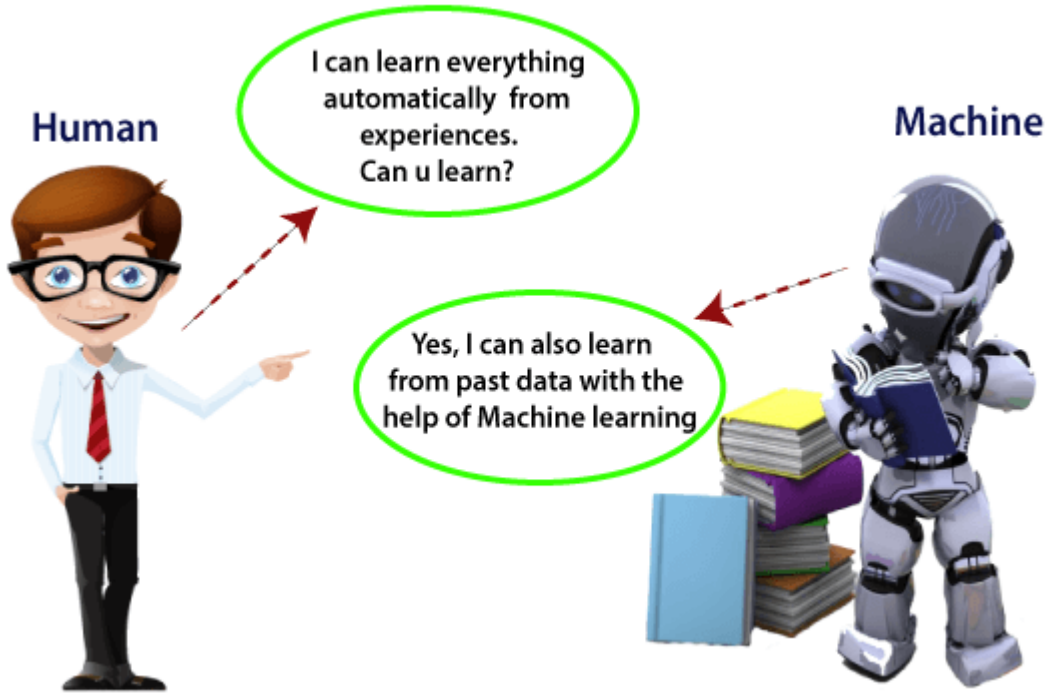


Fig 1.2.1: Machine Learning Introduction.

Machine Learning is said as a subset of artificial intelligence that is mainly concerned with the development of algorithms which allow a computer to learn from the data and past experiences on their own. The term machine learning was first introduced by Arthur Samuel in 1959. We can define it in a summarized way as:

Machine learning enables a machine to automatically learn from data, improve performance from experiences, and predict things without being explicitly programmed.

With the help of sample historical data, which is known as training data, machine learning algorithms build a mathematical model that helps in making predictions or decisions without being explicitly programmed. Machine learning brings computer science and statistics together for creating predictive models. Machine learning constructs or uses the algorithms that learn from historical data. The more we will provide the information, the higher will be the performance.

A machine has the ability to learn if it can improve its performance by gaining more data.

A Machine Learning system learns from historical data, builds the prediction models, and whenever it receives new data, predicts the output for it. The accuracy of predicted output depends upon the amount of data, as the huge amount of data helps to build a better model which predicts the output more accurately.

Suppose we have a complex problem, where we need to perform some predictions, so instead of writing a code for it, we just need to feed the data to generic algorithms, and with the help of these algorithms, machine builds the logic as per the data and predict the output. Machine learning has changed our way of thinking about the problem. The below block diagram explains the working of Machine Learning algorithm:

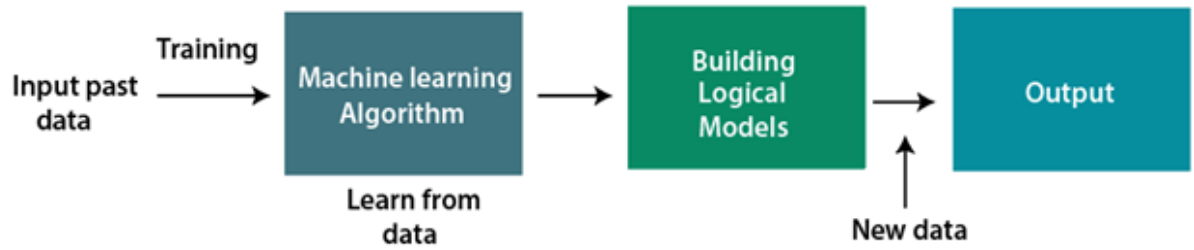


Fig 1.2.2: ML Model Training.

Features of Machine Learning:

- Machine learning uses data to detect various patterns in a given dataset.
- It can learn from past data and improve automatically.
- It is a data-driven technology.
- Machine learning is much similar to data mining as it also deals with the huge amount of the data.

Need for Machine Learning:

- Rapid increment in the production of data
- Solving complex problems, which are difficult for a human
- Decision making in various sector including finance
- Finding hidden patterns and extracting useful information from data.

Classification of Machine Learning:

At a broad level, machine learning can be classified into three types:

1 Supervised learning

2 Unsupervised learning

3 Reinforcement learning

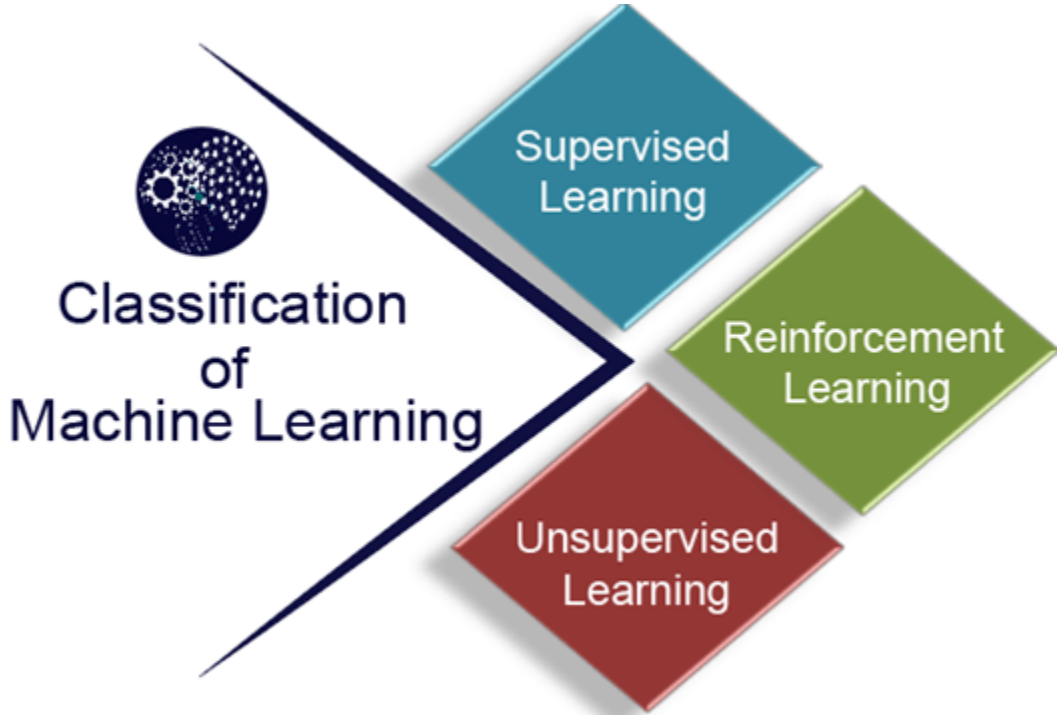


Fig 1.2.3: ML Classification.

1) Supervised Learning

Supervised learning is a type of machine learning method in which we provide sample labeled data to the machine learning system in order to train it, and on that basis, it predicts the output.

The system creates a model using labeled data to understand the datasets and learn about each data, once the training and processing are done then we test the model by providing a sample data to check whether it is predicting the exact output or not.

The goal of supervised learning is to map input data with the output data. The supervised learning is based on supervision, and it is the same as when a student learns things in the supervision of the teacher. The example of supervised learning is spam filtering.

Supervised learning can be grouped further in two categories of algorithms:

Classification
Regression
2) Unsupervised Learning
Unsupervised learning is a learning method in which a machine learns without any supervision.

The training is provided to the machine with the set of data that has not been labeled, classified, or categorized, and the algorithm needs to act on that data without any supervision. The goal of unsupervised learning is to restructure the input data into new features or a group of objects with similar patterns.

In unsupervised learning, we don't have a predetermined result. The machine tries to find useful insights from the huge amount of data. It can be further classifieds into two categories of algorithms:

Clustering
Association
3) Reinforcement Learning
Reinforcement learning is a feedback-based learning method, in which a learning agent gets a reward for each right action and gets a penalty for each wrong action. The agent learns automatically with these feedbacks and improves its performance. In reinforcement learning, the agent interacts with the environment and explores it. The goal of an agent is to get the most reward points, and hence, it improves its performance.

The robotic dog, which automatically learns the movement of his arms, is an example of Reinforcement learning.

1.3 VISUALIZATIONS

Data visualization is an essential tool for analyzing and communicating complex information effectively. It allows us to transform large datasets into meaningful insights that can be easily understood and interpreted by stakeholders. Whether you are working with financial data, marketing metrics, or scientific research findings, effective visualization techniques can help you uncover patterns, trends, and outliers that may not be apparent from a spreadsheet or a table of numbers. In this report, we will explore the benefits of data visualization and discuss some of the key techniques and best practices for creating clear, informative, and compelling visual representations of your data. By the end of this report, you will have a better understanding of how to leverage the power of data visualization to enhance your analysis and communication skills, and ultimately drive better decision-making.

CHAPTER 2

LITERATURE SURVEY

Given below are some strategies which was mentioned under literature review from the previous semester.

S.NO	TITLE	STRATEGY	ADVANTAGE	LIMITATION
1	Crime Analysis and Prediction Using Fuzzy C-Means Algorithm, B. Sivangaleela, S.Rajesh-2019	Fuzzy C-Means Algorithm to predict the crime rate	It gives the flexibility to express that data points can belong to more than one cluster	Apriori specification of the number of clusters
2	Crime Prediction Using KNN Algorithm, Akash Kumar, Aniket Verma, Gandhali Shinde, Yash Sukhdeve, Nidhi Lal-2020	KNN Algorithm to predict the crime rate	There's no need to build a model or tune several parameters	As dataset grows efficiency or speed of algorithm declines very fast
3	Crime Type and Occurrence	XGBoost, AdaBoost,	The accuracy has been high when	It does not perform so well

	Prediction Using Machine Learning Algorithm, ICAIS-2021	Random Forest and KNN	compared to other models.	on sparse and unstructured data
4	Multi-Step Occurrence Prediction of Urban Crimes with Enhanced GRU-ODE-Bayes Model, SMC-2022	Bayes Model to predict crime rate	It is extremely flexible to fit complex data sets	Bayesian inferences require skills to translate subjective prior beliefs into a mathematically formulated prior.
5	Crime Against Women : Analysis and Prediction Using Data Mining Techniques, COMITCon-2019	Data Mining Techniques to predict crime rate	It's an efficient, cost-effective solution compared to other data applications	A large database is required to go for mining thus making the process hard
6	Crime Analysis and Prediction Using Optimized K-Means	Optimized K-Means Algorithm to predict crime rate	Scales to large data sets and generalizes to different clusters	It has trouble clustering data where clusters are of varying sizes and density

	Algorithm, ICCMC-2020			

Table 2.1: Literature Survey

There are number of papers which we have analyzed in order to determine the technologies used. The review of literature will involve efficient and usable techniques such as fuzzy system and weka tool. [1] In the studied paper tool for forecasting the crime was created. The discrete choice model will be used in this case. The discrete choice model takes into account the choice of the criminal and location in order to forecast the crime. [2] In this paper a specific location Malaysia is considered. The crime location and mindset of the criminal will be considered in this case. Crime Forecasting is rarely used globally by police including Malaysia. In practice usually the police would target persons with their criminality and studying their strategy of implementing crime. The police will also monitor the current crime situation and will take necessary action when the crime index increases. Both of these scenarios require action taken after crime incurred. Therefore if crime forecasting can be adopted perhaps early crime prevention can be enforced. The aim of this study is to identify crime patterns in Kedah using univariate forecasting technique. Seventy six recent monthly data (January 2006 – April 2012) were obtained from IPK Alor Star with the permission from PDRM Bukit Aman. Exploratory Data Analysis (EDA) and adjusted decomposition technique were conducted in order to fulfill the objective of the study. The findings revealed that total crimes in Kedah were mainly contributed by type of property crime (80-85%) while violent crime has a small proportion only. Fortunately due to the productiveness of the police the property crime trend indicated curve declining pattern. [3]Special section of the crime forecasting is considered in this case. The crime forecasting is used so that crime can be controlled. No specific location is considered in this case. The GIS(Geographical Information System) is used inorder to detect the location where crime is happened. [4]This is book in which crime analysis is conducted. The mentally of criminal is considered in this case. The help of crime forecasting is also considered in this case.

Mugdha Sharma et al. proposed advanced ID3 algorithm for presenting importance-attribute significance on the attributes which has less values but higher importance, rather than the

attributes with more values and lower importance as well as solve the classification defect to choose attributions with more values. The analysis of the experimental data shows that the advanced ID3 algorithm gets more reasonable and more effective classification rules. In this Z- crime tool was also proposed to analyze the criminal activities through e-mailcommunication. SushantBharti et al. proposed hidden link algorithm to detect hidden links of the networks of co- offenders which show the possible future crime partner and different network beyond the real network. This paper also analyzes the centrality of node. This analysis describes the importance of node of the network. This is used to discover the strongest person, power of the person and role of the person in the network. This paper gave future approaches i.e. predictive approach in crime analysis which helps in stopping the crime before it occurs and also analyze the network of Co-offenders in India and predict the possible future network of offenders. ShijuSathyadevan et al. proposed Apriori algorithm to identify the trends and patterns in crime. This algorithm is also used to determine association rules highlighting general trends in the database. This paper has also proposed the naïve Bayes algorithm to create the model by training crime data. After testing, the result showed that Naive Bayes algorithm gave 90%accuracy. Prashant K. Khobragade et al. proposed Forensic Tool Kit 4.0 which provides remote data investigation and visualization analysis. In remote data, investigation includes to analyze process information, service information, driver information, network device, network information. This tool generates the file and analyzes the data. This tool is also used to analyze the victim system where the attack is occurring. With the help of crime investigation, the physical and logical memory data areanalyzed. K. ZakirHussain et al. used data mining techniques for analysis of the criminal behavior. This paper proposed criminal investigation analysis tool (CIA). This tool was used within the law enforcement community to help solve violent crimes. It was based on a review of evidence from the crime scene and from witnesses and victims. The analysis was done from both an investigative and a behavioral perspective. It provided insight into the unknown offender as well as investigative suggestions and strategies for interviews and trial.

Researchers have proposed a variety of data mining techniques to provide crime data analysis, crime prediction, criminal identification and crime hotspot area identification. Some of the papers are discussed here. Mehmet Sait, and Mustafa Gökpresented [5] the criminal prediction for finding the most probable criminal of a particular offense incident when the suspected list of offenders are provided with the criminal data which is generated synthetically using

Gaussian Mixture Model. The authors used Naïve Bayes Classifier and Decision tree for offender prediction method and compared its performance. As a result of the comparison the authors achieved that the Naïve Bayes Classifier consumed less execution time and performs better with 78.05% accuracy. AgarwalA., analyzed [7] various offenses done by offenders and predict the chance of each offense that can again be performed by that offenders. The authors used Apriori practice for frequent item set generation that can be done by the offenders. Ahishakiye, E., Anisha, and C.Dhanashreeapplied[6] J48 base model to predict crime category or level in certain location that will occur in the future. Sivaranjani, S., S. Sivakumari, and M. AashaPresented [8] crime analysis for six cities of Tamilnadu, India by using clustering practices kmeans, DBSCAN and Agglomerative clustering for grouping the similar patterns to recognize offenses and the authors conclude that DBSCAN clustering performs better with precision 0.95, recall 0.91 and F measure 0.93 for grouping the similar patterns to identify crimes in for six cities of Tamilnadu, India. The authors used KNN practice to extract and predicts future offenses that will occur in the future in six cities of Tamilnadu, India which have possibility of low, medium and high offense occurrence by visualizing on google map. Emmanuel A., et al. [8] Analyzed crime data by using support vector machine, naïve bayes, neural network and J48 and contrasts the techniques by using accuracy and execution time for predicting offense level as 'Low', 'Medium', and 'High'. As a result of the contrast the authors conclude that the decision tree (J48) consumed less execution time with 0.06 seconds and performs better with 100% accuracy for crime forecasting. Yerpude P., et al. applied [7] data mining practices from crime data for foreseeing features that affect the high or low crime rate in certain region. The authors used Random Forest, Naïve Bayes and Linear Regression for recognizing factors that affect the high crime rate and compared its performance. As a result of the comparison the authors conclude that the Random Forest performs better with 81.35% accuracy. Nafiz M., et al. introduced [6] CRIMECAST, a mathematical simulation tool that analyzes past crime trends, patterns, features affect crime, Crime occurrence frequency, crime taken place, crime happened time, type of crime and victims from past crime data up to 30 years to forecast future crime.Tahani A., et al. [8] analyzed two different crime data using Decision Tree and Naïve Bayesian classifier to locate the most probable crime locations and their frequent occurrence time using Apriori Algorithm. The authors introduced what kind of offense might happen next in a specific place within a certain time and combining crimes' dataset with its demographics information to capture the issues that might disturb the safety of neighborhoods. As a result of the comparison the authors conclude that the Naïve Bayes performs better with 51% accuracy for Denver and 54% for Los Angeles for crime prediction.

