

CRIME PREDICTION AND ANALYSIS USING MACHINE LEARNING

**A Project report submitted in partial
Fulfillment of the requirements for the award of the Degree**

**BACHELOR OF TECHNOLOGY
COMPUTER SCIENCE AND ENGINEERING**

Submitted By

K. AJAY KUMAR	19HT1A0550
B. YUVA KISHORE BABU	19HT1A0517
B. HARINADH	19HT1A0508
A. UPENDRA	19HT1A0504

**Under the Esteemed Guidance of
Mr. SK. KHADER BASHA
Associate Professor**



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

CHALAPATHI INSTITUTE OF TECHNOLOGY

(Affiliated to JNTU, Kakinada)

A.R. NAGAR, MOTHADAKA, GUNTUR(DIST), ANDHRA PRADESH

2019 - 2023



chalapathi
Institute of Technology
APPROVED BY AICTE | AFFILIATED TO JNTUK

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that K. AJAY KUMAR (19HT1A0550), B. YUVA KISHORE BABU (19HT1A0517), B. HARINADH (19HT1A0508), A. UPENDRA (19HT1A0504) are completed a Project entitled “**CRIME PREDICTION AND ANALYSIS USING MACHINE LEARNING**” for the partial fulfillment of the requirements for the award of **Bachelor of Technology** in **COMPUTER SCIENCE AND ENGINEERING** by JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY, KAKINADA.

Mr. SK. KHADER BASHA

Assistant Professor

PROJECT GUIDE

Dr. V. NAGA GOPIRAJU

Head of the Department

Submitted for Viva Voice Examination held on

External Examiner

DECLARATION

We hereby declare that the project entitled, “**CRIME PREDICTION AND ANALYSIS USING MACHINE LEARNING**” submitted in partial fulfillment of the requirements for the award of bachelor of technology in computer science and engineering, to Chalapathi Institute of Technology, (CITY-HT), permanently affiliated to JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY KAKINADA (JNTUK) is an authentic work and has not been submitted to university or institute for the award of the degree.

DATE:

PLACE:

SIGNATURE OF THE CANDIDATES

K. AJAY KUMAR - 19HT1A0550

B. YUVA KISHORE BABU - 19HT1A0517

B. HARINADH - 19HT1A0508

A. UPENDRA - 19HT1A0504

ACKNOWLEDGMENT

We Consider it our privilege to express our gratitude to all those who guided and inspired us in the completion of this Project .We express our sincere thanks to our beloved **Chairman Sri Y. V. ANJANEYULU Garu** for providing support and stimulating environment for developing the project.

We express deep sense of reverence and profound graduate **Dr. V. RANGARAO Garu, Principal, Chalapathi Institute of Technology**for providing us the great support in completing our resource for carrying out the project.

It is with immense pleasure that we would like to express our indebted gratitude to our guide **Mr. SK. KHADER BASHA** who has guided us alot and encouragedus in every step of the project work. He has given moral support and guidance throughout the project helped us to a great extent.

Our sincere thanks to **Dr. V. NAGAGOPIRAJU Garu, HEAD OF THE DEPT OF CSE** for his co-operation and guidance in helping us to make our project successful and complete in all aspects we are grateful to his precious guidance and suggestions.

We also place our floral gratitude to all other Teaching Staff and Labprogrammers for their constant support, advice throughout the project. Last but not least; we thank our PARENTS and FRIENDS who directly or indirectly helped us in the successful completion of our project.

PROJECT ASSOCIATES

K. AJAY KUMAR - 19HT1A0550

B. YUVA KISHORE BABU - 19HT1A0517

B. HARINADH - 19HT1A0508

A. UPENDRA - 19HT1A0504

ABSTRACT

Data mining and Machine learning have become a vital part of crime detection and prevention. The purpose of this project is to evaluate data mining methods and their performances that can be used for analyzing the collected data about the past crimes. Identified the most appropriate data mining methods to analyze the collected data from sources specialized in crime prevention by comparing them theoretically and practically. Some attributes of this dataset are gender, age, employment status, crime place. Methods are applied on these data to determine their effectiveness in analyzing and preventing crime. Evaluations on the data showed that the method with a higher performance is “Decision Tree”.

This was achieved by some performance measures, such as the number of instances correctly classified, accuracy or precision and recall that has brought better results compared to other methods. I come to the conclusion that the data mining methods contribute to the predictions on the possibility of occurrence of the crime and as a result in its prevention.

Keywords: Crime Prediction; Machine Learning; Decision tree; J48; Artificial Intelligence; Classification Algorithms.

LIST OF CONTENTS

SI.NO	TITLE	Page No
1	Introduction	1
2	Literature Survey	2-3
3	System Analysis	4-5
	3.1 Existing System	4
	3.2 Problem Statement	4
	3.3 Proposed System	4-5
	3.4 Hardware Requirements	5
	3.5 Software Requirements	5
4	Software Environment	6-12
	4.1 Machine Learning	6
	4.1.1 Relation to Data Mining	6
	4.1.2 Types of Learning Algorithms	7-8
	4.2 About Python	8-9
	4.3 Anaconda	9-10
	4.4 Jupyter Notebook	10
	4.5 Numpy	10-11
	4.6 Pandas	11
5	System Study	13-14
	5.1 Feasibility Study	13-14
6	Modules	15-17
	6.1 Data Collection	15
	6.2 Data Preprocessing	15
	6.3 Data Set	16
	6.4 Feature Extraction	16

	6.5 Evaluation Model	17
7	Algorithms	18-26
	7.1 Decision Tree	18-19
	7.2 Artificial Neural Networks	19-20
	7.3 Naïve Bayes Classifier	20-21
	7.4 Random Forest	21
	7.5 SVM	21-22
	7.6 K-Nearest Neighbors	22-23
	7.7 Logistic Regression	24-25
	7.8 Comparison of Algorithms	26
8	System Design	27-32
	8.1 System Architecture	27-28
	8.2 UML Diagrams	28
	8.2.1 Use Case Diagram	29
	8.2.2 Class Diagram	29-30
	8.2.3 Sequence Diagram	30-31
	8.2.4 Activity Diagram	31-32
9	Source Code	33-37
	Clustering	33-36
	Support Vector Machine	36-37
10	Screen Shots	38-42

11	System Testing	43-45
	11.1 Unit Testing	43
	11.2 Integration Testing	43-44
	11.3 Functional Testing	44
	11.4 White Testing	44
	11.5 Black Box Testing	44-45
	11.6 Software Testing Strategies	45
	11.6.1 User Acceptance Testing	45
	11.6.2 Output Testing	45
12	Conclusion	46
13	References	47
14	Future Scope	48

LIST OF FIGURES

Sl.NO	LIST OF FIGURES	Page No
5.1	Feasibility Study	14
7.6.1	KNN example before applying algorithm	22
7.6.2	KNN example after applying algorithm	23
8.1	Architecture Diagram	28
8.2.1	Use Case Diagram	29
8.2.2	Class Diagram	30
8.2.3	Sequence Diagram	31
8.2.4	Activity Diagram	32
10.1	Jupyter Notebook	38
10.2	Python Shell	39
10.3	Libraries	40
10.4	Decision Tree	40
10.5	Hierarchical Clustering Dendrogram	41
10.6	Graph Representation	42

Chapter – 1

INTRODUCTION

1. INTRODUCTION

The increase in crime data recording coupled with data analytics resulted in the growth of research approaches aimed at extracting knowledge from crime records to better understand criminal behavior and ultimately prevent future crimes.

Crime is a complex social phenomenon that has grown due to major changes in society. Law enforcement agencies need to learn the factors that lead to an increase in crime tendency. To curb this, there is always a need for strategies and policies to prevent crime. As a result of technology development, science and information, data mining and artificial intelligence tools are increasingly prevalent in the law enforcement community.

Law enforcement agencies face a large volume of data that needs to be processed and turned into useful information, and data mining can improve crime analysis by helping to predict and prevent it. By processing criminal data, law enforcement agencies can use models that may be important in the crime prevention process.

The use of data mining accelerates data analysis, and analysts can examine existing data to identify patterns and trends of crime. This project is structured as follows: It describes the relationship that exists between data mining, machine learning and criminology. The methodology and description of the dataset are described in it. Next, it represents a theoretical description of the methods and algorithms that will be applied practically to our data presents the results of the application of algorithms and an explanation for the algorithm with the best results.

Chapter - 2

LITERATURE SURVEY

2. LITERATURE SURVEY

2. Using machine learning algorithms to analyze crime data

AUTHORS: McClendon, Lawrence, and Natarajan Meghan than.

Data mining and machine learning have become a vital part of crime detection and prevention. In this research, we use WEKA, an open source data mining software, to conduct a comparative study between the violent crime patterns from the Communities and Crime Unnormalized Dataset provided by the University of California-Irvine repository and actual crime statistical data for the state of Mississippi that has been provided by neighborhoodscout.com.

We implemented the Linear Regression, Additive Regression, and Decision Stump algorithms using the same finite set of features, on the Communities and Crime Dataset. Overall, the linear regression algorithm performed the best among the three selected algorithms. The scope of this project is to prove how effective and accurate the machine learning algorithms used in data mining analysis can be at predicting violent crime patterns.

2.1 Learning to detect patterns of crime

AUTHORS: Wang, Tong, et al.

We introduce a novel, robust data-driven regularization strategy called Adaptive Regularized Boosting (AR-Boost), motivated by a desire to reduce overfitting. We replace AdaBoost's hard margin with a regularized soft margin that trades-off between a larger margins, at the expense of misclassification errors. Minimizing this regularized exponential loss results in a boosting algorithm that relaxes the weak learning assumption further: it can use classifiers with error greater than $\frac{1}{2}$. This enables a natural extension to multiclass boosting, and further reduces overfitting in both the binary and multiclass cases. We derive bounds for training and generalization errors, and relate them to Adaboost.

Finally, we show empirical results on benchmark data that establish the robustness of our approach and improved performance overall. 1 Introduction Boosting is a popular method for improving the accuracy of a classifier. In particular, AdaBoost is considered the most popular form of boosting and it has been shown to improve the performance of base learners both theoretically and empirically. The key idea behind AdaBoost is that it constructs a strong classifier using a set of weak classifiers.

2.2 Crime Analysis using K-Means Clustering

AUTHORS: Jyoti Agarwal, Renuka Nagpal and Rajni Sehgal

In today's world security is an aspect which is given higher priority by all political and government worldwide and aiming to reduce crime incidence. As data mining is the appropriate field to apply on high volume crime dataset and knowledge gained from data mining approaches will be useful and support police force. So In this paper crime analysis is done by performing k-means clustering on crime dataset using rapid miner tool.

The use of K-means data mining approach helps us identify patterns since it is very difficult for humans to process large amounts of data, especially if there are missing information to detect patterns. K-means clustering is one of the methods of cluster analysis. In the K-means algorithm, each point is assigned to the cluster whose centroid is the closest. K-means clustering aims to partition n observations into k clusters in which each observation belongs to the cluster with the nearest mean. It can be applied to relatively large sets of data.

Chapter – 3

SYSTEM ANALYSIS

3. SYSTEM ANALYSIS

3.1 EXISTING SYSTEM:

KNN, RF, SVM and Bayes models are existing methods although studies have been done in the medical field with an advanced data exploration using machine learning algorithms, orthopedic disease prediction is still a relatively new area and must be explored further for the accurate prevention and cure. It mines the double layers of hidden states of vehicle historical trajectories, and then selects the parameters of Hidden Markov Model (HMM) by the historical data. In addition, it uses a Viterbi algorithm to find the double layers hidden states sequences corresponding to the just driven trajectory. Finally, it proposes a new algorithm for vehicle trajectory prediction based on the hidden Markov model of double layers hidden states, and predicts the nearest neighbor unit of location information of the next k stages.

3.2 PROBLEM STATEMENT:

Crimes now a days are increasing day by day and with different level of Intensity and versatility. The result is a great loss to society in terms of monetary loss, Social loss and further it enhances the level of threat against the smooth livelihood in the society. To overcome this problem, the computing era can help to reduce the crime or even may be helpful in predicting the crime so that sufficient measures can be taken to minimize the loss to property and life. The crime rate prediction strategies can be applied on historical data available in the police records by examining the data at various angles like reason of crime, frequency of similar kind of crimes at specific location with other parameters to prepare the model crime prediction. It is a major challenge to understand the versatile data available with us, then model it to predict the future incidence with acceptable accuracy and further to reduce the crime rate.

