## CRIME RATE PREDICTION & ANALYSIS USING K-MEANS CLUSTERING ALGORITHM

***Project report Submitted in partial fulfillment of the requirements for the award of the***

#### Degree of

**Master of Computer Applications**

**By**

**Mr. MANDAPURI VAMSHIKRISHNA**

**H.T. No. 1251-21-862-026**

**Under the guidance of**

**LYNNET ALICE EZRA**

**Assistant Professor**



**DEPARTMENT OF MASTER OF COMPUTER APPLICATIONS**

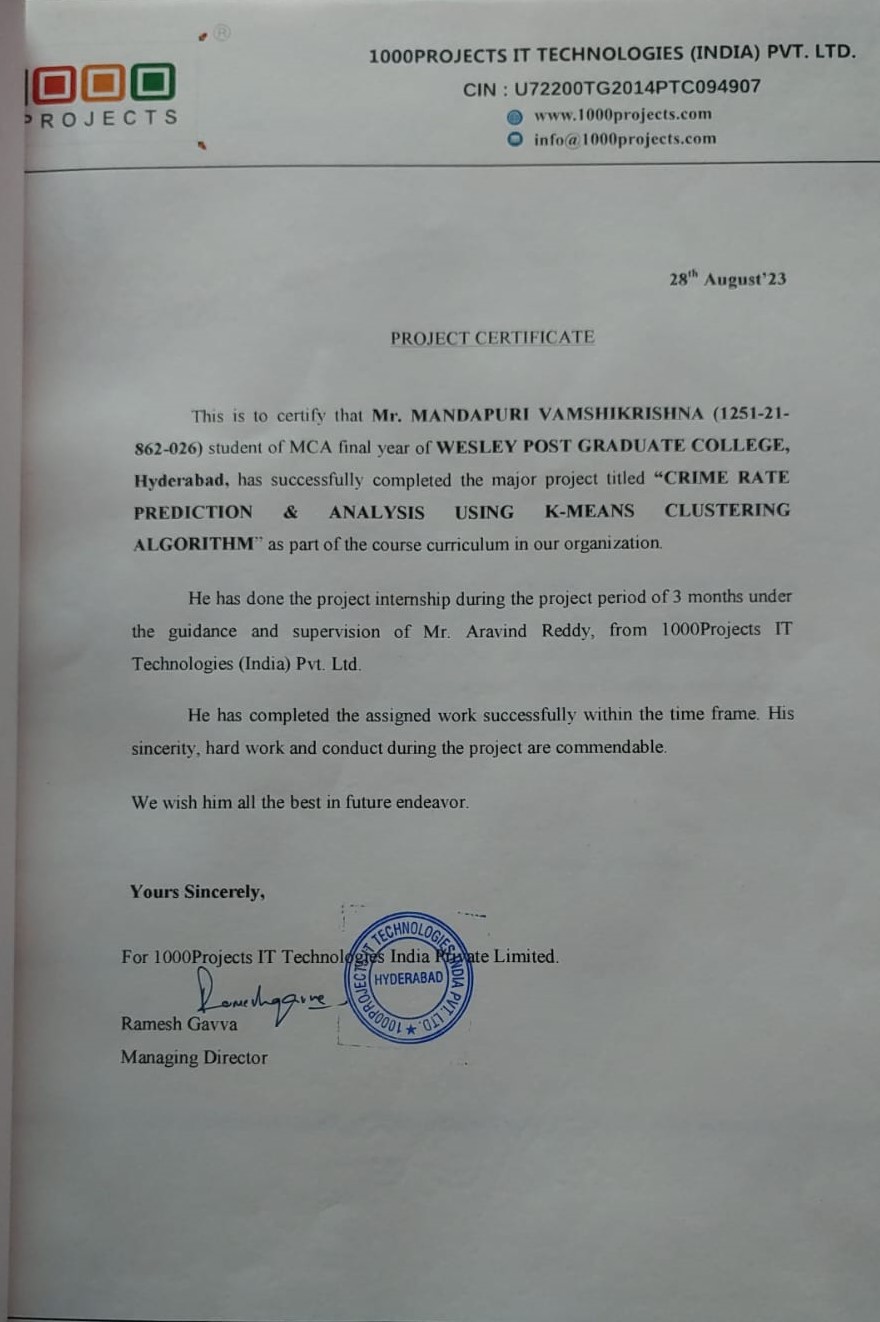
***CHURCH OF SOUTH INDIA- DIOCESE OF MEDAK***

**WESLEY POST GRADUATE COLLEGE**

**(Affiliated to Osmania University, Secunderabad & Approved by AICTE)**

**145, Mc Intyre Road, Secunderabad - 500 003**

**2021 - 2023**

****

**CERTIFICATE**

This is to certify that the major project work entitled **“ CRIME RATE PREDICTION & ANALYSIS USING K-MEANS CLUSTERING ALGORITHM”** is the bonafide work carried out by, **Mr. MANDAPURI VAMSHIKRISHNA, bearing H.T. No. 1251-21-862-026** is submitted in the partial fulfillment of the requirements for the award of the degree of Master of Computer Applications to the **Osmania University, Hyderabad** during the academic year **2022-2023**

**Internal Examiner External Examiner**

**Head of Department DIRECTOR**

**ACKNOWLEDGEMENT**

The successful completion of any task would be incomplete without the mention of people who made it possible through their constant guidance and encouragement.

I pay my sincere thanks to Prof**. Dr. B. VIMAL SUKUMAR**, **DIRECTOR WESLEY PG COLLEGE** for providing me with all the necessary amenities.

I wish to express my profound deep sense of gratitude **Dr. P.M.YOHAN**, **HEAD** **OF THE DEPARTMENT, WESLEY PG COLLEGE** for his effective encouragement and enthusiastic guidance for the successful completion of my project work.

I am also indebted to my project guide **Ms. LYNNET ALICE EZRA**, **ASST PROFESSOR** for her continuous support and valuable suggestions for successful completion of my project work.

I take this opportunity to acknowledge the importance of other faculty members in our department for their kindness and timely helping during my project work.

### DECLARATION

I **Mr. MANDAPURI VAMSHIKRISHNA** bearing **H.T No. 1251-21-862-026** here by declaring that the major project report entitled “**CRIME RATE ANALYSIS & PREDICTION USING K-MEANS CLUSTERING ALGORITHM”** is done by me under the guidance of **LYNNET ALICE EZRA** , Assistant. Professor, Department of MCA, Wesley PG College, is submitted in partial fulfillment of the requirements for the award of degree in Master of Computer Applications**.**

**Mr. MANDAPURI VAMSHIKRISHNA**

**H.T No. 1251-21-862-026**

**Place: Secunderabad**

**Date:**

**ABSTRACT**

In India, the crime rate is increasing each day. In the current situation, recent technological influence, effects of social media and modern approaches help the offenders to achieve their crimes. Both analysis and prediction of crime is a systematized method that classifies and examines the crime patterns. There exist various clustering algorithms for crime analysis and pattern prediction but they do not reveal all the requirements. Among these, K means algorithm provides a better way for predicting the results. The proposed research work mainly focused on predicting the region with higher crime rates and age groups with more or less criminal tendencies. We propose an optimized K means algorithm to lower the time complexity and improve efficiency in the result

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | **CONTENTS** |  |
|  | | | **PAGE NO** |
| **1. Introduction** | | | 01 |
| 1.1 Introduction to project | | | 02 |
| 1.2Purpose of the project | | | 03 |
| 1.3Existing system and its disadvantages | | | 05 |
| 1.4 Proposed system and its advantages | | | 06 |
| **2. System Analysis** | | | 07 |
| 2.1 | Study of system | | 08 |
| 2.2 | System architecture | | 11 |
| **3. Feasibility study** | | | 12 |
| 3.1 : Feasibility study | | | 13 |
| 3.2.Operational feasibility | | | 15 |
| **4. System Requirements Specifications** | | | 17 |
| 4.1 Hardware requirements | | | 18 |
| 4.2 Software Requirements | | | 19 |
| 4.2.1 Python | | | 19 |
| 4.2.2Machine Learning | | | 25 |
| **5. System Design** | | | 46 |
| 5.1 UML Diagram | | | 47 |
| 5.2 Use Case Diagrams | | | 49 |
| 5.3 Class Diagrams | | | 50 |
| 5.4 Sequence Diagrams | | | 51 |
| 5.5 Collaboration Diagram | | | 52 |
| 5.6 Activity Diagram | | | 53 |
| **6. Coding** | | | 55 |
| 6.1 Code | | | 56 |
| **7. Testing** | | | 72 |
| 7.1 System test | | | 73 |
| 7.2 Functionality Test | | | 74 |
| 7.3 Integration Test | | | 76 |
| **8. Output Screens** | | | 77 |
| 8.1 Output Screens | | | 78 |
| **9.Conclusion** | | | 88 |
| 9.1 Conclusion | | | 89 |
| **10. Future Enhancement** | | | 90 |
| 10.1 Future Enhancement | | | 91 |
| **11.References/Bibliography** | | | 95 |
| 11.1 References/Bibliography | | | 96 |

**INTRODUCTION**

**1.1 INTRODUCTION**

In present scenario criminals are becoming technologically sophisticated in committing crime and one challenge faced by intelligence and law enforcement agencies is difficulty in analyzing large volume of data involved in crime and terrorist activities therefore agencies need to know technique to catch criminal and remain ahead in the eternal race between the criminals and the law enforcement. So appropriate field need to chosen to perform crime analysis and as data mining refers to extracting or mining knowledge from large amounts of data, data mining is used here on high volume crime dataset and knowledge gained from data mining approaches is useful and support police forces. To perform crime analysis appropriate data mining approach need to be chosen and as clustering is an approach of data mining which groups a set of objects in such a way that object in the same group are more similar than those in other groups and involved various algorithms that differ significantly in their notion of what constitutes a cluster and how to efficiently find them. In this paper k means clustering technique of data mining used to extract useful information from the high volume crime dataset and to interpret the data which assist police in identify and analyze crime patterns to reduce further occurrences of similar incidence and provide information to reduce the crime. In this paper k mean clustering is implemented using open source data mining tool which are analytical tools used for analyzing data .Among the available open source data mining suite such as R, Tanagra ,WEKA ,KNIME ,ORANGE ,Rapid miner.k means clustering is done with the help of rapid miner tool which is an open source statistical and data mining package written in Java with flexible data mining support options. Also for crime analysis dataset used is Crime dataset an offences recorded by the police in England and Wales by offence and police force area from 1990 to 2011-12 .In this paper homicide which is crime committed by human by killing another human is being analyzed .

1.2 PURPOSE OF THE PROJECT

The purpose of the "Crime Rate Prediction & Analysis Using K-Means Clustering Algorithm" project is to leverage data-driven methodologies to address critical issues related to crime rates in specific geographic regions. This project aims to fulfill the following key objectives:

1. Crime Rate Prediction: Develop a predictive model using the K-Means clustering algorithm to anticipate future crime rates in various neighborhoods or regions. By analyzing historical crime data and identifying patterns, the project intends to provide law enforcement agencies, policymakers, and communities with insights that can help in proactively allocating resources and taking preventive measures.

2. Geospatial Analysis: Perform geospatial analysis to identify crime hotspots and cold spots within a given area. By utilizing K-Means clustering, the project aims to group regions with similar crime characteristics, enabling a better understanding of geographical variations in crime rates.

3. Resource Allocation: Assist law enforcement agencies and local governments in optimizing the allocation of resources such as police patrols, surveillance cameras, and community outreach programs. By accurately predicting crime rates, authorities can prioritize their efforts in areas with higher crime potential.

4. Community Engagement: Engage with local communities by sharing the results of the analysis in an easily understandable format. This promotes transparency and encourages community involvement in crime prevention efforts, fostering a sense of shared responsibility for public safety.

5. Data-Driven Policy Decisions: Provide policymakers with data-driven insights to formulate and adjust policies aimed at reducing crime rates. By understanding the underlying factors contributing to crime, policymakers can make informed decisions to create safer environments for residents.

6. Early Intervention: Enable early intervention by identifying trends and patterns in crime rates. By recognizing emerging hotspots, the project aims to facilitate timely intervention to prevent crime escalation.

7. Evaluation of Existing Strategies: Assess the effectiveness of existing crime prevention strategies and interventions. By comparing the predicted outcomes with actual crime rates, the project can help refine and improve existing approaches.

8. Public Safety Awareness: Raise public awareness about crime trends and safety concerns through the dissemination of easily accessible information. Empower individuals to make informed decisions regarding their safety and security.

In summary, the "Crime Rate Prediction & Analysis Using K-Means Clustering Algorithm" project seeks to harness the power of data analytics and machine learning to enhance public safety, inform policy decisions, and promote collaborative efforts between communities and law enforcement agencies to create safer living environment

**1.3 EXISTING SYSTEM :**

Crime analysis tool is developed using various distinct data mining methods. It supports the police officers for investigating crimes . Implementing a clustering algorithm on crime datasets enables analysis of crimes . It makes identification and analysis of various criminality trends over the years through their conclusion. The random initial starting points produced by K-means which gives results in the form of cluster that helps in reaching the local optima [8]. So to overcome this problem, the partitioned data along with the data axis with the highest variance for assigning the initial centroid for K-Means clustering was applied. So it is observed that the proposed technique uses a lesser number of iteration thereby reducing the clustering time. Using merge sort, K-means algorithm can be improved for clustering the Hidden Markov Model (HMM

**EXISTING SYSTEM DISADVANTAGES:**

1.LESS ACCURACY

2. LOW EFFICIENCY

**1.4 PROPOSED SYSTEM :**

We are working on Spyder for implementation. Here we use a Spyder 3.7 version. Spyder is an integrated development environment for systematic programming in Python. Here we implemented different packages like matplotlib,numpy,sklearn, pandas,etc. Which helps to plot elbow graph and data frame table using a K-means clustering algorithm? Dataset is collected from Kaggle datasets and import datasets into Spyder in CSV format as shown in Fig 1. We perform normalization for finding the accurate number of clusters (k) using the elbow method. The elbow method performs k- means clustering on the obtained dataset for a range of values of k (2-15) and calculates the SSE. A line chart of the SSE is plotted for each value of k

**PROPOSED SYSTEM ADVANTAGES:**

1.HIGH ACCURACY

2.HIGH EFFICIENCY

System Analysis

**2.1 STUDY OF SYSTEM**

The study of the system for the "Crime Rate Prediction & Analysis Using K-Means Clustering Algorithm" project involves a comprehensive analysis of the methodologies, components, and processes required to execute the project successfully. This study is essential to ensure that the project meets its objectives effectively. The following aspects will be covered in the system study:

1. Data Collection:

- Identify and collect relevant historical crime data, including types of crimes, locations, dates, and times.