Thongsatapornwatana U., studied 0, [10] different researches which are used data mining techniques for crime data analysis and foreseeing. The author identifies the research gap and challenges from different studies and recommends different data mining techniques for finding the patterns and trends in crime data to help the researcher on crime data. Rasoul K et al., [9] analyzed the crime data using the data mining techniques such as K-means Algorithm for grouping the similar crime patterns for identifying crime in different years based on amount of crime occurrence during different years and recognizing the crime patterns and trends to suggest this way can be used to decrease and avoid crime for the coming future years. The author presented the effect of parameters such as effect of outlier in data preprocessing and introduced GA for outlier detection in data preprocessing stage. The following table describes the remaining various papers which are done on crime data mining and ensemble learning.

We consider some previous works pertaining to time series analysis that have been carried out by researchers in the past. We look at and analyze some of the proposed methods. Based on the analysis, we perform a critical evaluation so as to find the limitations of the existing system. This would strengthen our proposed approach toward time series analysis for the carried out research. Sunil Bhaskaran [11] conducted time series analysis for the long-term forecasting and scheduling of organized resources. The various application areas considered for the research were currents through resistors, spikes in stock prices, computer system resource planning, and time series data mining in biological datasets. These case studies may be applied to domains such as voltage variations, ECG fluctuations, climatic variations, and weather forecasting. Although the research was well executed, it was restricted to a few case studies. Petitjean et al. [12] proposed a method to deal with satellite picture time arrangement investigations to manage unpredictable tested arrangements. This would permit the correlation of time arrangement sets with the end goal that every component of the match has a diverse number of tests. Dynamic time traveling was considered to approve the equivalent and may have a couple of restrictions in the type and predetermined number of layouts and the need for real preparing models. This depends on the Levenshtein distance, which has its own disadvantages. For example, the numerical analysis did not indicate the quality of the crime results. In addition, it also required explicit activities for the change of one image into another. Chujai et al. [13] carried out a period arrangement investigation of family unit electric utilization. The examination centered on finding a model to determine power utilization in a family unit to identify a reasonable determining period (i.e., daily, weekly, monthly, or quarterly). The investigation was

performed using the ARIMA and autoregressive moving average (ARMA) models. The Akaike information criterion (AIC) and root-mean-square error (RMSE) were considered to determine the most reasonable estimating techniques. The results showed that the ARIMA model was the most appropriate method for monthly and quarterly estimation. While the research led to anticipated and considerable results, estimating solely based on models may not be precise. In addition, the daily, weekly, monthly, and even quarterly estimations have gradually expanded to half-yearly and yearly estimates. Devi et al. [14] directed a time arrangement investigation for stock pattern expectations, utilizing the ARIMA model for Nifty Midcap 50. The testing rule to approve the precision of the model was the AIC or the Bayesian information criterion (BIC). The mean total rate mistake, percent mean absolute deviation (PMAD), and percentage error precision were considered to determine the contrast between genuine verifiable information and figure information. The mistake percentage precision was 16.26, which is insignificant. Smith and Agarwal [15] looked into a method to show the likelihood of creating innovation estimating models using patent gatherings. The time arrangement displaying systems were connected to a group of United States Patent and Trademark Office (USPTO) licenses from 1996 to 2013. Holt–Winters exponential smoothing (HWES) and ARIMA were the methods applied for the equivalent. A few cross-approving strategies decided the best fitting models [15]. RMSE, mean absolute error (MAE), mean absolute percentage error (MAPE), and mean absolute scaled error (MASE) were utilized as mistake measurements. The examination legitimized the creation of determining models; however, the legitimacy of collecting licenses is, as yet, unexplored. Patent-gathering affiliations may affect the rate of progress. Boubacar et al. [16] recommended building a forecast strategy for sustainable power hot spots to accomplish the smart administration of a micro-grid framework. The purpose was to advance the use of a sustainable power source in an associated matrix and disconnected power frameworks. The methodology used wavelet decay and counterfeit neural systems for the multi-goal investigation of the time arrangement. Every facet of information was examined using Hurst segments; segments with low consistency potential were excluded. Removing two out of seven parts prompted a decrease in assets by 29% without influencing the execution of the anticipated calculation. While the asset decrease rate was not excessive, counterfeit neural systems have their own weaknesses, such as problems in model translation and poor execution on little datasets, and are computationally difficult to prepare. Laptev et al. [17] presented event forecasting by means of neural networks for Uber using time series analysis. The objective was to accurately predict completed trips during special events with optimized resource allocation and less waiting time. The work proposed end-to-end recurrent neural network architecture,

long short-term memory (LSTM), and a public M3 dataset for time series forecasting competitions. Uncertainty estimation and heterogeneous forecasting were the modeling aspects, whereas the results were validated using special event forecasting accuracy and general time series forecasting accuracy. The experiments proved that a single generic neural network model can produce high-quality forecasts. The researchers relied on three criteria for choosing the appropriate neural network model: number of time series, length of time series, and correlation among time series. If the three criteria are high in value, then the approach is feasible. Hyland et al. [18] proposed recurrent generative adversarial networks (RGAN) and recurrent conditional generative adversarial networks (RCGAN) to generate realistic real-value multi-dimensional time series. Using novel evaluation methods, they validated that RCGANs can generate time series data that can be used for supervised training. The real test data, however, displayed minor degradation in terms of evaluation. Maximum mean discrepancy as well as the ‘‘train on synthetic, test on real’’ (TSTR) and ‘‘train on real, test on synthetic’’ (TRTS) methods were used. The evaluation was performed using comparative analysis and interpolation. Hosseini et al. [19] analyzed an energy system in Sweden using time series analysis. The research considered total energy consumption as well as energy consumption in the industrial and residential sectors. The additive HWES method and regression analysis showed that energy use will decrease from 2014 to 2024. The HWES method considered fewer parameters, and the error structure was not too sophisticated; hence, the forecasting was not too accurate. The regression can be lengthy and complicated and may lead to erroneous and misleading results. Karmakar et al. [20] studied a time series model to predict the demand of jute yarn based on the minimum values of forecasting errors. Various techniques such as simple moving average, single exponential smoothing, trend analysis, Winters method, and Holt’s method were used for forecasting. The analysis showed that in terms of forecasting accuracy, Winters method provided the best performance. Mean absolute deviation, mean square deviation, and MAPE were considered for validating the accuracy of the results. The research was a mere case study of the existing time series analysis methods. Incorporating more techniques and accuracy validation methods may lead to different results.

Crime data analysis is taking onward attention to explore hidden patterns in crime data. Based on existing research, it has been observed that data mining techniques assist the procedure of crime patterns detection. To analyze the crime data, classification and machine learning algorithms are used. Yu et al. [28], employ an ensemble of data mining classification

techniques to perform the crime forecasting. A variety of classification methods such as: One Nearest Neighbor (1NN), Decision Tree (J48), Support Vector Machine (SVM), Neural Network (Neural) with 2 layer network, and Naïve Bayesian (Bayes) were used to predict the crime "hotspot". Finally the best forecasting approach was proposed to achieve the most stable outcomes.

In [24], fuzzy association rule mining approach was used for community crime pattern discovery. The discovered rules were presented and discussed at regional and national levels for crime pattern investigation. Levine [22], developed a spatial statistical program for the analysis of crime incidents or other point locations. It was designed to operate with large crime – incident dataset collected by metropolitan police departments. In [[30]], different hotspot mapping techniques such as: point mapping, thematic mapping of geographic areas (e.g. Census areas), spatial ellipses, grid thematic mapping and kernel density estimation (KDE) were used for identifying hotspot of crimes. Finally, hotspot mapping accuracy was compared in relation to the mapping technique that was used to identify concentrations of crime events and by crime type – four crime types were compared (burglary, street crime, theft from vehicles and theft of vehicles).

In [21], a comparative study was conducted between some free available data mining and knowledge discovery tools and software packages. The result showed that, the performance of the tools for the classification task is affected by the kind of dataset used and by the way the classification algorithms were implemented within the toolkits. In [23], a clustering based model was used to anticipate crime trends. Performance of clustering technique was analyzed in forming accurate clusters, speed of creating clusters, efficiency in identifying crime trend, identifying crime zones, crime density of a state and efficiency of a state in controlling crime rate. In [25], an experiment was conducted to obtain better supervised classification learning algorithms (Naïve Bayesian (0.898), k Nearest Neighbor (k-NN) (0.895) and Neural Networks (0.892), Decision Tree (J48) (0.727), and Support Vector Machine (SVM) (0.678)) to predict crime status. Two different feature selection methods were tested on real dataset for prediction. Chi-square feature selection technique was used to improve the performance of mining results. Malathi et al. [26], applied anomalies detection and clustering algorithms to predict the crime patterns and speed up the process of solving crime. MV algorithm, DBScan and PAM outlier

detection algorithm were used to assist in the process of filling the missing value and investigation of crime patterns. In [27], a comparative study was conducted between the violent crime patterns from the communities and crime unnormalized dataset and actual crime statistical data for the state of Mississippi. Linear regression, additive regression, and decision stump algorithms were implemented on the communities and crime dataset. The linear regression algorithm performed the best among the three selected algorithms. By considering the geographical approach, Nath et al. [[29]], applied a combination of k-means and weighting algorithm which show regional crimes on a map and cluster crimes according to their types.

Bruin et al., created digital profiles for all offenders by extracting the factors: crime nature, frequency, duration and severity from the crime database. Comparison of all individuals were performed on these profiles by a new distance measure and cluster them accordingly. Thongtae et al., delivered a comprehensive survey of efficient and effective methods on data mining for crime data analysis. They explored the illegal activities of professional identity fraudsters based on knowledge discovered from their own histories. Bagui et al., used WEKA for mining association rules, developing a decision tree, and clustering to retrieve meaningful information about crime from a U.S. state database.

CHAPTER 3

SYSTEM ANALYSIS

3.1 PROBLEMS WITH EXISTING SYSTEM

The limitations of the Existing Systems are mentioned in the literature survey along with the existing work. Some of the limitations of these systems are as follows:

- The analysis is based on scarce data which generates bias for identifying the hotspots.
- Most of the existing systems requires data to be manually entered to perform analysis of a region.
- Most of the existing systems uses only one algorithm which might not be the best technique to get the better insights of a region.
- There are no systems which uses projection techniques to provide future input values to the Machine Learning models.
- Most of the existing systems lack visualization dashboards for the data which is an essential part of analysis.

3.2 PROPOSED SYSTEM

The Proposed System uses the below mentioned techniques to overcome the drawbacks of the Existing Systems. It:

- Identifies Crime Hotspots (Districts) by fetching the past data of 12 Years (2001-2012) and calculating the cluster for each year based on the existing clusters using K-Means and returns the mode of the cluster which helps in getting better insights.
- Predicts future cluster trends for districts based on Estimated Crime Rates (Exponential Smoothing) using Random Forest Classifier.
- Predicts the Total IPC Crime for the states based on Projected Population (Geometric Growth Method) using Linear Regression.
- Forecasts the Total IPC Crime and Crime Rate of India using fbprophet.
- Has visualization dashboards developed using Google Charts, Plotly and Choropleth.
- Web scrapes crime headlines and plots a heatmap to show the intensity of locations making it to the headlines frequently.