3.3 PROPOSED SYSTEM:

The proposed system is made on the basis of the research work that is done by going through various such documentations. Nearly all of the crimes are predicting based on the location and the types of crimes that are occurring in those areas. On surveying previous works, Linear Regression, Decision Tree and Random Forest tend to give good accuracy so these models are used in this paper to predict crimes. The data set contains

different types of crimes that being committed in India according to the state and year respectively. This papertakes types of crimes as input and gives the area in which crimes are committed as output.The data pre-processing involves data cleaning, features election, dropping null values, data scaling by normalizing and standardizing. After data preprocessing the data is free of null values which may alter the accuracy of the model significantly and feature selection is used to select only the required features that won't affect the accuracy of model.After data pre-processing the models chosen i.e. Logistic Regression, Decision Tree and Random Forest are trained by splitting the data into as train and test data. As the output required is a categorical value classification models are used here. Python language is used for the data prediction.

3.4 HARDWARE REQUIREMENTS:

Minimum hardware requirements are very dependent on the particular software beingdeveloped by a given Thought Python / Canopy / VS Code user. Applications that need to store large arrays/objects in memory will require more RAM, whereas applications that need to perform numerous calculations or tasks more quickly will require a faster processor

- Operating system : windows, Linux
- Processor : minimum intel i3 or higher processor
- Ram: minimum 4 GB
- Hard disk: minimum 250 GB

3.5 SOFTWARE REQUIREMENTS:

The functional requirements or the overall description documents include the productperspective and features, operating system and operating environment, graphics requirements, design constraints and user documentation. The appropriation of requirements and implementation constraints gives the general overview of the project in regards to what theareas of strength and deficit are and how to tackle them.

- 3.5.1 Python idle 3.7 version (or)
- 3.5.2 Anaconda 3.7 (or)
- 3.5.3 Jupiter (or)
- 3.5.4 Google colab or any Browser

Chapter - 4

SOFTWARE ENVIRONMENT

4. Software Environment

4.1 MACHINE LEARNING

Machine learning is the scientific study of algorithms and statistical models that computer systems use to perform a specific task without using explicit instructions, relying on patterns and inference instead. It is seen as a subset of artificial intelligence. Machine learning algorithms build a mathematical model based on sample data, known as "training data", in order to make predictions or decisions without being explicitly programmed to perform the task. Machine learning algorithms are used in a wide variety of applications, such as email filtering and computer vision, where it is difficult or infeasible to develop a conventional algorithm for effectively performing the task. Machine learning uses data to detect various patterns in a given dataset.

1. It can learn from past data and improve automatically.
2. It is a data-driven technology.
3. Machine learning is much similar to data mining as it also deals with the huge amount of the data.

4.1.1 Relation to Data Mining

Machine learning and data mining often employ the same methods and overlap significantly, but while machine learning focuses on prediction, based on known properties learned from the training data, data mining focuses on the discovery of (previously) unknown properties in the data (this is the analysis step of knowledge discovery in databases). Data mining uses many machine learning methods, but with different goals; on the other hand, machine learning also employs data mining methods as unsupervised learning or as a preprocessing step to improve learner accuracy. Much of the confusion between these (two research communities) comes from the basic assumptions they work with: in machine learning, performance is usually evaluated with respect to the ability to reproduce known knowledge, while in knowledge discovery and data mining (KDD) the key task is the discovery of previously unknown knowledge. Evaluated with respect to known knowledge, an uninformed (unsupervised) method will easily be outperformed by other supervised methods, while in a typical KDD task supervised methods cannot be used due to the unavailability of training data.

4.1.2. Types of learning algorithms:

The types of machine learning algorithms differ in their approach, the type of data they input and output, and the type of task or problem that they are intended to solve.

- Supervised learning
- Unsupervised learning
- Reinforcement learning

1. Supervised learning:

Supervised learning algorithms build a mathematical model of a set of data that contains both the inputs and the desired outputs. The data is known as training data, and consists of a set of training examples. Each training example has one or more inputs and the desired output, also known as a supervisory signal. In the mathematical model, each training example is represented by an array or vector, sometimes called a feature vector, and the training data is represented by a matrix. Supervised learning can be grouped further in two categories of algorithms:

1. Classification
2. Regression

2. Unsupervised learning:

Unsupervised learning algorithms take a set of data that contains only inputs, and find structure in the data, like grouping or clustering of data points. The algorithms, therefore, learn from test data that has not been labeled, classified or categorized. Instead of responding to feedback, unsupervised learning algorithms identify commonalities in the data and react based on the presence or absence of such commonalities in each new piece of data. Cluster analysis is the assignment of a set of observations into subsets (called clusters) so that observations within the same cluster are similar according to one or more predestinated criteria, while observations drawn from different clusters are dissimilar. Different clustering techniques make different assumptions on the structure of the data, often defined by some similarity metric and evaluated, for example, by internal compactness, or the similarity between members of the same cluster, and separation. It can be further classified into two categories of algorithms:

1. Clustering

2. Association

3. Reinforcement learning:

Reinforcement learning is an area of machine learning concerned with how software agents ought to take actions in an environment so as to maximize some notion of cumulative reward. Due to its generality, the field is studied in many other disciplines, such as game theory, control theory, operations research, information theory, simulation-based optimization, multi-agent systems, swarm intelligence, statistics and genetic algorithms.

Reinforcement learning algorithms do not assume knowledge of an exact mathematical model of the MDP, and are used when exact models are infeasible. Reinforcement learning algorithms are used in autonomous vehicles or in learning to play a game against a human opponent.

4.2 About PYTHON

Python is a translated, object-oriented, abnormal state programming language with dynamic semantics. Its abnormal state worked in information structures, joined with dynamic composing and dynamic authoritative, make it appealing for Rapid Application Development, just as for use as a scripting or paste language to interface existing segments together. Python's basic, simple to learn language structure underlines intelligibility and hence decreases the expense of program support. Python underpins modules and bundles, which empowers program seclusion and code reuse. The Python translator and the broad standard library are accessible in source or parallel structure without charge for every single significant stage, and can be openly appropriated. Frequently, software engineers begin to look all starry eyed at Python on account of the expanded efficiency it gives. Since there is no aggregation step, the alter test troubleshoot cycle is staggeringly quick. Troubleshooting Python programs is simple: a bug or awful information will never cause a division blame.

Rather, when the mediator finds a blunder, it raises a special case. At the point when the program doesn't get the special case, the translator prints a stack follow. A source level debugger permits assessment of nearby and worldwide factors, assessment of discretionary articulations, setting breakpoints, venturing through the code a line at any given moment, etc. The debugger is written in Python itself, vouching for Python's contemplative power.

Then again, frequently the speediest method to troubleshoot a program is to add a couple of print proclamations to the source: the quick alter test-investigate cycle makes this straight forward methodology successful.

Python is generally basic, so it's anything but difficult to learn since it requires a one of a kind language structure that centers around coherence. Designers can peruse and interpret Python code a lot simpler than different dialects. Thusly, this decreases the expense of program upkeep and improvement since it enables groups to work cooperatively without huge language and experience obstructions. A standout amongst the most encouraging advantages of Python is that both the standard library and the mediator are accessible for nothing out of pocket, in both parallel and source structure. There is no restrictiveness either, as Python and all the important instruments are accessible on every single real stage. In this way, it is a tempting alternative for designers who would prefer not to stress over paying highimprovement costs.

4.3 ANACONDA

Anaconda constrictor is bundle director. Jupiter is an introduction layer. Boa constrictor endeavors to explain the reliance damnation in python—where distinctive tasks have diverse reliance variants—in order to not influence distinctive venture conditions to require diverse adaptations, which may meddle with one another. Jupiter endeavors to fathomthe issue of reproducibility in investigation by empowering an iterative and hands-on way to deal with clarifying and imagining code; by utilizing rich content documentations joined withvisual portrayals, in a solitary arrangement.

Boa constrictor is like pyenv, venv and minconda; it's intended to accomplish apython situation that is 100% reproducible on another condition, autonomous of whatever different forms of a task's conditions are accessible. It's somewhat like Docker, however limited to the Python biological system.

Jupyter is an astounding introduction device for expository work; where you can display code in "squares," joins with rich content depictions among squares, and the consideration of organized yield from the squares, and charts created in an all around plannedissue by method for another square's code. Jupyter is extraordinarily great in expository workto guarantee reproducibility in somebody's exploration, so anybody can return numerous months after the fact and outwardly comprehend what somebody attempted to clarify, and seeprecisely which code drove which representation and end.

Regularly in diagnostic work you will finish up with huge amounts of half-completed note pads clarifying Proof-of-Concept thoughts, of which most won't lead anywhere at first. A portion of these introductions may months after the fact—or even years after the fact—present an establishment to work from for another issue.

4.4 JUPYTER NOTEBOOK

The Jupyter Notebook App is a server-customer application that permits altering and running note pad records by means of an internet browser. The Jupyter Notebook App can be executed on a nearby work area requiring no web access (as portrayed in this report) or can be introduced on a remote server and got to through the web. Notwithstanding showing/altering/running note pad archives, the Jupyter Notebook App has a "Dashboard" (Notebook Dashboard), a "control board" indicating nearby records and permitting to open note pad reports or closing down their portions.

When you open a Notebook report, the related part is consequently propelled. At the point when the scratch pad is executed (either cell-by-cell or with menu Cell - > Run All), the portion plays out the calculation and produces the outcomes. Contingent upon the sort of calculations, the piece may expend critical CPU and RAM. Note that the RAM isn't discharged until the part is closed down, the Notebook Dashboard is the part which is indicated first when you dispatch Jupyter Notebook App. The Notebook Dashboard is essentially used to open note pad archives, and to deal with the running portions (picture and shutdown). The Notebook Dashboard has different highlights like a record director, in particular exploring organizers and renaming/erasing documents.

4.5 NUMPY

NumPy is, much the same as SciPy, Scikit-Learn, Pandas, and so forth one of the bundles that you can't miss when you're learning information science, principally in light of the fact that this library gives you a cluster information structure that holds a few advantages over Python records, for example, being increasingly reduced, quicker access in perusing and composing things, being progressively advantageous and increasingly productive.

NumPy exhibits are somewhat similar to Python records, yet at the same time particularly unique in the meantime. For those of you who are new to the subject,

how about we clear up what it precisely is and what it's useful for. As the name gives away, a NumPy cluster is a focal information structure of the numpy library. The library's name is another way to say "Numeric Python" or "Numerical Python".

At the end of the day, NumPy is a Python library that is the center library for logical registering in Python. It contains an accumulation of apparatuses and strategies that can be utilized to settle on a PC numerical models of issues in Science and Engineering. One of these apparatuses is an elite multidimensional cluster object that is an incredible information structure for effective calculation of exhibits and lattices. To work with these clusters, there's a tremendous measure of abnormal state scientific capacities work on these grids and exhibits. Since you have set up your condition, it's the ideal opportunity for the genuine work. In fact, you have officially gone for some stuff with exhibits in the above DataCamp Light pieces. Be that as it may, you haven't generally gotten any genuine hands-on training with them, since you originally expected to introduce NumPy all alone pc. Since you have done this current, it's a great opportunity to perceive what you have to do so as to run the above code pieces without anyone else. To make a numpy exhibit, you can simply utilize the `np.array ()` work. You should simply pass a rundown to it, and alternatively, you can likewise indicate the information sort of the information. In the event that you need to find out about the conceivable information types that you can pick, go here or consider investigating Data Camp's NumPy cheat sheet.

There's no compelling reason to proceed to retain these NumPy information types in case you're another client; but you do need to know and mind what information you're managing. The information types are there when you need more power over how your information is put away in memory and on plate. Particularly in situations where you're working with broad information, it's great that you know to control the capacity type.

Remember that, so as to work with the `np.array ()` work, you have to ensure that the numpy library is available in your condition. The NumPy library pursues an import tradition: when you import this library, you need to ensure that you import it as `np`. By doing this, you'll ensure that different Pythonistas comprehend your code all the more effectively.