- Source data from law enforcement agencies, government databases, or other reliable sources.

- Ensure data quality and accuracy through data cleansing and preprocessing.

2. Data Analysis:

- Explore and visualize the collected data to gain insights into crime patterns and trends.

- Conduct statistical analysis to identify correlations and potential influencing factors.

- Determine the suitability of data for K-Means clustering.

3. K-Means Clustering:

- Explain the K-Means clustering algorithm and its suitability for crime rate analysis.

- Select appropriate features and parameters for clustering, such as the number of clusters (K).

- Implement the K-Means algorithm using programming languages like Python or R.

4. Geographic Information System (GIS) Integration:

- Incorporate geographic data (e.g., maps, coordinates) to perform geospatial analysis.

- Utilize GIS tools or libraries to visualize crime hotspots and coldspots.

- Overlay clustering results on maps for better visualization.

5. Model Training and Validation:

- Split the dataset into training and testing sets for model validation.

- Train the K-Means clustering model using historical crime data.

- Validate the model's accuracy and performance through appropriate metrics.

6. Predictive Modeling:

- Apply the trained K-Means model to predict future crime rates in different regions.

- Monitor model performance and update it periodically as new data becomes available.

7. Resource Allocation Strategy:

- Develop a resource allocation strategy based on clustering results and crime predictions.

- Recommend allocation of law enforcement resources to areas with higher predicted crime rates.

8. Community Engagement and Reporting:

- Develop user-friendly interfaces or dashboards to present the results to law enforcement agencies, policymakers, and the public.

- Conduct outreach and awareness campaigns to engage communities in crime prevention efforts.

9. Policy Recommendations:

- Analyze the data to identify factors contributing to crime.

- Provide data-driven policy recommendations to reduce crime rates.

10. Evaluation and Feedback Loop:

- Continuously evaluate the project's effectiveness in reducing crime rates and improving public safety.

- Collect feedback from law enforcement agencies and communities to make necessary adjustments and improvements.

11. Security and Privacy Measures:

- Implement stringent security measures to protect sensitive crime data.

- Ensure compliance with data privacy regulations and ethical considerations.

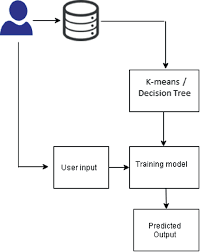
12. Documentation and Knowledge Transfer:

- Maintain comprehensive documentation of the system's architecture, algorithms, and processes.

- Facilitate knowledge transfer to ensure the project's sustainability beyond its initial implementation.

This study of the system provides a roadmap for the "Crime Rate Prediction & Analysis Using K-Means Clustering Algorithm" project, outlining the key components and activities required to achieve its objectives of enhancing public safety, informing policy decisions, and fostering community engagement in crime prevention efforts.

**2.2 SYSTEM ARCHITECTURE:**



**FEASIBILITY STUDY**

# 3.1 FEASIBILITY STUDY

# The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are

* ECONOMICAL FEASIBILITY
* TECHNICAL FEASIBILITY
* SOCIAL FEASIBILITY

**ECONOMICAL FEASIBILITY**

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

**TECHNICAL FEASIBILI**

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

**SOCIAL FEASIBILITY**

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

3.2 operational feasibility

Operational feasibility is a crucial aspect to consider when planning a project like "Crime Rate Prediction and Analysis Using K-Means Clustering." It assesses whether the project can be executed effectively within the constraints of resources, technology, and other practical considerations. Here is an operational feasibility analysis for this project:

1. Resource Availability:

- Data: Ensure access to a comprehensive and up-to-date dataset of crime incidents. Collaboration with law enforcement agencies or open data sources may be required.

- Computational Resources: Assess the availability of computing infrastructure to handle the data processing and clustering algorithms efficiently.

- \*\*Skilled Workforce\*\*: Ensure that there are individuals with the necessary skills in data analysis, machine learning, and programming to execute the project successfully.

2. Technology and Tools:

- K-Means Algorithm: Confirm that the K-Means clustering algorithm is an appropriate choice for crime rate analysis, considering the nature of the data.

- Software: Verify that the necessary software and tools for data preprocessing, analysis, and visualization are available and compatible.

3. Data Privacy and Legal Compliance:

- Consider legal and ethical aspects surrounding the use of crime data. Ensure that the project complies with data privacy regulations and does not infringe upon individuals' rights.

4. Data Quality:

- Assess the quality of the crime data, including completeness, accuracy, and consistency. Prepare strategies to handle missing or inconsistent data.

5. Scalability:

- Evaluate whether the project can handle an increase in the volume of data over time as crime data accumulates. Ensure that the system can scale as needed.

6. User Acceptance:

- Gather feedback from potential users, such as law enforcement agencies or community organizations, to ensure that the system's output aligns with their needs and expectations.

7. Maintenance and Support:

- Plan for the long-term maintenance and support of the system. This includes updates to the clustering model, data refresh, and troubleshooting.

8. Cost Analysis:

- Conduct a cost-benefit analysis to determine whether the project's benefits outweigh its costs. Consider the expenses associated with data acquisition, software licenses, personnel, and infrastructure.

9. Timeline and Milestones:

- Develop a realistic project timeline with well-defined milestones to track progress. Ensure that the project can be completed within the desired timeframe.

10. Risk Assessment:

- Identify potential risks that could impact the project's success, such as data breaches, technical challenges, or resource limitations. Develop mitigation strategies for these risks.

11. Integration with Existing Systems:

- If applicable, assess how the crime rate prediction and analysis system will integrate with existing law enforcement or community safety systems.

12. Training and Knowledge Transfer:

- Ensure that team members receive adequate training in using the system and interpreting its results. Consider knowledge transfer to future personnel who will maintain the project.

In summary, operational feasibility for the "Crime Rate Prediction and Analysis Using K-Means Clustering" project involves evaluating resources, technology, data quality, legal compliance, and user acceptance to determine whether the project can be successfully executed within practical constraints. Addressing these considerations early in the project planning phase can help ensure its success.

System Requirements Specification

**Software Requirement Specification**

**4.1 HARD REQUIRMENTS :**

* System    :   i3 or above.
* Ram    :   4 GB.
* Hard Disk : 40 GB

**4.2 SOFTWARE REQUIRMENTS :**

* Operating system   : Windows 8 or Above.
* Coding Language  : python

**SOFTWARE ENVIRONMENT :**

**4.2.1 PYTHON**

Below are some facts about Python.

Python is currently the most widely used multi-purpose, high-level programming language.

Python allows programming in Object-Oriented and Procedural paradigms. Python programs generally are smaller than other programming languages like Java.

Programmers have to type relatively less and indentation requirement of the language, makes them readable all the time.

Python language is being used by almost all tech-giant companies like – Google, Amazon, Facebook, Instagram, Dropbox, Uber… etc.

The biggest strength of Python is huge collection of standard library which can be used for the following –

* + [Machine Learning](https://www.geeksforgeeks.org/machine-learning/)
  + GUI Applications (like Kivy, Tkinter , PyQt etc. )
  + Web frameworks like Django (used by YouTube, Instagram, Dropbox)
  + Image processing (like Opencv, Pillow)
  + Web scraping (like Scrapy, BeautifulSoup, Selenium)
  + Test frameworks
  + Multimedia

**Advantages of Python :-**

Let’s see how Python dominates over other languages.

**1. Extensive Libraries**

Python downloads with an extensive library and it contain code for various purposes like regular expressions, documentation-generation, unit-testing, web browsers, threading, databases, CGI, email, image manipulation, and more. So, we don’t have to write the complete code for that manually.

**2. Extensible**

As we have seen earlier, Python can be**extended to other languages**. You can write some of your code in languages like C++ or C. This comes in handy, especially in projects.

## 3. Embeddable

Complimentary to extensibility, Python is embeddable as well. You can put your Python code in your source code of a different language, like C++. This lets us add **scripting capabilities**to our code in the other language.

## 4. Improved Productivity

The language’s simplicity and extensive libraries render programmers**more productive** than languages like Java and C++ do. Also, the fact that you need to write less and get more things done.

## 5. IOT Opportunities

Since Python forms the basis of new platforms like Raspberry Pi, it finds the future bright for the Internet Of Things. This is a way to connect the language with the real world.

When working with Java, you may have to create a class to print **‘Hello World’**. But in Python, just a print statement will do. It is also quite **easy to learn, understand,** and**code.** This is why when people pick up Python, they have a hard time adjusting to other more verbose languages like Java.

## 7. Readable

Because it is not such a verbose language, reading Python is much like reading English. This is the reason why it is so easy to learn, understand, and code. It also does not need curly braces to define blocks, and **indentation is mandatory.** This further aids the readability of the code.

## 8. Object-Oriented

This language supports both the **procedural and object-oriented**programming paradigms. While functions help us with code reusability, classes and objects let us model the real world. A class allows the **encapsulation of data** and functions into one.

## 9. Free and Open-Source

Like we said earlier, Python is **freely available.** But not only can you[**download Python**](https://data-flair.training/blogs/install-python-windows/) for free, but you can also download its source code, make changes to it, and even distribute it. It downloads with an extensive collection of libraries to help you with your tasks.

## 10. Portable

When you code your project in a language like C++, you may need to make some changes to it if you want to run it on another platform. But it isn’t the same with Python. Here, you need to**code only once**, and you can run it anywhere. This is called **Write Once Run Anywhere (WORA)**. However, you need to be careful enough not to include any system-dependent features.

## 11. Interpreted

Lastly, we will say that it is an interpreted language. Since statements are executed one by one, **debugging is easier** than in compiled languages.

Any doubts till now in the advantages of Python? Mention in the comment section.

# Advantages of Python Over Other Languages :

## 1. Less Coding

Almost all of the tasks done in Python requires less coding when the same task is done in other languages. Python also has an awesome standard library support, so you don’t have to search for any third-party libraries to get your job done. This is the reason that many people suggest learning Python to beginners.

## 2. Affordable

Python is free therefore individuals, small companies or big organizations can leverage the free available resources to build applications. Python is popular and widely used so it gives you better community support.

**The 2019 Github annual survey showed us that Python has overtaken Java in the most popular programming language category.**

## 3. Python is for Everyone

Python code can run on any machine whether it is Linux, Mac or Windows. Programmers need to learn different languages for different jobs but with Python, you can professionally build web apps, perform data analysis and [**machine learning**](https://data-flair.training/blogs/machine-learning-tutorials-home/), automate things, do web scraping and also build games and powerful visualizations. It is an all-rounder programming language.

## Disadvantages of Python

So far, we’ve seen why Python is a great choice for your project. But if you choose it, you should be aware of its consequences as well. Let’s now see the downsides of choosing Python over another language.

#### 1. Speed Limitations

We have seen that Python code is executed line by line. But since [Python](https://www.python.org/) is interpreted, it often results in **slow execution**. This, however, isn’t a problem unless speed is a focal point for the project. In other words, unless high speed is a requirement, the benefits offered by Python are enough to distract us from its speed limitations.

#### 2. Weak in Mobile Computing and Browsers

While it serves as an excellent server-side language, Python is much rarely seen on the **client-side**. Besides that, it is rarely ever used to implement smartphone-based applications. One such application is called **Carbonnelle**.

The reason it is not so famous despite the existence of Brython is that it isn’t that secure.

#### 3. Design Restrictions

As you know, Python is **dynamically-typed**. This means that you don’t need to declare the type of variable while writing the code. It uses **duck-typing**. But wait, what’s that? Well, it just means that if it looks like a duck, it must be a duck. While this is easy on the programmers during coding, it can**raise run-time errors**.

#### 4. Underdeveloped Database Access Layers

Compared to more widely used technologies like **JDBC (Java DataBase Connectivity)** and **ODBC (Open DataBase Connectivity)**, Python’s database access layers are a bit underdeveloped. Consequently, it is less often applied in huge enterprises.

#### 5. Simple

No, we’re not kidding. Python’s simplicity can indeed be a problem. Take my example. I don’t do Java, I’m more of a Python person. To me, its syntax is so simple that the verbosity of Java code seems unnecessary.

This was all about the Advantages and Disadvantages of Python Programming Language.

## History of Python : -

What do the alphabet and the programming language Python have in common? Right, both start with ABC. If we are talking about ABC in the Python context, it's clear that the programming language ABC is meant. ABC is a general-purpose programming language and programming environment, which had been developed in the Netherlands, Amsterdam, at the CWI (Centrum Wiskunde &Informatica). The greatest achievement of ABC was to influence the design of Python.Python was conceptualized in the late 1980s. Guido van Rossum worked that time in a project at the CWI, called Amoeba, a distributed operating system. In an interview with Bill Venners1, Guido van Rossum said: "In the early 1980s, I worked as an implementer on a team building a language called ABC at Centrum voor Wiskunde en Informatica (CWI). I don't know how well people know ABC's influence on Python. I try to mention ABC's influence because I'm indebted to everything I learned during that project and to the people who worked on it."Later on in the same Interview, Guido van Rossum continued: "I remembered all my experience and some of my frustration with ABC. I decided to try to design a simple scripting language that possessed some of ABC's better properties, but without its problems. So I started typing. I created a simple virtual machine, a simple parser, and a simple runtime. I made my own version of the various ABC parts that I liked. I created a basic syntax, used indentation for statement grouping instead of curly braces or begin-end blocks, and developed a small number of powerful data types: a hash table (or dictionary, as we call it), a list, strings, and numbers."

# 4.2.2 Machine Learning : -

Before we take a look at the details of various machine learning methods, let's start by looking at what machine learning is, and what it isn't. Machine learning is often categorized as a subfield of artificial intelligence, but I find that categorization can often be misleading at first brush. The study of machine learning certainly arose from research in this context, but in the data science application of machine learning methods, it's more helpful to think of machine learning as a means of building models of data.

