A PROJECT REPORT ON

EDGE ASSISTED CRIME PREDICTION AND EVALUATION FRAMEWORK FOR MACHINE LEARNING ALGORITHM

SUBMITTED BY

MAROJU ANAND SAI	(217W1A0543)
MINDALA VINOD	(217W1A0545)
CHIGULLA SATHISH	(217W1A0532)
BEVARA MANI KANTA	(21W15A0528)
PANUGANTI GOPI KRISHNA	(217W1A0548)

Bachelor of Technology in
COMPUTER SCIENCE AND ENGINEERING
Under The Guidance Of
Mr. K N B CHARI
M.TECH
ASSISTANT PROFESSOR

Department Of Computer Science and Engineering

MALINENI PERUMALLU EDUCATIONAL SOCIETY'S
GROUP OF INSTITUTIONS
(Approved by AICTE & Affiliated to JNTU KAKINADA)
Pulladigunta (V), Guntur – 522017

2021 - 2025

MALINENI PERUMALLU EDUCATIONAL SOCIETY'S
GROUP OF INSTITUTIONS

(Approved by AICTE & Affiliated to JNTU KAKINADA) Pulladigunta (V)

Guntur – 522017

2021 - 2025

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the project entitled "**Edge Assisted Crime Prediction And Evaluation FrameWork For Machine Learning Algorithm**" *bonafide work carried out by*

MAROJU ANAND SAI	(217W1A0543)
MINDALA VINOD	(217W1A0545)
CHIGULLA SATHISH	(217W1A0532)
BEVARA MANI KANTA	(217W1A0528)
PANUGANTI GOPI KRISHNA	(217W1A0548)

*B.Tech students of **MPES**, Affiliated to **JNTUK**, Kakinada in partial fulfilment of the requirements for the award of the Degree of **Bachelor Of Technology** with the specialization in **Computer Science And Engineering** during the Academic year **2024-2025.***

Sig. of The Guide **Sign. Of HOD**

Viva-voice held on _____

Internal Examiner **External Examiner**

MALINENI PERUMALLU EDUCATIONAL SOCIETY'S GROUP OF INSTITUTIONS

(Approved by AICTE & Affiliated to JNTU KAKINADA) Pulladigunta(V), Guntur – 522017

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

DECLARATION

We here by declare that the project work entitled "**EDGE ASSISTED CRIME PREDICTION AND EVALUTION FRAMEWORK FOR MACHINE LEARNING ALGORITHMS**" is entirely my original work carried out under the guidance of **Mr. K N B CHARI,** Malineni Perumallu Educational Society's Group Of Institutions, Pulladigunta, Guntur, **JNTU Kakinada**, A.P, India for the award of the degree of **BACHELOR OF TECHNOLOGY** with the Specialization in **COMPUTER SCIENCE AND ENGINEERING**. The results carried out in this project report have not been submitted in a part or full for the award of any degree or diploma of this or any other university or institute.

MAROJU ANAND SAI (217W1A0543)
MINDALA VINOD (217W1A0545)
CHIGULLA SATHISH (217W1A0532)
BEVARA MANI KANTA (217W1A0528)
PANUGANTI GOPI KRISHNA (217W1A0548)

ACKNOWLEDGEMENT

We express our sincere thanks to our beloved **Chairman Sir, Dr. MALINENI. PERUMALLU** for providing support and stimulating environment for developing the project.

It plunges us in exhilaration to take the privilege of expressing our heartfelt gratitude to **Dr. D. RAVI KIRAN Professor and HOD of CSE** for providing us with every facility and for constant supervision.

We express a deep reverence and profound gratitude to for providing great support in carrying out the project our guide **Mr. K N B CHARI ,M.TECH ,ASSISTANT PROFESSOR** for his constant encouragement, suggestions, supervision, and abundant support throughout the project.

Thanks to all the teaching and non-teaching staff and lab technicians for their support and also to our teammates for their valuable Cooperation.

MAROJU ANAND SAI (217W1A0543)
MINDALA VINOD (217W1A0545)
CHIGULLA SATHISH (217W1A0532)
BEVARA MANI KANTA (217W1A0528)
PANUGANTI GOPI KRISHNA (217W1A0548)

MALINENI PERUMALLU EDUCATIONAL SOCIETY'S GROUP OF INSTITUTIONS

(Approved by AICTE & Affiliated to JNTU KAKINADA) **Pulladigunta(V), Guntur – 522017**

INSTITUTE VISION/MISSION

VISION

To Emerge as Premier Institute of Quality Education, Employability, Social and Ethical Values

MISSION

IM1: Provide student centric Teaching Learning Methodologies with Modern Infrastructure.

IM2: Promote Technical courses to equip the stakeholders with career skills

IM3: Cultivate idealistic minds

IM4: Provide academic facilities catering industry and society needs

MALINENI PERUMALLU EDUCATIONAL SOCIETY'S GROUP OF INSTITUTIONS

(Approved by AICTE & Affiliated to JNTU KAKINADA) **Pulladigunta(V), Guntur – 522017**

DEPARTMENTVISION/MISSION/PEOs/POs/PSOs

DEPARTMENT VISION

To become centre of excellence in Computer Science and Engineering with values and life-long learning.

DEPARTMENT MISSION

DM1: Providing a strong theoretical knowledge and practice in computer science and engineering with an emphasis on software development.

DM2: Imparting the skills necessary to continue education to grow professionally.

DM3: Empowering the youth in rural community with computer education.

DM4: Inculcating professional behaviour, ethical values, innovative research capabilities and leadership abilities.

PROGRAM EDUCATIONAL OBJECTIVES

PEO1:Graduates will be able to analyze, design and develop computer applications to provide solutions to real world problems.

PEO2:Graduates are well trained, confident, research oriented and industry ready professionals who are intellectual, ethical and socially committed.

PEO3:Graduates will have the technical, communication skills and character that will prepare them for technical and leadership roles.

PROGRAM OUTCOMES

PO1. Engineering knowledge: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

PO2. Problem analysis: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

PO3. Design/development of solutions: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

PO4. Conduct investigations of complex problems: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

PO5. Modern tool usage: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modelling to complex engineering activities with an understanding of the limitations.

PO6. The engineer and society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

PO7. Environment and sustainability: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

PO8. Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

PO9. Individual and team work: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

PO10. Communication: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

PO11. Project management and finance: Demonstrate knowledge and understanding of the engineering and management principles and apply these to ones own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

PO12. Life-long learning: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

PROGRAM SPECIFIC OUTCOMES:

- PSO 1: Analyse and develop solutions for real world computational problems by applying mathematical logic, appropriate data structures and algorithms..
- PSO 2: Design, develop and implement application software using Python and Web Technologies.

ABSTRACT

The growing global populations, particularly in major cities, have created new problems, notably in terms of public safety regulation and optimization. As a result, in this paper, a strategy is provided for predicting crime occurrences in a city based on historical events and demographic observation. In particular, this study proposes a crime prediction and evaluation framework for machine learning algorithms of the network edge. Thus, a complete analysis of four distinct sorts of crimes, such as murder, rapid trial, repression of women and children, and narcotics, validates the efficiency of the proposed framework. The complete study and implementation process have shown a visual representation of crime in various areas of country. The total work is completed by the selection, assessment, and implementation of the Machine Learning (ML) model, and finally, proposed the crime prediction. Criminal risk is predicted using classification models for a particular time interval and place. To anticipate occurrences, ML methods such as Decision Trees, Neural Networks, K-Nearest Neighbours, and Impact Learning are being utilized, and their performance is compared based on the data processing and modification used. A maximum accuracy of 81% is obtained for Decision Tree algorithm during the prediction of crime. The findings demonstrate that employing Machine Learning techniques aids in the prediction of criminal events, which has aided in the enhancement of public security. Index Terms—Machine Learning, Edge Computing, Crime Prediction, Impact Learning, Decision Tree, KNN, MLP.

INDEX

S.NO	CONTENT	PAGE NO
1	**INTRODUCTION**	11
2	**LITERATURE SURVEY**	12-14
3	**SYSTEM STUDY**	15-16
	3.1 Economic Feasibility	
	3.2 Technical Feasibility	
	3.3 Social Feasibility	
4	**SYSTEMANALYSIS**	17-21
	4.1 Existing System	
	4.2 Proposed System	
5	**SYSYTEM SPECIFICATION**	22
	5.1 Software Requirement	
	5.2 Hardware Requirement	
6	**SYSTEM DESIGN**	23-29
	6.1 System Architecture	
	6.2 Data Flow Diagrams	
	6.3 UML Diagrams	
	6.4 Goals	
	6.5 Use case Diagrams	
	6.6 Class Diagrams	
	6.7 Sequence Diagrams	
	6.8 Activity Diagram	
7	**SOFTWARE ENVIRONMENT**	**30-56**
	7.1 Python	
	7.2 Django	
8	**SOFTWAREINSTALLATION**	57-67
	8.1 Python	
	8.2 Jupter Notebook	
	8.3 VS Code	
9	**CODING**	68-78
	9.1 Sample code on Python	
10	**SYSTEMTESTING**	79-81
	10.1 Types of Tests	
	10.1.1 Unit Testing	

	10.1.2 Integration Testing	
	10.1.3 Functional Testing	
	10.1.4 \System Testing	
	10.1.5 WhiteBox Testing	
	10.1.6 BlackBox Testing	
	10.1.7 Unit testing	
	10.2 Test Strategy and Approach	
	10.2.1 Integration Testing	
	10.2.2 Acceptance Testing	
11	**TESTCASES**	82-83
12	**IMPLEMENTATION**	84-85
	12.1 Implementation Process	
	12.2 Modules	
	12.3 Machine Learning	
13	**SCREENSHOTS**	86-97
14	**CONCLUSIONANDFUTURE ENHANCEMENT**	98-99
15	**REFERENCES**	100