CHAPTER 4

SYSTEM DESIGN

4.1 SYSTEM ARCHITECTURE

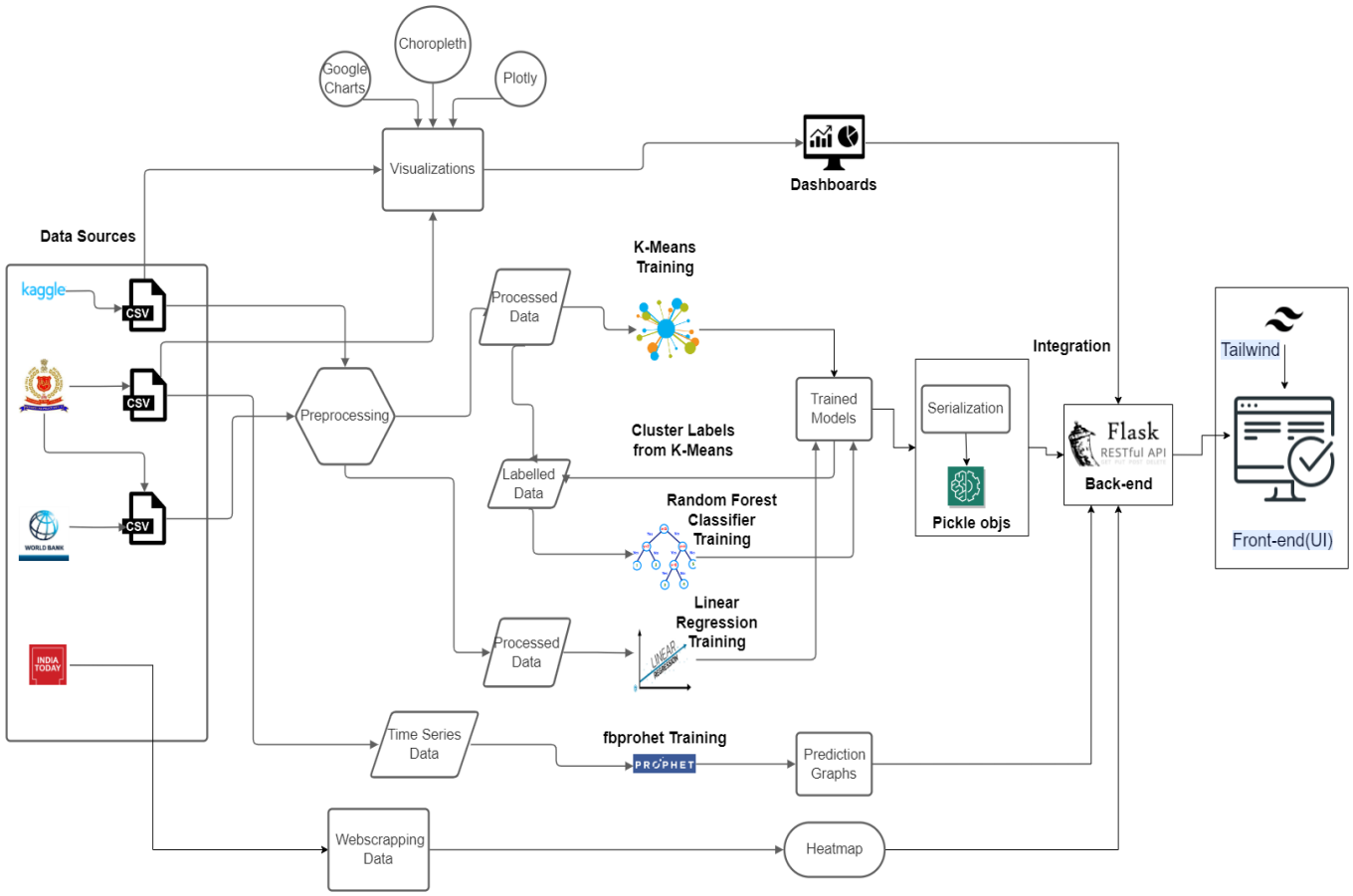


Fig 4.1.1: System Architecture

4.2 SYSTEM OVERVIEW

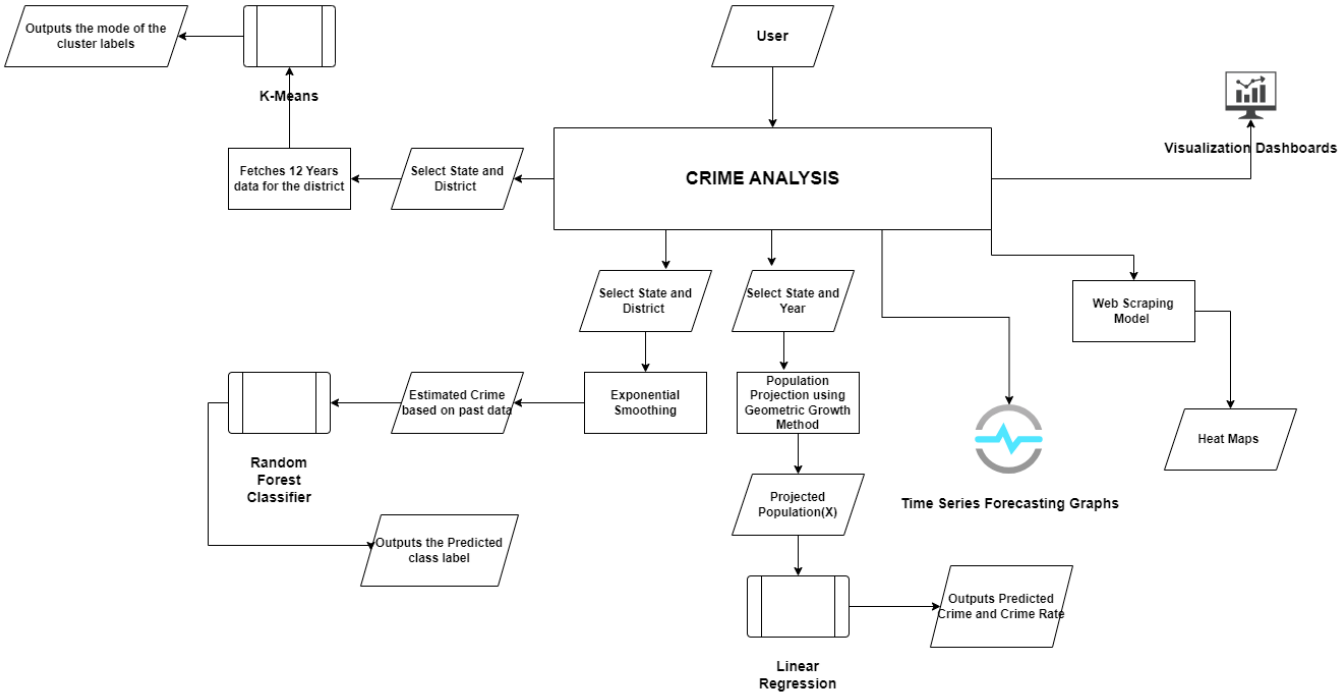


Fig 4.2.1: System Overview

4.3 UML DIAGRAMS

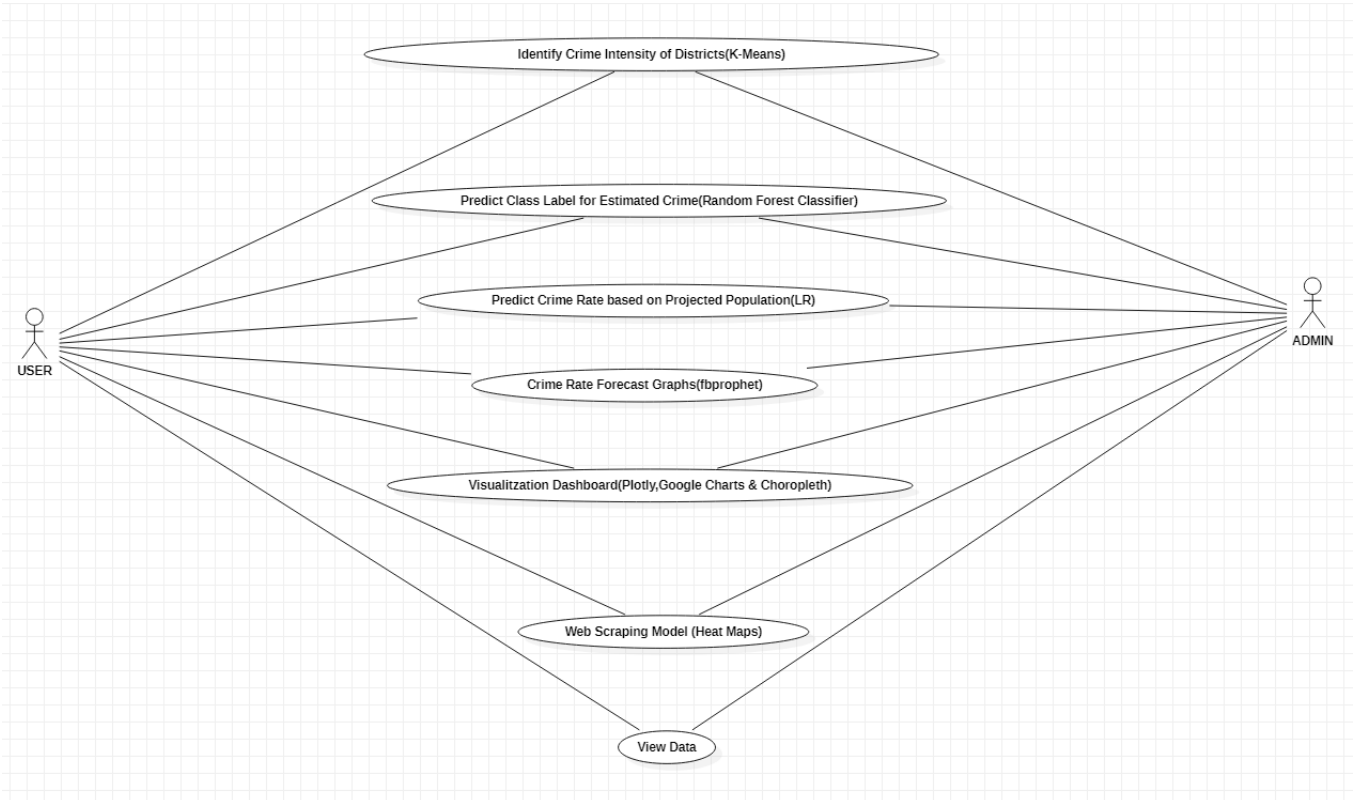


Fig 4.3.1: Use-Case Diagram

(i)

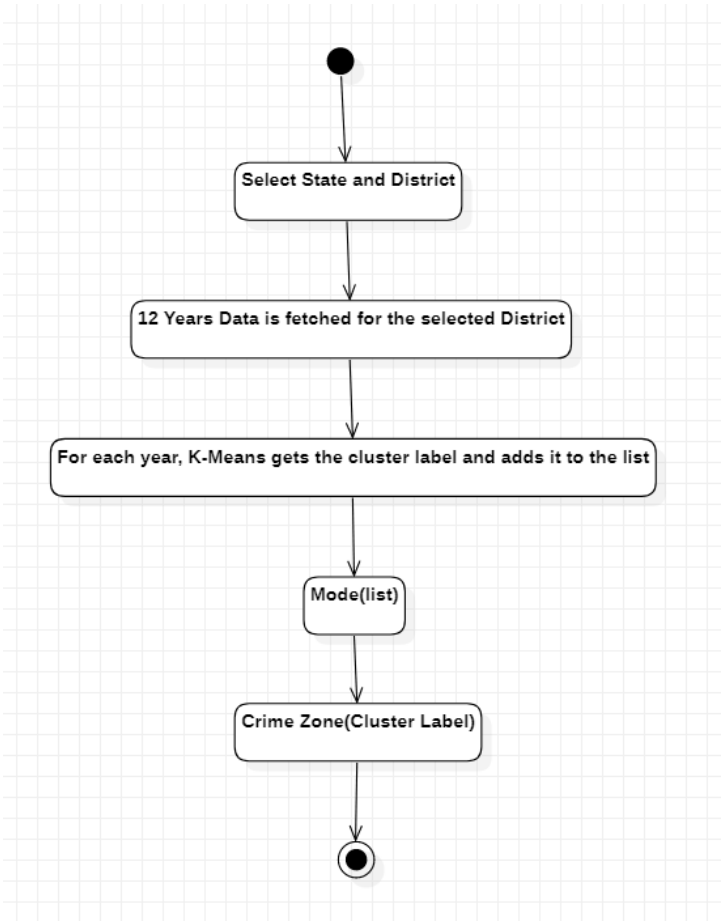


Fig 4.3.2: Activity Diagram for Clustering

(ii)

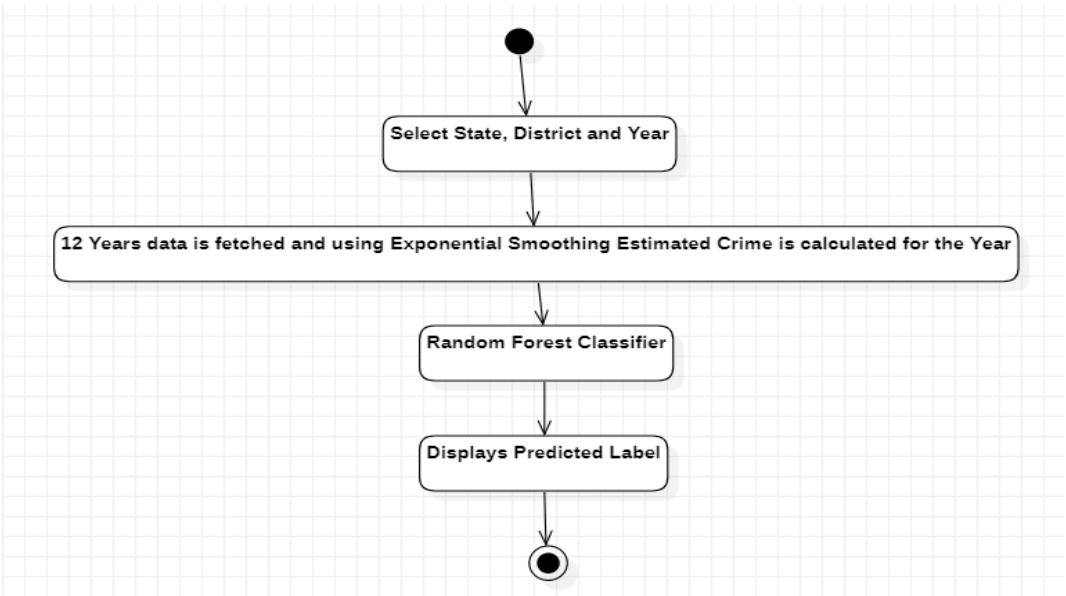


Fig 4.3.3: Activity Diagram for Classification

(iii)

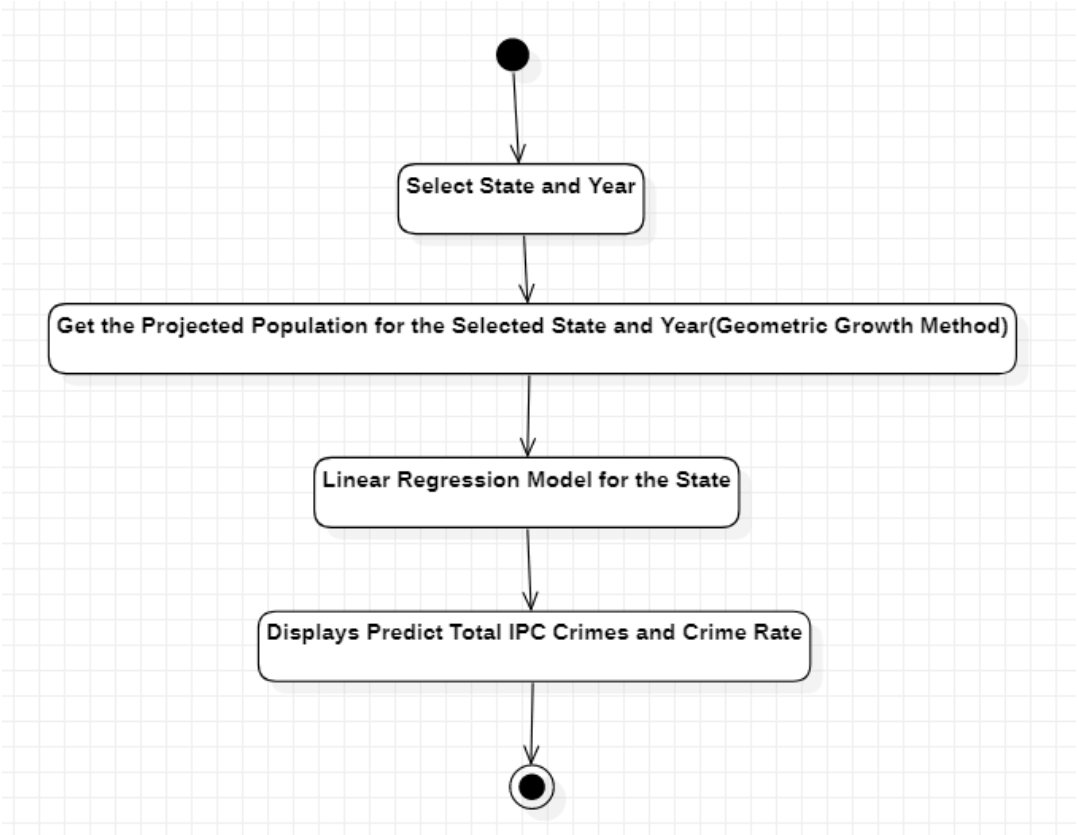


Fig 4.3.4: Activity Diagram for Linear Regression

(iv)

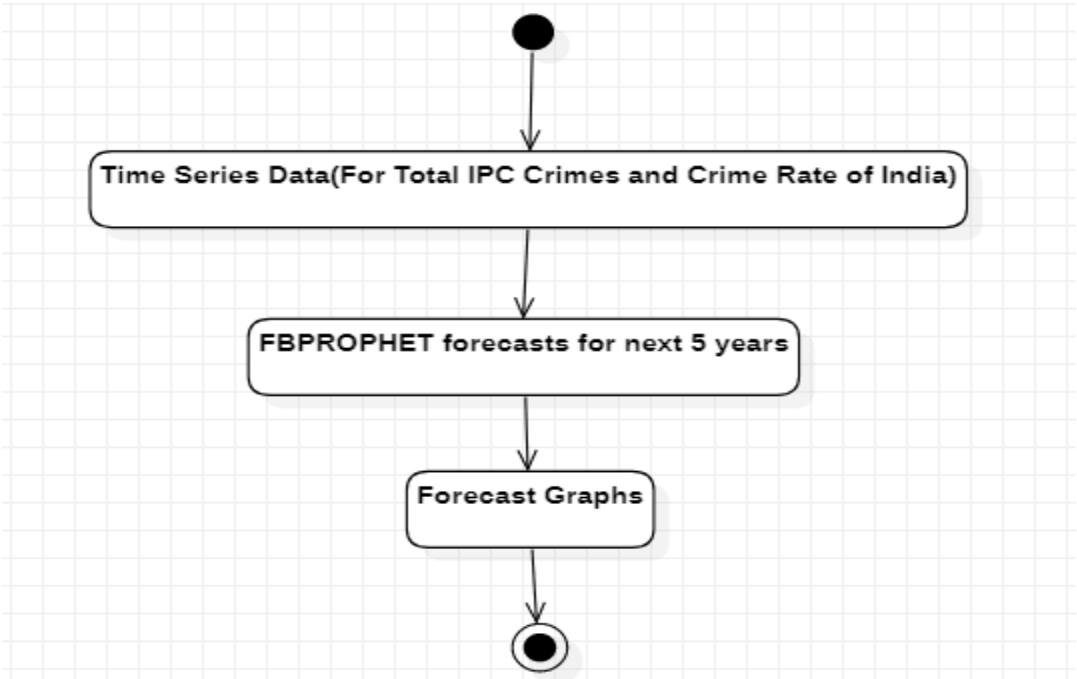


Fig 4.3.5: Activity Diagram for Time Series Forecasting

(vi)

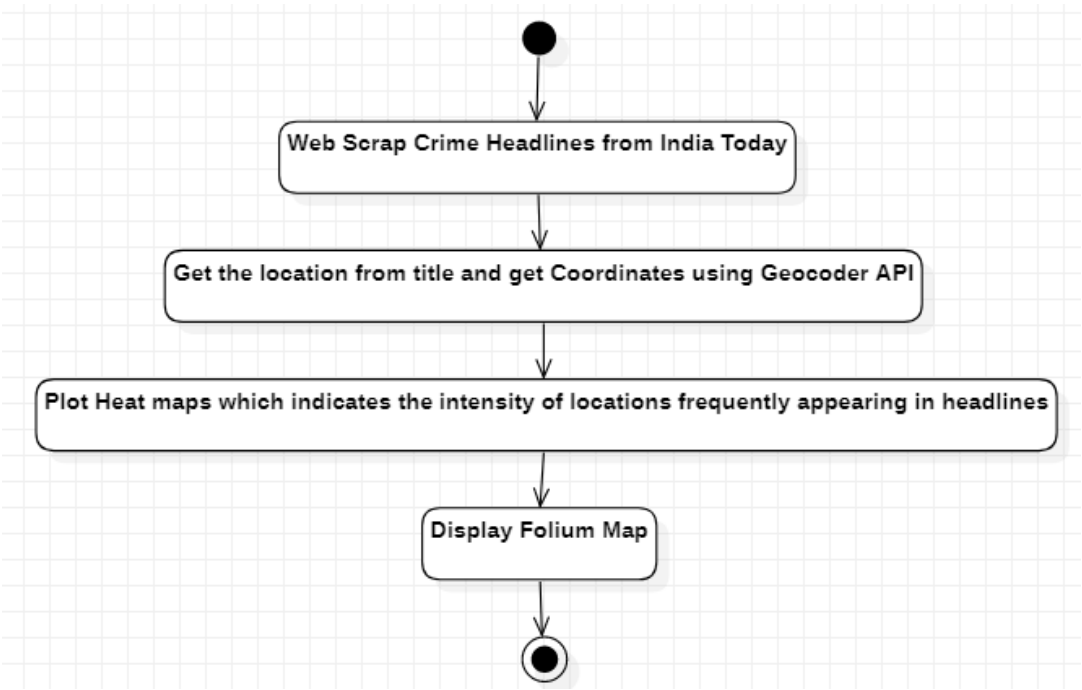


Fig 4.3.6: Activity Diagram for Web Scraping Module

(vii)