Fundamentally, machine learning involves building mathematical models to help understand data. "Learning" enters the fray when we give these models tunable parameters that can be adapted to observed data; in this way the program can be considered to be "learning" from the data. Once these models have been fit to previously seen data, they can be used to predict and understand aspects of newly observed data. I'll leave to the reader the more philosophical digression regarding the extent to which this type of mathematical, model-based "learning" is similar to the "learning" exhibited by the human brain.Understanding the problem setting in machine learning is essential to using these tools effectively, and so we will start with some broad categorizations of the types of approaches we'll discuss here.

# Categories Of Machine Leaning :-

At the most fundamental level, machine learning can be categorized into two main types: supervised learning and unsupervised learning.

Supervised learning involves somehow modeling the relationship between measured features of data and some label associated with the data; once this model is determined, it can be used to apply labels to new, unknown data. This is further subdivided into classification tasks and regression tasks: in classification, the labels are discrete categories, while in regression, the labels are continuous quantities. We will see examples of both types of supervised learning in the following section.

Unsupervised learning involves modeling the features of a dataset without reference to any label, and is often described as "letting the dataset speak for itself." These models include tasks such as clustering and dimensionality reduction. Clustering algorithms identify distinct groups of data, while dimensionality reduction algorithms search for more succinct representations of the data. We will see examples of both types of unsupervised learning in the following section.

# Need for Machine Learning

Human beings, at this moment, are the most intelligent and advanced species on earth because they can think, evaluate and solve complex problems. On the other side, AI is still in its initial stage and haven’t surpassed human intelligence in many aspects. Then the question is that what is the need to make machine learn? The most suitable reason for doing this is, “to make decisions, based on data, with efficiency and scale”.

Lately, organizations are investing heavily in newer technologies like Artificial Intelligence, Machine Learning and Deep Learning to get the key information from data to perform several real-world tasks and solve problems. We can call it data-driven decisions taken by machines, particularly to automate the process. These data-driven decisions can be used, instead of using programing logic, in the problems that cannot be programmed inherently. The fact is that we can’t do without human intelligence, but other aspect is that we all need to solve real-world problems with efficiency at a huge scale. That is why the need for machine learning arises.

# Challenges in Machines Learning :-

While Machine Learning is rapidly evolving, making significant strides with cybersecurity and autonomous cars, this segment of AI as whole still has a long way to go. The reason behind is that ML has not been able to overcome number of challenges. The challenges that ML is facing currently are −

**Quality of data** − Having good-quality data for ML algorithms is one of the biggest challenges. Use of low-quality data leads to the problems related to data preprocessing and feature extraction.

**Time-Consuming task** − Another challenge faced by ML models is the consumption of time especially for data acquisition, feature extraction and retrieval.

**Lack of specialist persons** − As ML technology is still in its infancy stage, availability of expert resources is a tough job.

**No clear objective for formulating business problems** − Having no clear objective and well-defined goal for business problems is another key challenge for ML because this technology is not that mature yet.

**Issue of overfitting & underfitting** − If the model is overfitting or underfitting, it cannot be represented well for the problem.

**Curse of dimensionality** − Another challenge ML model faces is too many features of data points. This can be a real hindrance.

**Difficulty in deployment** − Complexity of the ML model makes it quite difficult to be deployed in real life.

# Applications of Machines Learning :-

Machine Learning is the most rapidly growing technology and according to researchers we are in the golden year of AI and ML. It is used to solve many real-world complex problems which cannot be solved with traditional approach. Following are some real-world applications of ML −

* Emotion analysis
* Sentiment analysis
* Error detection and prevention
* Weather forecasting and prediction
* Stock market analysis and forecasting
* Speech synthesis
* Speech recognition
* Customer segmentation
* Object recognition
* Fraud detection
* Fraud prevention
* Recommendation of products to customer in online shopping

# How to Start Learning Machine Learning?

Arthur Samuel coined the term **“Machine Learning”** in 1959 and defined it as a **“Field of study that gives computers the capability to learn without being explicitly programmed”.**

And that was the beginning of Machine Learning! In modern times, Machine Learning is one of the most popular (if not the most!) career choices. According to [Indeed](http://blog.indeed.com/2019/03/14/best-jobs-2019/), Machine Learning Engineer Is The Best Job of 2019 with a 344% growth and an average base salary of **$146,085** per year.

But there is still a lot of doubt about what exactly is Machine Learning and how to start learning it? So this article deals with the Basics of Machine Learning and also the path you can follow to eventually become a full-fledged Machine Learning Engineer. Now let’s get started!!!

### How to start learning ML?

This is a rough roadmap you can follow on your way to becoming an insanely talented Machine Learning Engineer. Of course, you can always modify the steps according to your needs to reach your desired end-goal!

### Step 1 – Understand the Prerequisites

In case you are a genius, you could start ML directly but normally, there are some prerequisites that you need to know which include Linear Algebra, Multivariate Calculus, Statistics, and Python. And if you don’t know these, never fear! You don’t need a Ph.D. degree in these topics to get started but you do need a basic understanding.

#### (a) Learn Linear Algebra and Multivariate Calculus

Both Linear Algebra and Multivariate Calculus are important in Machine Learning. However, the extent to which you need them depends on your role as a data scientist. If you are more focused on application heavy machine learning, then you will not be that heavily focused on maths as there are many common libraries available. But if you want to focus on R&D in Machine Learning, then mastery of Linear Algebra and Multivariate Calculus is very important as you will have to implement many ML algorithms from scratch.

#### (b) Learn Statistics

Data plays a huge role in Machine Learning. In fact, around 80% of your time as an ML expert will be spent collecting and cleaning data. And statistics is a field that handles the collection, analysis, and presentation of data. So it is no surprise that you need to learn it!!!  
Some of the key concepts in statistics that are important are Statistical Significance, Probability Distributions, Hypothesis Testing, Regression, etc. Also, Bayesian Thinking is also a very important part of ML which deals with various concepts like Conditional Probability, Priors, and Posteriors, Maximum Likelihood, etc.

#### (c) Learn Python

Some people prefer to skip Linear Algebra, Multivariate Calculus and Statistics and learn them as they go along with trial and error. But the one thing that you absolutely cannot skip is [Python](https://www.geeksforgeeks.org/python-programming-language/)! While there are other languages you can use for Machine Learning like R, Scala, etc. Python is currently the most popular language for ML. In fact, there are many Python libraries that are specifically useful for Artificial Intelligence and Machine Learning such as [Keras](https://keras.io/" \t "_blank), [TensorFlow](https://www.tensorflow.org/), [Scikit-learn](https://scikit-learn.org/stable/" \t "_blank), etc.

So if you want to learn ML, it’s best if you learn Python! You can do that using various online resources and courses such as [**Fork Python**](https://practice.geeksforgeeks.org/courses/fork-python) available Free on GeeksforGeeks.

### Step 2 – Learn Various ML Concepts

Now that you are done with the prerequisites, you can move on to actually learning ML (Which is the fun part!!!) It’s best to start with the basics and then move on to the more complicated stuff. Some of the basic concepts in ML are:

#### (a) Terminologies of Machine Learning

* **Model –**A model is a specific representation learned from data by applying some machine learning algorithm. A model is also called a hypothesis.
* **Feature –**A feature is an individual measurable property of the data. A set of numeric features can be conveniently described by a feature vector. Feature vectors are fed as input to the model. For example, in order to predict a fruit, there may be features like color, smell, taste, etc.
* **Target (Label) –**A target variable or label is the value to be predicted by our model. For the fruit example discussed in the feature section, the label with each set of input would be the name of the fruit like apple, orange, banana, etc.
* **Training –**The idea is to give a set of inputs(features) and it’s expected outputs(labels), so after training, we will have a model (hypothesis) that will then map new data to one of the categories trained on.
* **Prediction –**Once our model is ready, it can be fed a set of inputs to which it will provide a predicted output(label).

#### (b) Types of Machine Learning

* **Supervised Learning –**This involves learning from a training dataset with labeled data using classification and regression models. This learning process continues until the required level of performance is achieved.
* **Unsupervised Learning –**This involves using unlabelled data and then finding the underlying structure in the data in order to learn more and more about the data itself using factor and cluster analysis models.
* **Semi-supervised Learning –**This involves using unlabelled data like Unsupervised Learning with a small amount of labeled data. Using labeled data vastly increases the learning accuracy and is also more cost-effective than Supervised Learning.
* **Reinforcement Learning –**This involves learning optimal actions through trial and error. So the next action is decided by learning behaviors that are based on the current state and that will maximize the reward in the future.

### Advantages of Machine learning :-

#### 1. Easily identifies trends and patterns -

Machine Learning can review large volumes of data and discover specific trends and patterns that would not be apparent to humans. For instance, for an e-commerce website like Amazon, it serves to understand the browsing behaviors and purchase histories of its users to help cater to the right products, deals, and reminders relevant to them. It uses the results to reveal relevant advertisements to them.

#### 2. No human intervention needed (automation)

With ML, you don’t need to babysit your project every step of the way. Since it means giving machines the ability to learn, it lets them make predictions and also improve the algorithms on their own. A common example of this is anti-virus softwares; they learn to filter new threats as they are recognized. ML is also good at recognizing spam.

#### 3. Continuous Improvement

As [**ML algorithms**](https://data-flair.training/blogs/machine-learning-algorithms/) gain experience, they keep improving in accuracy and efficiency. This lets them make better decisions. Say you need to make a weather forecast model. As the amount of data you have keeps growing, your algorithms learn to make more accurate predictions faster.

#### 4. Handling multi-dimensional and multi-variety data

Machine Learning algorithms are good at handling data that are multi-dimensional and multi-variety, and they can do this in dynamic or uncertain environments.

#### 5. Wide Applications

You could be an e-tailer or a healthcare provider and make ML work for you. Where it does apply, it holds the capability to help deliver a much more personal experience to customers while also targeting the right customers.

### Disadvantages of Machine Learning :-

#### 1. Data Acquisition

Machine Learning requires massive data sets to train on, and these should be inclusive/unbiased, and of good quality. There can also be times where they must wait for new data to be generated.

#### 2. Time and Resources

ML needs enough time to let the algorithms learn and develop enough to fulfill their purpose with a considerable amount of accuracy and relevancy. It also needs massive resources to function. This can mean additional requirements of computer power for you.

#### 3. Interpretation of Results

Another major challenge is the ability to accurately interpret results generated by the algorithms. You must also carefully choose the algorithms for your purpose.

#### 4. High error-susceptibility

[Machine Learning](https://en.wikipedia.org/wiki/Machine_learning) is autonomous but highly susceptible to errors. Suppose you train an algorithm with data sets small enough to not be inclusive. You end up with biased predictions coming from a biased training set. This leads to irrelevant advertisements being displayed to customers. In the case of ML, such blunders can set off a chain of errors that can go undetected for long periods of time. And when they do get noticed, it takes quite some time to recognize the source of the issue, and even longer to correct it.

**Python Development Steps : -**

Guido Van Rossum published the first version of Python code (version 0.9.0) at alt.sources in February 1991. This release included already exception handling, functions, and the core data types of list, dict, str and others. It was also object oriented and had a module system.  
Python version 1.0 was released in January 1994. The major new features included in this release were the functional programming tools lambda, map, filter and reduce, which Guido Van Rossum never liked.Six and a half years later in October 2000, Python 2.0 was introduced. This release included list comprehensions, a full garbage collector and it was supporting unicode.Python flourished for another 8 years in the versions 2.x before the next major release as Python 3.0 (also known as "Python 3000" and "Py3K") was released. Python 3 is not backwards compatible with Python 2.x. The emphasis in Python 3 had been on the removal of duplicate programming constructs and modules, thus fulfilling or coming close to fulfilling the 13th law of the Zen of Python: "There should be one -- and preferably only one -- obvious way to do it."Some changes in Python 7.3:

* Print is now a function
* Views and iterators instead of lists
* The rules for ordering comparisons have been simplified. E.g. a heterogeneous list cannot be sorted, because all the elements of a list must be comparable to each other.
* There is only one integer type left, i.e. int. long is int as well.
* The division of two integers returns a float instead of an integer. "//" can be used to have the "old" behaviour.
* Text Vs. Data Instead Of Unicode Vs. 8-bit

**Purpose :-**

We demonstrated that our approach enables successful segmentation of intra-retinal layers—even with low-quality images containing speckle noise, low contrast, and different intensity ranges throughout—with the assistance of the ANIS feature.

**Python**

Python is an interpreted high-level programming language for general-purpose programming. Created by Guido van Rossum and first released in 1991, Python has a design philosophy that emphasizes code readability, notably using significant whitespace.

Python features a dynamic type system and automatic memory management. It supports multiple programming paradigms, including object-oriented, imperative, functional and procedural, and has a large and comprehensive standard library.

* Python is Interpreted − Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.
* Python is Interactive − you can actually sit at a Python prompt and interact with the interpreter directly to write your programs.

Python also acknowledges that speed of development is important. Readable and terse code is part of this, and so is access to powerful constructs that avoid tedious repetition of code. Maintainability also ties into this may be an all but useless metric, but it does say something about how much code you have to scan, read and/or understand to troubleshoot problems or tweak behaviors. This speed of development, the ease with which a programmer of other languages can pick up basic Python skills and the huge standard library is key to another area where Python excels. All its tools have been quick to implement, saved a lot of time, and several of them have later been patched and updated by people with no Python background - without breaking.

**Modules Used in Project :-**

**Tensorflow**

TensorFlow is a [free](https://en.wikipedia.org/wiki/Free_software) and [open-source](https://en.wikipedia.org/wiki/Open-source_software) [software library for dataflow and differentiable programming](https://en.wikipedia.org/wiki/Library_(computing)) across a range of tasks. It is a symbolic math library, and is also used for [machine learning](https://en.wikipedia.org/wiki/Machine_learning) applications such as [neural networks](https://en.wikipedia.org/wiki/Neural_networks). It is used for both research and production at [Google](https://en.wikipedia.org/wiki/Google).‍

TensorFlow was developed by the [Google Brain](https://en.wikipedia.org/wiki/Google_Brain) team for internal Google use. It was released under the [Apache 2.0](https://en.wikipedia.org/wiki/Apache_License) [open-source license](https://en.wikipedia.org/wiki/Open-source_license) on November 9, 2015.

**Numpy**

Numpy is a general-purpose array-processing package. It provides a high-performance multidimensional array object, and tools for working with these arrays.