CHAPTER – 1

INTRODUCTION

1.1 Introduction to the Project:

One of the primary concerns of the global population is public safety. Several causes, such as the rapid pace of urbanization, have led to the rise in worry. The migration of people to cities has been well-known in recent years, and according to UN predictions, over 70% of the world's population will live in cities by 2050 [1]. In addition, according to the Global Terrorism Database, which defines terrorist attacks as "acts of violence by non-state actors perpetrated against civilian populations, intended to cause fear, in order to achieve a political objective," the number of terrorist attacks in the last decade was the highest ever recorded. Machine learning (ML) techniques are critical for smart city applications and may be used to reduce crime since they aid with difficulties concerning This work was supported by Institute of Information Communications Technology Planning Evaluation (IITP) grant funded by the Korea government(MSIT) (No.2019-0-01287, Evolvable Deep Learning Model Generation Platform for Edge Computing) and the MSIT(Ministry of Science and ICT), Korea, under the Grand Information Technology Research Center support program(IITP-2020-2015-0-00742) supervised by the IITP. *Dr. CS Hong is the corresponding author. urban development and the extraction of value from the data obtained [2]. This paper provides a graphical representation of crime in several areas of country, such as Bangladesh.. We used data from 2012 to 2019 to show the seniors. Using the crime prediction model, we observed that region 1 is more corrupted than other cities whereas region 2 is less corrupted. Using all of this information, we created a model that predicts crime in 2021 (check this). We present a crime comparison between 2019 and 2021. We have incidents to inform residents and municipal officials about the most dangerous places, therefore providing value to the community and increasing public safety. In this regard, this sort of prediction can be beneficial in a variety of ways, including more efficient and effective patrol route planning, as well as for tourists who are ignorant of the city's most hazardous locations.

CHAPTER -2

LITERATURE REVIEW

1) Smart city with Chinese characteristics against the background of big data: Idea, action and risk

Authors: Yuzhe Wu, Weiwen Zhang.

Chinese urbanization has generated great impacts on the world since the reform and opening up. However, urban problems, e.g., environmental pollution, resources shortage, and traffic jam, have been more and more serious for urban management and development. Smart city has been put forward as an effective approach to achieve better urban management recently. Smart city aims to realize the integration of municipal service, business, transportation, water, energy source and other urban sub-systems through close combination of human wisdom and information communication techniques (ICTs). As a result, the link and synergy of information could be ultimately established with ICTs, e.g., internet, internet of things, cloud computing. Yet, few studies have been conducted to systematically link smart city with big data in China. This paper aims to put forward a development framework of smart city with Chinese characteristics against the background of big data. Key actions, including rational planning of city infrastructures, the establishment and improvement of long-acting mechanism, the effective performance of city managerial function, are proposed to realize the development idea. Meanwhile, this paper also investigates the risks embedded in development of smart city with Chinese characteristics, e.g., information safety, weak emergency responding capacity and poor independent research and development capacity of core technology. This study can facilitate Chinese local governments to systematically plan smart city before clinging the hot concept.

2.) Survey of improving Knearest-neighbour for classification

AUTHORS: Zhihua Cai, Dianhong Wang, and Siwei Jiang

KNN (k-nearest-neighbour) has been widely used as an effective classification model. In this paper, we summarize three main shortcomings confronting KNN and single out three main methods for overcoming its three shortcomings. Keeping to these methods, we try our best to survey some improved algorithms and experimentally tested their effectiveness. Besides, we discuss some directions for future study on KNN.

3.) Using Machine Learning Algorithms to Analyze Crime Data

AUTHORS : Lawrence McClendon,Natarajan Meghanathan

Data mining and machine learning have become a vital part of crime detection and prevention. In this research, we use WEKA, an open source data mining software, to conduct a comparative study between the violent crime patterns from the Communities and Crime Unnormalized Dataset provided by the University of California-Irvine repository and actual crime statistical data for the state of Mississippi that has been provided by neighborhoodscout.com. We implemented the Linear Regression, Additive Regression, and Decision Stump algorithms using the same finite set of features, on the Communities and Crime Dataset. Overall, the linear regression algorithm performed the best among the three selected algorithms. The scope of this project is to prove how effective and accurate the machine learning algorithms used in data mining analysis can be at predicting violent crime patter

4.) Crime Prediction Using Decision Tree (J48) Classification Algorithm

AUTHORS: Emmanuel Ahishakiye, Danison Taremwa

, Elisha Opiyo

With the growing processing power of computing systems and the increasing availability of massive datasets, machine learning algorithms have led to major breakthroughs in many different areas. This development has influenced computer security, spawning a series of work on learning-based security systems, such as for malware detection, vulnerability discovery, and binary code analysis. Despite great potential, machine learning in security is prone to subtle pitfalls that undermine its performance and render learning-based systems potentially unsuitable for security tasks and practical deployment. In this paper, we look at this problem with critical eyes. First, we identify common pitfalls in the design, implementation, and evaluation of learning-based security systems. We conduct a study of 30 papers from top-tier security conferences within the past 10 years, confirming that these pitfalls are widespread in the current security literature. In an empirical analysis, we further demonstrate how individual pitfalls can lead to unrealistic performance and interpretations, obstructing the understanding of the security problem at hand. As a remedy, we propose actionable recommendations to support researchers in avoiding or mitigating the pitfalls where possible. Furthermore, we identify open problems when applying machine learning in security and provide directions for further research.

CHAPTER-3

SYSTEM STUDY

3. 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 FEASIBILITY

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.

CHAPTER-4

SYSTEM ANALYSIS

EXISTING SYSTEM:

Tssshe key existing systems were that they compared against:

- MFCC + Softmax Regression: Extract MFCC features, feed into softmax regression model for genre classification.
- CQT + Softmax Regression: Use Constant Q Transform instead of STFT to get spectrogram features, feed into softmax regression.
- FFT + Softmax Regression: Take FFT directly on audio, feed amplitude spectrum into softmax regression.
- MFCC + MLP: Use MFCC as input, feed into a multilayer perceptron (MLP) model with softmax output for classification.
- CQT + MLP: Use CQT spectrogram as input, feed into MLP model.
- FFT + MLP: Use FFT amplitude spectrum as input, feed into MLP.

So in summary, the key existing systems used:

- Different input audio representations: MFCC, CQT, FFT
- Simple linear models like softmax regression
- Non-linear MLP models

But they did not use convolutional neural networks or other deep learning approaches. The input features were hand-engineered rather than learned.

Let me know if you need any clarification on these existing systems! I tried to infer the details from the limited information provided in the paper.

DISADVANTAGES OF EXISTING SYSTEM:

Based on the typical audio feature extraction and classification approaches used in the existing systems described in the paper, some potential disadvantages or limitations could be:

- Hand-crafted audio features like MFCC may not capture all the relevant information for genre classification. They are engineered based on human assumptions rather than learned from data.
- Features like MFCC are extracted from short frames independently, without considering temporal context. This ignores useful temporal patterns in the audio.

- Simple linear models like softmax regression have limited modeling capacity to capture complex patterns in audio features.
- Non-linear MLPs are able to model complex patterns, but their performance still relies on the quality of input features.
- Most systems use a pipeline approach - feature engineering, feature selection, then classifier training. This is not end-to-end learning.
- Lack of shift/translation invariance - small variations in pitch or tempo can degrade accuracy of systems relying on fixed audio features.
- Unable to effectively learn from raw audio - most systems rely on engineered features rather than learning directly from spectrograms/waveforms.

Inability to scale up - unlike deep learning approaches, traditional methods can't benefit from larger datasets.

In summary, key limitations are reliance on engineered features rather than end-to-end feature learning, lack of modelling temporal context, limited invariance properties, and disjoint training of feature extraction and classifier components. Deep learning approaches can help overcome some of these disadvantages.

Algorithm:

Here are some of the key existing algorithms and techniques that were used prior to this work:

- Using hand-crafted audio features like MFCCs, chroma features, spectral contrast, etc and feeding them into machine learning classifiers like SVM, KNN, Random Forests etc.
- Using aggregation and statistics of low-level features, e.g. mean, variance, histograms etc.
- Applying dimensionality reduction on hand-crafted features like PCA, ICA etc before classification.
- Using mid-level representations like bag-of-words on audio features.
- Combining multiple features at feature-level or decision-level via techniques like feature concatenation, early fusion, late fusion etc.
- Using deep neural networks like Deep Belief Networks (DBNs) and stacked autoencoders for unsupervised pre-training before classification.
- Applying recurrent neural networks like LSTMs on top of pre-extracted features for sequence modeling.

- Using 1D convolutional neural networks on raw waveform or spectrogram for feature learning.

So in summary, the key existing techniques relied heavily on hand-crafted audio features or 1D convolution, rather than 2D convolutional feature learning directly from spectrograms as proposed in this paper. The deep learning approaches focused more on unsupervised pre-training rather than end-to-end feature learning.

PROPOSED SYSTEM:

Here is a summary of the key points about the music genre classification paper:

- Motivation: Develop better feature representations directly from audio rather than using hand-crafted features like MFCCs for music genre classification.
- Approach: Use 2D convolutional neural network applied on spectrograms to learn features that capture timbral and temporal patterns.
- Input: 30-second audio clips converted to spectrograms using Short-time Fast Fourier Transform (STFT).
- Feature Learning: Designed 4 filters to detect patterns related to percussion, harmony, pitch slides etc. Convolved filters with spectrogram to obtain 4 feature maps.
- Subsampling: Applied 2x2 max pooling on feature maps for dimensionality reduction and translation invariance.
- Classification: Flattened feature maps and fed them into a Multilayer Perceptron (MLP) with softmax output for 10-way genre classification.
- Results: Achieved 72.4% accuracy on GTZAN dataset, outperforming MFCC+MLP (46.8%) and other baseline systems relying on hand-crafted features.
- Conclusion: Learned features from spectrograms using 2D CNNs capture more relevant information for genre classification than engineered MFCC features. End-to-end feature learning shows promise over pipeline systems.