4.6 PANDAS

Pandas is an open-source, BSD-authorized Python library giving elite, simple to-utilize information structures and information examination instruments for the Python programming language. Python with Pandas is utilized in a wide scope of fields including scholastic and business areas including money, financial matters, Statistics, examination, and so on. In this instructional exercise, we will get familiar with the different highlights of Python Pandas and how to utilize them practically speaking.

This instructional exercise has been set up for the individuals who try to become familiar with the essentials and different elements of Pandas. It will be explicitly valuable for individuals working with information purging and examination. In the wake of finishing this instructional exercise, you will wind up at a moderate dimension of ability from where you can take yourself to more elevated amounts of skill. You ought to have a fundamental comprehension of Computer Programming phrasings. A fundamental comprehension of any of the programming dialects is an or more. Panda's library utilizes the vast majority of the functionalities of NumPy. It is recommended that you experience our instructional exercise on NumPy before continuing with this instructional exercise.

Chapter - 5

SYSTEM STUDY

5.SYSTEM STUDY

Crime encompasses a number of disciplines, drawing on methods and techniques developed in both the natural and the social sciences. As do other disciplines, crime distinguishes between pure and applied research and between statistical and intuitive ways of thinking. More than most other disciplines, however, criminological research depends upon the willing cooperation of governmental agencies and other public authorities for the provision of essential data.

5.1. FEASABILITY STUDY

The manner and extent of data collection differ considerably from country to country and even within countries that have federal systems. Variables include how often data are collected and published, what items are given importance, whether the choice is between complete listings or sample surveys, and what the ratio between governmental and private research is. These differences, combined with differences in law and legal administration and in popular views and habits, have made it difficult to devise a meaningful system of international criminal statistics and to compare national statistics that are collected separately.

The most common data used in criminological research are official statistics, which are collected as part of the operation of criminal justice agencies.

For example, police collect data on the crimes they know about and on the people they arrest for committing those crimes; courts collect data on the cases that are brought to them and on the outcomes of those cases, including convictions; and prisons and jails, as well as probation and parole agencies, collect data on the people under their jurisdiction. In all cases the usefulness of official criminal statistics depends on human factors such as the willingness of private individuals to report criminal events to the police, of the police to officially respond to the criminal event, and of court officials to prosecute the case.

Because these decisions depend on a variety of factors—including whether the criminal laws at issue are popular or unpopular, whether the criminal event occurs in a high-crime or low-crime area, and whether the victim or offender is a member of a minority group—they are not very reliable as a measure of the amount of crime in a society or of changes in the amount of crime over time.

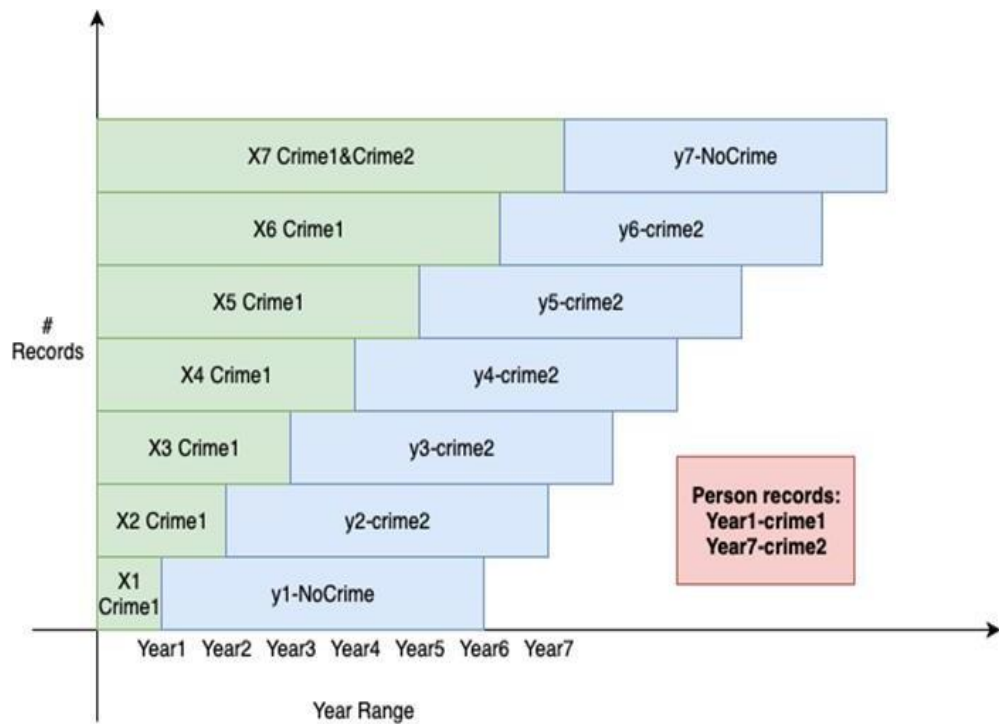


Fig: 5.1. 5 year look ahead window for labeling

Chapter - 6

MODULES

6. MODULES

6.1. Data Collection

The crime data collection is achieved using the third-party API provided by New York City Police Department (NYPD) at the NYC Open Data portal which is reserved for free federal data to involve civilians in the reports generated and managed by the city administration. This dataset comprises all credible transgressions delivered to the New York City Police Department (NYPD). The dataset is updated every three months. The variables stored in the dataset comprise of the following:- the name of the borough in which the incident occurred, the date and time of occurrence for the reported event, an intimation of whether the crime was interrupted prematurely, attempted but failed, or completed successfully, the level of offense, the specific location of occurrence in or around the premises, the description of the crime, the date of reporting of the event, the victim and the suspect's age group, race description, and sex description, and the latitude and longitude of the crime incident.

6.2 Data Preprocessing

Data preprocessing includes reconstructing primary data to proper data sets since machines cannot use data that they cannot interpret. Primary data is usually deficient and has incongruous formatting. The adequacy or inadequacy of data preparation is associated with the success of every project that requires analysis or prediction of data. Data Preprocessing comprises both validation and imputation of data. The purpose of validation is to evaluate whether the data is both comprehensive and precise. The purpose of the imputation of data is to rectify errors and input missing values for the preprocessing of the dataset, we split it into two separate datasets, one dataset for analysis and the other for prediction. For the prediction dataset, there were quite a few missing values. Instead of dropping entire rows, we replaced missing values with "UNKNOWN" values to avoid the loss of data. We also used the date and time to add the year, month, day of the week, the part of the day, and the hour at which the crime took place for better analysis. For the analysis dataset, we took the date, time, latitude, longitude, category, and description of the crime.

Since analysis prefers the data to be in numerical format, we created dummies for the crime categories and descriptions. We also used the date and time to add the year, month, and day of the week, part of the day, and the hour in a numerical format.

6.3 Data Set

A data set (or dataset) could be an assortment of knowledge. most ordinarily a knowledge set corresponds to the contents of one information table, or one applied mathematics information matrix, wherever each column of the table represents a specific variable, and every row corresponds to a given member of the information set in question. the information set lists values for every of the variables, like height Associate in Nursing weight of an object, for every member of the information set. every price is understood as a data point. Set might comprise data for one or additional members, appreciate the quantity of rows.

The dataset consists of the following details about the crime incidents:

Category - category of the crime incident. This is the target variable which is going to be predicted.

- Descript - description of the crime incident.
- Day Of Week - the day of the week.
- PD District - name of the Police Department District
- Resolution - how the crime incident was resolved.
- Address - the approximate street address of the crime incident.

6.4 FEATURE EXTRACTION

Next thing is to do Feature extraction is an attribute reduction process. Unlike feature selection, which ranks the existing attributes according to their predictive significance, feature extraction actually transforms the attributes. The transformed attributes, or features, are linear combinations of the original attributes. Finally, our models are trained using Classifier algorithm.

6.5 EVALUATION MODEL

Model Evaluation is an integral part of the model development process. It helps to find the best model that represents our data and how well the chosen model will work in the future. Evaluating model performance with the data used for training is not acceptable in data science because it can easily generate overoptimistic and over fitted models. Performance of each classification model is estimated base on its averaged. The result will be in the visualized form. Representation of classified data in the form of graphs.

CHAPTER - 7

ALGORITHMS

7. ALGORITHMS

Classification is a data mining technique that categorizes data in order to assist in more accurate predictions and analyzes. It is one of the data mining methods that aims to analyze very large datasets. It is used to derive patterns that accurately define the important data classes within the data set. Classification consists in predicting a given result based on a given input. Classification algorithms attempt to detect relationships between attributes that would make it possible to predict the result. They analyze the input and produce a prediction.

7.1 Decision Tree

A Decision Tree algorithm creates prediction models by following non-parametric supervised learning approaches. Here, from the training data, simple decision rules are presumed for the creation of training models that can predict the classes and the values of actual variables. By slabbing down a dataset into more petite subsets, this algorithm progressively develops an associated decision tree. The resultant tree is the one with decision nodes and leaf nodes.

The algorithms commonly used to construct decision trees are; ID3 and C4.5. The ID3 (Iterative Dichotomiser 3) algorithm induces classification models, or decision trees, from data. It is a supervised learning algorithm that is trained by examples for different classes. After being trained, the algorithm should be able to predict the class of a new item. ID3 identifies attributes that differentiate one class from another. All attributes must be known in advance, and must also be either continuous or selected from a set of known values. For instance, temperature (continuous), and country of citizenship (set of known values) are valid attributes. To determine which attributes are the most important, ID3 uses the statistical property of entropy.

The C4.5 algorithm overcomes this problem by using another statistical property known as information gain. Information gain measures how well a given attribute separates the training sets into the output classes. This algorithm has input in the form of training samples and samples.

Training samples in the form of sample data that will be used to build a tree that has been substantiated. C4.5 algorithms are algorithms result of the development of the algorithm ID3. C4.5 algorithm works by grouping several training sample data that will result in a decision tree based on the facts on the training data.

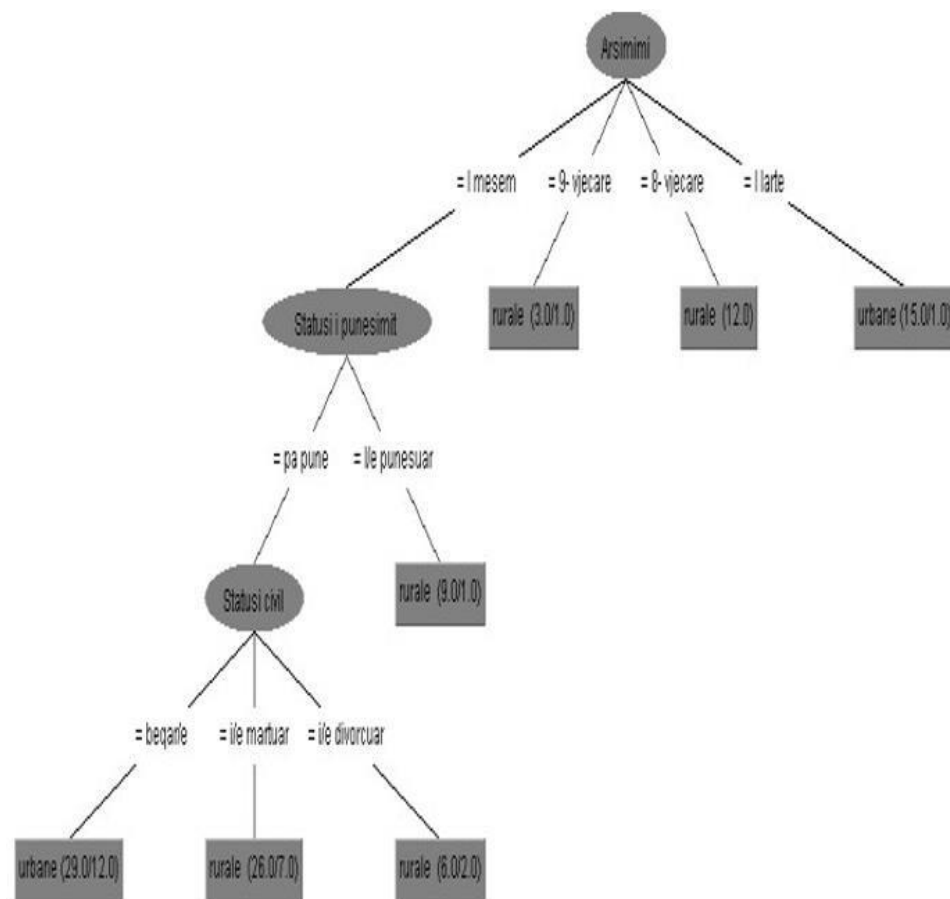


Fig: 7.1 Decision Tree

7.2 Artificial Neural Networks

Neural networks are an area of Artificial Intelligence (AI) based on the inspiration from the human brain. I use them to find data structures and algorithms for learning and classifying data. By applying neural network techniques, a program can learn from the examples and create an internal set of rules for classifying different inputs. Artificial Neural Networks (ANNs) are capable of predicting new observations from existing observations. A neural network consists of interconnected processing elements also called units, nodes.