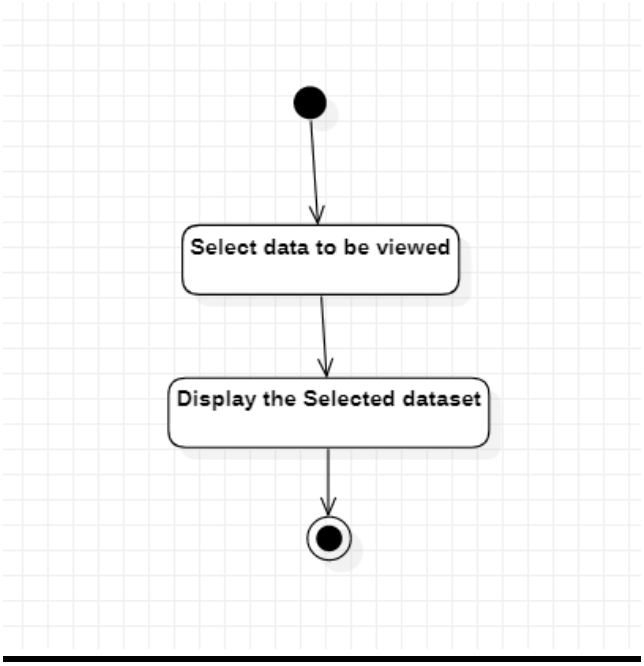


Fig 4.3.7: Activity Diagram for Viewing Data

(viii)

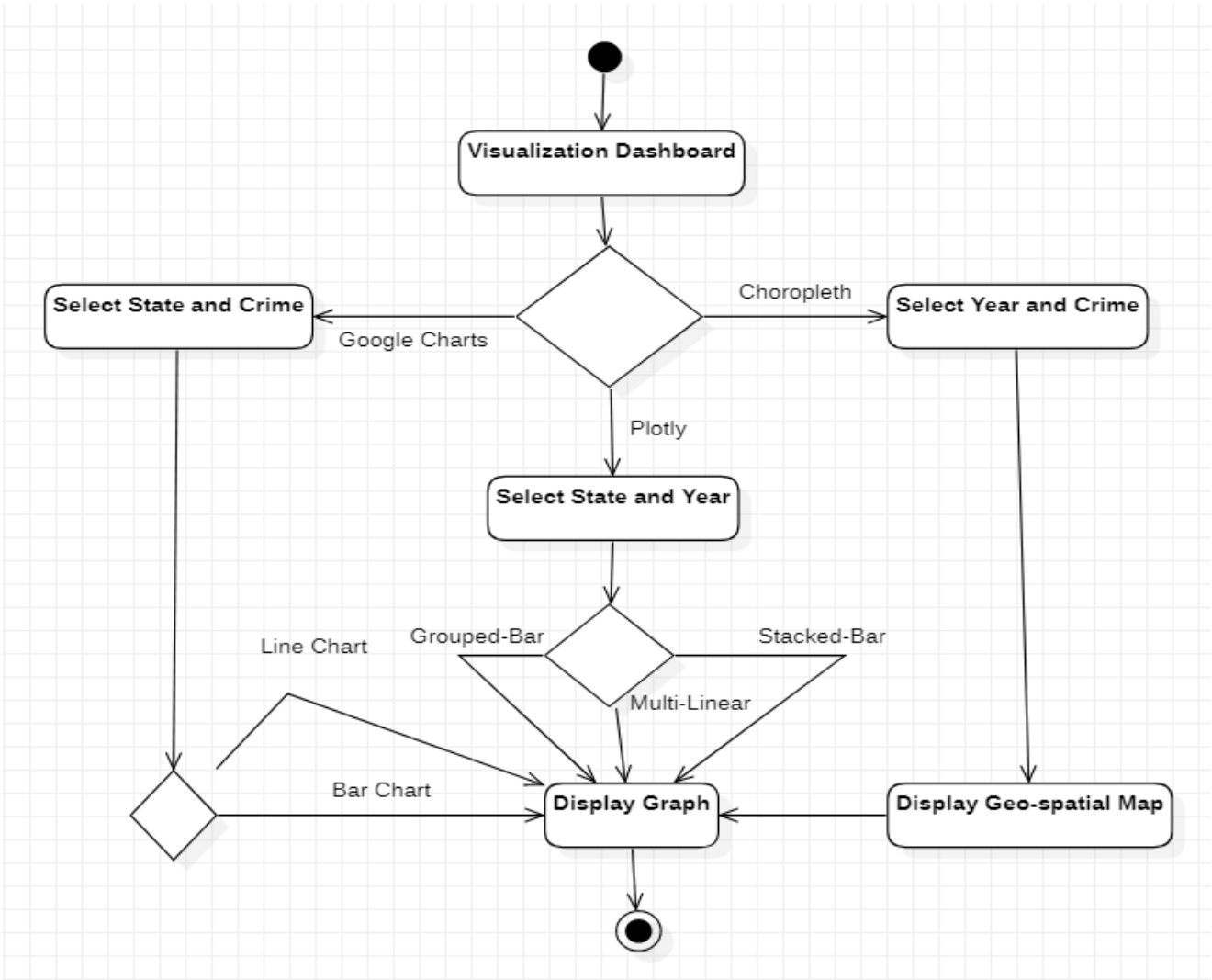


Fig 4.3.8: Activity Diagram for Visualization

CHAPTER 5

IMPLEMENTATION

5.1 K-MEANS CLUSTERING ALGORITHM

5.1.1 AIM:

To Identify Crime Hotspots (Districts) by fetching the past data of 12 Years (2001-2012) and calculating the cluster for each year based on the existing clusters using K-Means and return the mode of the cluster list which helps in getting better insights.

5.1.2 DESCRIPTION:

K-means is a popular clustering algorithm used in unsupervised machine learning. The goal of clustering is to group similar data points together into clusters, where the data points within a cluster are more similar to each other than to those in other clusters.

K-means works by randomly assigning K initial cluster centroids, where K is a user-defined parameter that represents the number of clusters to form. The algorithm then iteratively assigns each data point to its nearest centroid and updates the centroid location based on the new cluster assignments. This process is repeated until the centroids no longer move or a predefined stopping criterion is met.

The distance metric used to determine the nearest centroid can vary, but the most commonly used is Euclidean distance. Once the algorithm converges, the data points will be partitioned into K clusters, and each data point will be assigned to the cluster with the nearest centroid.

K-means has several advantages, including its simplicity, speed, and scalability. However, it also has some limitations, such as sensitivity to the initial centroid positions and the need to specify the number of clusters K in advance. Additionally, K-means may not perform well on data with irregular or non-convex shapes, and it can struggle to handle data with varying cluster densities.

Overall, K-means is a useful clustering algorithm that can be applied to a wide range of datasets and is particularly effective when the data has well-defined clusters.

5.1.3 ALGORITHM:

Step-1: Select the number K to decide the number of clusters.

Step-2: Select random K points or centroids. (It can be other from the input dataset).

Step-3: Assign each data point to their closest centroid, which will form the predefined K clusters.

Step-4: Calculate the variance and place a new centroid of each cluster.

Step-5: Repeat the third steps, which means reassign each datapoint to the new closest centroid of each cluster.

Step-6: If any reassignment occurs, then go to step-4 else go to FINISH.

Step-7: The model is ready.

5.1.4 IMPLEMENTATION:

Libraries Used: pandas, numpy, matplotlib, sklearn, seaborn, pickle, plotly, requests, geocoder and folium.

Dataset Used: 2001 – 2012 India (District Level) Crime Dataset from Kaggle(NCRB). First 5 rows of the dataset: *(Table 5.1.4.1:2001-2012 Dataset)*

STATE/UT	DISTRICT	YEAR	MURDER	ATTEMPT TO MURDER	RAPE	KIDNAPPING & ABDUCTION	DACOITY	ROBBERY	THEFT	HURT/GREVIOUS HURT
ANDHRA PRADESH	ADILABAD	2001	101	60	50	46	9	41	199	1131
ANDHRA PRADESH	ANANTAPUR	2001	151	125	23	53	8	16	366	1543
ANDHRA PRADESH	CHITTOOR	2001	101	57	27	59	4	14	723	2088
ANDHRA PRADESH	CUDDAPAH	2001	80	53	20	25	1	4	173	795
ANDHRA PRADESH	EAST GODAVARI	2001	82	67	23	49	4	25	1021	1244

Data Preprocessing:

- Significant Crime Categories Selected: Murder, Attempt to Murder, Rape, Kidnapping & Abduction, Dacoity, Robbery, Theft and Hurt / Grevious Hurt.
- "STATE/UT" and "DISTRICT" are encoded using LabelEncoder().
- Data Scaled using preprocessing.scale().

Elbow Method:

The elbow method is a way to determine the optimal number of clusters for a dataset in clustering analysis. It involves plotting the within-cluster sum of squares (WCSS) against the number of clusters and finding the point where the rate of decrease in WCSS slows down significantly, forming an "elbow" shape. This point represents the optimal number of clusters as it balances the trade-off between capturing the underlying patterns in the data and not overfitting the data.

Optimal number of clusters : 3

WCCS(kmeans.inertia_) = 67611.35

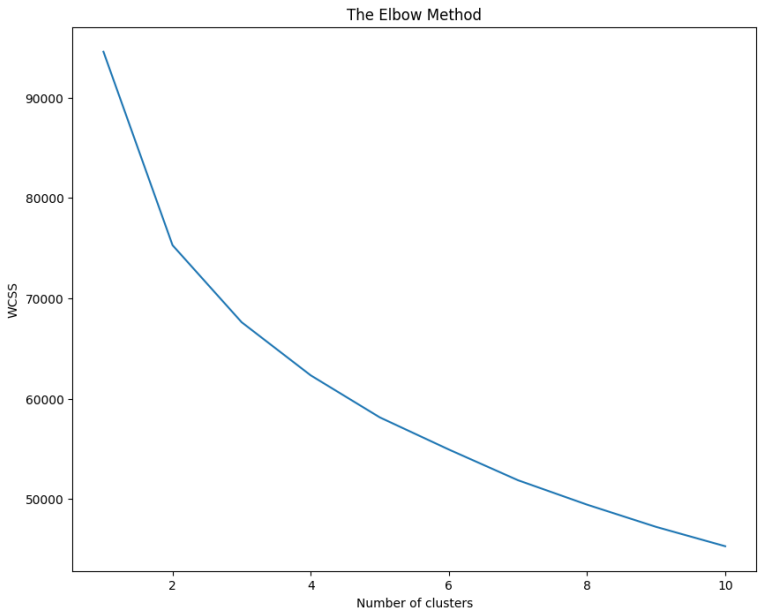


Fig 5.1.4.1: Elbow Method Curve

Model Training:

kmeans = KMeans(n_clusters = 3, init = 'k-means++', random_state = 42)

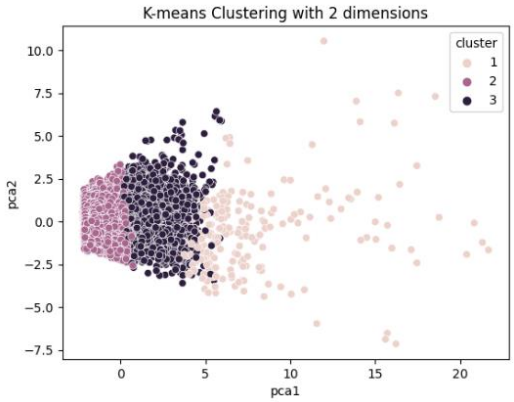
y_kmeans = kmeans.fit (ss)

Serialization:

Serialization is the process of converting an object into a format that can be easily stored or transmitted across different computing environments. Pickle is a module in Python that provides a simple way to serialize and deserialize Python objects.

- Dump : pickle.dump()
- Load : pickle.load()

Clusters Plotted Using PCA and Seaborn and Geospatially Plotted Using Folium and Geocoder:



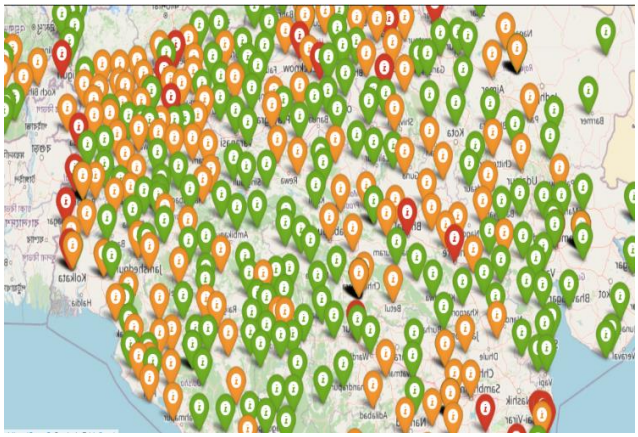


Fig 5.1.4.2: Clusters Plot (PCA)

Fig 5.1.4.3: Geospatial Clusters Plot

Cluster Centroids:

Cluster	Murder	Attempt to Murder	Rape	Kidnapping & Abduction	Dacoity	Robbery	Theft	Hurt
RED ZONE	158.9	160.1	96.8	232.6	26.0	212.9	3463.7	1391.2
GREEN ZONE	27.7	22.7	16.6	21.9	3.6	14.5	211.0	215.93
ORANGE ZONE	84.0	75.3	49.4	72.6	13.3	47.5	610.4	738.3

Table *5.1.4.2: Cluster Centroids*

Identification of Cluster Centroids:

1. Select State and District.
2. Data is fetched in backend for 12 Years for the selected District.
3. For each year, K-Means calculates the cluster label and appends it to the list clusters[].
4. Display Mode(clusters[]).

5.2 RANDOM FOREST CLASSIFIER

5.2.1 AIM:

To Predict future cluster trends for Districts based on Estimated Crime Rates (calculated using Exponential Smoothing based on the past data) using Random Forest Classifier.

5.2.2 DESCRIPTION:

A Random Forest Classifier is a type of supervised machine learning algorithm used for classification tasks. It's a type of ensemble learning method that combines multiple decision trees to improve the accuracy and robustness of the model.

In a random forest classifier, multiple decision trees are created using different subsets of the training data and different random combinations of the input features. Each decision tree produces its own prediction, and the final prediction is determined by combining the predictions of all the individual trees.

The random forest classifier is a powerful and versatile algorithm that can be used for both binary and multi-class classification problems. It can handle a large number of input features and is resistant to overfitting, making it a popular choice in many real-world applications.

5.2.3 ALGORITHM:

1. Randomly select a subset of the training data from the original dataset.
2. Randomly select a subset of the input features.
3. Build a decision tree using the selected data and features.
4. Repeat steps 1-3 to build multiple decision trees.
5. To make a prediction for a new data point, pass it through all the individual decision trees, and take the majority vote of the predictions as the final prediction.
6. Evaluate the accuracy of the model using a validation set or cross-validation.
7. Optionally, adjust the hyperparameters of the model (e.g., the number of trees, the depth of each tree, etc.) to optimize performance.
8. Once the model is trained, it can be used to make predictions on new, unseen data.

5.2.4 IMPLEMENTATION:

Libraries Used: pandas, numpy, sklearn, and joblib.

Dataset Used: 2001 – 2012 India (District Level) Crime Dataset from Kaggle(NCRB) (Unlabelled Data) + Cluster Labels from K-Means.

Target Variable(Y): Cluster Label.

Test Size: 0.2 (or) 20%.

Model Training:

```
random_forest = RandomForestClassifier(n_estimators=100)
random_forest.fit(X_train,Y_train)
```

Accuracy: 0.9732 (or) 97.32%.

Classification Report:

	precision	recall	f1-score	support
1	0.88	1.00	0.94	38
2	0.99	0.98	0.98	1248
3	0.94	0.95	0.95	434
accuracy			0.97	1720
macro avg	0.94	0.98	0.96	1720
weighted avg	0.97	0.97	0.97	1720

Fig 5.2.4.1: Classification Report

Serialization:

Serialization is the process of converting an object into a format that can be easily stored or transmitted across different computing environments. Joblib is a module in Python that

provides a simple way to serialize and deserialize Python objects.

- Dump : joblib.dump()
- Load : joblib.load()

Exponential Smoothing:

Exponential smoothing is a technique used in time series analysis to forecast future values of a variable based on past observations. It is a widely used method for smoothing out random variations or noise in the data and identifying underlying trends or patterns.

The basic idea behind exponential smoothing is to give more weight to recent observations while gradually decreasing the weight of older observations as they become less relevant. This is done by assigning a weight (also known as a smoothing factor) to each observation, where the weight of the most recent observation is highest and the weight of the oldest observation is lowest.