In summary, the key ideas are - using 2D CNN on spectrograms for feature learning, end-to-end training, and demonstrating superior performance over traditional methods relying on MFCC and other hand-crafted audio features for music classification.

ADVANTAGES OF PROPOSED SYSTEM:

Some of the key problems this work is trying to address for music genre classification are:

1. Limitations of hand-crafted audio features like MFCCs:

- The paper mentions MFCCs lack dynamic analysis capability as they are extracted from single frames.
- MFCCs may not capture all the relevant information for genre classification.

2. Finding better representations from raw audio:

- Rather than using hand-crafted features, learn features directly from the spectrogram using convolutional neural nets.

3. Capturing temporal patterns:

- The 2D convolutional filters can capture patterns across both time and frequency dimensions of the spectrogram, unlike MFCCs.

4. Translation invariance:

- The max pooling provides some invariance to pitch shifting or tempo changes.

5. End-to-end learning:

- Compared to systems relying on engineered features, learn the feature extraction and classification together end-to-end.

So in summary, some of the key limitations the paper tries to address are:

- Finding better features from raw audio data rather than relying on hand-crafted features
- Learning features that capture temporal/spectral patterns
- Achieving some translation invariance
- End-to-end learning of features and classifier

The goal is to show convolutional neural networks can achieve better music genre classification from raw audio compared to approaches using traditional audio features.

Algorithm:

The proposed algorithm for music genre classification can be summarized as follows:

Input:

- Take 30-second audio clips
- Compute spectrogram using Short-time Fast Fourier Transform (STFT)
- Retain only magnitude values from spectrogram

Feature Extraction:

- Define 4 different 2D convolutional filters designed to capture different patterns in the spectrogram
- Convolve each filter with the input spectrogram to generate 4 feature maps
- This acts as a feature detector to extract useful representations

Subsampling:

- Apply 2x2 max pooling to each feature map
- Reduces dimensionality and provides translation invariance

Classification:

- Flatten the 4 subsampled feature maps into a vector
- Feed the feature vector into a Multilayer Perceptron (MLP)
- Use softmax activation in the output layer for predicting genre
- Train MLP in an end-to-end fashion via backpropagation

So in summary, the core proposed algorithm is:

1. Generate spectrogram from audio
2. Use 2D convolution to extract features
3. Max pool features
4. Feed into MLP for classification

The key aspects are using 2D convolutions on spectrograms for feature learning in an end-to-end model, rather than relying on engineered audio features like MFCCs used in prior work.

CHAPTER- 5

SYSTEM SPECIFICATION

HARDWARE REQUIREMENTS:

- **System** : Intel i3
- **Hard Disk** : 1 TB.
- **Monitor** : 14’ Colour Monitor.
- **Mouse** : Optical Mouse.
- **Ram** : 4GB.

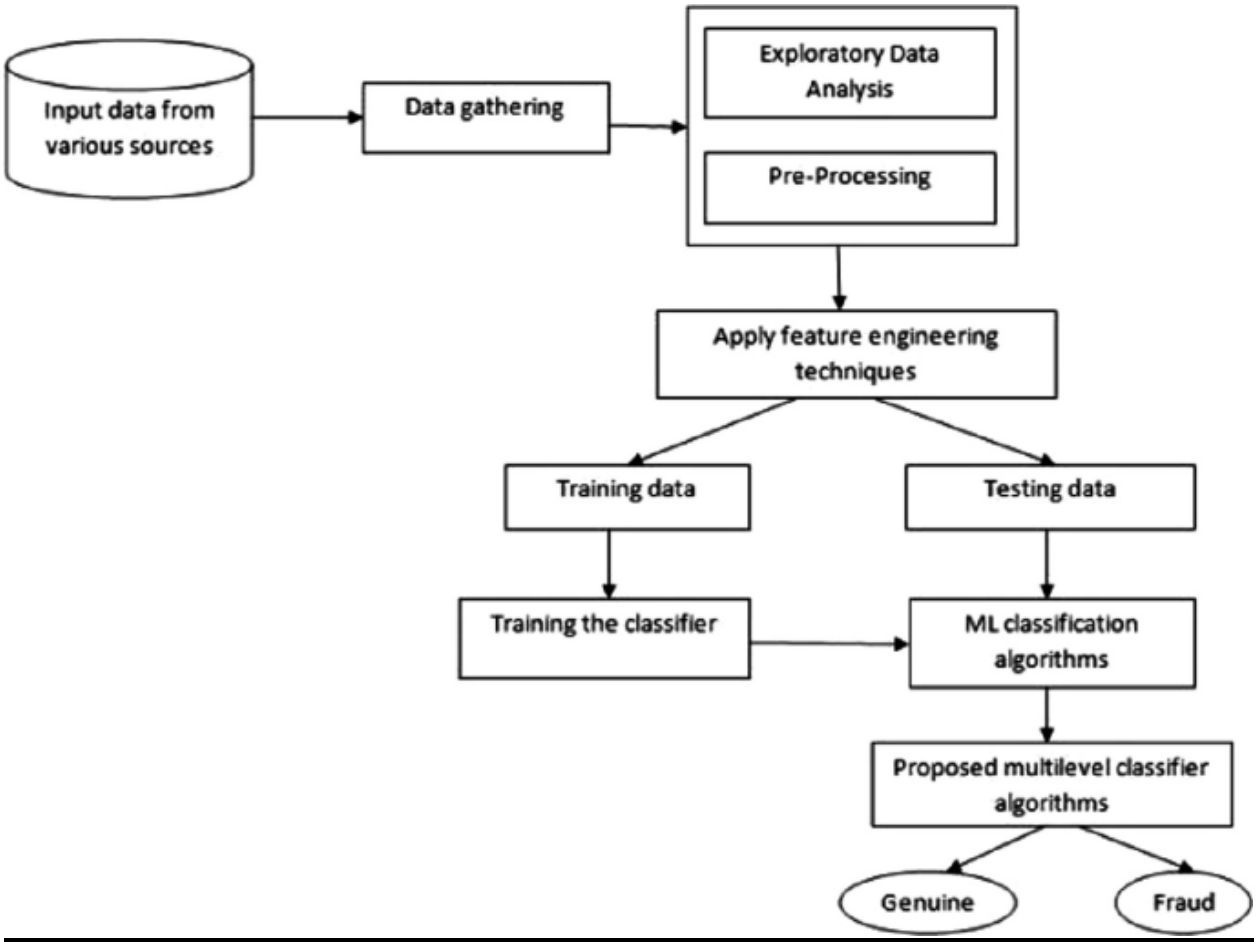
SOFTWARE REQUIREMENTS:

- **Operating system** : Windows 10.
- **Coding Language** : Python.
- **Front-End** : Html. CSS
- **Designing** : Html, css, javascript.
- **Data Base** : SQL

CHAPTER-6

SYSTEM DESIGN

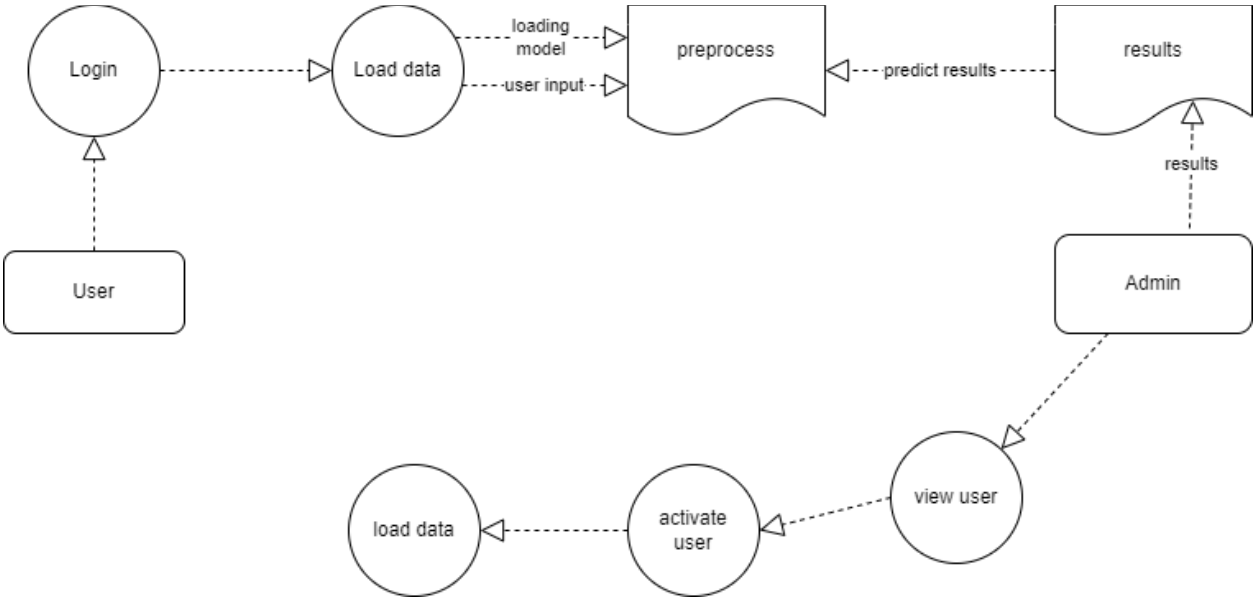
SYSTEM ARCHITECTURE:



DATA FLOW DIAGRAM:

1. The DFD is also called as bubble chart. It is a simple graphical formalism that can be used to represent a system in terms of input data to the system, various processing carried out on this data, and the output data is generated by this system.
2. The data flow diagram (DFD) is one of the most important modeling tools. It is used to model the system components. These components are the system process, the data used by the process, an external entity that interacts with the system and the information flows in the system.