All processes of a neural network are performed by this group of neurons or units. Each neuron is a separate communication device, making its operation relatively simple. The function of one unit is simply to receive data from other units, as a function of the inputs it receives to calculate an output value, which it sends to other units. In artificial neural networks, neurons are organized in layers which process information using dynamic state responses to external inputs. The Multilayer Perceptron (MLP) is a feed-forward artificial neural network model that maps sets of input data to a set of appropriate outputs. In a feed-forward neural network, the input signal traverses the neural network in a forward direction from the input layer to the output layer through the hidden layers.

7.3 Naive Bayes Classifier

Bayesian classification represents a supervised learning method as well as a statistical classification method. It assumes a high-probability underlying model, which allows us to determine in principle the uncertainties for the model, thus determining the probability of the results. The Naive Bayes Classifier technique is based on the Bayesian theorem and is used especially when the dimensionality of the inputs is high. Naive Bayes classifier is a term in Bayesian statistics dealing with a simple probabilistic classifier based on applying Bayes' theorem with strong (naive) independence assumptions. Bayesian classification provides practical learning algorithms and prior knowledge, here the observed data can be combined. Bayesian classification provides a useful perspective for understanding and evaluating many learning algorithms. It calculates the apparent hypothetical probability. The algorithm works as follows. Bayes' theorem offers a way to calculate the probability of a hypothesis based on our prior knowledge.

$$P(c | x) = \frac{P(x | c)P(c)}{P(x)}$$

Likelihood
Class Prior Probability

Posterior Probability
Predictor Prior Probability

$$P(c | X) = P(x_1 | c) \times P(x_2 | c) \times \cdots \times P(x_n | c) \times P(c)$$

- $P(c/x)$ is the posterior probability of *class (target)* given *predictor (attribute)*.
- $P(c)$ is the prior probability of *class*.
- (x/c) is the likelihood which is the probability of *predictor* given *class*.
- $P(x)$ is the prior probability of *predictor*.

7.4 Random Forest

The Random Forest algorithm is a supervised algorithm for classification built from decision tree algorithms. In this, the accuracy and precision are directly proportional to the number of trees that the model builds. Both classifications, as well as regression problems, are answered by the Random Forest Algorithm. It also helps to bypass overfitting and deals competently with the missing values in a dataset. It applies the bagging or bootstrap aggregation technique of ensemble learning methods to solve complex problems by consolidating several classifiers together taking the average or mean of the outputs.

7.5 Support Vector Machine

Support Vector Machines are based on the concept of decision making plans that set the boundaries of decisions. A decision plan is one that divides a group of objects that have different class memberships. Classification tasks that are based on the dividing lines between different class membership objects are known as hyper-plane Classifiers. SVMs are a set of related supervised learning methods used for classification and regression. Support Vector Machine (SVM) is primarily a classification method that performs classification tasks by constructing hyper-plane in a multidimensional space. The SVM uses statistical learning theory to search for a regularized hypothesis that fits the available data well without over-fitting. SVM also supports regression and classification techniques and can handle multiple continuous and categorical variables.

The efficiency of SVM-based classification is not directly dependent on the dimension of the classified entities. SVM can also be extended to learn nonlinear decision functions by first projecting the input data into a high dimensional space using kernel functions and formulating a linear classification problem in that space. SMO (Sequential Minimal Optimization) implements John C. Platt's sequential minimal optimization algorithm for training a Support Vector classifier using

polynomial or RBF (Radial Basis Function) kernels. This implementation globally replaces all lost values and transforms nominal attributes into binary ones. It can be seen that the choice of kernel function and best value of parameters for particular kernel is critical for a given amount of data. It also normalizes all attributes by default.

7.6 K NEAREST NEIGHBOURS

KNN may be used for each classification and regression prognostic issues.

However, it is a largely utilized in classification issues of the industry. To gauge any technique we usually look into 3 necessary aspects:

- i) Ease to interpret results.
- ii) Computation time.
- iii) Prediction capabilities.

KNN algorithm fares across all parameters of considerations. It is commonly used for its easy of interpretation and low calculation time. To understand the functioning of the K nearest neighbour's algorithm we need to consider the following situation: The following diagram contains a graph that consists of a group of squares (GS) and a group of circles (RC) and a star (BS). The aim is to find out which class does the star belong to Squares(S) or Circle(C).

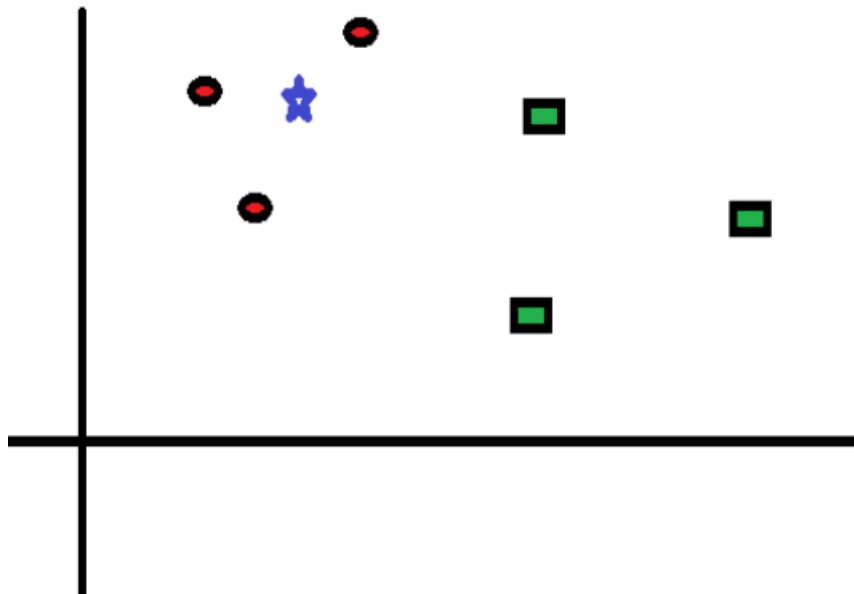


Fig: 7.6.1 KNN example before applying algorithm

The KNN algorithm is given in the pseudocode form as mentioned below. The KNN model can be built by following the steps that are mentioned below:

- Load the information
- Initialize the worth of k
- For obtaining the expected category, reiterate from one to total range of coaching knowledge points
- Calculate the space between take a look at knowledge and every row of coaching data. Here we are going to use geometric distance as our distance metric since it's the foremost in style technique. The opposite metrics which will be used are Chebyshev, cosine, etc.
- Sort the calculated distances in ascending order supported distance values
- Get prime k rows from the sorted array
- Get the foremost frequent category of those rows
- Return the expected category

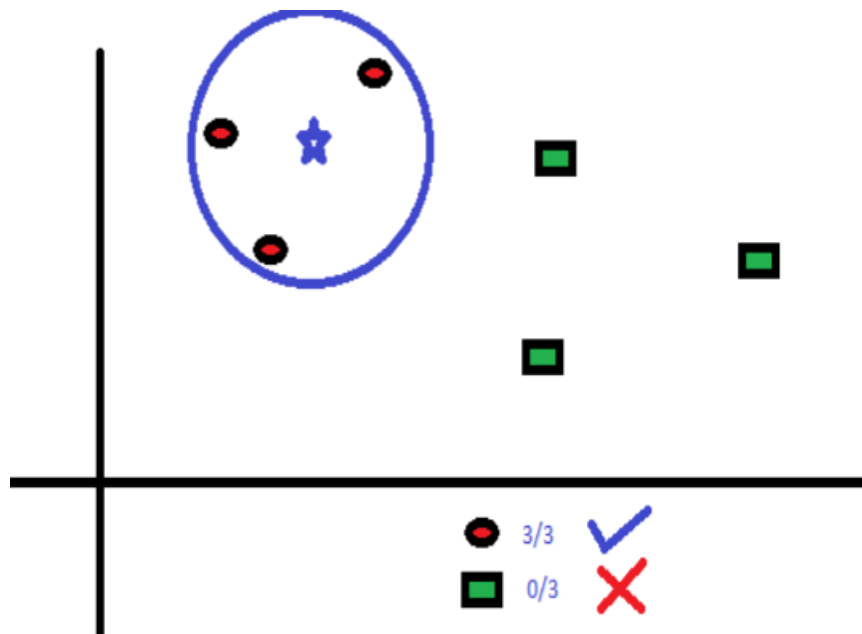


Fig: 7.6.2.KNN example after applying algorithm

7.7 LOGISTIC REGRESSION

In statistics, the logistical model (or logit model) could be a wide used appliedmath model. In its basic kind it uses a logistical operate to model a binary variable, though more advanced extensions exist. In multivariate analysis, logistical regression (or logit regression) is estimating the parameters of a logistic model (a sort of binomial regression).

Mathematically, a binary logistical model incorporates a variable with 2 doable values, like pass/fail, win/lose, alive/dead or healthy/sick; these are depicted by associate degree indicator variable, wherever the 2 values are tagged "0" and "1". Within the logistical model, the log-odds (the index of the odds) for the worth tagged "1" could be a linear combination of 1 or additional freelance variables ("predictors"); the independent variables will every be a binary variable (two categories, coded by associate degree indicator variable) or never-ending variable (any real value). The corresponding chance of the worth tagged "1" will vary between zero (certainly the value "0") and one (certainly the value "1"), thence the labeling; the operate that converts log-odds to chance is that the logistical function, hence the name.

The unit of measuring for the log-odds scale is termed a logit, from logistical unit, thence the choice names. Analogous models with a special sigmoid operate rather than the logistical function may be used, like the probit model; the shaping characteristic of the logistical model is that increasing one in all the freelance variables multiplicatively scales the percentages of the given outcome at a continuing rate, with every variable having its own parameter; for a binary experimental variable this generalizes the percentages magnitude relation.

In rectilinear regression, the relationships are sculptural victimizationlinear predictor functions whose unknown model parameters are calculable from the information. Such models are referred to as linear models. Most typically, the conditional mean of the response given the values of the instructive variables (or predictors) is assumed to be associate affine perform of these values; less ordinarily, the conditional median or another quantile is employed. Like all kinds of multivariate analysis, rectilinear regression focuses on the probability distribution of the response given the values of the predictors,

instead of on the chance distribution of all of those variables, that is that the domain of statistical procedure.

The technique may be utilized in engineering, particularly for predicting the chance of failure of a given method, system or product. It's conjointly utilized in selling applications like prediction of a customer's propensity to buy a product or halt a subscription, etc. In economic science it are often wont to predict the probability of a person'sselecting to be within the proletariat, and a business application would be to predict the probability of a house owner defaulting on a mortgage. Conditional random fields, associate degree extension of supplying regression to consecutive knowledge, are utilized in tongue process.

a. Comparison of Algorithms

Method	Algorithm	Correctly Classified Instances	Incorrectly Classified Instances	Recall	Precision	F-Measure
Regression	Simple Logistic	68 (68%)	32 (32 %)	0.680	0.680	0.665
	Logistic	71 (71%)	29 (29 %)	0.710	0.707	0.707
Bayes Classifier	Naive Bayes	73 (73%)	27 (27%)	0.730	0.737	0.732
	Bayes Net	72 (72%)	28 (28%)	0.720	0.725	0.721
SVM	SMO	67(67%)	33 (33 %)	0.670	0.666	0.666
Decision tree	C4.5	76 (76%)	24 (24 %)	0.760	0.762	0.761
Artificial Neural Network	Multilayer Perceptron	63 (63%)	37 (37%)	0.630	0.637	0.632

Fig: 7.8.Comparison of the results of the algorithms applied in WEKA

Chapter – 8

SYSTEM DESIGN

8. SYSTEM DESIGN

The System Design Document describes the system requirements, operating environment, system and subsystem architecture, files and database design, input formats, output layouts, human-machine interfaces, detailed design, processing logic, and external interfaces. This section describes the system in narrative form using non-technical terms. It should provide a high-level system architecture diagram showing a subsystem breakout of the system, if applicable. The high-level system architecture or subsystem diagrams should, if applicable, show interfaces to external systems. Supply a high-level context diagram for the system and subsystems, if applicable. Refer to the requirements traceability matrix (RTM) in the Functional Requirements Document (FRD). The organization code and title of the key points of contact (and alternates if appropriate) for the information system development effort. These points of contact should include the Project Manager, System Proponent, User Organization, Quality Assurance (QA) Manager, Security Manager, and Configuration Manager, as appropriate.