It is the fundamental package for scientific computing with Python. It contains various features including these important ones:

* A powerful N-dimensional array object
* Sophisticated (broadcasting) functions
* Tools for integrating C/C++ and Fortran code
* Useful linear algebra, Fourier transform, and random number capabilities

Besides its obvious scientific uses, Numpy can also be used as an efficient multi-dimensional container of generic data. Arbitrary data-types can be defined using Numpy which allows Numpy to seamlessly and speedily integrate with a wide variety of databases.

**Pandas**

Pandas is an open-source Python Library providing high-performance data manipulation and analysis tool using its powerful data structures. Python was majorly used for data munging and preparation. It had very little contribution towards data analysis. Pandas solved this problem. Using Pandas, we can accomplish five typical steps in the processing and analysis of data, regardless of the origin of data load, prepare, manipulate, model, and analyze. Python with Pandas is used in a wide range of fields including academic and commercial domains including finance, economics, Statistics, analytics, etc.

**Matplotlib**

Matplotlib is a Python 2D plotting library which produces publication quality figures in a variety of hardcopy formats and interactive environments across platforms. Matplotlib can be used in Python scripts, the Python and [IPython](http://ipython.org/) shells, the [Jupyter](http://jupyter.org/) Notebook, web application servers, and four graphical user interface toolkits. Matplotlib tries to make easy things easy and hard things possible. You can generate plots, histograms, power spectra, bar charts, error charts, scatter plots, etc., with just a few lines of code. For examples, see the [sample plots](https://matplotlib.org/tutorials/introductory/sample_plots.html) and [thumbnail gallery](https://matplotlib.org/gallery/index.html).

For simple plotting the pyplot module provides a MATLAB-like interface, particularly when combined with IPython. For the power user, you have full control of line styles, font properties, axes properties, etc, via an object oriented interface or via a set of functions familiar to MATLAB users.

**Scikit – learn**

Scikit-learn provides a range of supervised and unsupervised learning algorithms via a consistent interface in Python. It is licensed under a permissive simplified BSD license and is distributed under many Linux distributions, encouraging academic and commercial use. **Python**

Python is an interpreted high-level programming language for general-purpose programming. Created by Guido van Rossum and first released in 1991, Python has a design philosophy that emphasizes code readability, notably using significant whitespace.

Python features a dynamic type system and automatic memory management. It supports multiple programming paradigms, including object-oriented, imperative, functional and procedural, and has a large and comprehensive standard library.

* Python is Interpreted − Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.
* Python is Interactive − you can actually sit at a Python prompt and interact with the interpreter directly to write your programs.

Python also acknowledges that speed of development is important. Readable and terse code is part of this, and so is access to powerful constructs that avoid tedious repetition of code. Maintainability also ties into this may be an all but useless metric, but it does say something about how much code you have to scan, read and/or understand to troubleshoot problems or tweak behaviors. This speed of development, the ease with which a programmer of other languages can pick up basic Python skills and the huge standard library is key to another area where Python excels. All its tools have been quick to implement, saved a lot of time, and several of them have later been patched and updated by people with no Python background - without breaking.

**Install Python Step-by-Step in Windows and Mac :**

Python a versatile programming language doesn’t come pre-installed on your computer devices. Python was first released in the year 1991 and until today it is a very popular high-level programming language. Its style philosophy emphasizes code readability with its notable use of great whitespace.

The object-oriented approach and language construct provided by Python enables programmers to write both clear and logical code for projects. This software does not come pre-packaged with Windows.

## How to Install Python on Windows and Mac :

There have been several updates in the Python version over the years. The question is how to install Python? It might be confusing for the beginner who is willing to start learning Python but this tutorial will solve your query. The latest or the newest version of Python is version 3.7.4 or in other words, it is Python 3.

**Note:** The python version 3.7.4 cannot be used on Windows XP or earlier devices.

Before you start with the installation process of Python. First, you need to know about your **System Requirements**. Based on your system type i.e. operating system and based processor, you must download the python version. My system type is a **Windows 64-bit operating system**. So the steps below are to install python version 3.7.4 on Windows 7 device or to install Python 3. [Download the Python Cheatsheet here.](https://myelearninghub.com/python-cheat-sheet/)The steps on how to install Python on Windows 10, 8 and 7 are **divided into 4 parts** to help understand better.

### Download the Correct version into the system

**Step 1:** Go to the official site to download and install python using Google Chrome or any other web browser. OR Click on the following link: [https://www.python.org](https://www.python.org/)



Now, check for the latest and the correct version for your operating system.

**Step 2:** Click on the Download Tab.

****

**Step 3:** You can either select the Download Python for windows 3.7.4 button in Yellow Color or you can scroll further down and click on download with respective to their version. Here, we are downloading the most recent python version for windows 3.7.4

****

**Step 4:** Scroll down the page until you find the Files option.

**Step 5:** Here you see a different version of python along with the operating system.



• To download Windows 32-bit python, you can select any one from the three options: Windows x86 embeddable zip file, Windows x86 executable installer or Windows x86 web-based installer.

•To download Windows 64-bit python, you can select any one from the three options: Windows x86-64 embeddable zip file, Windows x86-64 executable installer or Windows x86-64 web-based installer.

Here we will install Windows x86-64 web-based installer. Here your first part regarding which version of python is to be downloaded is completed. Now we move ahead with the second part in installing python i.e. Installation

**Note:** To know the changes or updates that are made in the version you can click on the Release Note Option.

### Installation of Python

**Step 1:** Go to Download and Open the downloaded python version to carry out the installation process.



**Step 2:** Before you click on Install Now, Make sure to put a tick on Add Python 3.7 to PATH.



**Step 3:** Click on Install NOW After the installation is successful. Click on Close.



With these above three steps on python installation, you have successfully and correctly installed Python. Now is the time to verify the installation.

**Note:** The installation process might take a couple of minutes.

### Verify the Python Installation

**Step 1:** Click on Start

**Step 2:** In the Windows Run Command, type “cmd”.



**Step 3:** Open the Command prompt option.

**Step 4:** Let us test whether the python is correctly installed. Type **python –V** and press Enter.



**Step 5:** You will get the answer as 3.7.4

**Note:** If you have any of the earlier versions of Python already installed. You must first uninstall the earlier version and then install the new one.

### Check how the Python IDLE works

**Step 1:** Click on Start

**Step 2:** In the Windows Run command, type “python idle”.



**Step 3:** Click on IDLE (Python 3.7 64-bit) and launch the program

**Step 4:** To go ahead with working in IDLE you must first save the file. **Click on File > Click on Save**



**Step 5:** Name the file and save as type should be Python files. Click on SAVE. Here I have named the files as Hey World.

**Step 6:** Now for e.g. **enter print**

SYSTEM DESIGN

**5.1 UML DIAGRAM’S :**

UML stands for Unified Modeling Language. UML is a standardized general-purpose modeling language in the field of object-oriented software engineering. The standard is managed, and was created by, the Object Management Group.

The goal is for UML to become a common language for creating models of object oriented computer software. In its current form UML is comprised of two major components: a Meta-model and a notation. In the future, some form of method or process may also be added to; or associated with, UML.

The Unified Modeling Language is a standard language for specifying, Visualization, Constructing and documenting the artifacts of software system, as well as for business modeling and other non-software systems.

The UML represents a collection of best engineering practices that have proven successful in the modeling of large and complex systems.

The UML is a very important part of developing objects oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects.

**GOALS:**

The Primary goals in the design of the UML are as follows:

1. Provide users a ready-to-use, expressive visual modeling Language so that they can develop and exchange meaningful models.
2. Provide extendibility and specialization mechanisms to extend the core concepts.
3. Be independent of particular programming languages and development process.
4. Provide a formal basis for understanding the modeling language.
5. Encourage the growth of OO tools market.
6. Support higher level development concepts such as collaborations, frameworks, patterns and components.
7. Integrate best practices.

**5.2 USE CASE DIAGRAM:**

A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.

v

# 5.3 CLASS DIAGRAM:

In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. It explains which class contains information.



**5.4 SEQUENCE DIAGRAM:**

A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.



**5.5 COLLABRATION DIAGRAM:**

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.



**5.6 ACTIVITY DIAGRAM :**

Activity diagram is another important behavioral diagram in [**UML**](https://en.wikipedia.org/wiki/Unified_Modeling_Language) diagram to describe dynamic aspects of the system. Activity diagram is essentially an advanced version of flow chart that modeling the flow from one activity to another activity.

If yes

user

If no

Cluster prediction

Future prediction

analysis

**IMPLEMENTATION:**

**MODULES:**

Cluster prediction : use this module to predict cluster.

Future prediction : use this module to predict future.

Analysis : use this module to anylize.

Coding

**6.1 CODE**

#!/usr/bin/env python

import os

import sys

if \_\_name\_\_ == '\_\_main\_\_':

os.environ.setdefault('DJANGO\_SETTINGS\_MODULE', 'CrimeAnalysis.settings')

try:

from django.core.management import execute\_from\_command\_line

except ImportError as exc:

raise ImportError(

"Couldn't import Django. Are you sure it's installed and "

"available on your PYTHONPATH environment variable? Did you "

"forget to activate a virtual environment?"

) from exc

execute\_from\_command\_line(sys.argv)

from django.shortcuts import render

from django.template import RequestContext

from django.contrib import messages

from django.http import HttpResponse

import os

import matplotlib.pyplot as plt

import pandas as pd

import numpy as np

from sklearn.preprocessing import LabelEncoder

from sklearn.preprocessing import MinMaxScaler

from sklearn.ensemble import RandomForestRegressor

from sklearn.model\_selection import train\_test\_split

from sklearn.cluster import KMeans

from sklearn.metrics import mean\_squared\_error

from sklearn.model\_selection import train\_test\_split

from math import sqrt

global dataset, kmeans\_cluster, theft\_cls, rape\_cls, murder\_cls

sc = MinMaxScaler(feature\_range = (0, 1))

le1 = LabelEncoder()

le2 = LabelEncoder()

le3 = LabelEncoder()

le4 = LabelEncoder()

global mse, rmse

def calculateError(alg, X\_test, y\_test):

predict = alg.predict(X\_test)

#predict = predict.reshape(predict.shape[0],1)

#predict = sc.inverse\_transform(predict)

predict = predict.ravel()

#labels = sc.inverse\_transform(y\_test)

labels = y\_test.ravel()

mse\_error = mean\_squared\_error(labels,predict)

rmse\_error = sqrt(mse\_error)

mse.append(mse\_error/1000)

rmse.append(rmse\_error)

def UploadDatasetAction(request):

if request.method == 'POST':

global dataset, kmeans\_cluster, theft\_cls, rape\_cls, murder\_cls, mse, rmse

mse = []

rmse = []

myfile = request.FILES['t1']

dataset = pd.read\_csv("Dataset/Dataset.csv", usecols=['States/UTs','District', 'Murder', 'Rape', 'Theft', 'Dowry\_Deaths', 'Year'])

dataset.fillna(0, inplace = True)

cols = ['States/UTs', 'District']

dataset[cols[0]] = pd.Series(le1.fit\_transform(dataset[cols[0]].astype(str)))

dataset[cols[1]] = pd.Series(le2.fit\_transform(dataset[cols[1]].astype(str)))

X = dataset.values

X = sc.fit\_transform(X)

kmeans\_cluster = KMeans(n\_clusters=2, n\_init=1200)

kmeans\_cluster.fit(X)

dataset = pd.read\_csv("Dataset/Dataset.csv", usecols=['States/UTs','District', 'Year', 'Theft', 'Murder', 'Rape'])

dataset.fillna(0, inplace = True)

print(dataset)

cols = ['States/UTs', 'District']

dataset[cols[0]] = pd.Series(le3.fit\_transform(dataset[cols[0]].astype(str)))

dataset[cols[1]] = pd.Series(le4.fit\_transform(dataset[cols[1]].astype(str)))

theft\_Y = dataset['Theft'].values

murder\_Y = dataset['Murder'].values

rape\_Y = dataset['Rape'].values

dataset.drop(['Theft'], axis = 1,inplace=True)

dataset.drop(['Murder'], axis = 1,inplace=True)

dataset.drop(['Rape'], axis = 1,inplace=True)

X = dataset.values

X\_train1, X\_test1, y\_train1, y\_test1 = train\_test\_split(X, theft\_Y, test\_size = 0.2)

X\_train2, X\_test2, y\_train2, y\_test2 = train\_test\_split(X, rape\_Y, test\_size = 0.2)

X\_train3, X\_test3, y\_train3, y\_test3 = train\_test\_split(X, murder\_Y, test\_size = 0.2)

theft\_cls = RandomForestRegressor()

theft\_cls.fit(X, theft\_Y)

calculateError(theft\_cls, X\_test1, y\_test1)

rape\_cls = RandomForestRegressor()

rape\_cls.fit(X, rape\_Y)

calculateError(rape\_cls, X\_test2, y\_test2)

murder\_cls = RandomForestRegressor()

murder\_cls.fit(X, murder\_Y)

calculateError(murder\_cls, X\_test3, y\_test3)

dataset = pd.read\_csv("Dataset/Dataset.csv")

dataset.fillna(0, inplace = True)

columns = list(dataset.columns)

strdata = '<table border=1 align=center width=100%><tr><th><font size="" color="black">'+columns[0]+'</th>'

for i in range(1,len(columns)):

strdata+='<th><font size="" color="black">'+columns[i]+'</th>'

strdata += "</tr>"

dataset = dataset.values

for i in range(len(dataset)):

strdata += "<tr>"

for j in range(len(dataset[i])):

strdata+='<td><font size="" color="black">'+str(dataset[i,j])+'</td>'

strdata += "</tr>"

context= {'data':strdata}

return render(request, 'ViewDataset.html', context)

def AdminLogin(request):

if request.method == 'POST':

user = request.POST.get('t1', False)

password = request.POST.get('t2', False)

if user == 'admin' and password == 'admin':

context= {'data':user}

return render(request, 'AdminScreen.html', context)

else:

context= {'data':"Invalid login details"}

return render(request, 'Admin.html', context)

def index(request):

if request.method == 'GET':

return render(request, 'index.html', {})

def Admin(request):

if request.method == 'GET':

return render(request, 'Admin.html', {})

def UploadDataset(request):

if request.method == 'GET':

return render(request, 'UploadDataset.html', {})

def MSEGraph(request):

if request.method == 'GET':

strdata = '<table border=1 align=center width=100%><tr><th><font size="" color="black">Algorithm Name</th><th><font size="" color="black">RMSE</th>'

strdata+='<th><font size="" color="black">MSE</th></tr>'

strdata+='<tr><td><font size="" color="black">Random Forest Theft Prediction</td><td><font size="" color="black">'+str(rmse[0])+'</td>'

strdata+='<td><font size="" color="black">'+str(mse[0])+'</td>'

strdata+='<tr><td><font size="" color="black">Random Forest Rape Prediction</td><td><font size="" color="black">'+str(rmse[1])+'</td>'

strdata+='<td><font size="" color="black">'+str(mse[1])+'</td>'

strdata+='<tr><td><font size="" color="black">Random Forest Murder Prediction</td><td><font size="" color="black">'+str(rmse[2])+'</td>'

strdata+='<td><font size="" color="black">'+str(mse[2])+'</td></tr></table>'

df = pd.DataFrame([['Random Forest Theft Prediction','RMSE',rmse[0]],['Random Forest Theft Prediction','MSE',mse[0]],