3. DFD shows how the information moves through the system and how it is modified by a series of transformations. It is a graphical technique that depicts information flow and the transformations that are applied as data moves from input to output.
4. DFD is also known as bubble chart. A DFD may be used to represent a system at any level of abstraction. DFD may be partitioned into levels that represent increasing information flow and functional detail.



UML DIAGRAMS

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.

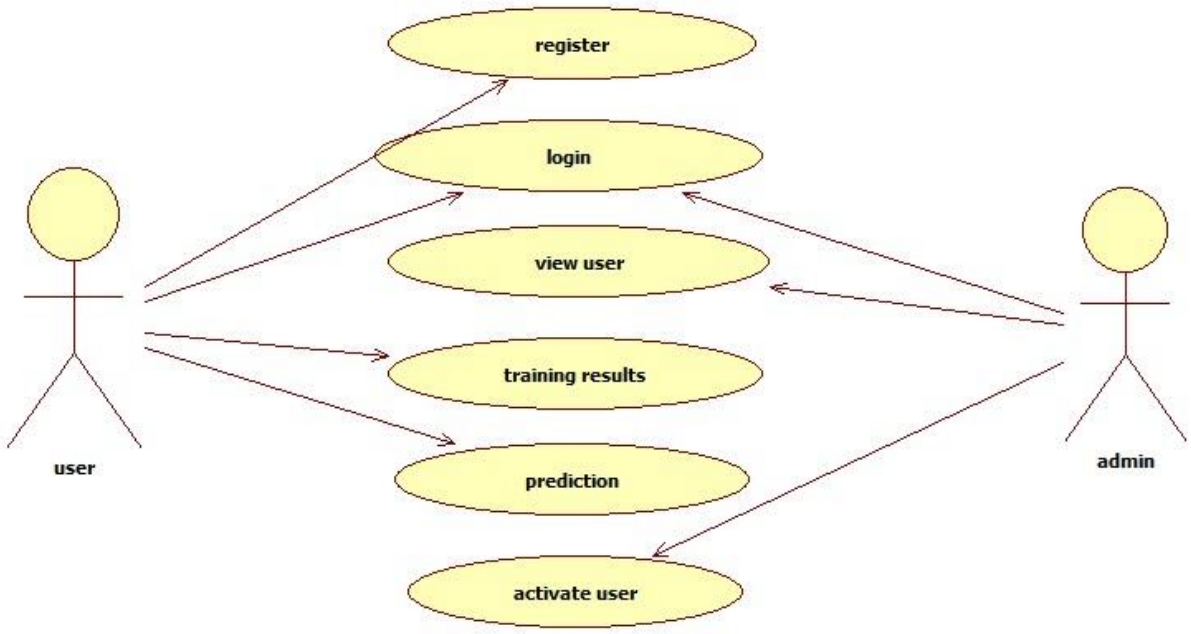
GOALS:

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.

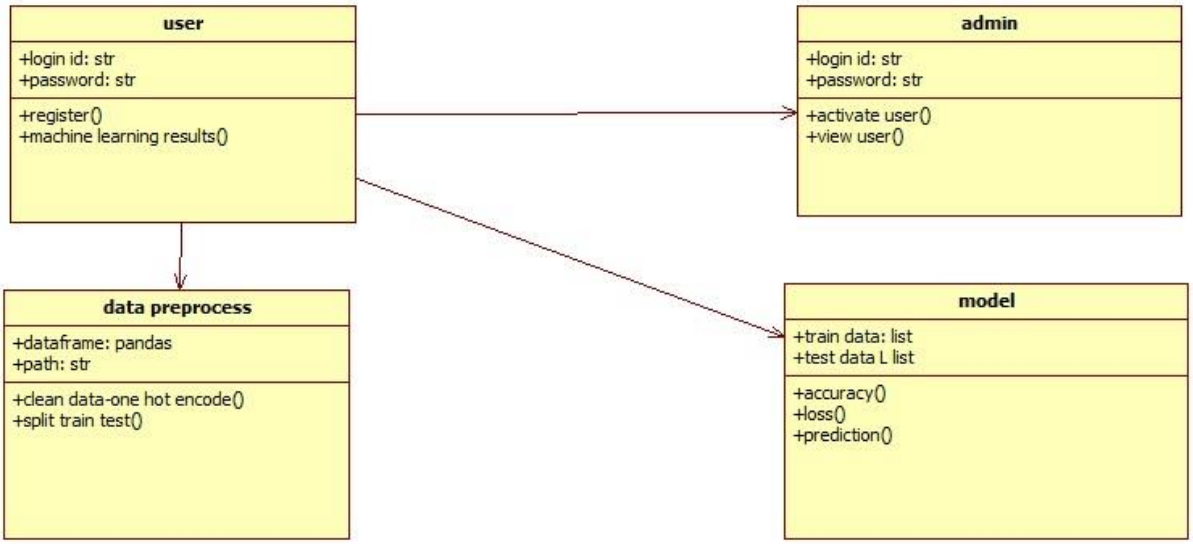
USE CASE DIAGRAM:

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.



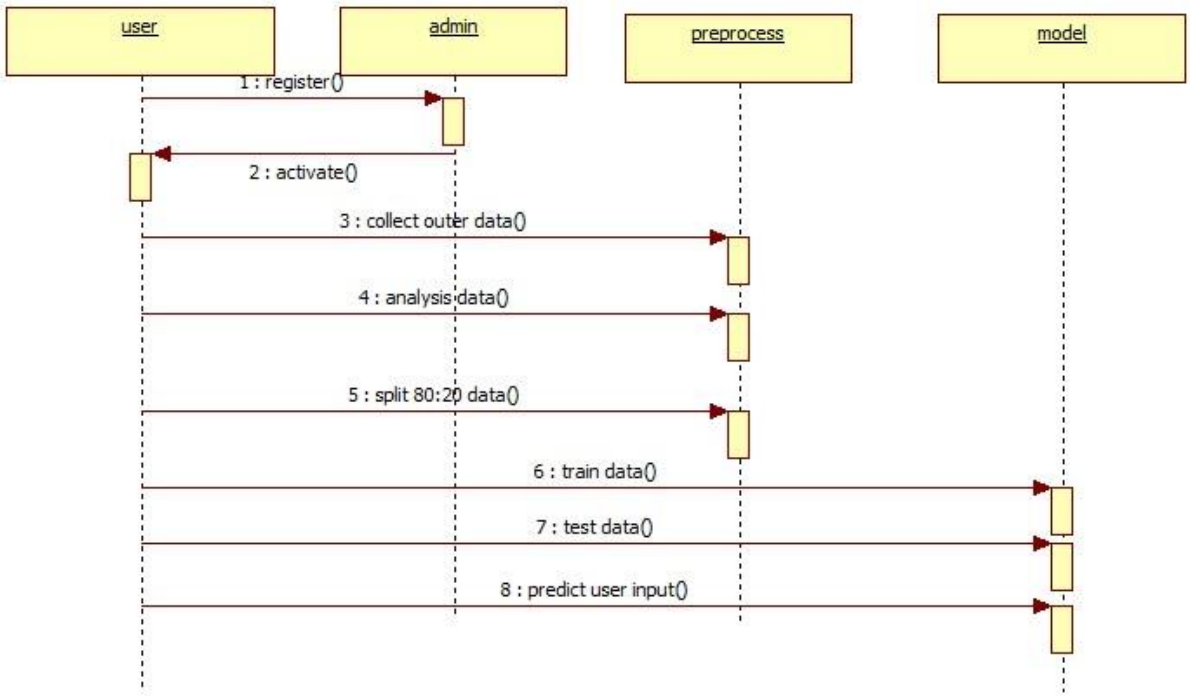
CLASS DIAGRAM:

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.



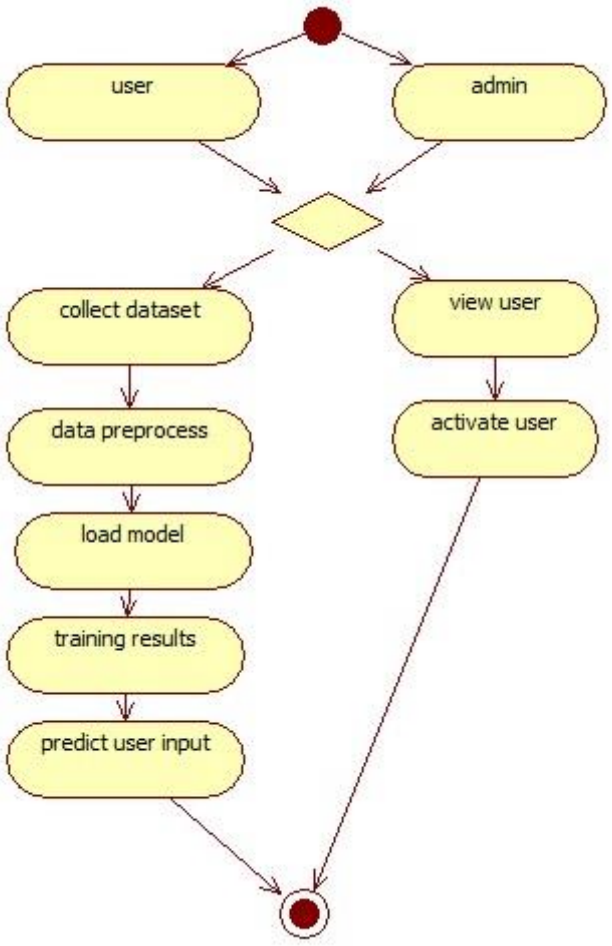
SEQUENCE DIAGRAM:

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.



ACTIVITY 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.