8.1 SYSTEM ARCHITECTURE

An unsupervised machine learning model used for making clusters of crime type as crime head labels further a supervised machine learning model is trained by using crime head data. The model is trained to predict the probability that a new crime head that should be reported and we can avoid crimes to be occurred.

- To utilize the resources identify the hotspots of crimes and allocate vigilante resources such as policeman, police cars, weapons etc. reschedule patrols according to the vulnerability of a place.
- Through that avoid crimes Ensure better civilization through avoiding happening crimes such as murder, rapes, thefts, drug, smugglings etc.

System Architecture:

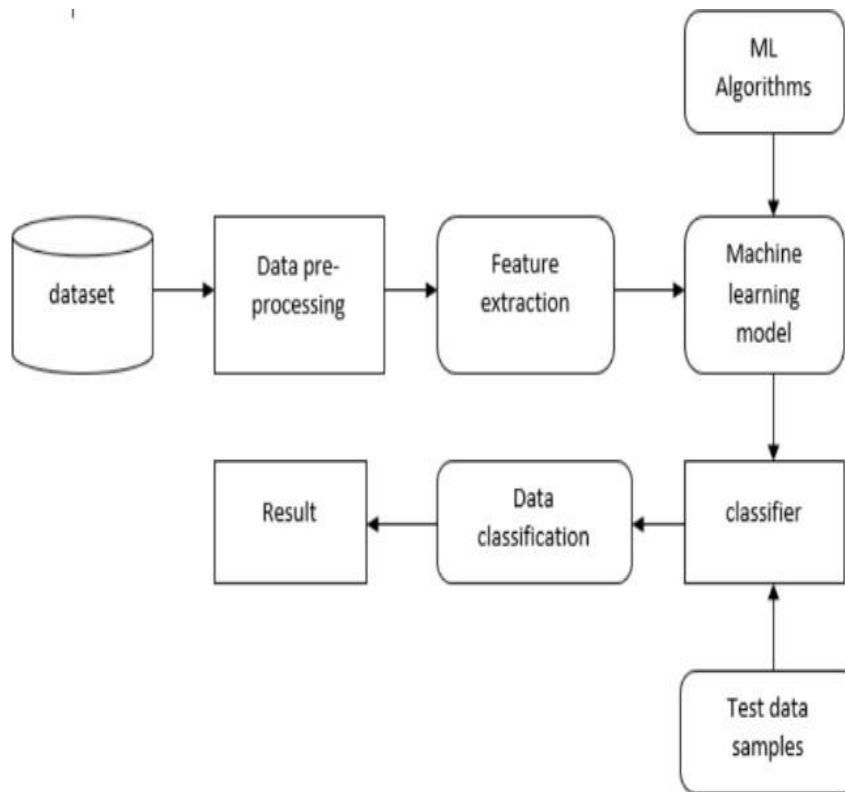


Fig: 8.1. Architecture Diagram

8.2 UML Diagrams

UML (Unified Modeling Language) is a standard vernacular for choosing, envisioning, making, and specifying the collectibles of programming structures. UML is a pictorial vernacular used to make programming blue prints. It is in like way used to exhibit non programming structures similarly like process stream in a gathering unit andso forth.²⁰ UML is not a programming vernacular yet rather instruments can be utilized to make code in different tongues utilizing UML graphs. UML has an incite relationship with question composed examination and outline. UML expect a fundamental part in portraying trade viewpoints of a structure.

8.2.1 Use case Diagram

The use case graph is for demonstrating the direct of the structure. This chart contains the course of action of use cases, performing pros and their relationship. This chart might be utilized to address the static perspective of the structure.

Use Case Diagram:

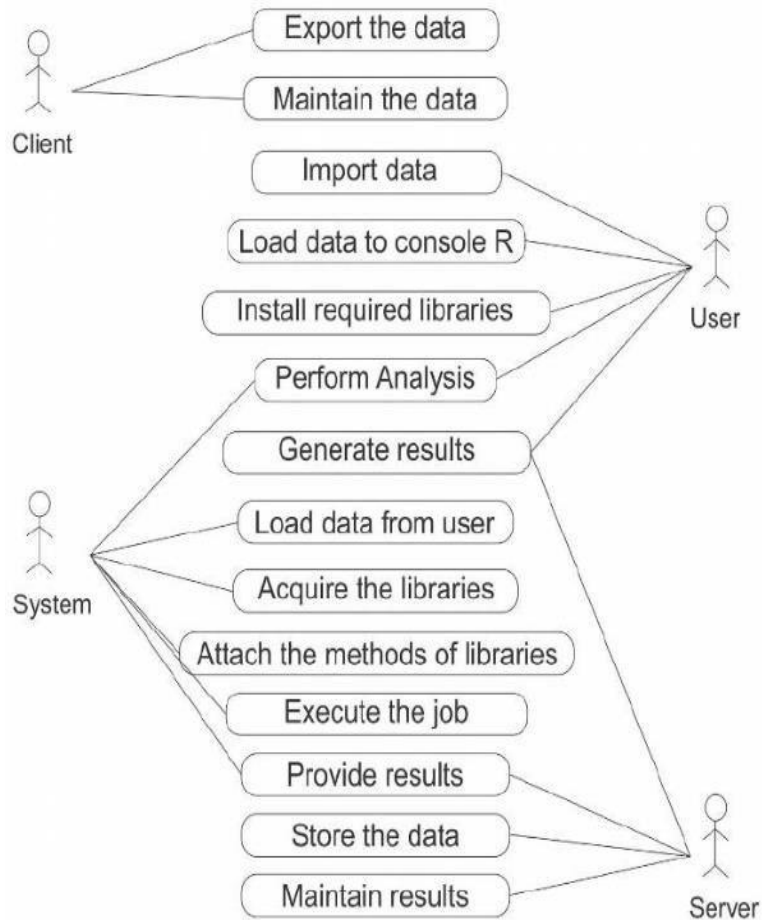


Fig: 8.2.1. Use Case Diagram

8.2.2 Class Diagram

The class graph is the most normally pulled in layout UML. It addresses the static course of action perspective of the structure. It solidifies the strategy of classes, interfaces, joint attempts and their affiliation.

Class Diagram:

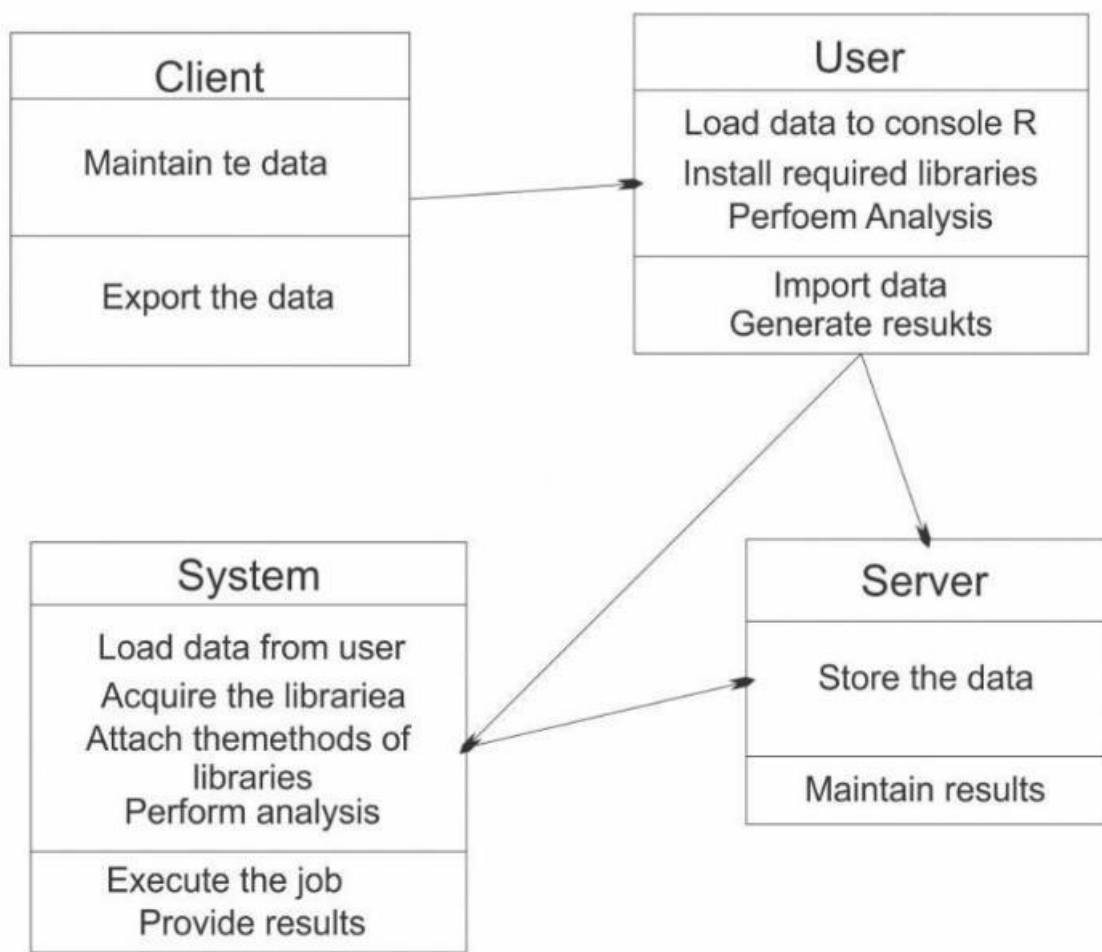


Fig: 8.2.2. Class Diagram

8.2.3 Sequence Diagram

This is a cooperation design which tends to the time requesting of messages. It includes set of parts and the messages sent and gotten by the instance of parts. This chart is utilized to address the dynamic perspective of the structure.

Sequence Diagram:

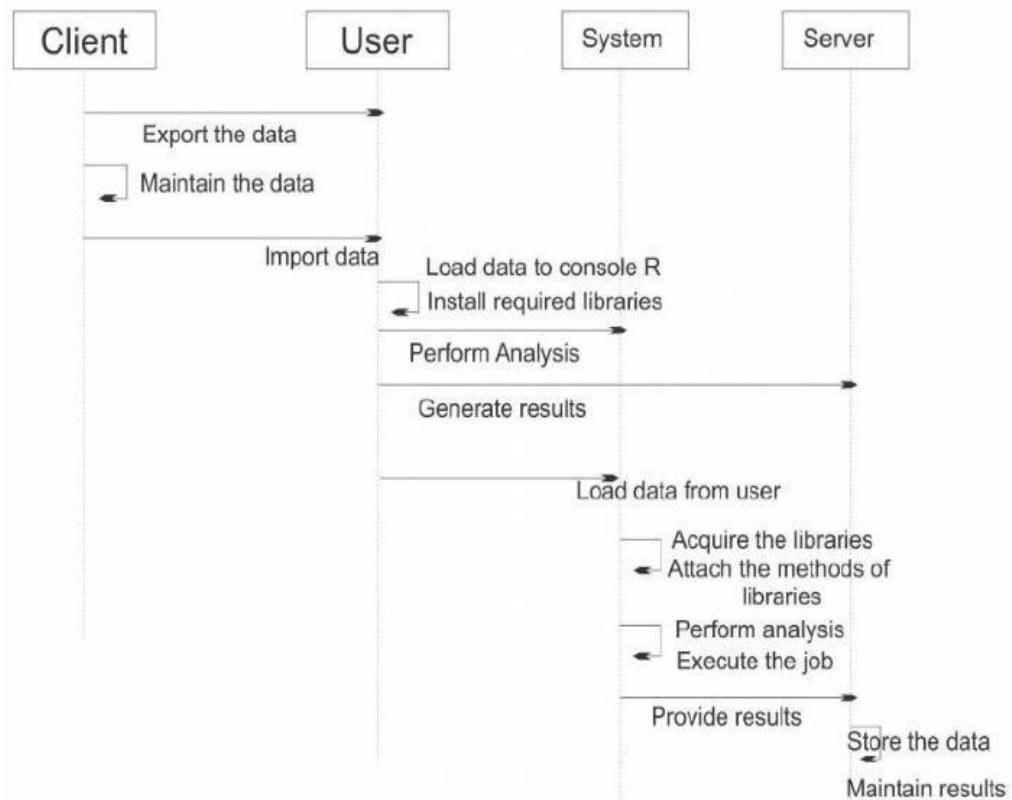


Fig: 8.2.3. Sequence Diagram

8.2.4 Activity Diagrams

Activity diagrams are graphical representations of Workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.