There are different types of exponential smoothing techniques, including simple exponential smoothing, double exponential smoothing, and triple exponential smoothing (also known as Holt-Winters forecasting). The choice of technique depends on the nature of the data and the specific forecasting problem.

Simple exponential smoothing is used for data with no trend or seasonality, while double and triple exponential smoothing are used for data with trend and seasonality. These techniques involve more complex calculations and take into account the trend and seasonality components of the data to generate more accurate forecasts.

Overall, exponential smoothing is a useful tool for forecasting future values of a time series based on past observations, and it can be particularly helpful for predicting short-term trends or identifying underlying patterns in the data. Simple Exponential Smoothing Algorithm:

1. Choose a smoothing factor alpha (0 < alpha < 1), which represents the weight given to the most recent observation.
2. Set the initial forecast F1 equal to the first observation in the time series.
3. For each subsequent observation in the time series, calculate the smoothed forecast Ft as follows:

4. Ft = alpha * Yt + (1 - alpha) * Ft-1; where Yt is the actual observation at time t, and Ft-1 is the previous forecast.
5. Use the most recent smoothed forecast Ft as the forecast for the next period.

Prediction of Future Cluster Trends:

1. Select the State, District and Year.
2. Past data is fetched for the selected District.
3. Estimated values for the selected Year for every Crime Category is calculated using Exponential Smoothing class based on the past data.
4. Input to Random Forest : Estimated Crime Values.
5. Displays the predicted label.

5.3 LINEAR REGRESSION

5.3.1 AIM:

- To Predict the Total IPC Crime for the States based on the Projected Population (Using Geometric Growth Method) using Linear Regression.

5.3.2 DESCRIPTION:

Linear regression is a statistical technique used to model the relationship between a dependent variable (usually denoted by "Y") and one or more independent variables (usually denoted by "X"). The objective of linear regression is to find a linear equation that best describes the relationship between the variables. The equation takes the form: Y = a + bX

where "a" is the intercept or constant term, and "b" is the slope of the line. The slope represents the change in Y for a unit change in X. The linear regression model estimates the values of "a" and "b" that minimize the difference between the predicted Y values and the actual Y values in the data set.

Linear regression can be used to make predictions about the dependent variable based on the values of the independent variables. It can also be used to test hypotheses about the relationship

between the variables. For example, linear regression can be used to determine if there is a statistically significant relationship between a person's age and their income.

Linear regression is widely used in many fields, including economics, finance, engineering, and social sciences, among others. It is a relatively simple and easy-to-understand method for modeling linear relationships between variables, but it is important to note that it assumes a linear relationship between the variables and that the data is normally distributed.

5.3.3 ALGORITHM:

1. Collect and prepare the data.
2. Select the independent variables.
3. Fit the linear regression model using a method such as ordinary least squares.
4. Evaluate the performance of the model by analyzing the residuals.
5. Use the model to make predictions.
6. Test the model on new data to assess its performance.

5.3.4 IMPLEMENTATION:

Libraries Used: pandas, numpy, sklearn, seaborn and joblib.

Dataset Used: 2001 – 2019 India (State Level) Total IPC Crime Dataset from NCRB.

Serial Number	Category	State/UT	Year	Crime Count	Population (in lakhs)	Crime Rate
29	Union Territory	A & N Islands	2001	658	3.56	184.831461
1	State	Andhra Pradesh	2001	130089	757.28	171.784545
2	State	Arunachal Pradesh	2001	2342	10.91	214.665444

3	State	Assam	2001	36877	266.38	138.437570
4	State	Bihar	2001	88432	828.79	106.700129

Table 5.3.4.1: 2001-2019 Dataset

Correlation:

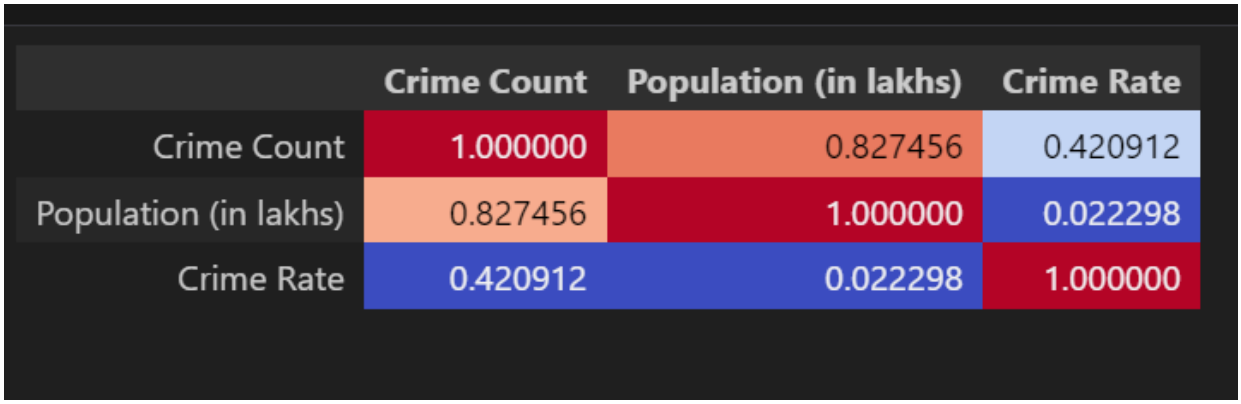


Fig 5.3.4.1: Correlation Plot

Independent Variable(X): Population(in Lakhs).

Dependent Variable(Y): Crime Count.

Model Training:

```
models = {}
states = list(set(df["State/UT"].values))
years = list(set(df["Year"].values))
for i in states:
    X = pd.DataFrame(df.loc[df["State/UT"]==i]["Population(inlakhs)"].values)
    Y = pd.DataFrame(df.loc[df["State/UT"]==i]["Crime Count"].values)
    linreg = LinearRegression()
    linreg.fit(X,Y)
    models[i] = linreg
```

Serialization:

Serialization is the process of converting an object into a format that can be easily stored or transmitted across different computing environments. Joblib is a module in Python that provides a simple way to serialize and deserialize Python objects.

- Dump : joblib.dump()
- Load : joblib.load()

Population Projection(Geometric Growth Method):

Population projection using geometric growth method is a way of predicting the future population of a particular area or region based on the assumption that the population will continue to grow at a constant rate over a given period. This method assumes that the rate of population growth remains constant and is expressed as a percentage or a decimal.

The formula for calculating population projection using geometric growth method is:

$$Nt = N0 * (1 + r)^t$$

;where Nt is the estimated population at time t, N0 is the initial population, r is the annual growth rate expressed as a decimal or percentage, and t is the number of years into the future.

For example, suppose the current population of a city is 1,000 and the annual growth rate is 2%. Using the geometric growth method, we can project the population of the city in 10 years as follows:

$$Nt = 1,000 * (1 + 0.02)^10$$

$$Nt = 1,219.91$$

Therefore, the projected population of the city in 10 years using geometric growth method is approximately 1,220.

It is important to note that population projection using geometric growth method is based on several assumptions, such as a constant growth rate and no significant changes in birth or death

rates. These assumptions may not always hold true in real-world scenarios, and the actual population growth may differ from the projected values.

Prediction of Total IPC Crime Count:

1. Select State and Year.
2. The Linear Regression model for the selected State is loaded.
3. 2001 and 2019 population is fetched for the selected State.
4. Based on this the Projected Population is calculated using Projection function.
5. Input to Regression model (X): Projected Population (in lakhs).
6. Displays the predicted Crime Count(Y) and Crime Rate(Y / X).

5.4 TIME SERIES FORECASTING

5.4.1 AIM:

- To Forecast the Total IPC Crime and Crime Rate of India for the next 5 years using fbprophet.

5.4.2 DESCRIPTION:

Time series forecasting is a technique used in statistics and data analysis to predict future values of a variable based on its past values. A time series is a set of observations that are measured at regular intervals over time, such as daily, weekly, monthly, or yearly. Examples of time series data include stock prices, weather patterns, and website traffic.

Time series forecasting can be used to identify trends, patterns, and seasonal fluctuations in the data, as well as to forecast future values. There are several methods that can be used for time series forecasting, including moving averages, exponential smoothing, and autoregressive integrated moving average (ARIMA) models.

The accuracy of time series forecasting depends on several factors, including the quality and quantity of historical data, the choice of forecasting method, and the complexity of the underlying patterns in the data. Time series forecasting is widely used in industries such as

finance, economics, marketing, and manufacturing to make informed decisions about future business strategies.

FBProphet is a time series forecasting library developed by Facebook's Core Data Science team. It is an open-source tool that uses a decomposable time series model with three main components: trend, seasonality, and holidays. FBProphet is designed to be easy to use, even for users without a background in time series forecasting or statistics.

One of the advantages of FBProphet is its ability to handle missing data and outliers in the time series. It also includes built-in functionality to detect and incorporate holiday effects, which can be important for accurately forecasting certain types of time series data.

FBProphet is implemented in Python and integrates well with popular data analysis libraries such as Pandas and NumPy. It provides a simple interface for specifying the time series data and configuring the forecasting model. Additionally, FBProphet includes visualization tools to help users better understand the underlying patterns in the data and the forecasted values.

FBProphet has been widely adopted by businesses and researchers for a variety of applications, including forecasting sales, website traffic, and stock prices, among others.

5.4.3 ALGORITHM:

1. Input the time series data and preprocess it to a format that can be used by FBProphet.

2. Decompose the time series into its three main components: trend, seasonality, and holidays.

3. Model the trend component using a piecewise linear or logistic growth curve.

4. Model the seasonality component using Fourier series with a specified number of terms.

5. Model the holiday component by detecting and incorporating holiday effects based on a user-defined list of dates and associated prior scale values.

6. Fit the model using a Bayesian approach to estimate the parameters of the trend, seasonality, and holiday components.

7. Generate future forecasts based on the fitted model, including point estimates and uncertainty intervals.

8. Visualize the forecasted values and underlying patterns in the data using interactive plots and trend/seasonality components.

5.4.4 IMPLEMENTATION:

Libraries Used: pandas, numpy, sklearn, datetime, fbprophet and plotly.

Dataset Used: 1981 – 2021 India Total IPC Crime and Crime Rate Data from NCRB.

Year(ds)	Total IPC(y1)	Crimeate(IPC per 100k)(y2)
1981-12-31	1385757	200.8
1982-12-31	1353904	192.0
1983-12-31	1349866	187.4
1984-12-31	1358660	184.7
1985-12-31	1384731	184.4

Table 5.4.4.1: 1981-2021 Dataset

Accuracy(For Crime Rate): 0.8885 (or) 88.85 %

Accuracy(Total IPC Crime): 0.9666 (or) 96.66 %

Time Series (ds): Year.

Variable: (i) Total IPC Crime(y)

(ii) Crime Rate(y)

Model Training:

```
m = Prophet()

m.fit(df)

future = m.make_future_dataframe(periods=5, freq='Y')

forecast = m.predict(future)

forecast_ouptut=forecast[['ds','trend','yhat_lower','yhat_upper','yhat']]

plot_plotly(m, forecast)
```

Forecast Visualization:

1. Select between Total IPC Crime and Crime Rate.
2. Display the plotly graphs in <iframe> .
3. The forecast for the selected option is displayed for the next 5 years.
4. Also the forecast components(trends and seasonality) graph is displayed.

Fig 5.4.4.1: Forecast Graphs

5.5 VISUALIZATION DASHBOARDS

5.5.1 AIM:

- To build Visualization Dashboards using Google Charts, Plotly and Choropleth to visualize large volume of data in the simplest possible way to better analysis.

5.5.2 DESCRIPTION:

Google Charts:

Google Charts is a web-based data visualization tool provided by Google. It allows users to create interactive charts and graphs that can be embedded on a website or shared via a URL.

Google Charts supports a variety of chart types including line charts, bar charts, area charts, scatter charts, pie charts, and more. Users can customize the appearance of their charts by selecting from a range of colors, fonts, and other design options.

Data can be inputted into Google Charts using a variety of methods, including manually entering data into a spreadsheet, importing data from a file, or connecting to a data source via an API.

Google Charts is easy to use and does not require any programming knowledge. However, advanced users can also utilize the Google Charts API to further customize and integrate the charts into their web applications.

Overall, Google Charts is a powerful and versatile tool for creating visual representations of data that can be easily shared and embedded on websites.

Plotly:

Plotly is a data visualization and analytics tool that provides a range of graphing and charting tools for creating interactive, web-based visualizations. It offers a variety of chart types including scatter plots, line charts, bar charts, heatmaps, and more. Plotly also provides a wide range of customization options that allow users to modify the look and feel of their visualizations, as well as the ability to embed and share their visualizations on websites, blogs, and social media platforms. In addition to its charting capabilities, Plotly also offers a cloud-based platform for data science collaboration and exploration. The platform includes tools for

data import, cleaning, and analysis, as well as machine learning and artificial intelligence algorithms for predictive modeling. It also provides a range of integrations with popular programming languages such as Python, R, and MATLAB, as well as with popular data analytics and visualization tools such as Tableau and Power BI.

Overall, Plotly is a powerful data visualization and analytics tool that offers a range of features for creating, customizing, and sharing interactive visualizations. Its cloud-based platform also provides a comprehensive set of tools for data science collaboration and exploration.

Choropleth:

A choropleth is a type of thematic map used to visualize data based on geographic regions. The regions are typically defined by administrative boundaries such as countries, states, or counties.

In a choropleth map, each region is shaded or colored to represent a specific data value. The shading or coloring is typically done on a gradient scale, with lighter colors representing lower values and darker colors representing higher values. This allows viewers to quickly see patterns and variations in the data across different regions.

Choropleth maps are commonly used to display a wide range of data, including population density, election results, and economic indicators such as GDP or unemployment rates. They are a powerful tool for visualizing spatial data and can help to identify trends and patterns that might not be immediately apparent from raw data.

5.5,3 IMPLEMENTATION:

Libraries Used: pandas, numpy, plotly and google charts library.

Front-end: HTML5, CSS, JS, Tailwind CSS and iframes.

Back-end: flask.

Dataset Used: (i) 2001-12 India (State Level) Dataset from NCRB (For Google Charts).

(ii) 2001-19 India (State) Dataset from NCRB (For Plotly and Choropleth).

Working:

1. Select the visualization from the options.
2. Select the State / type of crime / year.
3. Select the chart :
    - Bar graph
    - Line chart
    - Multi-Linear chart
    - Stacked-Bar graph
    - Grouped-Bar chart
4. Chart is displayed in the <iframe>.

5.6 WEB SCRAPING MODEL

5.6.1 AIM:

- To Web Scrap Crime Headlines and Visualize the intensity of locations making it to the headlines frequently on a heat map(Folium).

5.6.2 DESCRIPTION:

Web Scraping:

Web scraping is the process of extracting data from websites using automated software or tools. It involves writing code that can access and parse the HTML and CSS of web pages, allowing you to extract information such as text, images, links, and other data.

Web scraping is used for a variety of purposes, including market research, competitive analysis, data analysis, and content aggregation. Some common examples of web scraping include:

- Extracting product information from e-commerce sites

- Scraping job listings from career sites

- Collecting data on social media platforms

- Gathering news articles from media outlets

Web scraping can be done using programming languages like Python, JavaScript, and Ruby, and there are many libraries and tools available to help with the process. However, it is important to note that not all websites allow web scraping, and it is important to respect any terms of use or restrictions set by website owners.

BeautifulSoup and requests:

BeautifulSoup and Requests are two popular Python libraries used in web scraping.

Requests is a Python library that simplifies the process of making HTTP requests to web servers. It allows you to send HTTP/1.1 requests with ease, including GET, POST, PUT, DELETE, and other methods. With Requests, you can add content like headers, form data, multipart files, and parameters via simple Python libraries and access the server's response data.

BeautifulSoup, on the other hand, is a Python library for parsing HTML and XML documents. It provides a convenient way to extract and navigate data from HTML and XML files by creating a parse tree from HTML or XML source code. BeautifulSoup makes it easy to search for specific elements, extract data from tags and attributes, and navigate through the document hierarchy.

In summary, Requests is used to retrieve data from web pages and BeautifulSoup is used to parse the retrieved data to extract specific information. Together, they provide a powerful toolset for web scraping in Python.

Folium Open Street Maps:

Folium is a Python library used for visualizing geospatial data. It is built on top of the OpenStreetMap (OSM) platform, which is an open-source project that provides free map data and services.

Folium allows users to create interactive maps with a variety of customizable features such as markers, popups, layers, and various tilesets. These maps can be embedded in web pages or used for data analysis in Jupyter notebooks.

The library uses the Leaflet JavaScript mapping library under the hood, which provides a powerful set of tools for working with interactive maps. Folium also supports the use of GeoJSON and TopoJSON formats for displaying and manipulating geospatial data.

Folium makes it easy to add data to a map by allowing users to pass data as Pandas DataFrames or GeoPandas GeoDataFrames. This makes it simple to create choropleth maps, heatmaps, and other visualizations based on geospatial data. Overall, Folium is a powerful and flexible library for working with geospatial data in Python.