['Random Forest Rape Prediction','RMSE',rmse[1]],['Random Forest Rape Prediction','MSE',mse[1]],

['Random Forest Murder Prediction','RMSE',rmse[2]],['Random Forest Murder Prediction','MSE',mse[2]],

],columns=['Parameters','Algorithms','Value'])

df.pivot("Parameters", "Algorithms", "Value").plot(kind='bar')

plt.show()

context= {'data':strdata}

return render(request, 'Comparison.html', context)

def ClusterPrediction(request):

if request.method == 'GET':

dataset = pd.read\_csv("Dataset/Dataset.csv", usecols=['States/UTs','District', 'Year'])

dataset.fillna(0, inplace = True)

states = np.unique(dataset['States/UTs'].values)

output = '<tr><td><font size="" color="black">States</b></td><td><select name="t1">'

for i in range(len(states)):

output += '<option value="'+states[i]+'">'+states[i]+'</option>'

output += "</select></td></tr>"

output += '<tr><td><font size="" color="black">District</b></td><td><select name="t2">'

for i in range(len(states)):

district = dataset[dataset['States/UTs'] == states[i]]['District']

district = district.values

output += '<option value="'+states[i]+'"><b>--'+states[i]+'--</b></option>'

for j in range(len(district)):

output += '<option value="'+district[j]+'">'+district[j]+'</option>'

output += "</select></td></tr>"

output += '<tr><td><font size="" color="black">Year</b></td><td><select name="t3">'

year = np.unique(dataset['Year'].values)

for i in range(len(year)):

output += '<option value="'+str(year[i])+'">'+str(year[i])+'</option>'

output += "</select></td></tr>"

context= {'states':output}

return render(request, 'ClusterPrediction.html', context)

def ClusterPredictionAction(request):

if request.method == 'POST':

state = request.POST.get('t1', False)

district = request.POST.get('t2', False)

year = request.POST.get('t3', False)

murder = request.POST.get('t4', False)

rape = request.POST.get('t5', False)

theft = request.POST.get('t6', False)

dowry = request.POST.get('t7', False)

temp = []

temp.append([state, district, murder, rape, theft, dowry, year])

test = pd.DataFrame(temp, columns=['States/UTs','District', 'Murder', 'Rape', 'Theft', 'Dowry\_Deaths', 'Year'])

test.fillna(0, inplace = True)

test['States/UTs'] = pd.Series(le1.transform(test['States/UTs'].astype(str)))

test['District'] = pd.Series(le2.transform(test['District'].astype(str)))

test = test.values

test = sc.transform(test)

predict = kmeans\_cluster.predict(test)

print(predict)

output = district+" Low Crime Rate Area"

if predict == 1:

output = district+" High Crime Rate Area"

context= {'data':output}

return render(request, 'index.html', context)

def FuturePrediction(request):

if request.method == 'GET':

dataset = pd.read\_csv("Dataset/Dataset.csv", usecols=['States/UTs','District', 'Year'])

dataset.fillna(0, inplace = True)

states = np.unique(dataset['States/UTs'].values)

output = '<tr><td><font size="" color="black">States</b></td><td><select name="t1">'

for i in range(len(states)):

output += '<option value="'+states[i]+'">'+states[i]+'</option>'

output += "</select></td></tr>"

output += '<tr><td><font size="" color="black">District</b></td><td><select name="t2">'

for i in range(len(states)):

district = dataset[dataset['States/UTs'] == states[i]]['District']

district = district.values

output += '<option value="'+states[i]+'"><b>--'+states[i]+'--</b></option>'

for j in range(len(district)):

output += '<option value="'+district[j]+'">'+district[j]+'</option>'

output += "</select></td></tr>"

output += '<tr><td><font size="" color="black">Year</b></td><td><select name="t3">'

year = np.unique(dataset['Year'].values)

for i in range(len(year)):

output += '<option value="'+str(year[i])+'">'+str(year[i])+'</option>'

output += '<option value="2022">2022</option>'

output += '<option value=2023>2023</option>'

output += "</select></td></tr>"

context= {'states':output}

return render(request, 'FuturePrediction.html', context)

def FuturePredictionAction(request):

if request.method == 'POST':

state = request.POST.get('t1', False)

district = request.POST.get('t2', False)

year = request.POST.get('t3', False)

classify\_type = request.POST.get('t4', False)

temp = []

temp.append([state, district, year])

test = pd.DataFrame(temp, columns=['States/UTs','District', 'Year'])

test.fillna(0, inplace = True)

test['States/UTs'] = pd.Series(le1.transform(test['States/UTs'].astype(str)))

test['District'] = pd.Series(le2.transform(test['District'].astype(str)))

test = test.values

predict = 0

if classify\_type == "Theft":

predict = "Future Predicted Thefts = "+str(int(theft\_cls.predict(test)[0]))

if classify\_type == "Murder":

predict = "Future Predicted Murders = "+str(int(murder\_cls.predict(test)[0]))

if classify\_type == "Rape":

predict = "Future Predicted Rapes = "+str(int(rape\_cls.predict(test)[0]))

context= {'data':predict}

return render(request, 'index.html', context)

def Analysis(request):

if request.method == 'GET':

return render(request, 'Analysis.html', {})

def AnalysisAction(request):

if request.method == 'POST':

classify\_type = request.POST.get('t1', False)

strdata = '<table border=1 align=center width=100%>'

if classify\_type == "Theft":

strdata += '<tr><td><img src="static/analysis/theft\_bar.png" height="300" width="500"/></td></tr>'

strdata += '<tr><td><img src="static/analysis/theft\_pie.png" height="300" width="500"/></td></tr>'

if classify\_type == "Murder":

strdata += '<tr><td><img src="static/analysis/murder\_bar.png" height="300" width="500"/></td></tr>'

strdata += '<tr><td><img src="static/analysis/murder\_pie.png" height="300" width="500"/></td></tr>'

if classify\_type == "Rape":

strdata += '<tr><td><img src="static/analysis/rape\_bar.png" height="300" width="500"/></td></tr>'

strdata += '<tr><td><img src="static/analysis/rape\_pie.png" height="300" width="500"/></td></tr>'

context= {'data':strdata}

return render(request, 'ViewGraphs.html', context)

"""CrimeAnalysis URL Configuration

The `urlpatterns` list routes URLs to views. For more information please see:

https://docs.djangoproject.com/en/2.1/topics/http/urls/

Examples:

Function views

1. Add an import: from my\_app import views

2. Add a URL to urlpatterns: path('', views.home, name='home')

Class-based views

1. Add an import: from other\_app.views import Home

2. Add a URL to urlpatterns: path('', Home.as\_view(), name='home')

Including another URLconf

1. Import the include() function: from django.urls import include, path

2. Add a URL to urlpatterns: path('blog/', include('blog.urls'))

"""

from django.contrib import admin

from django.urls import path, include

urlpatterns = [

path('admin/', admin.site.urls),

path('', include('CrimeAnalysisApp.urls')),

]

Testing

**7.1 SYSTEM TEST :**

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

### TYPES OF TESTS

**Unit testing :**

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

**Integration testing**

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

**7.2 Functional test**

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

Systems/Procedures : interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

**System Test**

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

**White Box Testing**

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

**Black Box Testing**

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box .you cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works.

**Unit Testing**

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

**Test strategy and approach**

Field testing will be performed manually and functional tests will be written in detail.

**Test objectives**

* All field entries must work properly.
* Pages must be activated from the identified link.
* The entry screen, messages and responses must not be delayed.

**Features to be tested**

* Verify that the entries are of the correct format
* No duplicate entries should be allowed
* All links should take the user to the correct page.

# 7.3 Integration Testing

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

**Test Results:** All the test cases mentioned above passed successfully. No defects encountered.

**Acceptance Testing**

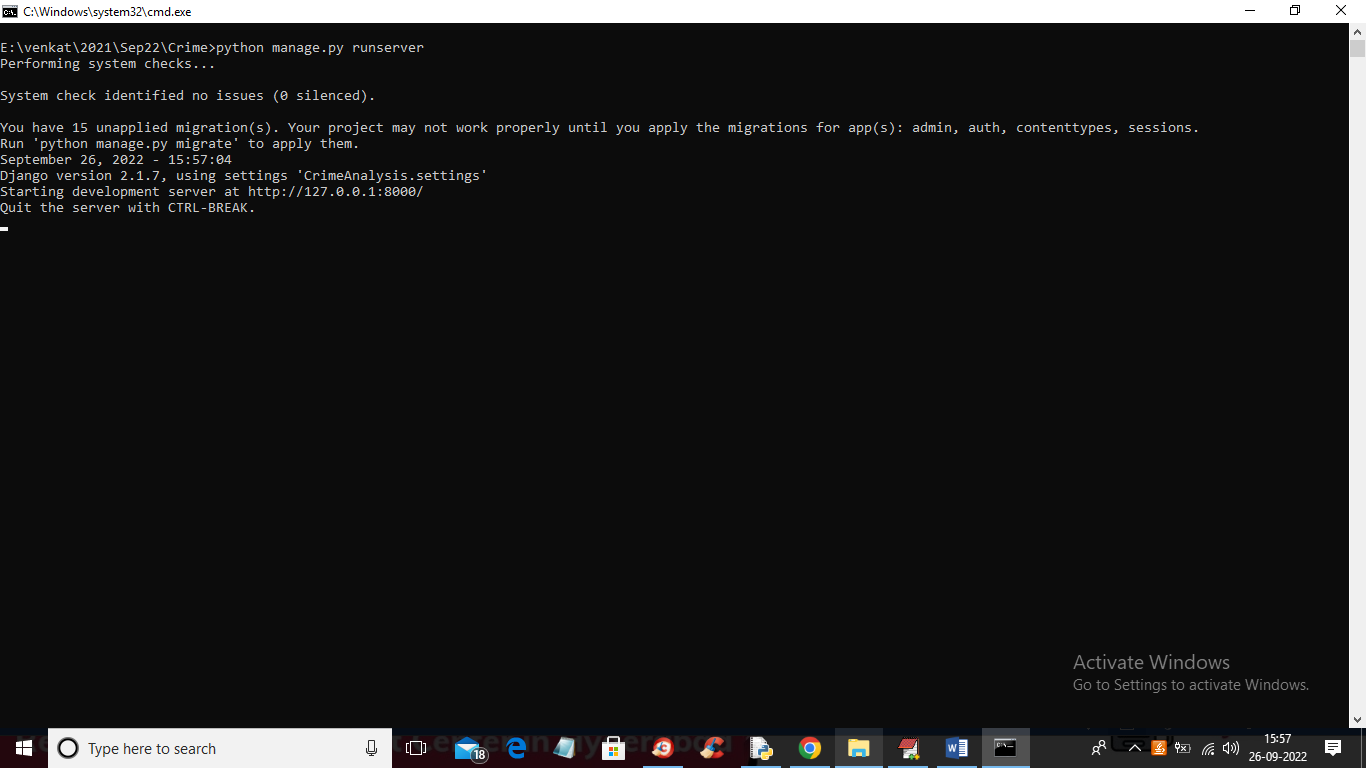
User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