Activity Diagram:

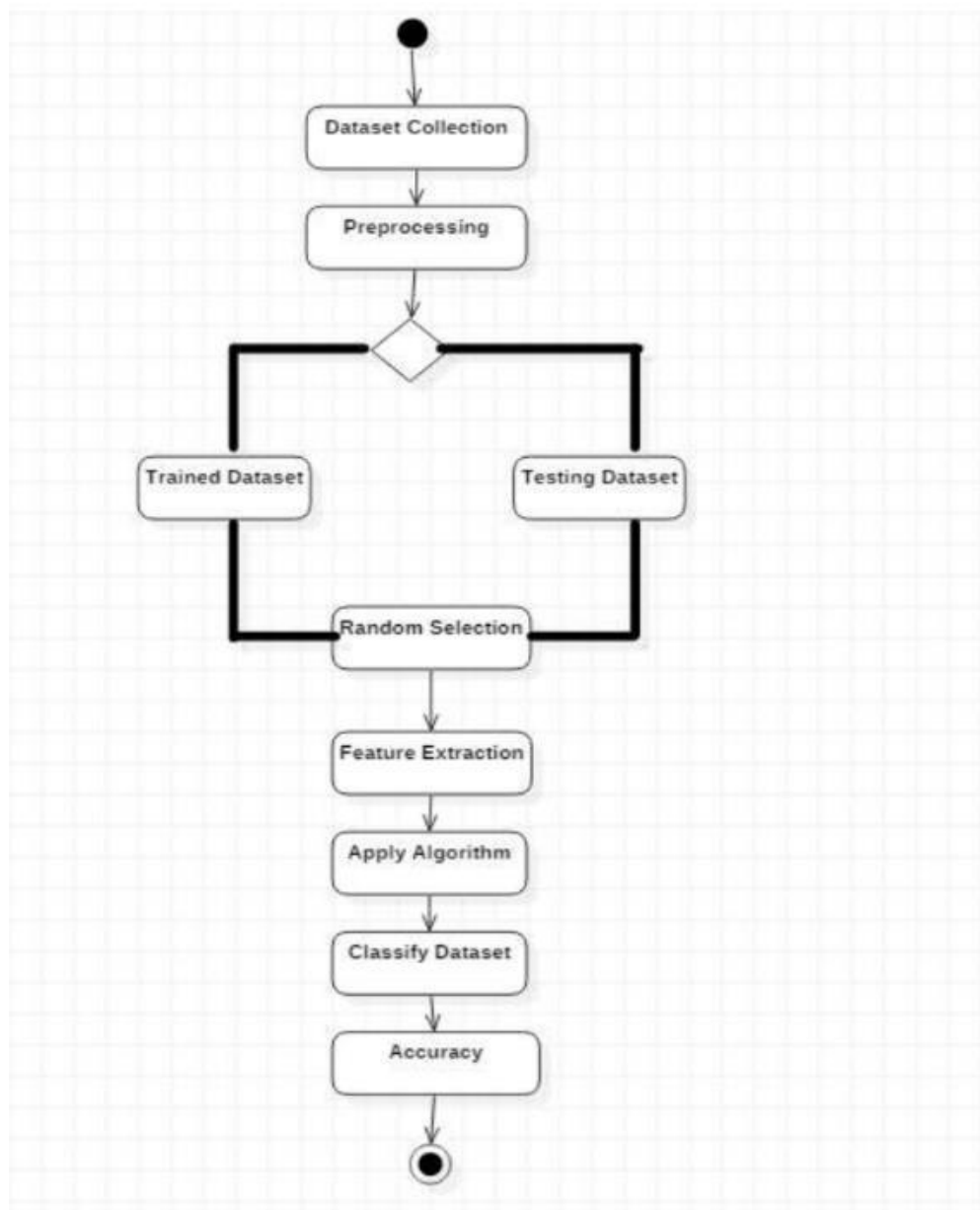


Fig: 8.2.4. Activity Diagram

Chapter – 9

SAMPLE CODE

SAMPLE CODE

9.1 CLUSTERING

```
from sklearn.cluster import KMeans
import pandas as pd
from sklearn.preprocessing import MinMaxScaler
import seaborn as sns
import numpy as np
from matplotlib import pyplot as plt
%matplotlib inline
df=pd.read_excel(r'DataFinal17-20.xlsx',sheet_name='2020')
df2=df.iloc[:,2:16]
data=df.iloc[:,1:23]
data=data.drop(['Total','Crime Rate','Sex Ratio','Literacy','Density','Latitude','Longitude'],
axis=1)
df2=df2.drop(['Total'], axis=1)
df2.head()
from sklearn import preprocessing
df_standardized = preprocessing.scale( df2 )
df1 = pd.DataFrame( df_standardized )
data.head()
#Elbow Method
from sklearn.cluster import KMeans

#create a list for the wcss parameter
wcss = []
#test with 14 clusters
for i in range(1, 15):
    kmeans = KMeans(n_clusters = i, init = 'k-means++', random_state = 0)
    kmeans.fit(df1)
    wcss.append(kmeans.inertia_)
wcss
plt.plot(range(1, 15), wcss)
plt.title('The Elbow Method')
plt.xlabel('Number of clusters')
plt.ylabel('WCSS')
plt.show()
import seaborn as sns
#Silhouette Method
from sklearn.metrics import silhouette_score

sil = []
kmax = 14

# dissimilarity would not be defined for a single cluster, thus, minimum number of clusters
should be 2
```

```

for k in range(2, kmax+1):
    kmeans = KMeans(n_clusters = k).fit(df1)
    labels = kmeans.labels_
    sil.append(silhouette_score(df1, labels, metric = 'euclidean'))
plt.plot(range(1, 14), sil)
plt.title('Silhouette Method')
plt.xlabel('Number of clusters')
plt.ylabel('Silhouette Coefficient')
plt.show()
#K-Means Clustering using k=6
km=KMeans(n_clusters=6)
y_pred=km.fit_predict(df1)
y_pred
df['cluster']=y_pred
df.head()
plt.scatter(df['SrNo'],df['cluster'])
#for col in df.columns:
#    print(col)
#Hierarchical Clustering
from scipy.cluster.hierarchy import linkage
import scipy.cluster.hierarchy as sch # for creating dendrogram
z = linkage(df1, method="complete",metric="euclidean")
plt.figure(figsize=(15,7))
plt.title('Hierarchical Clustering Dendrogram')
plt.xlabel('Features')
plt.ylabel('Crime')
sch.dendrogram(z,
    leaf_rotation=0., # rotates the x axis labels
    leaf_font_size=8., # font size for the x axis labels
)
plt.show()
#df = pd.read_excel (r'Path where the Excel file is stored\File name.xlsx',
sheet_name='your Excel sheet name')
X = df2[['Homicide/Murder','Causing death by negligence','Hurt','Assault on
woman','Kidnapping and abduction','Human trafficking ','Rape','Offence against public
tranquility','Offences against property','Offences relating to documents and property
marks','Miscellaneous','Others']]
clusters = KMeans(6) # 6 clusters
clusters.fit( X )
clusters.cluster_centers_
clusters.labels_
df2['Crime_clusters'] = clusters.labels_
df2.head()
df2.sort_values(by=['Crime_clusters'],ascending = True)
X.head()
stats =df2.sort_values("Hurt", ascending=True)
stats
sns.lmplot(x='Hurt',y='Assault on woman',data=df2,hue =
'Crime_clusters',fit_reg=False);
sns.lmplot(x='Hurt',y='Rape',data=df2,hue = 'Crime_clusters',fit_reg=False);
sns.lmplot(x='Assault on woman',y='Rape',data=df2,hue =

```

```

'Crime_clusters',fit_reg=False);
sns.lmplot(x='Kidnapping and abduction',y='Human trafficking ',data=df2,hue =
'Crime_clusters',fit_reg=False);
sns.lmplot(x='Hurt',y='Homicide/Murder',data=df2,hue = 'Crime_clusters',fit_reg=False);
sns.pairplot(data,hue='cluster')
df3 = pd.DataFrame(data=df2, columns=['Hurt', 'Homicide/Murder', 'Kidnapping and
abduction']) df3.plot.kde()
#kernel density estimate (KDE) plot is a method for visualizing the distribution of
observations in a
dataset
df2.head()
a1=data['Hurt']
b1=data['District']
f = plt.figure()
f.set_figwidth(10)
f.set_figheight(10)
plt.barh(b1, a1, align='center', alpha=0.5)
plt.ylabel('District')
plt.title('Top District with highest Crime rates')
plt.show()
#Cluster plotting of crime vs crime/diff metrics
df=pd.read_excel(r'DataFinal17-20.xlsx',sheet_name='2020')
df2=df.iloc[:,2:23]
for i in range(len(df2.columns)-1):
    titles=df2.columns[i]+" vs \n"+df2.columns[i+1]
    plt.figure(i)

sns.lmplot(x=df2.columns[i],y=df2.columns[i+1],data=df2,hue="cluster",legend=1,fit_re
g=False).set(title=titles)
#District vs Crime in 4 yrs
df=pd.read_excel(r'DataFinal17-20.xlsx',sheet_name='practice')
df2=df.iloc[:,0:6]
years=[2017,2018,2019,2020]
Districts=df['District'].unique()
Crimes=df['Crime'].unique()
for district in Districts:
    fig=plt.figure(figsize=(20,10),dpi=80,facecolor='w',edgecolor='k')
    plt.title(district)
    plt.xlabel('Years')
    plt.ylabel('No. Of Cases')
    for case in Crimes:
        temp_df=df[(df['District']==district)&(df['Crime']==case)]
        N_cases=[temp_df[c].values[0] for c in years]
        plt.plot(years,N_cases)
        plt.legend(Crimes)
#Crime vs Time(4yrs) for all district
for Crime in Crimes:
    fig=plt.figure(figsize=(20,10),dpi=80,facecolor='w',edgecolor='k')
    plt.title(Crime)
    plt.xlabel('Years')
    plt.ylabel('No. Of Cases')
    for district in Districts:

```

```
temp_df=df[(df['District']==district)&(df['Crime']==Crime)]
N_cases=[temp_df[c].values[0] for c in years]
plt.plot(years,N_cases)
plt.legend(Districts)
```