Geocoder API:

A geocoder API (Application Programming Interface) is a software tool that allows developers to access location data by sending requests to the API using programming code.

The geocoder API typically takes an address or location name as input and returns the corresponding geographic coordinates (latitude and longitude) for that location. Some geocoder APIs may also provide additional information such as address validation, geocoding quality indicators, or nearby points of interest.

Geocoder APIs can be used for a variety of purposes, such as mapping and navigation applications, location-based marketing, or analyzing geographic patterns in data. They are commonly used by developers building web or mobile applications that need to incorporate location data.

Examples of popular geocoder APIs include Google Maps Geocoding API, Bing Maps REST Services, and OpenStreetMap Nominatim. These APIs may have different pricing models, usage limits, and data quality, so developers should carefully consider their needs and budget when selecting a geocoder API.

RSS FEED:

An RSS feed, also known as Rich Site Summary or Really Simple Syndication, is a format used for distributing and sharing web content, such as blog posts, news articles, and podcasts, in a standardized way.

An RSS feed is essentially an XML file that includes the title, description, and link to each piece of content, along with other metadata such as the author, date of publication, and category. Users can subscribe to an RSS feed using a feed reader or aggregator, which will automatically pull the latest content from the feed and display it in a user-friendly format.

RSS feeds are commonly used by websites and blogs to syndicate their content and make it easily accessible to a wider audience. By subscribing to an RSS feed, users can stay up-to-date with their favorite websites and topics without having to manually check for updates.

5.6,3 IMPLEMENTATION:

Libraries Used: pandas, numpy, beautifulsoup, requests and folium.

Front-end: HTML5, CSS, JS, Tailwind CSS and iframes.

Back-end: flask.

Data: Web Scraping from https://www.indiatoday.in/crime

Working:

1. Select the Web Scraping Model.
2. RSS Feed from https://www.indiatoday.in/crime is displayed.
3. Crime Headlines are extracted from https://www.indiatoday.in/crime.
4. The location is extracted from the title and the occurrence of the location in all the titles is counted as frequency.
5. Coordinates of the location are fetched using geocoder() API.
6. Heat Map is plotted using Folium Open Street map.
7. Heat Map is displayed in the <iframe>.

5.6 FRONT-END

5.6.1 HTML5

HTML5 (Hypertext Markup Language version 5) is the latest version of the HTML standard used for creating and structuring content on the World Wide Web. It is a markup language that defines the structure and layout of web pages, and it is the foundation upon which most modern websites and web applications are built.

HTML5 includes several new features and improvements over previous versions of HTML, such as improved multimedia support, better form handling, and more advanced styling capabilities. It also introduces several new semantic tags that provide a more meaningful and structured way to describe the content of a web page.

One of the major improvements of HTML5 is its multimedia support. HTML5 provides built-in support for audio and video playback without the need for third-party plugins like Flash. It also includes the `<canvas>` element, which allows developers to create dynamic graphics and animations using JavaScript.

HTML5 also introduces several new form elements and attributes that make it easier to create and validate forms. For example, it includes new input types like "email", "date", and "range", and it supports new attributes like "required" and "pattern" for more advanced validation.

HTML5 also includes several new semantic tags that provide a more meaningful and structured way to describe the content of a web page. For example, it includes tags like `<header>`, `<nav>`, `<footer>`, and `<article>`, which help to improve the accessibility and SEO (search engine optimization) of a website.

Overall, HTML5 is a powerful and versatile language that offers many new features and capabilities for creating dynamic and interactive web applications. Its improved multimedia support, better form handling, and more advanced styling capabilities make it an essential tool for modern web development.

Sure, here are some technical descriptions of HTML5:

1. Improved multimedia support: HTML5 introduces the `<audio>` and `<video>` elements, which provide native support for playing audio and video content directly in the browser

without the need for plugins like Flash. These elements also support several different codecs, making it easier to deliver multimedia content across different devices and platforms.

2. New form elements and attributes: HTML5 includes several new form elements and attributes that make it easier to create and validate forms. These include input types like "email", "date", and "range", as well as new attributes like "required" and "pattern" for more advanced validation.

3. New semantic tags: HTML5 introduces several new semantic tags like `<header>`, `<nav>`, `<footer>`, and `<article>`. These tags provide a more meaningful and structured way to describe the content of a web page, which can improve accessibility, SEO, and overall page structure.

4. Canvas and WebGL: HTML5 includes the `<canvas>` element, which allows developers to create dynamic graphics and animations using JavaScript. It also includes the WebGL API, which provides hardware-accelerated 3D graphics in the browser.

5. Offline capabilities: HTML5 provides a way to store data locally on the user's device, which allows web applications to work offline. This is achieved through the use of the `localStorage` and `sessionStorage` APIs.

6. Responsive design: HTML5 provides new features that make it easier to create responsive web designs that adapt to different screen sizes. These include the `<picture>` element, which allows you to specify different image sources for different screen sizes, and the `<video>` element, which allows you to specify different video sources for different devices.

7. Better accessibility: HTML5 includes several new features that improve accessibility for users with disabilities, such as the `alt` attribute for images and the `<audio>` and `<video>` elements with built-in closed captioning and transcript support.

5.6.2 CSS

CSS (Cascading Style Sheets) is a language used to describe the presentation and styling of HTML and XML documents, including the layout, colors, fonts, and other visual aspects of a webpage. It works by associating style rules with HTML elements, allowing developers to separate the content and structure of a webpage from its presentation.

Some key features of CSS include:

1. Selectors: CSS uses selectors to target specific HTML elements and apply styles to them. Selectors can be based on element type, class, ID, and other attributes.

2. Style rules: CSS uses style rules to define the visual properties of HTML elements, such as font size, color, margin, and padding.

3. Cascading: CSS styles can cascade, meaning that multiple style rules can be applied to the same element, and the final style is determined by a set of rules based on specificity and order.

4. Inheritance: CSS styles can be inherited from parent elements to child elements, reducing the need for redundant style declarations.

5. Media queries: CSS includes support for media queries, which allow developers to apply different styles based on the screen size or other properties of the device being used to view the webpage.

6. Responsive design: By using CSS to create responsive designs, developers can ensure that web pages look and function well on a variety of devices and screen sizes.

7. Preprocessors: CSS preprocessors such as Sass and Less allow developers to use advanced features like variables, mixins, and functions to streamline their CSS code and make it more modular and reusable.

Overall, CSS is a powerful tool for creating visually appealing and responsive web pages, and is an essential part of modern web development. More efficient and effective stylesheets can be created that enhance the user experience and help achieve the desired visual design.

5.6.3 JAVASCRIPT

JavaScript is a high-level programming language that is widely used in web development to create interactive and dynamic web pages. Here are some key features and benefits of JavaScript in web development:

1. Client-side scripting: JavaScript runs on the client-side, which means that it is executed by the user's web browser rather than on the server. This allows for faster response times and more dynamic user interfaces.

2. Cross-platform compatibility: JavaScript is supported by all major web browsers and can run on any operating system, making it a versatile and widely used language for web development.

3. Interactivity: JavaScript enables the creation of interactive web applications, such as form validation, animations, and user-triggered events. This makes web pages more engaging and user-friendly.

4. Dynamic content: JavaScript can be used to dynamically modify web page content without requiring a full page refresh. This allows for a more seamless user experience and can improve performance by reducing server requests.

5. Third-party libraries and frameworks: There are many third-party libraries and frameworks available for JavaScript, such as jQuery, React, and Angular, which can significantly speed up the development process and provide additional functionality.

6. Server-side scripting: In addition to client-side scripting, JavaScript can also be used for server-side scripting using technologies such as Node.js. This allows for the creation of server-side applications using a single language.

Overall, JavaScript is a powerful and versatile language that is essential for modern web development. Dynamic, interactive, and engaging web pages can be created that provide a great user experience.

5.6.4 TAILWIND CSS

Tailwind CSS is a utility-first CSS framework that allows developers to rapidly build responsive, mobile-first user interfaces with minimal effort. It offers a set of pre-designed CSS classes that can be used to apply styling and layout to HTML elements, enabling developers to create complex layouts and styles with ease.

One of the main benefits of Tailwind CSS is that it eliminates the need to write custom CSS, saving time and reducing the likelihood of errors. Instead, developers can use pre-designed utility classes to quickly add styles to their HTML, such as changing the font size, color, padding, or margin of an element. Tailwind CSS also includes a range of responsive utility classes, which can be used to adjust styles based on the size of the screen or viewport.

Tailwind CSS is highly customizable, with options to customize colors, typography, and other design elements. This allows developers to create their own design system and brand identity, while still benefiting from the speed and consistency of the utility-first approach.

Another advantage of Tailwind CSS is that it can be easily integrated into different front-end frameworks or libraries, such as React or Vue.js, making it a versatile choice for web development.

Overall, Tailwind CSS offers a modern and efficient approach to CSS styling and layout, allowing developers to quickly build beautiful and responsive user interfaces without sacrificing flexibility or customizability.

5.7 BACK-END

5.7.1 FLASK

Flask is a popular Python web framework that is used to develop web applications quickly and easily. It is a micro-framework, which means it is minimalistic in nature and provides only the basic tools required for web development. Flask is also highly flexible and modular, allowing developers to easily customize and extend its functionality to fit their specific needs.

Some key features of Flask include:

1. Routing: Flask provides a simple and intuitive routing system that allows developers to map URLs to view functions. This makes it easy to create dynamic web pages that respond to user requests.

2. Templates: Flask comes with a built-in template engine called Jinja2, which allows developers to create HTML templates that can be rendered with dynamic content. This makes it easy to create consistent and reusable layouts for web pages.

3. Database integration: Flask supports a variety of database systems, including SQLite, MySQL, and PostgreSQL. It also provides an object-relational mapping (ORM) tool called SQLAlchemy, which makes it easy to work with databases in Python.

4. Extensions: Flask has a large ecosystem of extensions that can be used to add additional functionality to the framework. These extensions include tools for handling forms, authentication, and caching, among others.

5. Lightweight: Flask is designed to be lightweight and unopinionated, meaning it does not impose any particular structure or way of doing things on developers. This allows for maximum flexibility and freedom in developing web applications.

6. RESTful API development: Flask provides a simple way to build RESTful APIs using the `flask-restful` extension, allowing developers to create scalable and modular APIs quickly and easily.

Flask is widely used for building web applications and RESTful APIs in Python, and its flexibility and simplicity make it a popular choice among developers.

CHAPTER 6

TESTING

6.1 TESTING APPROACHES

6.1.1 UNIT TESTING

Unit testing is a software testing technique that verifies the functionality of individual units or components of a software application in isolation from the rest of the application. It helps to catch and correct errors early in the development process, improve code quality and maintainability, and increase confidence in the code.

6.1.2 BLACK-BOX TESTING

Black box testing is a software testing technique where the tester tests the functionality of a software application without knowing its internal workings. The tester creates test cases based on the requirements, executes them, and reports any defects found to the development team for resolution. The goal is to ensure that the software application meets the requirements and functions as expected.

6.1.3 TEST CASES

K-Means:

TEST CASE	UNIT TESTING
TEST NAME	CLUSTER IDENTIFICATION
ITEMS BEING TESTED	CLUSTERING MODEL
SAMPLE INPUT	“ANDHRA PRADESH”, “HYDERABAD” (Actual input data is fetched in backend)

EXPECTED OUTPUT	RED ZONE
ACTUAL OUTPUT	RED ZONE
REMARKS	PASS

Table 6.1.3.1: Test Case for K-Means

Random Forest Classifier:

TEST CASE	UNIT TESTING
TEST NAME	LABEL PREDICTION
ITEMS BEING TESTED	CLASSIFICATION
EXPECTED OUTPUT	RED ZONE
ACTUAL OUTPUT	RED ZONE
REMARKS	PASS

Table 6.1.3.2: Test Case for Classifier

Linear Regression:

TEST CASE	UNIT TESTING
TEST NAME	CRIME COUNT PREDICTION
ITEMS BEING TESTED	LINEAR REGRESSION MODELS
SAMPLE INPUT	"ANDHRA PRADESH", 2019 (Actual Population Input is fetched in backend)
EXPECTED OUTPUT	237567
ACTUAL OUTPUT	235008
REMARKS	PASS

Table 6.1.3.3: Test Case for Regression

Time Series Forecasting:

TEST CASE	UNIT TESTING
TEST NAME	CRIME RATE TEST
ITEMS BEING TESTED	FORECASTING MODEL
SAMPLE INPUT	2021
EXPECTED OUTPUT	269
ACTUAL OUTPUT	268
REMARKS	PASS

Table 6.1.3.4: Test Case for Time Series Forecasting

6.1.4 TEST ON DATASETS:

Random Forest Classifier:

Test Data Size = 0.2 (or) 20%

Accuracy = 0.9732 (or) 97.32%

Time Series Forecasting:

Accuracy (For Crime Rate): 0.8885 (or) 88.85 %

Accuracy (Total IPC Crime): 0.9666 (or) 96.66 %

CHAPTER 7

SCREENSHOTS

HOME PAGE:

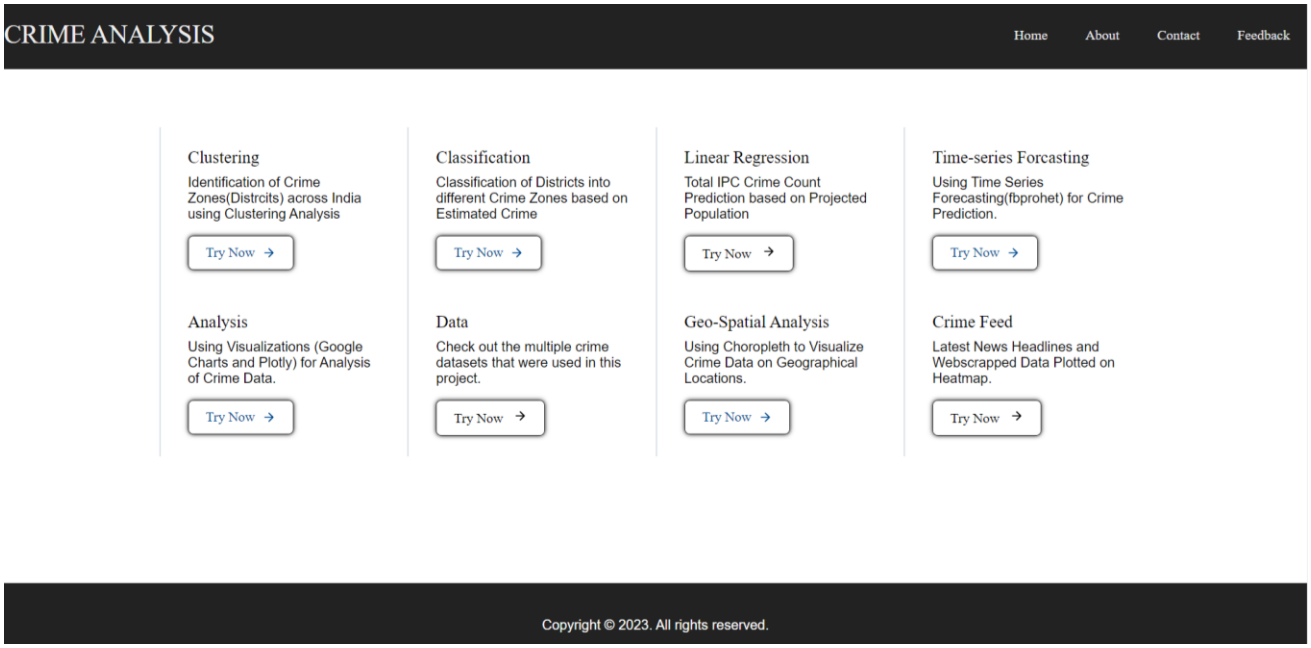


Fig 7.1: Homepage

CLUSTERING:

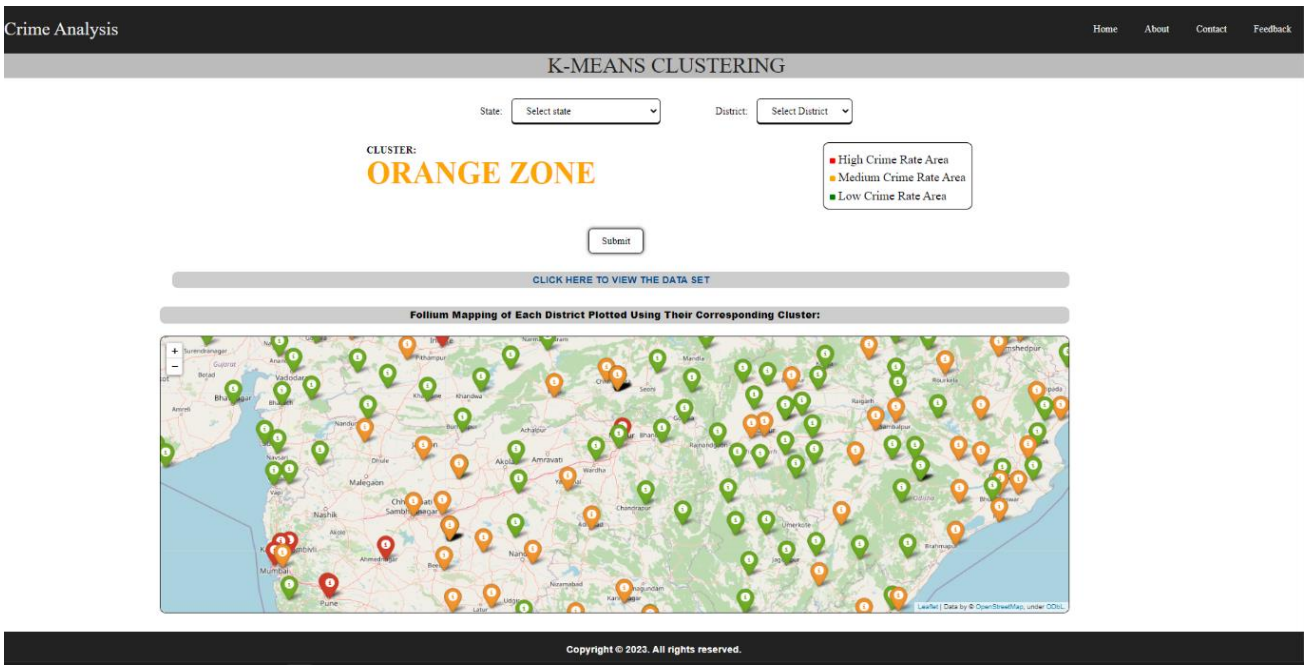


Fig 7.2: Clustering

CLASSIFICATION:

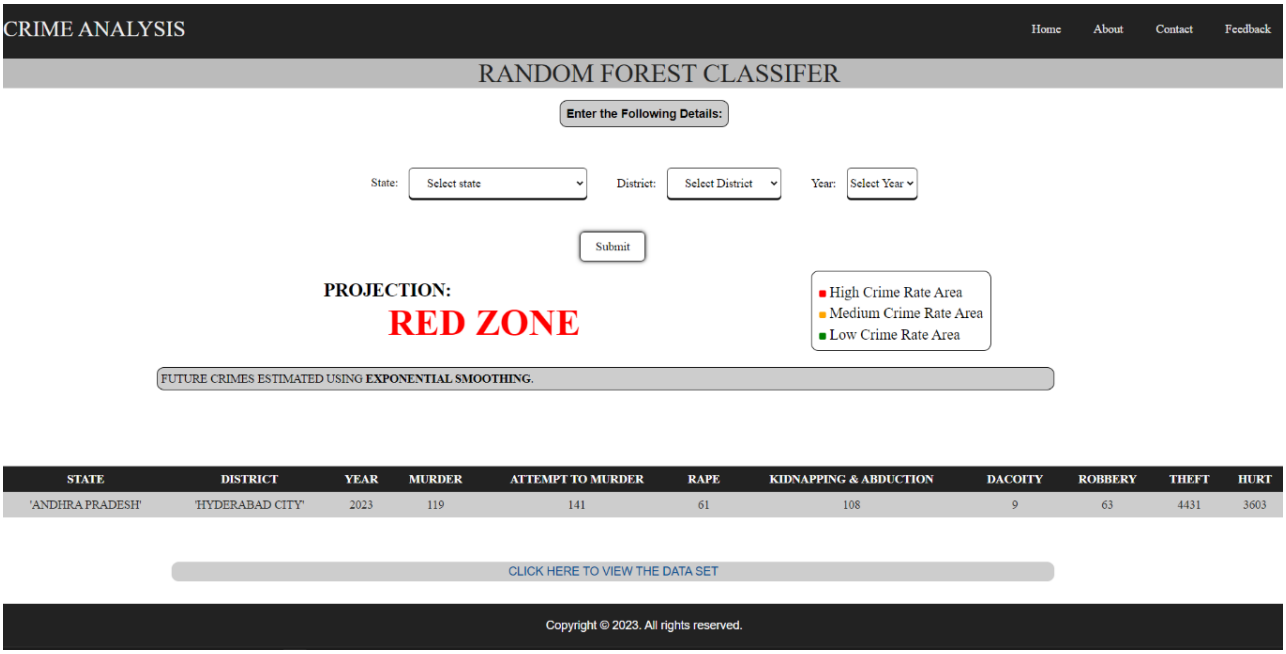


Fig 7.3: Classification

LINEAR REGRESSION:

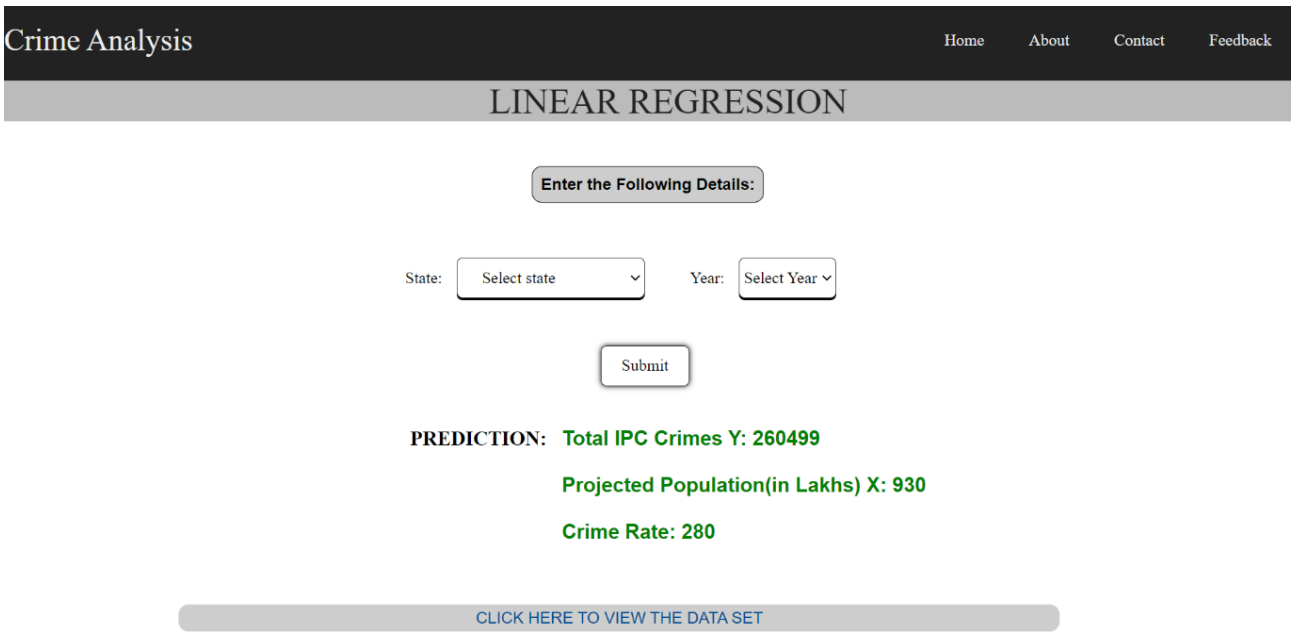


Fig 7.4: Linear Regression

TIME SERIES FORECASTING:

Fig 7.5: Time Series Forecasting

GOOGLE CHARTS LINE CHARTS):

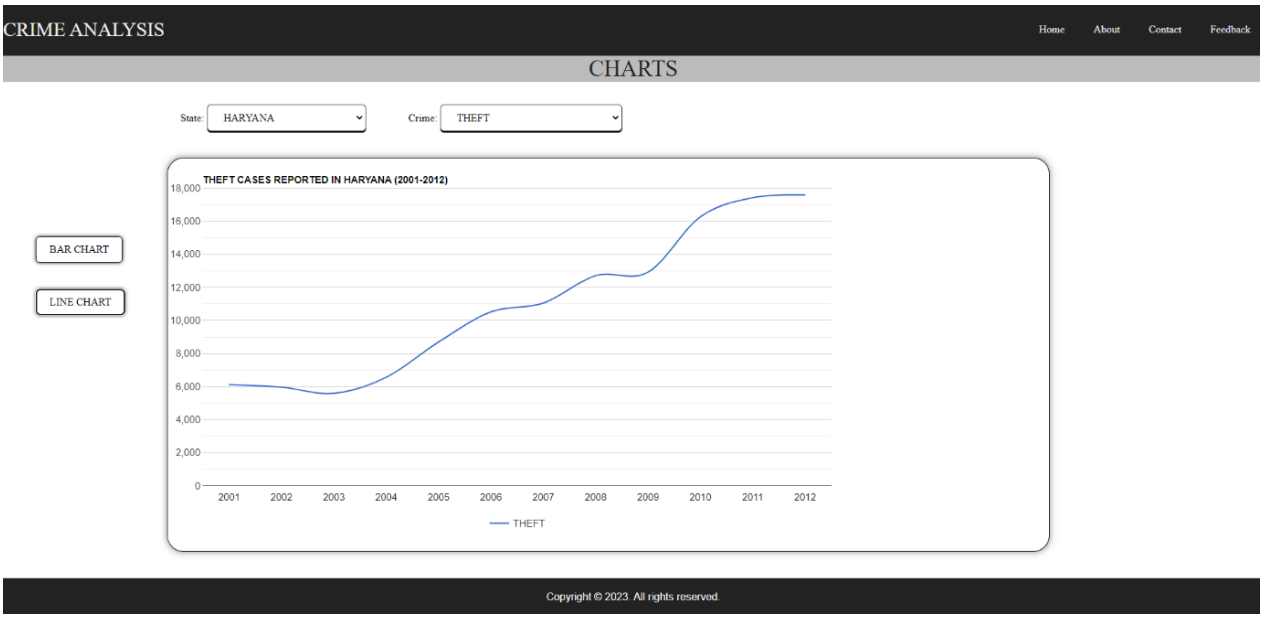


Fig 7.6: Google Charts (Line Charts)

GOOGLE CHARTS (BAR CHARTS):

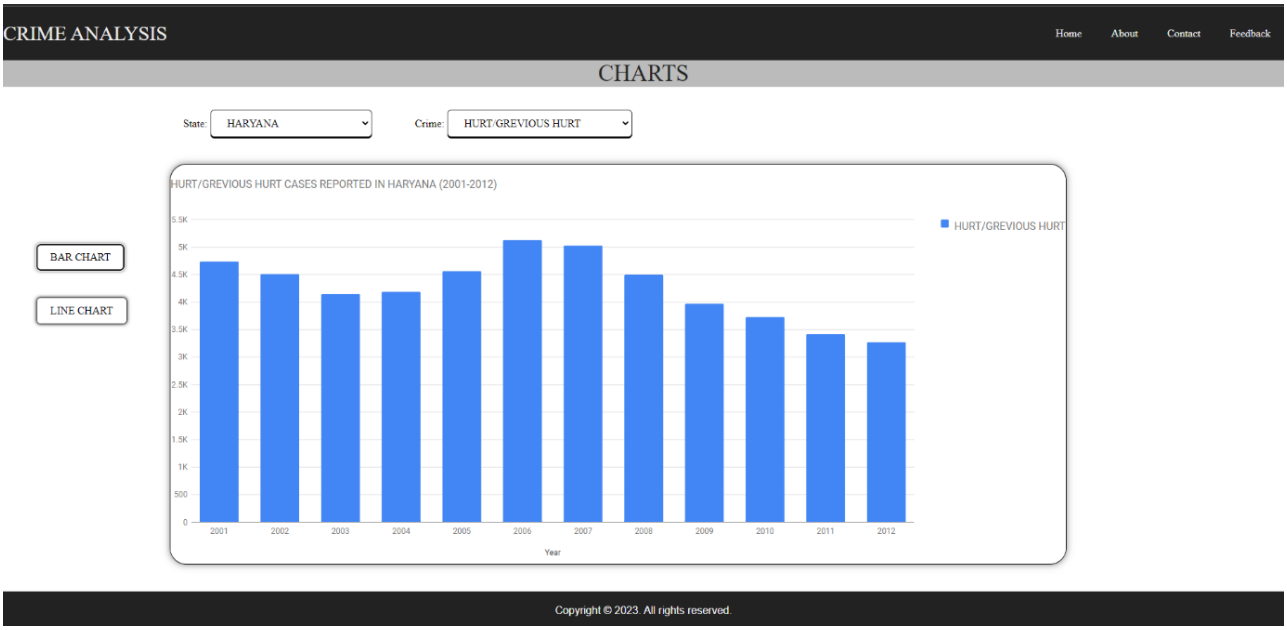


Fig 7.7: Google Charts (Bar Charts)

PLOTLY (MULTI-LINEAR):

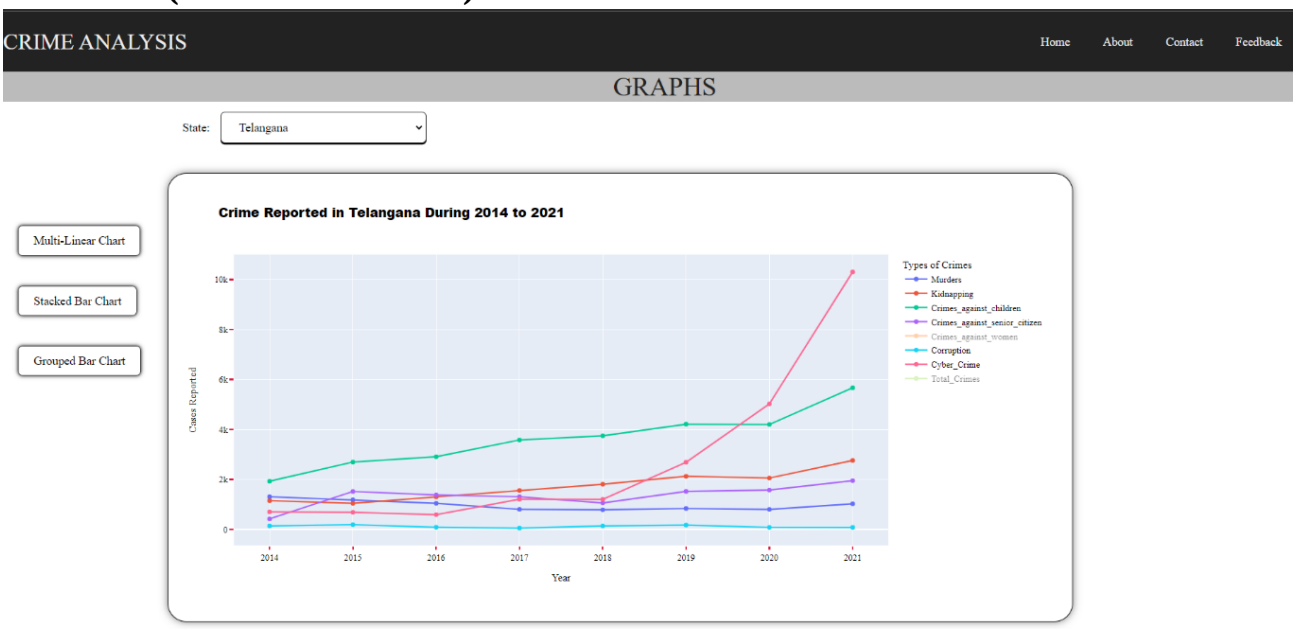


Fig 7.8: Plotly (Line Charts)

PLOTLY (STACKED-BAR):

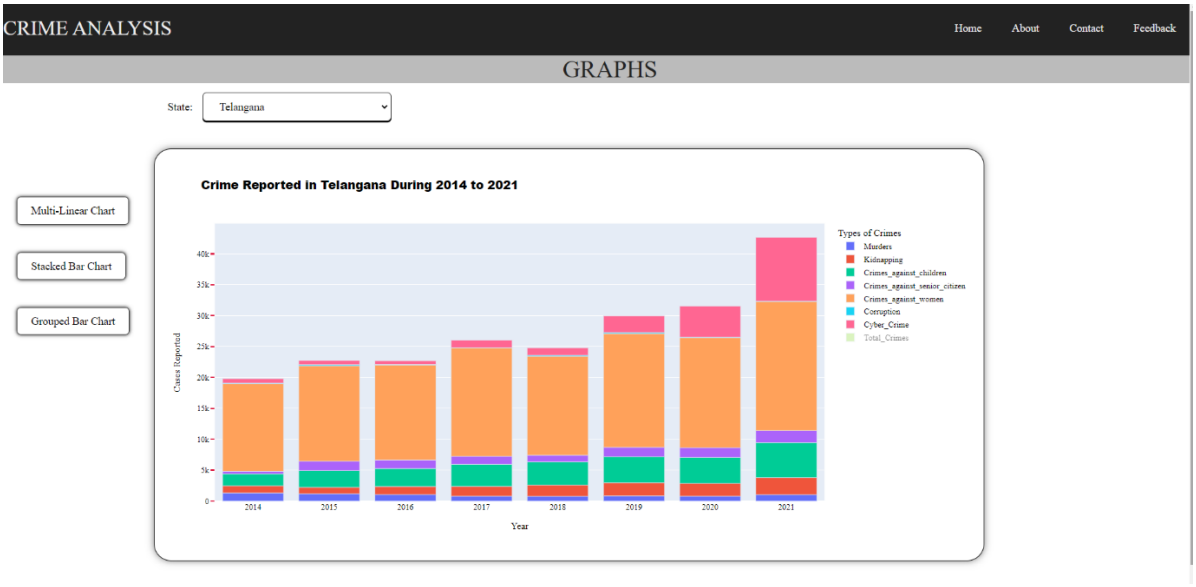


Fig 7.9: Plotly (Stacked Bar)