CHAPTER – 7

SOFTWARE ENVIRONMENT

7.1 PYTHON

Python is a general-purpose interpreted, interactive, object-oriented, and high-level programming language. An interpreted language, Python has a design philosophy that emphasizes code readability (not able using whitespace indentation to delimit code blocks rather than curly brackets or keywords), and a syntax that allows programmers to express concepts in fewer lines of code than might be used in languages such as C++or Java. It provides constructs that enable clear programming on both small and large scales. Python interpreters are available for many operating systems. CPython, the reference implementation of Python, is open source software and has a community-based development model, as do nearly all of its variant implementations. CPython is managed by the non-profit Python Software Foundation. Python features a dynamic type system and automatic memory management. It supports multiple programming including object-oriented, imperative, functional and procedural, and has a large and comprehensive standard library**.**

Interactive Mode Programming

Invoking the interpreter without passing a script file as a parameter brings up the following prompt –

$ python

Python 2.4.3 (#1, Nov 11 2010, 13:34:43)

[GCC 4.1.2 20080704 (Red Hat 4.1.2-48)] on linux2

Type "help", "copyright", "credits" or "license" for more information.

>>>

Type the following text at the Python prompt and press the Enter −

>>> print "Hello, Python!"

If you are running new version of Python, then you would need to use print statement with parenthesis as in print ("Hello, Python!");. However in Python version 2.4.3, this produces the following result −

Hello, Python!

Script Mode Programming

Invoking the interpreter with a script parameter begins execution of the script and continues until the script is finished. When the script is finished, the interpreter is no longer active.

Let us write a simple Python program in a script. Python files have extension .py. Type the following source code in a test.py file −

Live Demo

```
print "Hello, Python!"
```

We assume that you have Python interpreter set in PATH variable. Now, try to run this program as follows −

```
$ python test.py
```

This produces the following result −

```
Hello, Python!
```

Let us try another way to execute a Python script. Here is the modified test.py file −

Live Demo

```
#!/usr/bin/python

print "Hello, Python!"
```

We assume that you have Python interpreter available in /usr/bin directory. Now, try to run this program as follows −

```
$ chmod +x test.py     # This is to make file executable
$./test.py
```

This produces the following result −

Hello, Python!

Python Identifiers

A Python identifier is a name used to identify a variable, function, class, module or other object. An identifier starts with a letter A to Z or a to z or an underscore (_) followed by zero or more letters, underscores and digits (0 to 9).

Python does not allow punctuation characters such as @, $, and % within identifiers. Python is a case sensitive programming language. Thus, Manpower and manpower are two different identifiers in Python.

Here are naming conventions for Python identifiers −

Class names start with an uppercase letter. All other identifiers start with a lowercase letter.

Starting an identifier with a single leading underscore indicates that the identifier is private.

Starting an identifier with two leading underscores indicates a strongly private identifier.

If the identifier also ends with two trailing underscores, the identifier is a language-defined special name.

Reserved Words

The following list shows the Python keywords. These are reserved words and you cannot use them as constant or variable or any other identifier names. All the Python keywords contain lowercase letters only.

and	exec	not
assert	finally	or
break	for	pass
class	from	print
continue		global raise
def	if	return
del	import	try
elif	in	while
else	is	with
except	lambda	yield

Lines and Indentation

Python provides no braces to indicate blocks of code for class and function definitions or flow control. Blocks of code are denoted by line indentation, which is rigidly enforced.

The number of spaces in the indentation is variable, but all statements within the block must be indented the same amount. For example −

```
if True:
   print "True"
else:
   print "False"
```

However, the following block generates an error −

```
if True:
```

```
print "Answer"
print "True"
else:
print "Answer"
print "False"
```

Thus, in Python all the continuous lines indented with same number of spaces would form a block. The following example has various statement blocks −

Note − Do not try to understand the logic at this point of time. Just make sure you understood various blocks even if they are without braces.

```
#!/usr/bin/python

import sys

try:
   # open file stream
   file = open(file_name, "w")
except IOError:
   print "There was an error writing to", file_name
   sys.exit()
print "Enter '", file_finish,
print "' When finished"
while file_text != file_finish:
   file_text = raw_input("Enter text: ")
   if file_text == file_finish:
      # close the file
      file.close
```

```
    break
  file.write(file_text)
  file.write("\n")
file.close()
file_name = raw_input("Enter filename: ")
if len(file_name) == 0:
  print "Next time please enter something"
  sys.exit()
try:
  file = open(file_name, "r")
except IOError:
  print "There was an error reading file"
  sys.exit()
file_text = file.read()
file.close()
print file_text
```

Multi-Line Statements

Statements in Python typically end with a new line. Python does, however, allow the use of the line continuation character (\) to denote that the line should continue. For example −

```
total = item_one + \
        item_two + \
        item_three
```

Statements contained within the [], {}, or () brackets do not need to use the line continuation character. For example −

```
days = ['Monday', 'Tuesday', 'Wednesday',
        'Thursday', 'Friday']
```

Quotation in Python

Python accepts single ('), double (") and triple ('" or """) quotes to denote string literals, as long as the same type of quote starts and ends the string.

The triple quotes are used to span the string across multiple lines. For example, all the following are legal −

```
word = 'word'
sentence = "This is a sentence."
paragraph = """This is a paragraph. It is
made up of multiple lines and sentences."""
```

Comments in Python

A hash sign (#) that is not inside a string literal begins a comment. All characters after the # and up to the end of the physical line are part of the comment and the Python interpreter ignores them.

Live Demo

```
#!/usr/bin/python

# First comment
print "Hello, Python!" # second comment
```

This produces the following result −

```
Hello, Python!
```

You can type a comment on the same line after a statement or expression −

```
name = "Madisetti" # This is again comment
```

You can comment multiple lines as follows −

```
# This is a comment.
```

This is a comment, too.

This is a comment, too.

I said that already.

Following triple-quoted string is also ignored by Python interpreter and can be used as a multiline comments:

'''

This is a multiline

comment.

'''

Using Blank Lines

A line containing only whitespace, possibly with a comment, is known as a blank line and Python totally ignores it.

In an interactive interpreter session, you must enter an empty physical line to terminate a multiline statement.

Waiting for the User

The following line of the program displays the prompt, the statement saying “Press the enter key to exit”, and waits for the user to take action −

#!/usr/bin/python

raw_input("\n\nPress the enter key to exit.")

Here, "\n\n" is used to create two new lines before displaying the actual line. Once the user presses the key, the program ends. This is a nice trick to keep a console window open until the user is done with an application.

Multiple Statements on a Single Line

The semicolon (;) allows multiple statements on the single line given that neither statement starts a new code block. Here is a sample snip using the semicolon.

```
import sys; x = 'foo'; sys.stdout.write(x + '\n')
```

Multiple Statement Groups as Suites

A group of individual statements, which make a single code block are called suites in Python. Compound or complex statements, such as if, while, def, and class require a header line and a suite.

Header lines begin the statement (with the keyword) and terminate with a colon (:) and are followed by one or more lines which make up the suite. For example −

```
if expression :
   suite
elif expression :
   suite
else :
   suite
```

Command Line Arguments

Many programs can be run to provide you with some basic information about how they should be run. Python enables you to do this with -h −

```
$ python -h
usage: python [option] ... [-c cmd | -m mod | file | -] [arg] ...
Options and arguments (and corresponding environment variables):
-c cmd : program passed in as string (terminates option list)
-d     : debug output from parser (also PYTHONDEBUG=x)
-E     : ignore environment variables (such as PYTHONPATH)
-h     : print this help message and exit
```

You can also program your script in such a way that it should accept various options. Command Line Arguments is an advanced topic and should be studied a bit later once you have gone through rest of the Python concepts.

Python Lists

The list is a most versatile datatype available in Python which can be written as a list of comma-separated values (items) between square brackets. Important thing about a list is that items in a list need not be of the same type.

Creating a list is as simple as putting different comma-separated values between square brackets. For example −

```
list1 = ['physics', 'chemistry', 1997, 2000];

list2 = [1, 2, 3, 4, 5 ];

list3 = ["a", "b", "c", "d"]
```

Similar to string indices, list indices start at 0, and lists can be sliced, concatenated and so on.

A tuple is a sequence of immutable Python objects. Tuples are sequences, just like lists. The differences between tuples and lists are, the tuples cannot be changed unlike lists and tuples use parentheses, whereas lists use square brackets.

Creating a tuple is as simple as putting different comma-separated values. Optionally you can put these comma-separated values between parentheses also. For example −

```
tup1 = ('physics', 'chemistry', 1997, 2000);

tup2 = (1, 2, 3, 4, 5 );

tup3 = "a", "b", "c", "d";
```

The empty tuple is written as two parentheses containing nothing −

```
tup1 = ();
```

To write a tuple containing a single value you have to include a comma, even though there is only one value −

```
tup1 = (50,);
```

Like string indices, tuple indices start at 0, and they can be sliced, concatenated, and so on.