**Test Results:** All the test cases mentioned above passed successfully. No defects encountered.

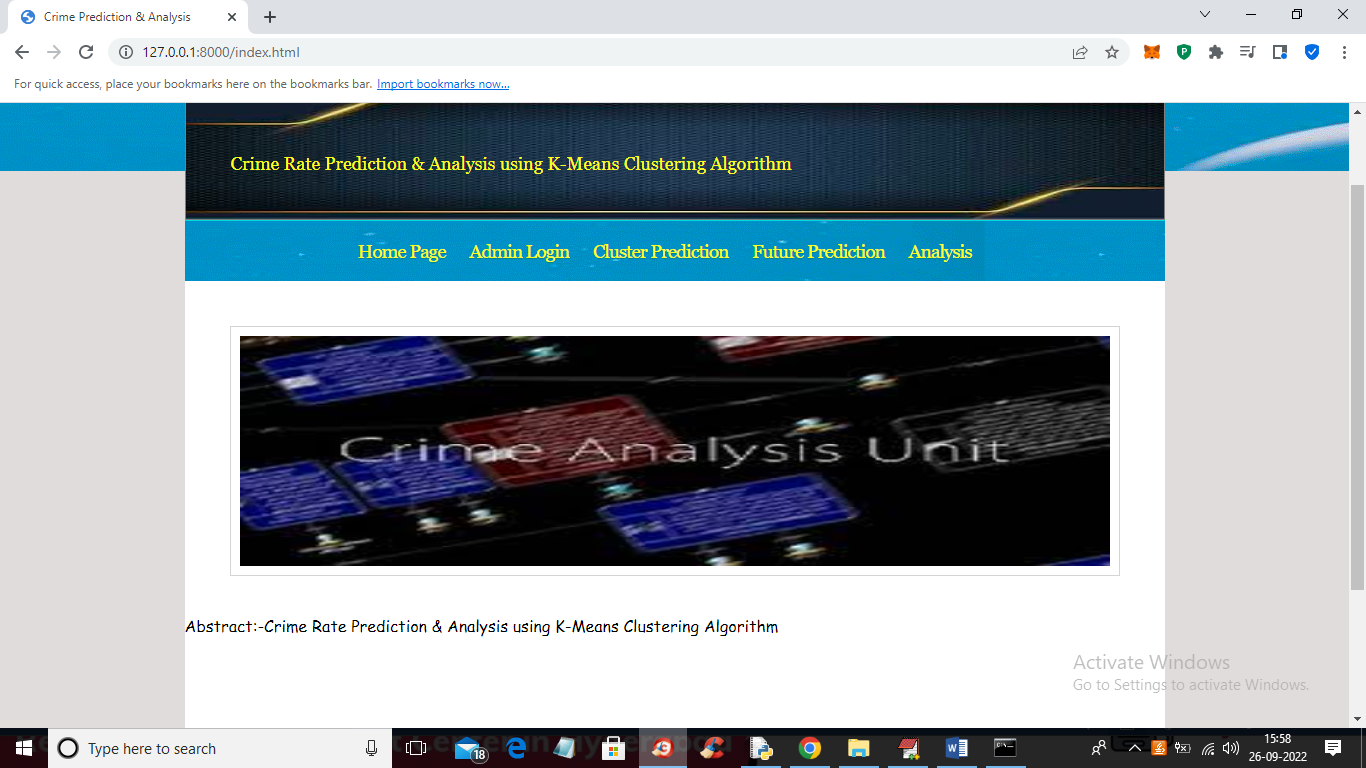
Output Screens

**8.1 Output Screens:**

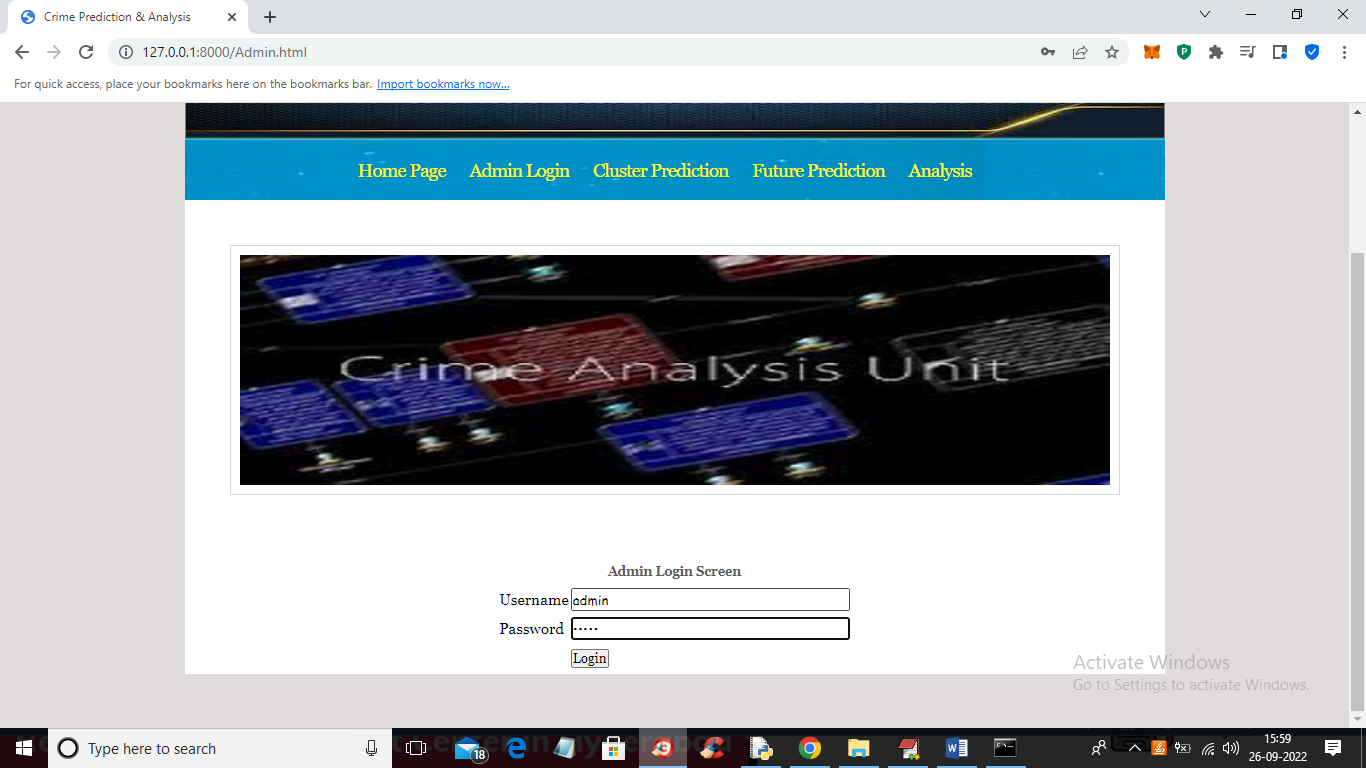
Double click on run.bat file to start python server and get below screen



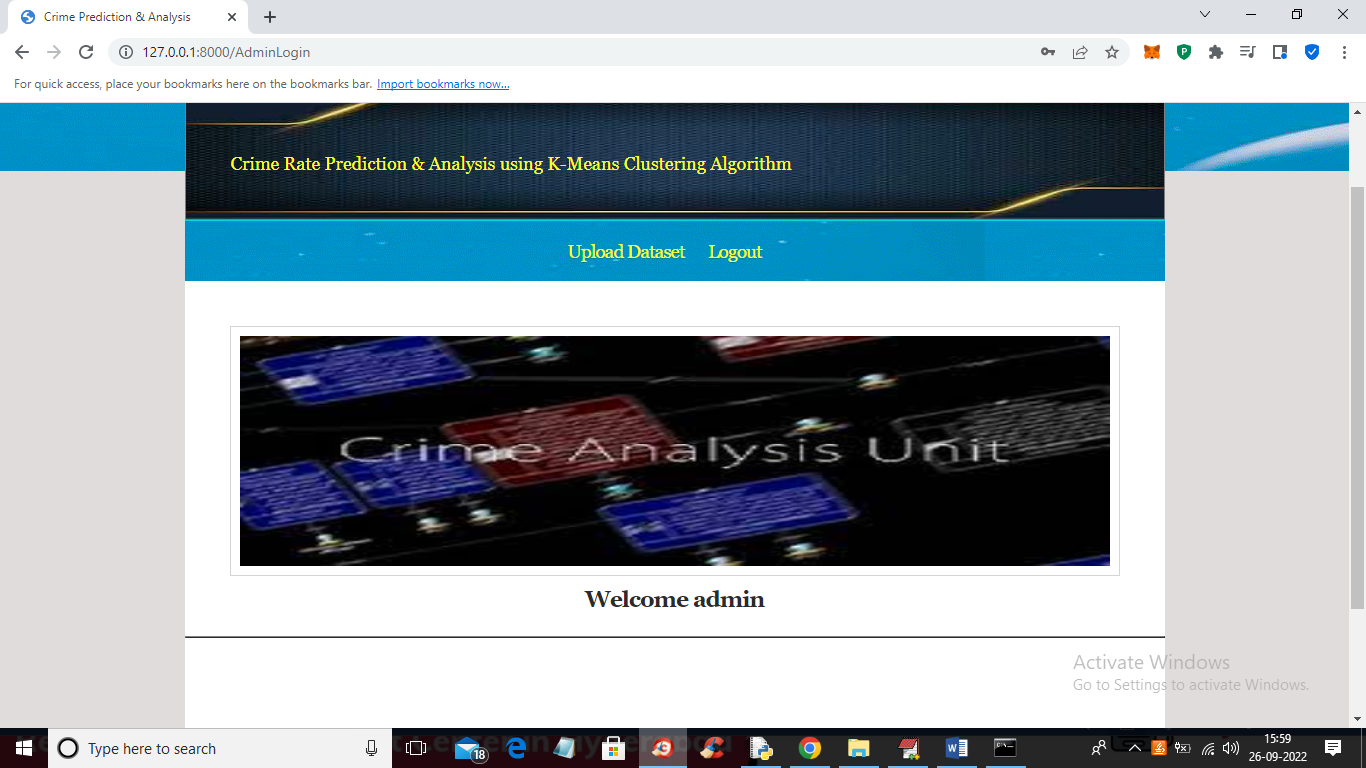
In above screen python server started and now open browser and enter URL as ‘http://127.0.0.1:8000/index.html’ and press enter key to get below page



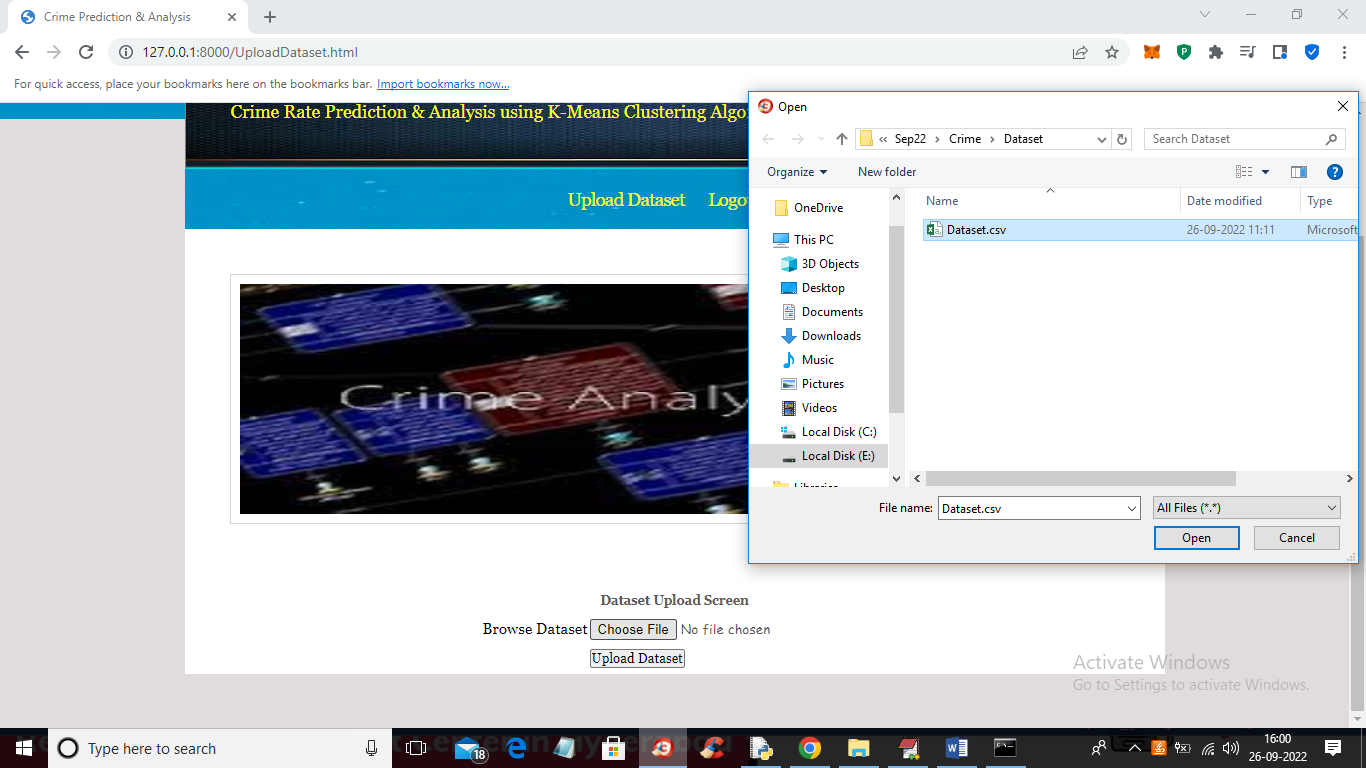
In above screen click on ‘Admin Login’ link to get below login page



In above screen admin is login and after login will get below screen



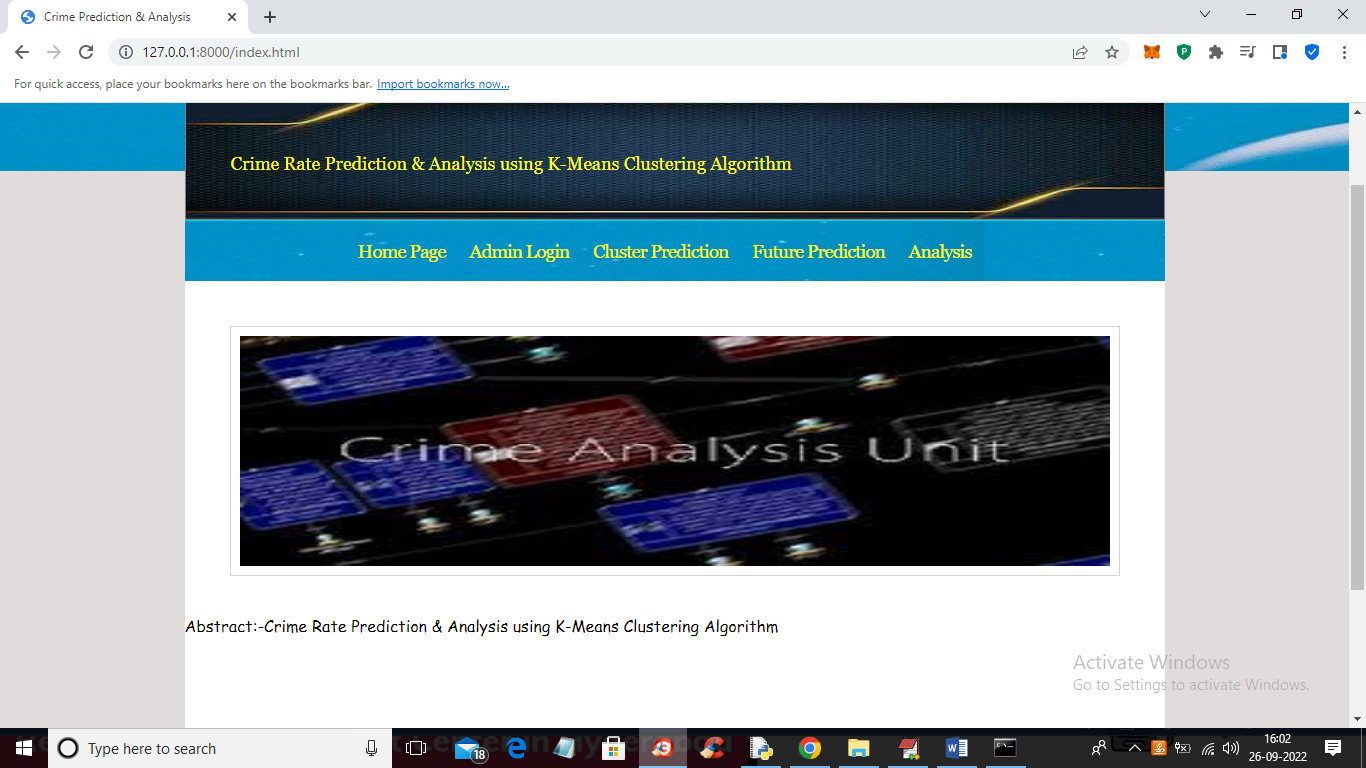
In above screen admin can click on ‘Upload Dataset’ link to upload dataset and then click submit button to load dataset and then train it with machine learning algorithms