9.2 SUPPORT VECTOR MACHINE

```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
dataset = pd.read_excel(r'DataFinal17-20.xlsx',sheet_name='2020')
dataset = dataset.drop(['Total'], axis=1)
print(dataset)
X = dataset.iloc[:, 2:14].values
y = dataset.iloc[:, 19:20].values
y2 = dataset.iloc[:, 20:21].values
print(X)
print(y)
print(y2)
print(dataset)
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2, random_state = 0)
X_train, X_test, y2_train, y2_test = train_test_split(X, y2, test_size = 0.2, random_state = 0)
# Feature Scaling
from sklearn.preprocessing import StandardScaler
sc_X = StandardScaler()
sc_y = StandardScaler()
sc_y2 = StandardScaler()
X = sc_X.fit_transform(X)
y = sc_y.fit_transform(y)
y2 = sc_y2.fit_transform(y2)
# Fitting SVR to the dataset
from sklearn.svm import SVR
regressor = SVR(kernel = 'rbf')
#rbf = Gaussian Radial Basis Function Kernel
yfit = regressor.fit(X, y)
y2fit =regressor.fit(X, y2)
# Predicting a new result

y_pred =
regressor.predict(sc_X.transform(np.array([[254,620,1826,543,369,0,92,905,3104,220,30
535,221]])))
y_pred2 =
regressor.predict(sc_X.transform(np.array([[254,620,1826,543,369,0,92,905,3104,220,30
535,221]])))
```

```

y2_pred =
regressor.predict(sc_X.transform(np.array([[84,352,433,80,98,1,14,205,781,77,1237,175]
])))

y2_pred2 =
regressor.predict(sc_X.transform(np.array([[84,352,433,80,98,1,14,205,781,77,1237,175]
])))

y3_pred =
regressor.predict(sc_X.transform(np.array([[62,133,255,44,57,0,17,116,429,38,2930,305
3]])))
y3_pred2 =

regressor.predict(sc_X.transform(np.array([[62,133,255,44,57,0,17,116,429,38,2930,305
3]])))
y_pred = sc_y.inverse_transform(np.array(y_pred).reshape(1,-1))
y_pred2 = sc_y2.inverse_transform(np.array(y_pred2).reshape(1,-1))

y2_pred = sc_y.inverse_transform(np.array(y2_pred).reshape(1,-1))
y2_pred2 = sc_y2.inverse_transform(np.array(y2_pred2).reshape(1,-1))

y3_pred = sc_y.inverse_transform(np.array(y3_pred).reshape(1,-1))
y3_pred2 = sc_y2.inverse_transform(np.array(y3_pred2).reshape(1,-1))
def district_pred(a,b):
    target_district = ""
    min = 999
    num = 0
    for i in range(len(dataset['Latitude'])):
        if (abs(a-dataset['Latitude'][i])+abs(b-dataset['Longitude'][i]))<min:
            min = abs(a-dataset['Latitude'][i])+abs(b-dataset['Longitude'][i])
            target_district = dataset['District'][i]
            num = i
    return target_district,num
y_preds = [y_pred,y2_pred,y3_pred]
y_pred2s = [y_pred2,y2_pred2,y3_pred2]

for i in range(len(y_preds)):
    str, x = district_pred(y_preds[i],y_pred2s[i])
    print(str)
    print(dataset['Latitude'][x])
    print(dataset['Longitude'][x])

```

Chapter – 10

SCREEN SHOTS

10. SCREEN SHOTS

Jupyter notebook is the original web application for creating and sharing computational documents. It offers a simple, streamlined, document-centric experience.

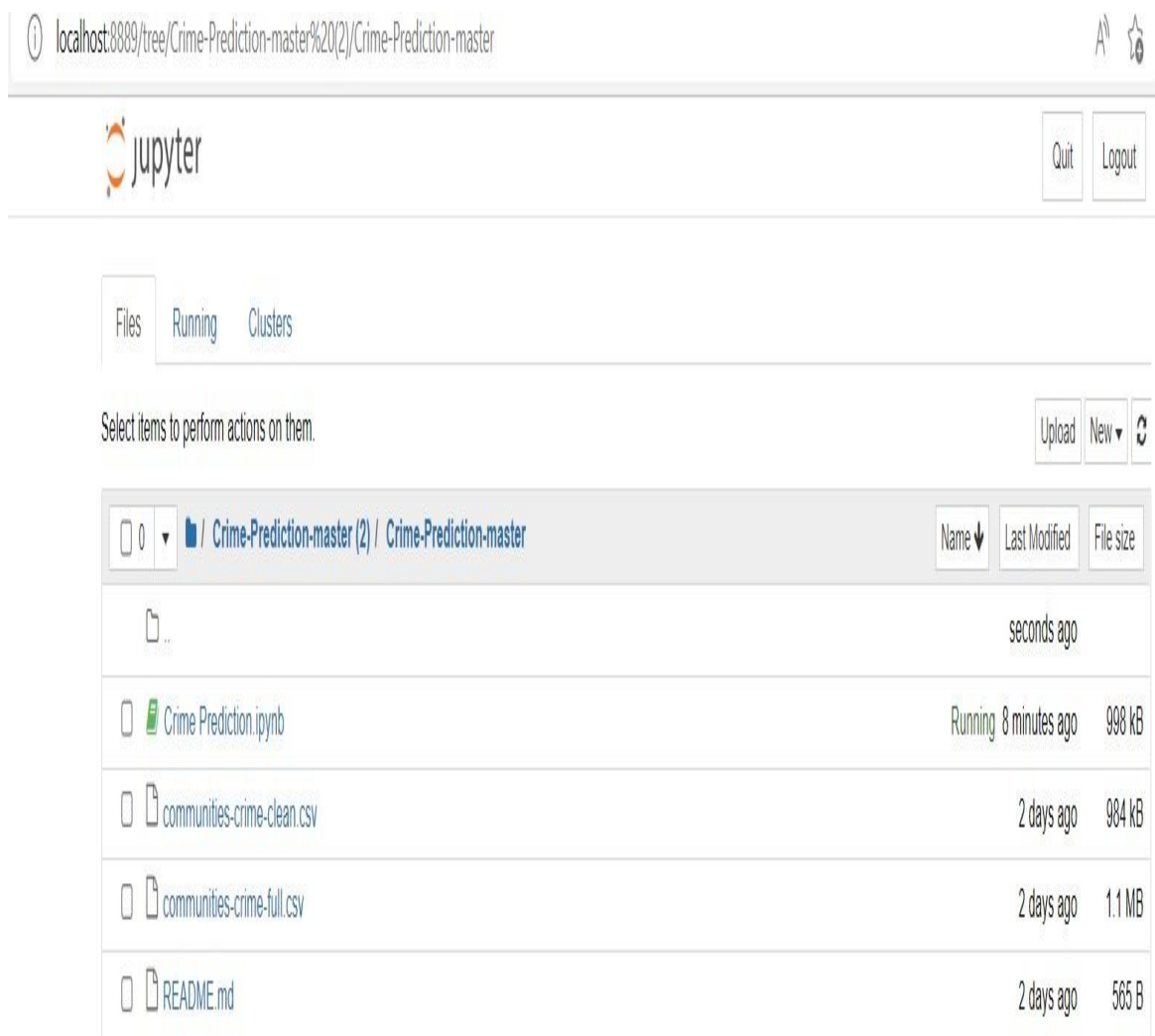
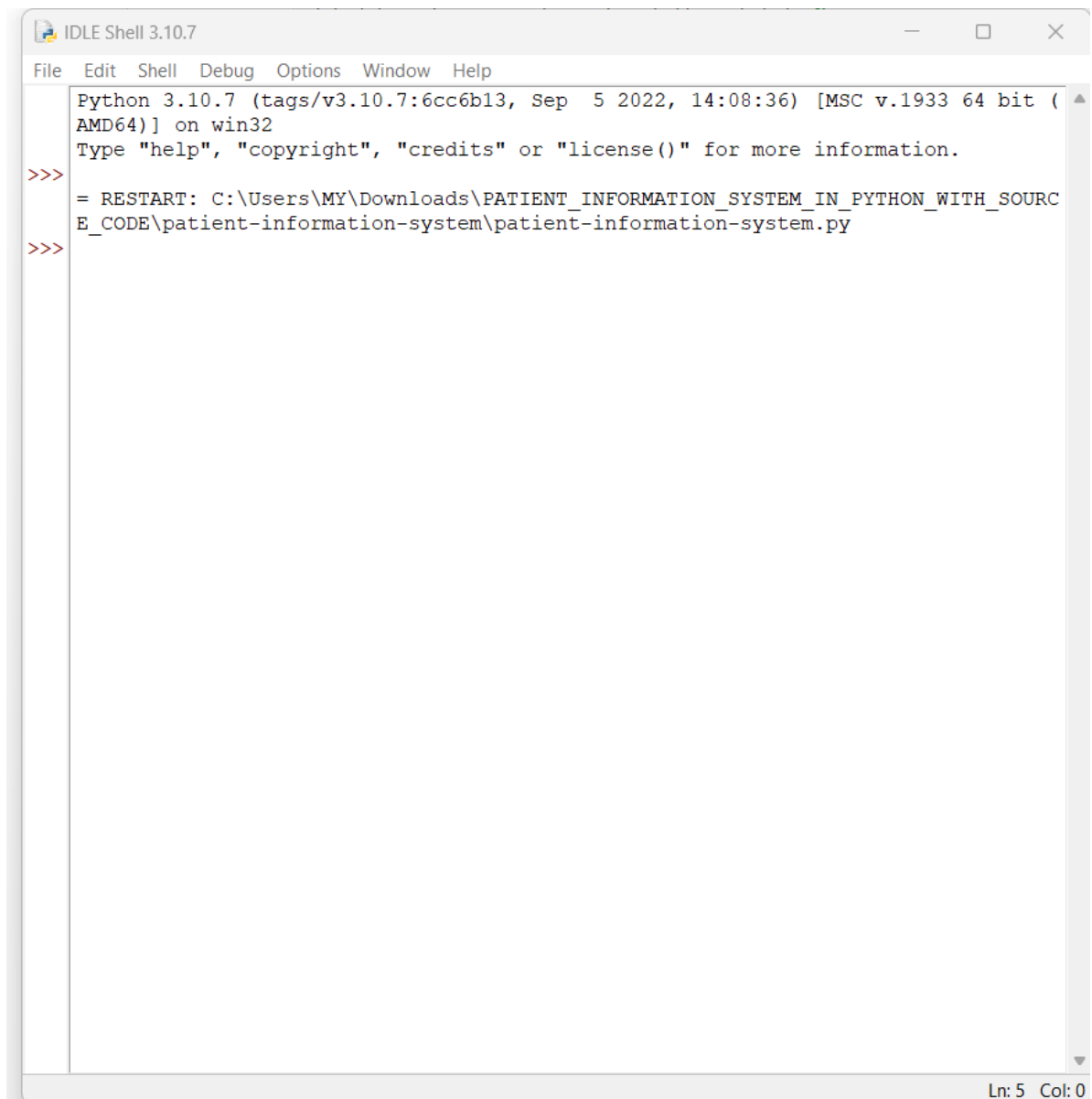


Fig: 10.1 Jupyter Notebook

Python is an interpreter language. It means it executes the code line by line. Python provides a Python Shell, which is used to execute a single Python command and display the result.

It is also known as REPL (Read, Evaluate, Print, Loop), where it reads the command, evaluates the command, prints the result, and loop it back to read the command again.

The image shows a screenshot of the IDLE Shell 3.10.7 window. The window has a title bar with the text "IDLE Shell 3.10.7" and standard window controls (minimize, maximize, close). Below the title bar is a menu bar with the following items: File, Edit, Shell, Debug, Options, Window, and Help. The main area of the window is a text editor displaying the Python REPL interface. The text in the editor is as follows:

```
Python 3.10.7 (tags/v3.10.7:6cc6b13, Sep 5 2022, 14:08:36) [MSC v.1933 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:\Users\MY\Downloads\PATIENT_INFORMATION_SYSTEM_IN_PYTHON_WITH_SOURCE_CODE\patient-information-system\patient-information-system.py
>>>
```

At the bottom right of the window, there is a status bar that reads "Ln: 5 Col: 0".