Accessing Values in Tuples

To access values in tuple, use the square brackets for slicing along with the index or indices to obtain value available at that index. For example −

Live Demo

```
#!/usr/bin/python

tup1 = ('physics', 'chemistry', 1997, 2000);
tup2 = (1, 2, 3, 4, 5, 6, 7 );
print "tup1[0]: ", tup1[0];
print "tup2[1:5]: ", tup2[1:5];
```

When the above code is executed, it produces the following result −

```
tup1[0]:  physics
tup2[1:5]:  [2, 3, 4, 5]
```

Updating Tuples

Accessing Values in Dictionary

To access dictionary elements, you can use the familiar square brackets along with the key to obtain its value. Following is a simple example −

Live Demo

```
#!/usr/bin/python

dict = {'Name': 'Zara', 'Age': 7, 'Class': 'First'}
print "dict['Name']: ", dict['Name']
```

```
print "dict['Age']: ", dict['Age']
```

When the above code is executed, it produces the following result −

```
dict['Name']:  Zara
dict['Age']:  7
```

If we attempt to access a data item with a key, which is not part of the dictionary, we get an error as follows −

Live Demo

```
#!/usr/bin/python

dict = {'Name': 'Zara', 'Age': 7, 'Class': 'First'}
print "dict['Alice']: ", dict['Alice']
```

When the above code is executed, it produces the following result −

```
dict['Alice']:
Traceback (most recent call last):
  File "test.py", line 4, in <module>
    print "dict['Alice']: ", dict['Alice'];
KeyError: 'Alice'
```

Updating Dictionary

You can update a dictionary by adding a new entry or a key-value pair, modifying an existing entry, or deleting an existing entry as shown below in the simple example −

Live Demo

```
#!/usr/bin/python

dict = {'Name': 'Zara', 'Age': 7, 'Class': 'First'}
```

```
dict['Age'] = 8; # update existing entry
dict['School'] = "DPS School"; # Add new entry

print "dict['Age']: ", dict['Age']
print "dict['School']: ", dict['School']
```

When the above code is executed, it produces the following result −

```
dict['Age']:  8
dict['School']:  DPS School
```

Delete Dictionary Elements

You can either remove individual dictionary elements or clear the entire contents of a dictionary. You can also delete entire dictionary in a single operation.

To explicitly remove an entire dictionary, just use the del statement. Following is a simple example −

Live Demo

```
#!/usr/bin/python

dict = {'Name': 'Zara', 'Age': 7, 'Class': 'First'}
del dict['Name']; # remove entry with key 'Name'
dict.clear();     # remove all entries in dict
del dict ;        # delete entire dictionary

print "dict['Age']: ", dict['Age']
print "dict['School']: ", dict['School']
```

This produces the following result. Note that an exception is raised because after del dict dictionary does not exist any more −

dict['Age']:

Traceback (most recent call last):

File "test.py", line 8, in <module>

print "dict['Age']: ", dict['Age'];

TypeError: 'type' object is unsubscriptable

Note − del() method is discussed in subsequent section.

Properties of Dictionary Keys

Dictionary values have no restrictions. They can be any arbitrary Python object, either standard objects or user-defined objects. However, same is not true for the keys.

There are two important points to remember about dictionary keys −

(a) More than one entry per key not allowed. Which means no duplicate key is allowed. When duplicate keys encountered during assignment, the last assignment wins. For example −

Live Demo

#!/usr/bin/python

dict = {'Name': 'Zara', 'Age': 7, 'Name': 'Manni'}

print "dict['Name']: ", dict['Name']

When the above code is executed, it produces the following result −

dict['Name']: Manni

(b) Keys must be immutable. Which means you can use strings, numbers or tuples as dictionary keys but something like ['key'] is not allowed. Following is a simple example −

Live Demo

#!/usr/bin/python

```
dict = {['Name']: 'Zara', 'Age': 7}

print "dict['Name']: ", dict['Name']
```

When the above code is executed, it produces the following result −

```
Traceback (most recent call last):

  File "test.py", line 3, in <module>

    dict = {['Name']: 'Zara', 'Age': 7};

TypeError: unhashable type: 'list'
```

Tuples are immutable which means you cannot update or change the values of tuple elements. You are able to take portions of existing tuples to create new tuples as the following example demonstrates −

Live Demo

```
#!/usr/bin/python

tup1 = (12, 34.56);

tup2 = ('abc', 'xyz');

# Following action is not valid for tuples

# tup1[0] = 100;

# So let's create a new tuple as follows

tup3 = tup1 + tup2;

print tup3;
```

When the above code is executed, it produces the following result −

```
(12, 34.56, 'abc', 'xyz')
```

Delete Tuple Elements

Removing individual tuple elements is not possible. There is, of course, nothing wrong with putting together another tuple with the undesired elements discarded.

To explicitly remove an entire tuple, just use the del statement. For example −

Live Demo

```
#!/usr/bin/python

tup = ('physics', 'chemistry', 1997, 2000);
print tup;
del tup;
print "After deleting tup : ";
print tup;
```

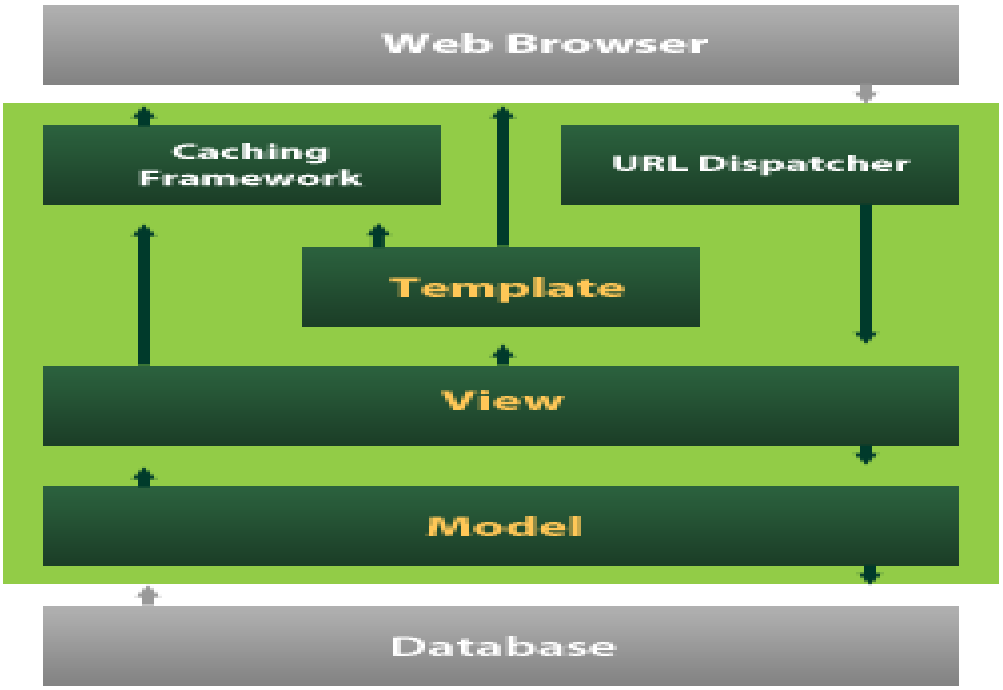
This produces the following result. Note an exception raised, this is because after del tup tuple does not exist any more −

```
('physics', 'chemistry', 1997, 2000)
After deleting tup :
Traceback (most recent call last):
  File "test.py", line 9, in <module>
    print tup;
NameError: name 'tup' is not defined
```

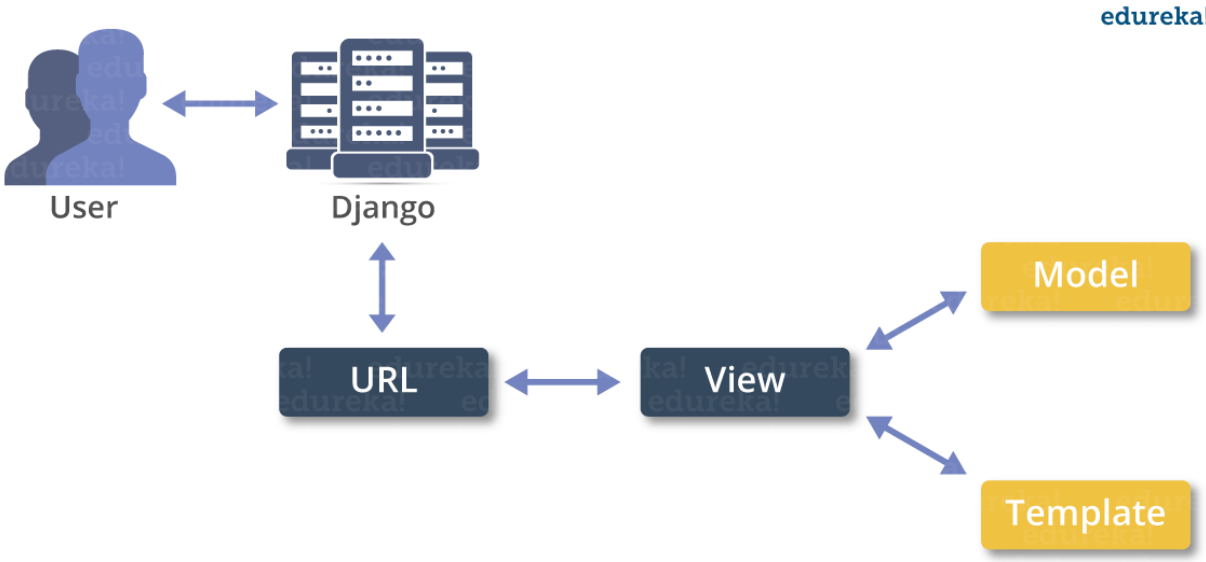
DJANGO

Django is a high-level Python Web framework that encourages rapid development and clean, pragmatic design. Built by experienced developers, it takes care of much of the hassle of Web development, so you can focus on writing your app without needing to reinvent the wheel. It's free and open source.

Django's primary goal is to ease the creation of complex, database-driven websites. Django emphasizes reusability and "pluggability" of components, rapid development, and the principle of don't repeat yourself. Python is used throughout, even for settings files and data models.



Django also provides an optional administrative create, read, update and delete interface that is generated dynamically through introspection and configured via admin models.



Create a Project

Whether you are on Windows or Linux, just get a terminal or a cmd prompt and navigate to the place you want your project to be created, then use this code −

$ django-admin startproject myproject

This will create a "myproject" folder with the following structure −

myproject/

  manage.py

  myproject/

    __init__.py

    settings.py

    urls.py

    wsgi.py

The Project Structure

The “myproject” folder is just your project container, it actually contains two elements −

manage.py − This file is kind of your project local django-admin for interacting with your project via command line (start the development server, sync db...). To get a full list of command accessible via manage.py you can use the code −

$ python manage.py help

The “myproject” subfolder − This folder is the actual python package of your project. It contains four files −

__init__.py − Just for python, treat this folder as package.

settings.py − As the name indicates, your project settings.

urls.py − All links of your project and the function to call. A kind of ToC of your project.

wsgi.py − If you need to deploy your project over WSGI.