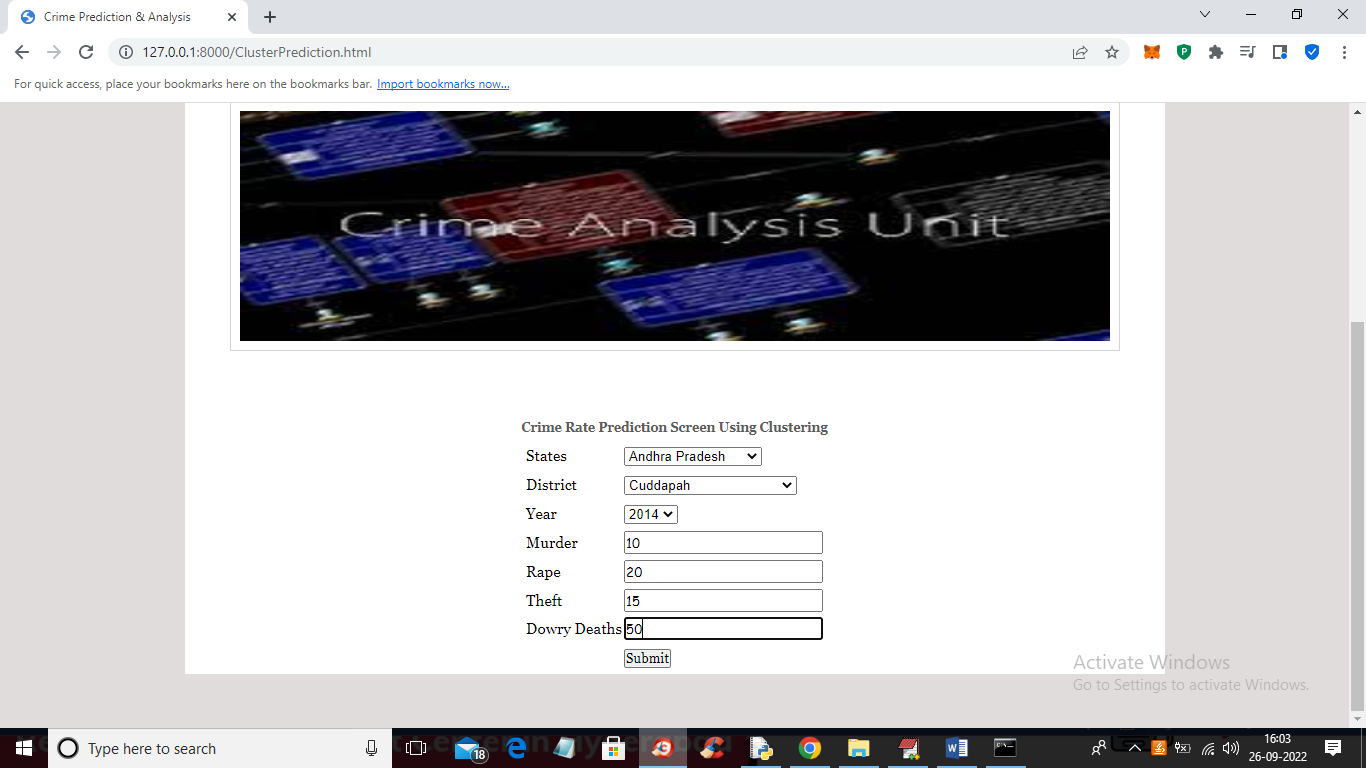
In above screen selecting and upload dataset and then click on ‘Open’ and ‘Upload Dataset’ button to load and complete training process and get below output



In above screen training is completed and then we got all dataset details and now click on ‘Logout’ link to get below screen



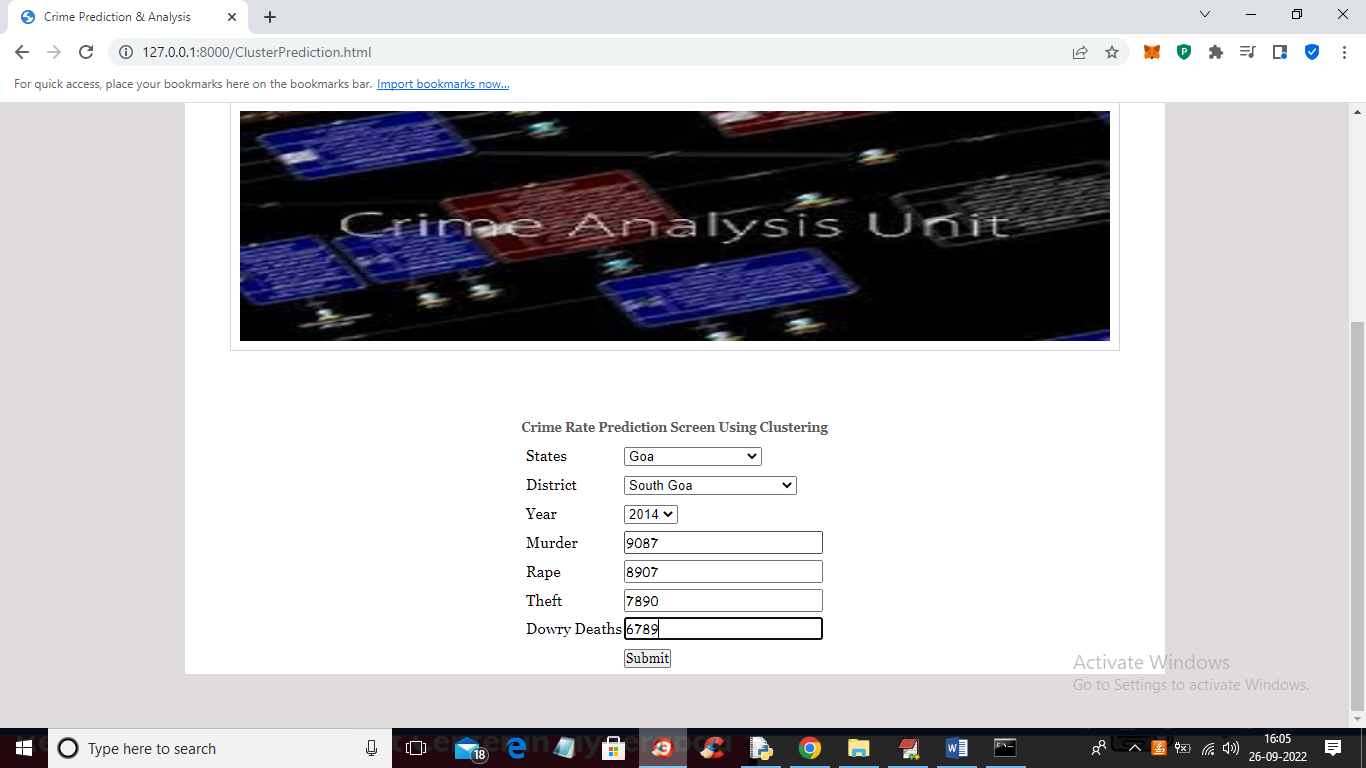
In above screen click on ‘Cluster Prediction’ link to get below screen



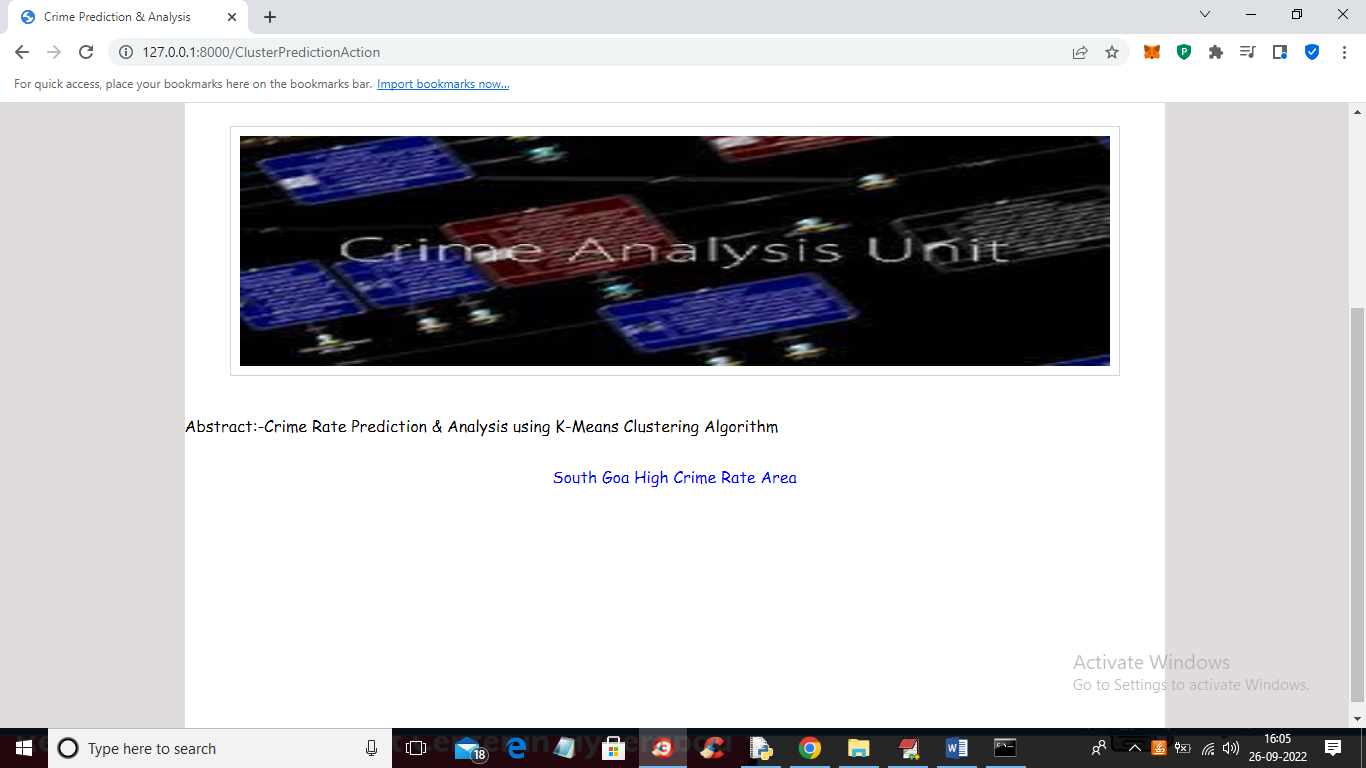
In above screen select state and district name and then enter details of crime and then press ‘Submit’ button to get below output



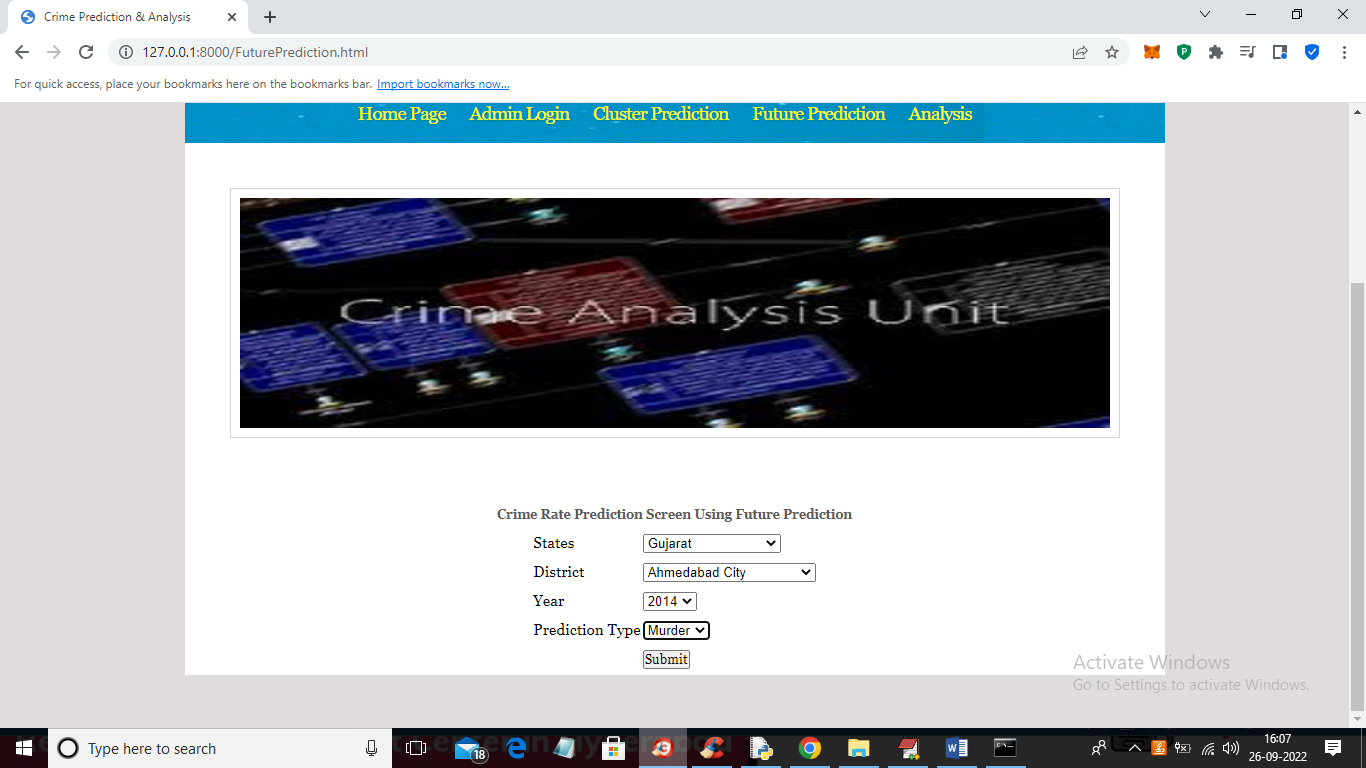
In above screen in blue colour we got output as ‘Cuddapah is the Low Crime Area’ and similarly we can test any other state