PLOTLY (GROUPED-BAR):

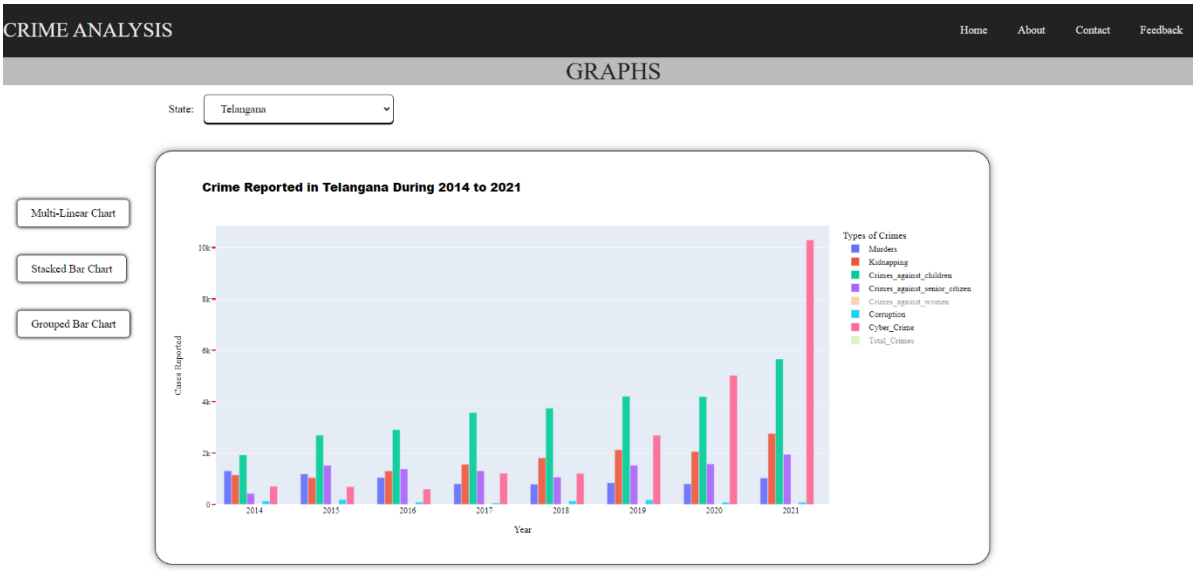


Fig 7.10: Plotly (Grouped Bar)

CHOROPLETH:

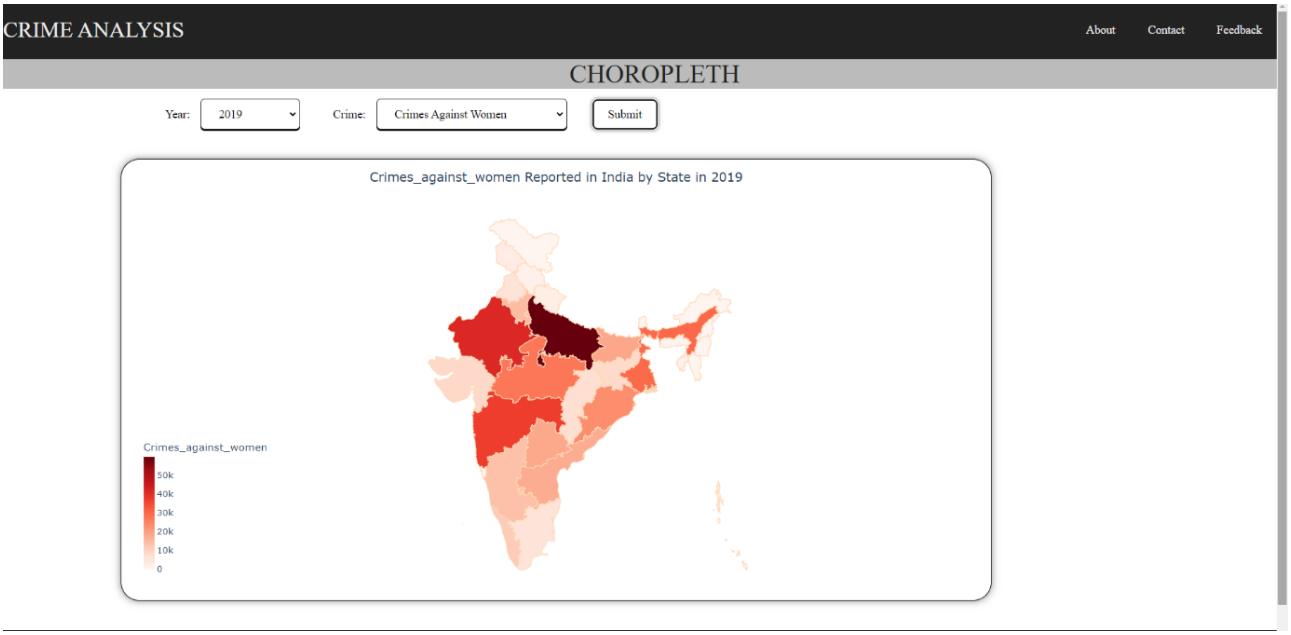


Fig 7.11: Geospatial Visualization (Choropleth)

WEB SCRAPING MODEL:

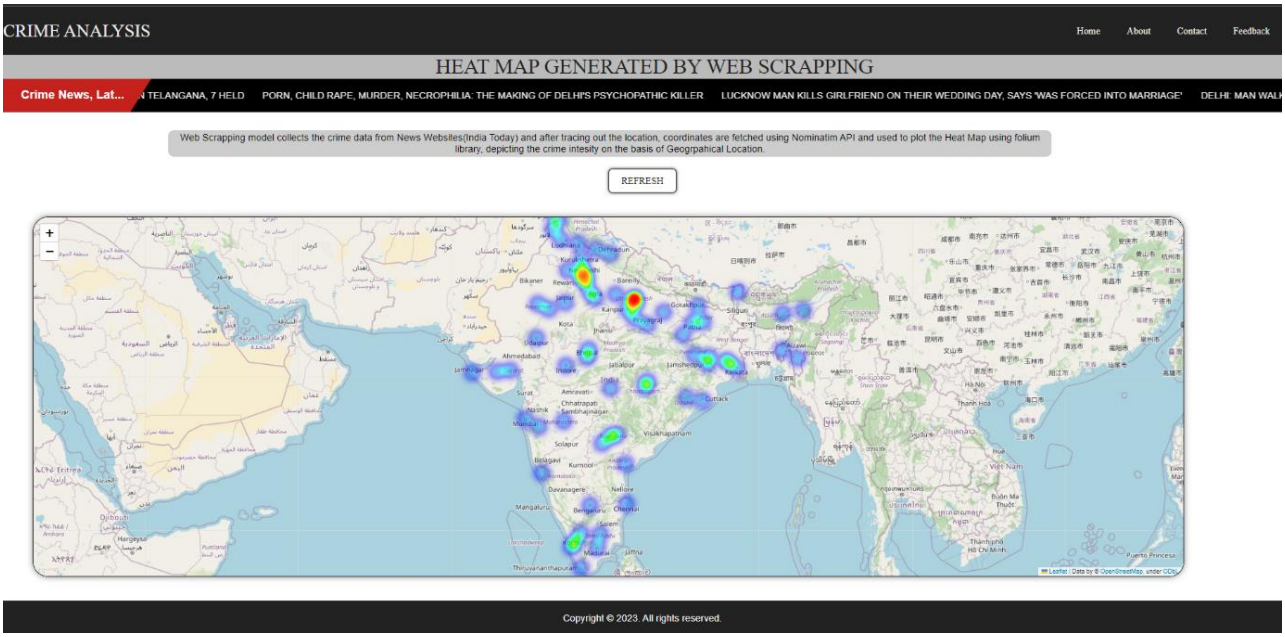


Fig 7.12: Web Scraping Module

CHAPTER 8

CONCLUSION

- In conclusion, this project aimed to analyze crime data and provide insights into crime patterns, trends, and hotspots. The project utilized a range of techniques, including :

  - K-Means clustering
  - Random Forest Classifier
  - Exponential Smoothing
  - Linear Regression
  - Geometric Growth Method
  - Fbprophet

  , to analyze the data and make predictions about future crime rates and cluster trends.

- Through data visualization using Google Charts, Plotly, and Choropleth, the project presented the findings in a visually engaging manner, allowing users to explore and interact with the data.

- The project also employed web scraping techniques to extract crime headlines and plot them on a heatmap, providing an additional perspective on crime patterns and hotspots.

- The project offers valuable insights into crime analysis and can help policymakers, law enforcement agencies, and the general public to understand and address crime issues more effectively.

- Different approaches or strategies can be deployed in different crime zone areas in order to enforce laws and curb higher crime rates, and avoid potential future crime hotspots.

- The techniques and methods used in this project can also be applied to other domains, such as healthcare, finance, and marketing, to extract insights from data and make informed decisions.

CHAPTER 9

FUTURE ENHANCEMENTS

- While the project has provided useful insights into crime analysis, there are several areas where future enhancements could be made to further improve the accuracy and usability of the findings.

- Firstly, the project can be expanded to include more data sources, such as social media, CCTV footage, real-time data and police reports. Integrating data from various sources can provide a more comprehensive understanding of crime patterns and help identify potential hotspots.

- Secondly, the project can be enhanced by incorporating advanced machine learning techniques, such as deep learning and neural networks, to analyze the data. These techniques can capture more complex relationships between variables and improve the accuracy of predictions.

- Thirdly, the project can be extended to include predictive modeling for specific types of crimes, such as robbery, burglary, or cybercrime. This would allow for more targeted interventions to prevent or reduce specific types of crime.

- Fourthly, the project can be enhanced by integrating real-time data feeds, which would allow for real-time analysis and visualization of crime patterns. This would enable law enforcement agencies to respond more quickly to emerging crime hotspots and take proactive measures to prevent crime.

- Lastly, the project can be made more accessible to the general public by developing a user-friendly mobile application. The application could provide users with real-time crime alerts, crime prevention tips, and a dashboard for exploring crime patterns in their local area.

- Overall, these future enhancements can further improve the accuracy, relevance, and usability of the project's findings and provide a more comprehensive and holistic understanding of crime patterns and trends.

REFERENCES

[1] M. B. Mitchell, D. E. Brown, and J. H. Conklin, DzA Crime Forecasting Tool for the Web-Based Crime Analysis Toolkit,dz ´TT7 IEEE Syst. Inf. Eng. Des. Symp.,pp.Ť– 5,2007.

[2] S.Ismail and N. Ramli,DzShort-Term Crime Forecast ting In Keedah,dz Procedia-Soc. Behav. Sci., vol.91,pp.654– 660,2013.

[3] W. Gorr and R. Harries, DzIntroduction to crime forecasting,dz vol. Ť9, pp. ииŤ–555, 2003.

[4] K. H. Vellani, DzCrime Analysis: for Problem Solving Security Professionals in ´и Small Steps,dz Facilities,pp.Ť– 56,2010.

[5] Mehmet Sait, and Mustafa Gök. "Criminal prediction using Naive Bayes theory." Springer 28.9 (2016): 2581-2592.

[6] Ahishakiye, E., Anisha, C., Dhanashree., and , I., 2017. Crime Prediction Using Decision Tree (J48) Classification Algorithm. International Journal Computer and Information Technology, 6(03).

[7] A.Anisha., C.Dhanashree, and A.Arpita ,. Application for analysis and prediction of crime data using data mining. International journal of advanced computational engineering and networking, 2017.

[8] Sivaranjani, S., S. Sivakumari, and M. Aasha. "Crime prediction and forecasting in Tamilnadu using clustering approaches." Emerging Technological Trends (ICETT), International Conference on. IEEE, 2016.

[9] Kiani, Rasoul, Siamak Mahdavi, and Amin Keshavarzi. "Analysis and prediction of crimes by clustering and classification." international journal of advanced research in artificial intelligence (2015).

[10] Cesario, Cesario E, Catlett C, Talia D. Forecasting Crimes Using Autoregressive Models. InDependable, Autonomic and Secure Computing, 2016 IEEE 14th Intl C 2016 Aug 8 (pp. 795-802). IEEE.

[10] Bhaskaran S (2012) Time series data analysis for long term forecasting and scheduling of organizational resources-few cases. Int J Comput Appl 41(12):4–9

[11] Bhaskaran S (2012) Time series data analysis for long term forecasting and scheduling of organizational resources-few cases. Int J Comput Appl 41(12):4–9

[12] Petitjean F, Inglada J, Ganc¸arski P (2012) Satellite image time series analysis under time warping. IEEE Trans Geosci Remote Sens 50:3081–3095. https://doi.org/10.1109/TGRS.2011.2179050

[13] Chujai P, Kerdprasop N, Kerdprasop K (2013) Time series analysis of household electric consumption with ARIMA and ARMA models. In: Proceedings of the international multi conference of engineers and computer scientists, IMECS 2013, Hong Kong, 13–15 Mar 2013, vol 1, pp 295–300

[14] Devi BU, Sundar D, Alli P (2013) An effective time series analysis for stock trend prediction using ARIMA model for nifty midcap-50. Int J Data Min Knowl Manag Process 3:65–78. https://doi.org/10.5121/ijdkp.2013.3106

[15] Smith M, Agrawal R (2015) A comparison of time series model forecasting methods on patent groups. In: CEUR workshop proceedings, pp 167–173

[16] Doucoure B, Agbossou K, Cardenas A (2016) Time series prediction using artificial wavelet neural network and multi-resolution analysis: application to wind speed data. Renew Energy 92:202–211. https://doi.org/10.1016/j.renene.2016.02.003

[17] Laptev N, Yosinski J, Li LE, Smyl S (2017) Time-series extreme event forecasting with neural networks at Uber. In: International conference on machine learning, Sydney, Australia, 11 Aug 2017, vol 34, pp 1–5

[18] Esteban C, Hyland SL, Ra¨tsch G (2017) Real-valued (medical) time series generation with recurrent conditional GANs. In: ICLR 2018 conference on blind submiss, pp 1–15

[19] Hosseini SM, Saifoddin A, Shirmohammadi R, Aslani A (2019) Forecasting of CO2 emissions in Iran based on time series and regression analysis. Energy Rep 5:619–631. https://doi.org/10. 1016/j.egyr.2019.05.004

[20] Karmaker CL, Halder PK, Sarker E (2017) A study of time series model for predicting jute yarn demand: case study. J Ind Eng 2017:1–8. https://doi.org/10.1155/2017/2061260

[21] http://www.dmp.gov.bd/application/index/page/crimedata

[22] http://www.police.gov.bd/Crime-Statisticsyearly.php?id=337

[23] A. H. Wahbeh, Q. A. Al-Radaideh, M. N. Al-Kabi, E. M. Al-Shawakfa, "A Comparison Study between Data Mining Tools over some Classification Methods", International Journal of Advanced Computer Science and Applications, The SAI Organization, Special Issue on Artificial Intelligence, pp. 18-26, 2011.

[24] N. Levine, "CrimeStat: A Spatial Statistic Program for the Analysis of Crime Incident Locations (v 2.0)", Ned Levine & Associates, Houston, TX, and the National Institute of Justice, Washington, DC, May 2002.

[25] A. L. Buczak, C. M. Gifford, "Fuzzy Association Rule Mining for Community Crime Pattern Discovery", In ACM SIGKDD Workshop on Intelligence and Security Informatics (ISIKDD' 10), 2012.

[26] J. Han, M. Kamber, J. Pei, "Data Mining: Concepts and Techniques", vol. 5. Morgan Kaufmann Publishers, USA, 2012.

[27] S. Shojaee, A. Mustapha, F. Sidi, M. A. Jabar, "A study on classification learning algorithms to predict crime status", In: International Journal of Digital Content Technology and its Applications (JDCTA) 7.9 (May 2013), pp. 361–369, issn: 1975-9339.

[28] A. Malathi, S. S. Baboo, "Enhanced Algorithms to Identify Change in Crime Patterns", International Journal of Combinatorial Optimization Problems and 337 Informatics, Aztec Dragon Academic Publishing, vol. 2, no.3, pp. 32-38, 2011.

[29] L. McClendon, N. Meghanathan, "Using Machine Learning Algorithms to Analyze Crime Data", Machine Learning and Applications: An International Journal (MLAIJ) vol. 2, no. 1, March 2015.

[30] C. H. Yu, M. W. Ward, M. Morabito, W. Ding, "Crime Forecasting Using Data Mining Techniques", In Proceedings of the 2011 IEEE 11th International Conference on Data Mining Workshops (ICDMW' 11), pp. 779-786, 2011.

[31] https://www.javatpoint.com/machine-learning

[32] https://www.javatpoint.com/k-means-clustering-algorithm-in-machine-learning

[33] https://www.geeksforgeeks.org/

[34] https://towardsdatascience.com/