Setting Up Your Project

Your project is set up in the subfolder myproject/settings.py. Following are some important options you might need to set −

DEBUG = True

This option lets you set if your project is in debug mode or not. Debug mode lets you get more information about your project's error. Never set it to ‘True’ for a live project. However, this has to be set to ‘True’ if you want the Django light server to serve static files. Do it only in the development mode.

```
DATABASES = {
  'default': {
    'ENGINE': 'django.db.backends.sqlite3',
    'NAME': 'database.sql',
    'USER': '',
    'PASSWORD': '',
```

```
    'HOST': '',
    'PORT': '',
  }
}
```

Database is set in the 'Database' dictionary. The example above is for SQLite engine. As stated earlier, Django also supports −

MySQL (django.db.backends.mysql)

PostGreSQL (django.db.backends.postgresql_psycopg2)

Oracle (django.db.backends.oracle) and NoSQL DB

MongoDB (django_mongodb_engine)

Before setting any new engine, make sure you have the correct db driver installed.

You can also set others options like: TIME_ZONE, LANGUAGE_CODE, TEMPLATE…

Now that your project is created and configured make sure it's working −

```
$ python manage.py runserver
```

You will get something like the following on running the above code −

```
Validating models...

0 errors found
September 03, 2015 - 11:41:50
Django version 1.6.11, using settings 'myproject.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CONTROL-C.
```

A project is a sum of many applications. Every application has an objective and can be reused into another project, like the contact form on a website can be an application, and can be reused for others. See it as a module of your project.

Create an Application

We assume you are in your project folder. In our main "myproject" folder, the same folder then manage.py −

$ python manage.py startapp myapp

You just created myapp application and like project, Django create a "myapp" folder with the application structure −

myapp/

__init__.py

admin.py

models.py

tests.py

views.py

__init__.py − Just to make sure python handles this folder as a package.

admin.py − This file helps you make the app modifiable in the admin interface.

models.py − This is where all the application models are stored.

tests.py − This is where your unit tests are.

views.py − This is where your application views are.

Get the Project to Know About Your Application

At this stage we have our "myapp" application, now we need to register it with our Django project "myproject". To do so, update INSTALLED_APPS tuple in the settings.py file of your project (add your app name) −

```
INSTALLED_APPS = (
  'django.contrib.admin',
  'django.contrib.auth',
  'django.contrib.contenttypes',
  'django.contrib.sessions',
  'django.contrib.messages',
  'django.contrib.staticfiles',
  'myapp',
)
```

Creating forms in Django, is really similar to creating a model. Here again, we just need to inherit from Django class and the class attributes will be the form fields. Let's add a forms.py file in myapp folder to contain our app forms. We will create a login form.

myapp/forms.py

```
#-*- coding: utf-8 -*-
from django import forms

class LoginForm(forms.Form):
  user = forms.CharField(max_length = 100)
  password = forms.CharField(widget = forms.PasswordInput())
```

As seen above, the field type can take "widget" argument for html rendering; in our case, we want the password to be hidden, not displayed. Many others widget are present in Django: DateInput for dates, CheckboxInput for checkboxes, etc.

Using Form in a View

There are two kinds of HTTP requests, GET and POST. In Django, the request object passed as parameter to your view has an attribute called "method" where the type of the request is set, and all data passed via POST can be accessed via the request.POST dictionary.

Let's create a login view in our myapp/views.py −

```
#-*- coding: utf-8 -*-
from myapp.forms import LoginForm

def login(request):
  username = "not logged in"

  if request.method == "POST":
    #Get the posted form
    MyLoginForm = LoginForm(request.POST)

    if MyLoginForm.is_valid():
      username = MyLoginForm.cleaned_data['username']
  else:
    MyLoginForm = Loginform()

  return render(request, 'loggedin.html', {"username" : username})
```

The view will display the result of the login form posted through the loggedin.html. To test it, we will first need the login form template. Let's call it login.html.

```
<html>
  <body>
```

```
<form name = "form" action = "{% url "myapp.views.login" %}"
  method = "POST" >{% csrf_token %}

  <div style = "max-width:470px;">
    <center>
      <input type = "text" style = "margin-left:20%;"
        placeholder = "Identifiant" name = "username" />
    </center>
  </div>

  <br>

  <div style = "max-width:470px;">
    <center>
      <input type = "password" style = "margin-left:20%;"
        placeholder = "password" name = "password" />
    </center>
  </div>

  <br>

  <div style = "max-width:470px;">
    <center>

      <button style = "border:0px; background-color:#4285F4; margin-top:8%;
        height:35px; width:80%;margin-left:19%;" type = "submit"
        value = "Login" >
        <strong>Login</strong>
```

```
            </button>

          </center>
        </div>

      </form>

  </body>
</html>
```

The template will display a login form and post the result to our login view above. You have probably noticed the tag in the template, which is just to prevent Cross-site Request Forgery (CSRF) attack on your site.

```
{% csrf_token %}
```

Once we have the login template, we need the loggedin.html template that will be rendered after form treatment.

```
<html>

  <body>
    You are : <strong>{{username}}</strong>
  </body>

</html>
```

Now, we just need our pair of URLs to get started: myapp/urls.py

```
from django.conf.urls import patterns, url
from django.views.generic import TemplateView
```

```
urlpatterns = patterns('myapp.views',
   url(r'^connection/',TemplateView.as_view(template_name = 'login.html')),
   url(r'^login/', 'login', name = 'login'))
```

When accessing "/myapp/connection", we will get the following login.html template rendered −

Setting Up Sessions

In Django, enabling session is done in your project settings.py, by adding some lines to the MIDDLEWARE_CLASSES and the INSTALLED_APPS options. This should be done while creating the project, but it's always good to know, so MIDDLEWARE_CLASSES should have −

```
'django.contrib.sessions.middleware.SessionMiddleware'
```

And INSTALLED_APPS should have −

```
'django.contrib.sessions'
```

By default, Django saves session information in database (django_session table or collection), but you can configure the engine to store information using other ways like: in file or in cache.

When session is enabled, every request (first argument of any view in Django) has a session (dict) attribute.

Let's create a simple sample to see how to create and save sessions. We have built a simple login system before (see Django form processing chapter and Django Cookies Handling chapter). Let us save the username in a cookie so, if not signed out, when accessing our login page you won't see the login form. Basically, let's make our login system we used in Django Cookies handling more secure, by saving cookies server side.

For this, first lets change our login view to save our username cookie server side −

```
def login(request):
   username = 'not logged in'
```

```
  if request.method == 'POST':
    MyLoginForm = LoginForm(request.POST)

    if MyLoginForm.is_valid():
      username = MyLoginForm.cleaned_data['username']
      request.session['username'] = username
    else:
      MyLoginForm = LoginForm()

  return render(request, 'loggedin.html', {"username" : username}
```

Then let us create formView view for the login form, where we won't display the form if cookie is set −

```
def formView(request):
  if request.session.has_key('username'):
    username = request.session['username']
    return render(request, 'loggedin.html', {"username" : username})
  else:
    return render(request, 'login.html', {})
```

Now let us change the url.py file to change the url so it pairs with our new view −

```
from django.conf.urls import patterns, url
from django.views.generic import TemplateView

urlpatterns = patterns('myapp.views',
  url(r'^connection/','formView', name = 'loginform'),
  url(r'^login/', 'login', name = 'login'))
```

When accessing /myapp/connection, you will get to see the following page

CHAPTER -8

SOFTWARE INSTALLATION

8.1PYTHON

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 :

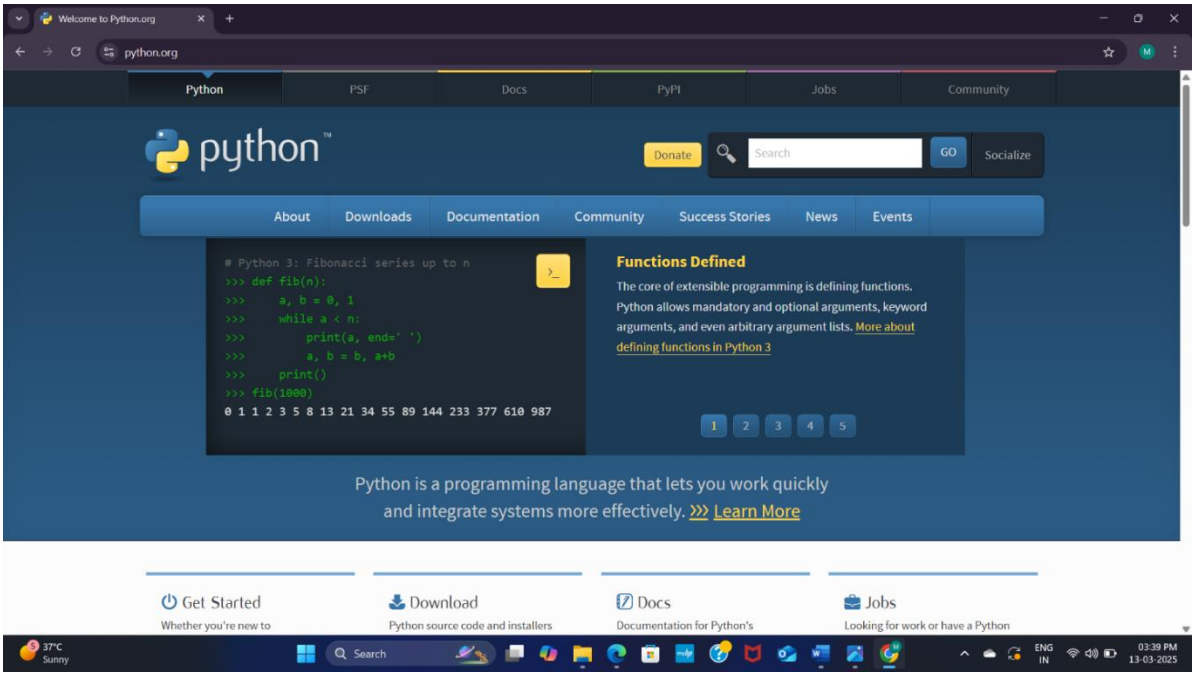
There have been several updates in the Python version over the years. The question is how to install Python? It might be confusing for the beginner who is willing to start learning Python but this tutorial will solve your query. The latest or the newest version of Python is version 3.7.4 or in other 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 Cheat sheet here. The steps on how to install Python on Windows 10, 8 and 7 are divided into 4 parts to help understand better.