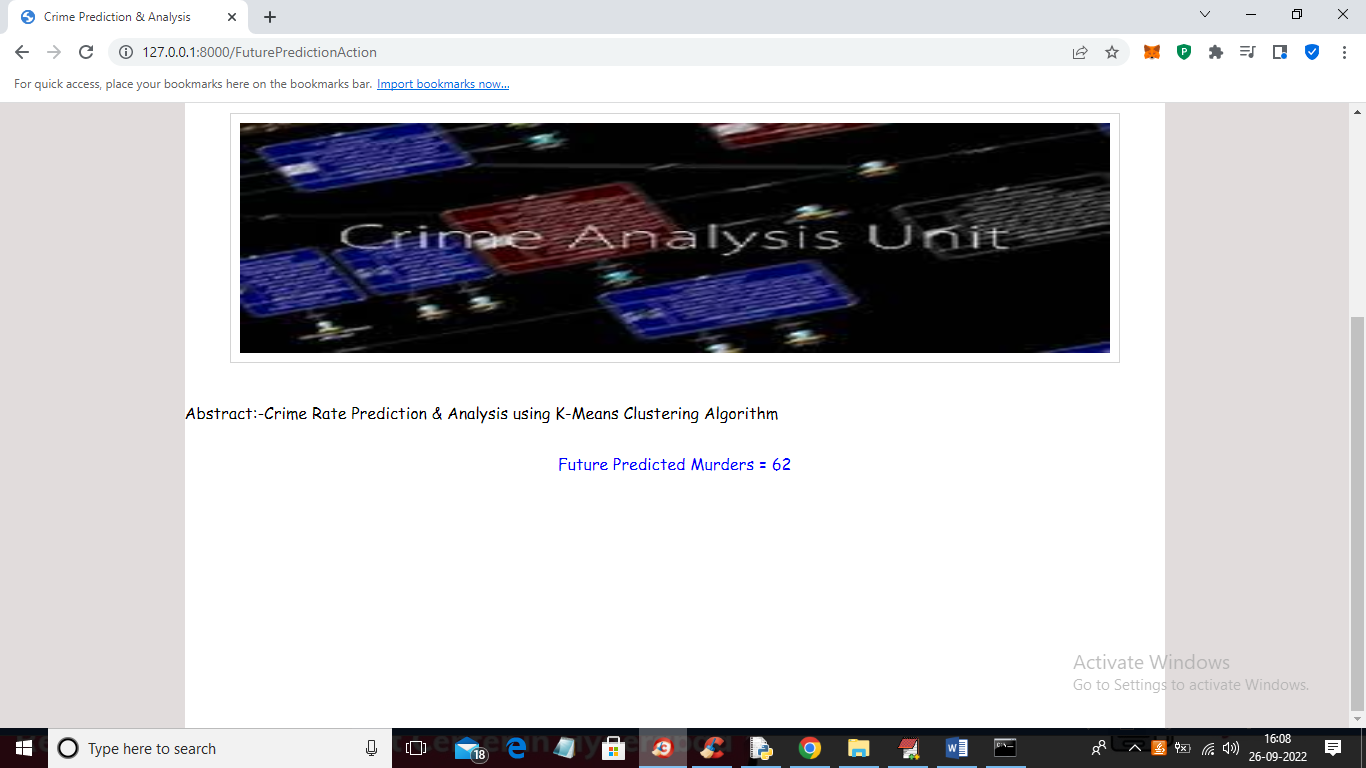
In above screen I entered some other state and crime rate and press button to get below output



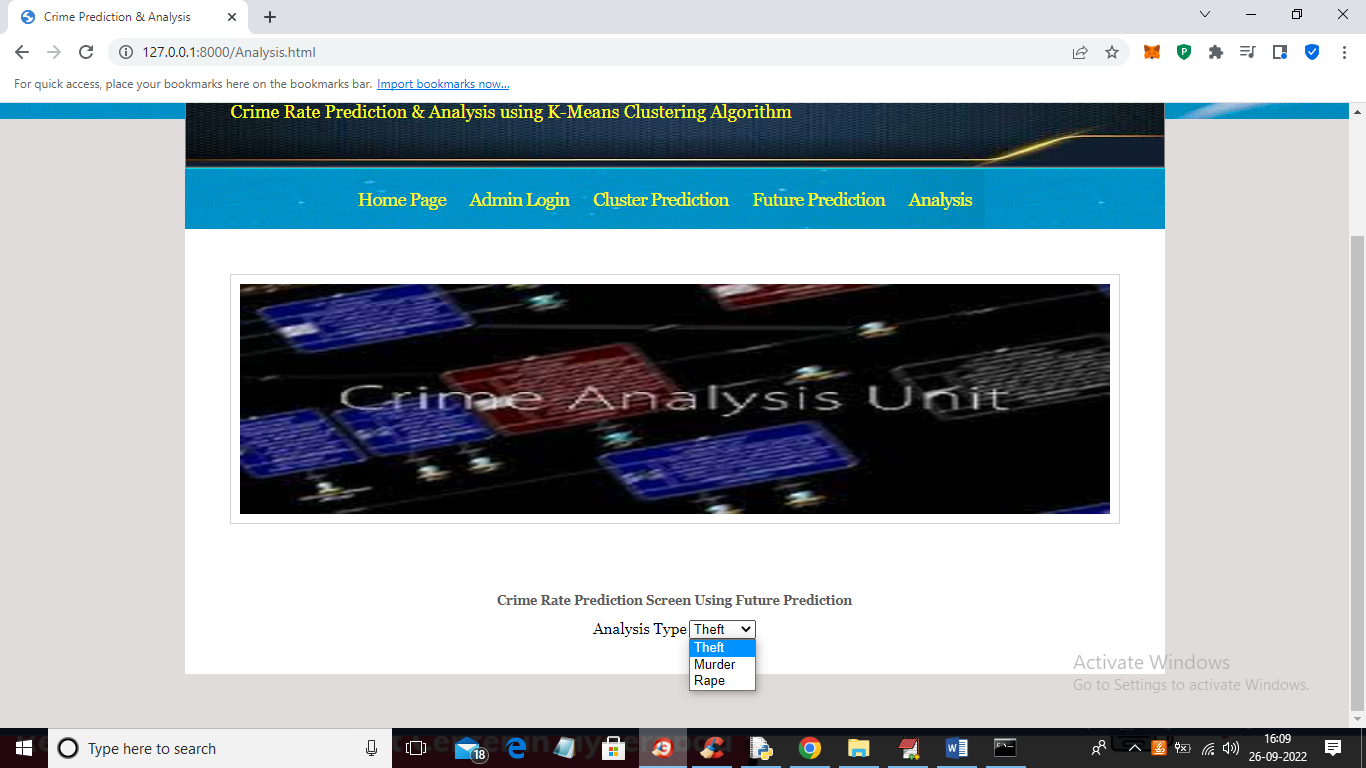
In above screen in blue colour text we got output as ‘Goa is High Crime area’ and now click on ‘Future Prediction’ link to get below screen



In above screen I selected state, district and then select crime as ‘Murder’ and then press ‘Submit’ button to get below output



In above screen future predicted Murders for Gujarat state and Ahmadabad district is 62 and similarly you can select option and get future prediction and now click on ‘Analysis’ link to get below screen



In above screen select the type of analysis and press button to get below graphs

****

**Conclusion**

**9.1 CONCLUSION :**

This project focuses on crime analysis by implementing clustering algorithm on crime dataset using rapid miner tool and here we do crime analysis by considering crime homicide and plotting it with respect to year and got into conclusion that homicide is decreasing from 1990 to 2011 .From the clustered results it is easy to identify crime trend over years and can be used to design precaution methods for future.

Ultimately, the success of the "Crime Rate Prediction and Analysis Using K-Means Clustering" project will not only depend on the technical aspects but also on the collaboration and engagement of stakeholders, including law enforcement agencies and community organizations.

As we move forward with the project, it is imperative to adhere to ethical and legal standards, maintain data privacy and security, and prioritize the well-being and safety of the communities we aim to serve. With a well-thought-out plan, dedicated personnel, and a commitment to continuous improvement, we have the potential to create a valuable tool that contributes to safer communities and more effective law enforcement efforts

FUTURE ENHANCEMENT

**10.1 FUTURE ENHANCEMENTS:**

From the encouraging results, we believe that crime data mining has a promising future for increasin the effectiveness and efficiency of criminal and intelligence analysis. Visual and intuitive criminal and intelligence investigation techniques can be developed for crime pattern. As we have applied clustering technique of data mining for crime analysis we can also perform other techniques of data mining such as classification. Also we can perform analysis on various dataset such as enterprise survey dataset, poverty dataset, aid effectiveness dataset, etc.

Future enhancements for the "Crime Rate Prediction and Analysis Using K-Means Clustering" project can significantly improve its effectiveness and relevance. Here are some potential avenues for future development:

1. Real-time Data Integration:

- Enhance the system's capabilities by integrating real-time data feeds from law enforcement agencies. This will enable more dynamic and up-to-date crime rate predictions and analysis.

2. Advanced Machine Learning Models:

- Explore more advanced machine learning models beyond K-Means clustering, such as deep learning and ensemble methods, to improve the accuracy and granularity of crime rate predictions.

3. Geospatial Analysis:

- Incorporate geospatial analysis techniques to identify crime hotspots and trends with greater precision. This can involve utilizing geographic information systems (GIS) for spatial data visualization.

4. Predictive Policing:

- Develop predictive policing algorithms that not only identify historical crime patterns but also forecast future criminal activities based on historical data, weather conditions, and other relevant factors.

5. Social and Demographic Data Integration:

- Integrate social and demographic data to better understand the underlying factors contributing to crime rates, such as income levels, education, and unemployment rates.

6. Mobile Application:

- Create a mobile application for law enforcement officers to access crime predictions and incident data while on patrol, enabling more proactive policing strategies.

7. Community Engagement Features:

- Develop features that allow community members to access crime data and safety information, fostering community engagement and collaboration with law enforcement.

8. Predictive Analytics for Resource Allocation:

- Implement predictive analytics to optimize resource allocation for law enforcement agencies, helping them deploy personnel and resources more efficiently in high-risk areas.

9. Machine Learning Explainability:

- Enhance the system's transparency by incorporating machine learning explainability techniques, allowing users to understand the rationale behind crime predictions.

10. Multimodal Data Integration:

- Incorporate data from various sources, including surveillance cameras, social media, and IoT devices, to provide a more comprehensive view of crime-related activities.

11. Natural Language Processing (NLP):

- Use NLP techniques to analyze unstructured text data, such as police reports and news articles, to extract valuable insights and context related to crimes.

12. Feedback Mechanism:

- Establish a feedback mechanism that allows law enforcement and community members to provide input on the system's accuracy and usefulness, facilitating continuous improvement.

13. Machine Learning Model Updates:

- Implement a system for regular updates of machine learning models to adapt to changing crime patterns and improve predictive accuracy.

14. Cross-agency Collaboration:

- Encourage collaboration among different law enforcement agencies and jurisdictions to create a more comprehensive and accurate crime prediction system.

15. Ethical and Bias Mitigation:

- Continuously monitor and address potential biases in the data and algorithms to ensure fair and ethical use of the technology.

16.\*User Training and Education:

- Provide training and educational resources to law enforcement and community users to maximize the effectiveness of the system and promote responsible use.

These future enhancements will transform the "Crime Rate Prediction and Analysis Using K-Means Clustering" project into a more powerful and comprehensive tool for crime prevention, law enforcement, and community safety, ultimately contributing to safer and more secure communities.

**REFERENCES**

**11.1 REFERENCES :**

1] De Bruin ,J.S.,Cocx,T.K,Kosters,W.A.,Laros,J. and Kok,J.N(2006) Data mining approaches to criminal carrer analysis ,”in Proceedings of the Sixth International Conference on Data Mining (ICDM”06) ,Pp. 171-177

[2] Manish Gupta1\*, B.Chandra1 and M. P. Gupta1,2007 Crime Data Mining for Indian Police Information System

[3] Nazlena Mohamad Ali1, Masnizah Mohd2, Hyowon Lee3, Alan F. Smeaton3, Fabio Crestani4 and Shahrul Azman Mohd Noah2 ,2010 Visual Interactive Malaysia Crime News Retrieval System

[4] Sutapat Thirprungsri Rutgers University .USA ,2011 Cluster Analysis of Anomaly Detection in Accounting Data : An Audit Approach 1

[5] A.Malathi ,Dr.S.Santhosh Baboo. D.G. Vaishnav College,Chennai ,2011 Algorithmic Crime Prediction Model Based on the Analysis of Crime Clusters.

[6] Malathi.A 1 ,Dr.S.Santhosh Baboo 2 and Anbarasi . A 31 Assistant professor ,Department of Computer Science ,Govt Arts College ,Coimbatore , India . 2 Readers , Department of Computer science , D.G. Vaishnav Collge ,Chennai , India , 2011 An intelligent Analysis of a city Crime Data Using Data Mining

[7] Malathi , A; Santhosh Baboo , S, 2011 An Enhanced Algorithm to Predict a Future Crime using Data Mining

[8] Kadhim B.Swadi al-Janabi . Department of Computer Science . Faculty of Mathematics and Computer Science .University of Kufa/Iraq , 2011 A Proposed Framework for Analyzing Crime DataSet using Decision Tree and Simple K-means Mining Algorithms.

[9] Aravindan Mahendiran, Michael Shuffett, Sathappan Muthiah, Rimy Malla, Gaoqiang Zhang,2011 Forecasting Crime Incidents using Cluster Analysis and Bayesian Belief Networks

[10] Sutapat Thiprungsri,2012 Cluster Analysis for Anomaly Detection in Accounting Data : An Audit Approach1

[11] K. Zakir Hussain, M. Durairaj and G. Rabia Jahani Farzana ,2012 Application of Data Mining Techniques for Analyzing Violent Criminal Behavior by Simulation Model

[12] <https://www.gov.uk/government/publications/offences-recorded-by-the-police-in-england-and-wales-by-offence-and-police-force-area-1990-to-2011-12>