Fig: 10.2 Python Shell

```

from sklearn.cluster import KMeans
import pandas as pd
from sklearn.preprocessing import MinMaxScaler
import seaborn as sns
import numpy as np
from matplotlib import pyplot as plt
%matplotlib inline

```

Fig: 10.3 Libraries

```

from sklearn.metrics import accuracy_score
print ('Accuracy is', accuracy_score(Y,y_pred)*100)
from sklearn.metrics import precision_score
print ('Precesion is', precision_score(Y,y_pred)*100)
from sklearn.metrics import recall_score
print ('Recall is', recall_score(Y,y_pred)*100)

```

```

Accuracy is 83.592574009
Precesion is 90.0260190807
Recall is 83.04

```

Fig: 10.4 Decision Tree

```
plt.figure(figsize=(15,7))
plt.title('Hierarchical Clustering Dendrogram')
plt.xlabel('Features')
plt.ylabel('Crime')
sch.dendrogram(z,
    leaf_rotation=0., # rotates the x axis labels
    leaf_font_size=8., # font size for the x axis labels
)
plt.show()
```

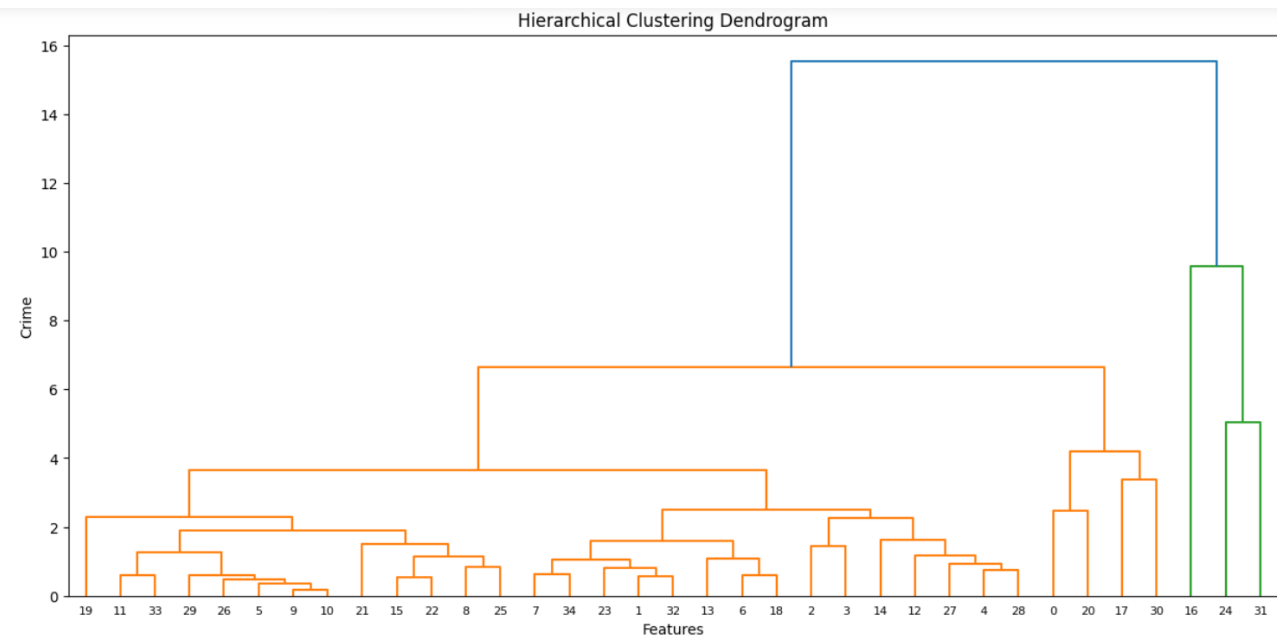


Fig: 10.5 Hierarchical Clustering Dendrogram

```
df3 = pd.DataFrame(data=df2, columns=['Hurt', 'Homicide/Murder', 'Kidnapping and abduction'])
df3.plot.kde()
#kernel density estimate (KDE) plot is a method for visualizing the distribution of observations in a dataset
```

<Axes: ylabel='Density'>

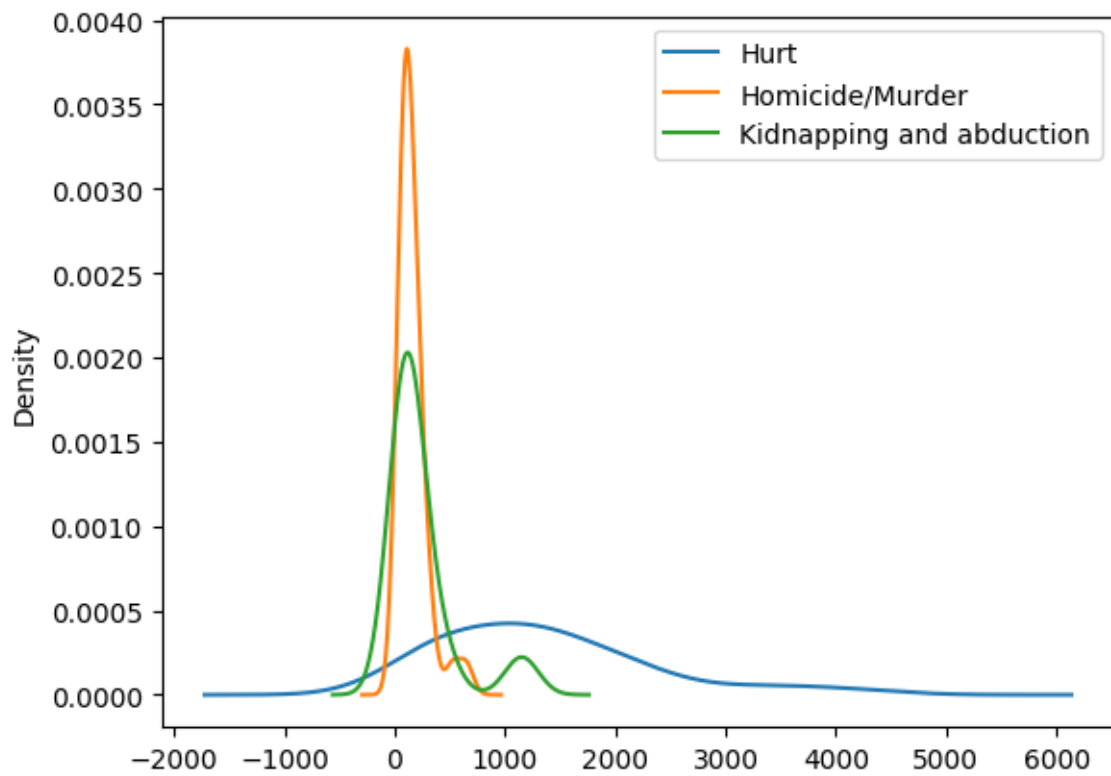


Fig: 10.6 Graph Representation

Chapter – 11

SYSTEM TESTING

11. SYSTEM TESTING

Testing:

Software testing is an investigation conducted to provide stakeholders with information about the quality of the product or service under test. Software Testing also provides an objective, independent view of the software to allow the business to appreciate and understand the risks at implementation of the software. Test techniques include, but are not limited to, the process of executing a program or application with the intent of finding software bugs.

Software Testing can also be stated as the process of validating and verifying that a software program/application/product:

- Meets the business and technical requirements that guided its design and Development.
- Works as expected and can be implemented with the same characteristics.

11.1. UNIT TESTING:

Unit testing is the testing of each module and the integration of the overall system is done. Unit testing becomes verification efforts on the smallest unit of software design in the module. This is also known as 'module testing'. The modules of the system are tested separately. This testing is carried out during the programming itself. In this testing step, each model is found to be working satisfactorily as regard to the expected output from the module. There are some validation checks for the fields. For example, the validation check is done for verifying the data given by the user where both format and validity of the data entered is included. It is very easy to find error and debug the system.

11.2 INTEGRATION TESTING:

Data can be lost across an interface, one module can have an adverse effect on the other sub function, when combined, may not produce the desired major function. Integrated testing is systematic testing that can be done with sample data. The need for the integrated test is to find the overall system performance.

There are two types of integration testing. They are:

Top-down integration testing.

Bottom-up integration testing.

11.3 FUNCTIONAL TESTING:

Functional testing is a type of testing that seeks to establish whether each application feature works as per the software requirements. Each function is compared to the corresponding requirement to ascertain whether its output is consistent with the end user's expectations. Functional testing focuses on the results of processing and not the mechanics of the processing, and determines whether the application satisfies the basic minimum user expectations.

11.4 WHITE BOX TESTING:

White Box testing is a test case design method that uses the control structure of the procedural design to drive cases. Using the white box testing methods, we derived test cases that guarantee that all independent paths within a module have been exercised at least once.

11.5 BLACK BOX TESTING:

In 'functional testing', is performed to validate an application conforms to its specifications of correctly performs all its required functions. So this testing is also called 'black box testing'. It tests the external behavior of the system. Here the engineered product can be tested knowing the specified function that a product has been designed to perform, tests can be conducted to demonstrate that each function is fully operational,

1. Black box testing is done to find incorrect or missing function
2. Interface error
3. Errors in external database access
4. Performance errors.

5. Initialization and termination error

11.6. SOFTWARE TESTING STRATEGIES

11.6.1 VALIDATION TESTING:

After the culmination of black box testing, software is completed assembly as a package, interfacing errors have been uncovered and corrected and final series of software validation tests begin validation testing can be defined as many, but a single definition is that validation succeeds when the software functions in a manner that can be reasonably expected by the customer.

11.6.2 USER ACCEPTANCE TESTING:

User acceptance of the system is the key factor for the success of the system. The system under consideration is tested for user acceptance by constantly keeping in touch with prospective system at the time of developing changes whenever required.

11.6.3 OUTPUT TESTING:

After performing the validation testing, the next step is output asking the user about the format required testing of the proposed system, since no system could be useful if it does not produce the required output in the specific format. The output displayed or generated by the system under consideration. Here the output format is considered in two ways. One is screen and the other is printed format. The output format on the screen is found to be correct as the format was designed in the system phase according to the user needs. For the hard copy also output comes out as the specified requirements by the user. Hence the output testing does not result in any connection in the system.

Chapter – 12

CONCLUSION

12. CONCLUSION

Crime prediction is one the current trends in the society. Crime prediction intends to reduce crime occurrences. It does this by predicting which type of crime may occur in future. Here, analysis of crime and prediction are performed with the help of various approaches. From the results obtained we saw that the training time of SVM is very high thus it should be avoided for this dataset. However which model will work best is totally dependent on the dataset that is being used. In this system, we get to classify and cluster to improve the accuracy of location and pattern-based crimes. This software predicts frequently occurring crimes, especially for particular state, and occurrences.

Chapter – 13

REFERENCES

13. REFERENCES

- [1] K. Zakir Hussain, M. Durairaj and G. R. J. Farzana, "Criminal behavior analysis by using data mining techniques," IEEE-International Conference on Advances in Engineering, Science and Management (ICAESM -2012), Nagapattinam, Tamil Nadu, 2012, pp. 656-658.
- [2] Keyvanpour, Mohammad & Javideh, Mostafa & Ebrahimi, Mohammadreza. (2011). Detecting and investigating crime by means of data mining: A general crime matching framework. *Procedia CS*. 3. 872-880. 10.1016/j.procs.2010.12.143.
- [3] Ioannis Kavakiotis OlgaTsava Athanasios Salifoglou Nicos Maglaveras Ioannis VlahavasIoanna Chouvarda, *Machine Learning and Data Mining Methods in Diabetes Research*, Computational and Structural Biotechnology Journal Volume 15, 2017, Pages 104-116
- [4] Frank, Eibe & Hall, Mark & Holmes, Geoffrey & Kirkby, Richard & Pfahringer, Bernhard & Witten, Ian & Trigg, Len. (2010). *Weka-A Machine Learning Workbench for Data Mining*. 10.1007/978-0-387- 09823-4_66.
- [5] Pang-Ning Tan; Michael Steinbach; Anuj Karpatne; Vipin Kuma *Introduction to Data Mining* 2 nd ed, Publisher: Pearson, 2019, Print ISBN: 9780133128901, 0133128903 e-textISBN: 9780134080284, 013408028.
- [6] M. Kantardzic, *Data Mining Concepts, Models, Methods, and Algorithms*, 2 nd ed, JohnWiley & Sons, Inc., Hoboken, New Jersey 2011, ISBN 978-0-470-89045-5, oBook ISBN: 978-1-118-02914-5, ePDF ISBN: 978-1-118-02912-1, ePub ISBN: 978-1-118-02913-8.

Chapter – 14

FUTURE SCOPE

14. FUTURE SCOPE

As of now, the project relies on manual input from a human (a police officer) in order to enter details in the database. If we can make this a centralized system and connect it to all the police stations countrywide and make FIR reporting digital, then it would be quite easier to predict crimes in that particular location and recognize patterns in them. It would also encourage citizens to track their E-FIR online. We can also avoid corruption as the government can keep a track on the number of cases registered and their solvability rate which can help them utilize their resources better.