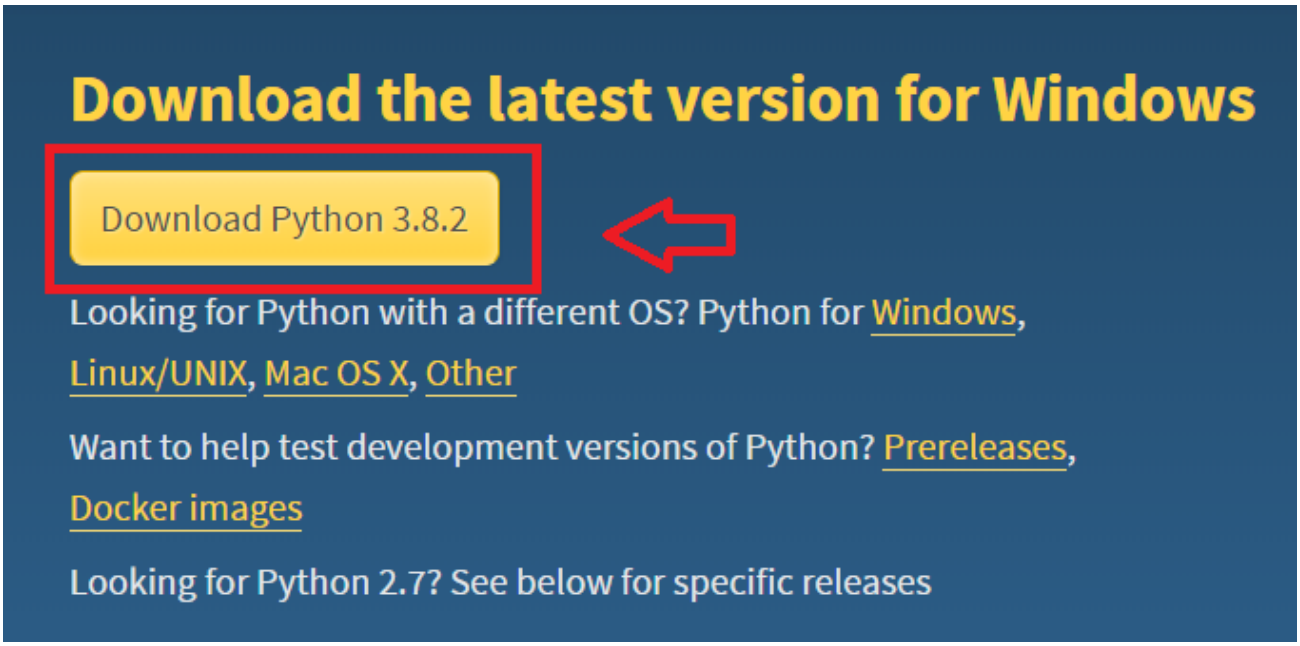
8.1.1Download the Correct version into the system

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:



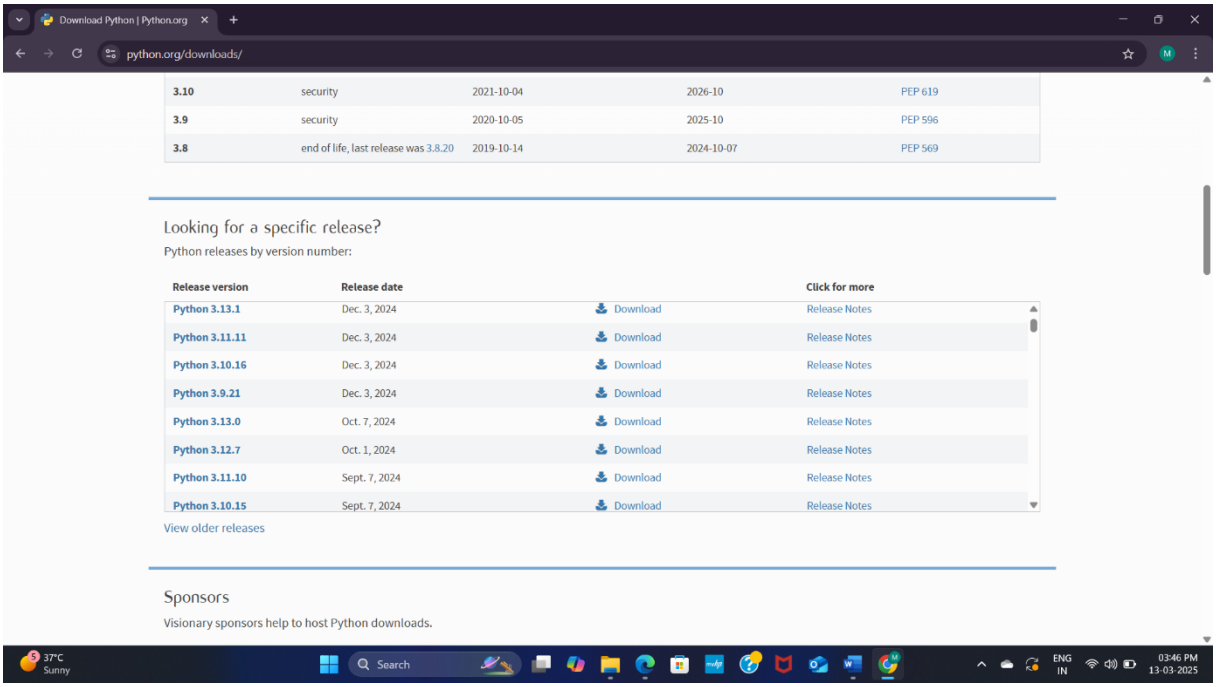
Now, check for the latest and the correct version for your operating system.

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

Step4: 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

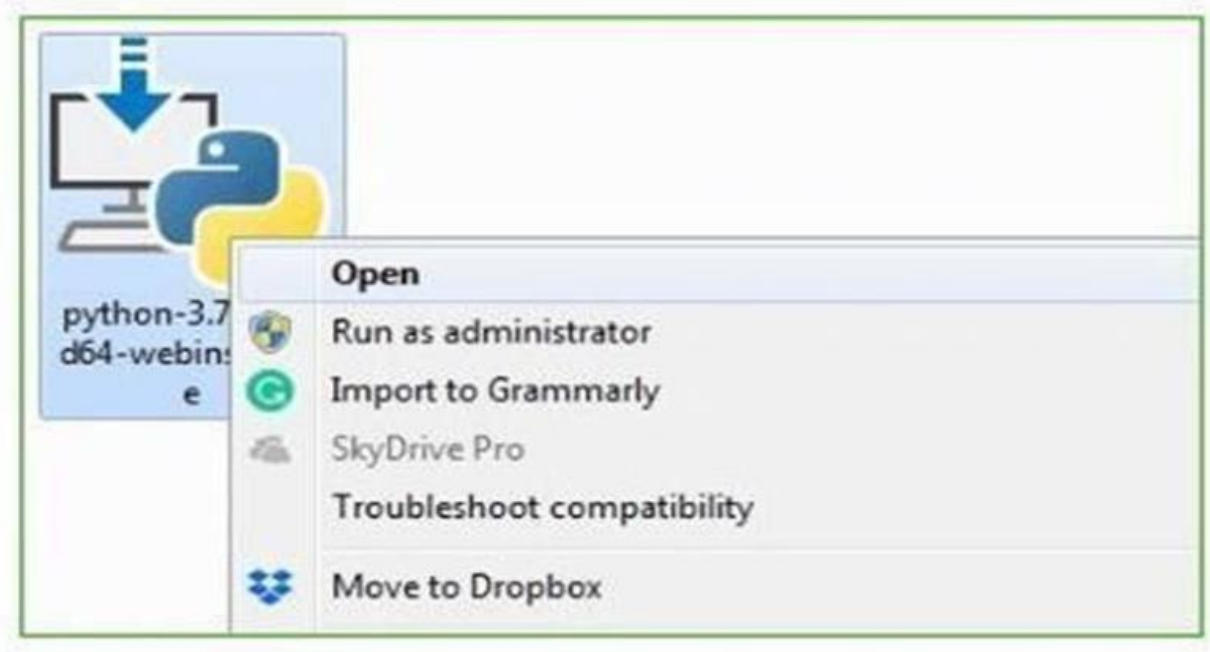
Files

Version	Operating System	Description
Gzipped source tarball	Source release	
XZ compressed source tarball	Source release	
Mac OS X 64-bit/32-bit installer	macOS	for Mac OS X 10.6 and later
Windows help file	Windows	
Windows x86-64 embeddable zip file	Windows	for AMD64/EM64T/x64
Windows x86-64 executable installer	Windows	for AMD64/EM64T/x64
Windows x86-64 web-based installer	Windows	for AMD64/EM64T/x64
Windows x86 embeddable zip file	Windows	
Windows x86 executable installer	Windows	
Windows x86 web-based installer	Windows	

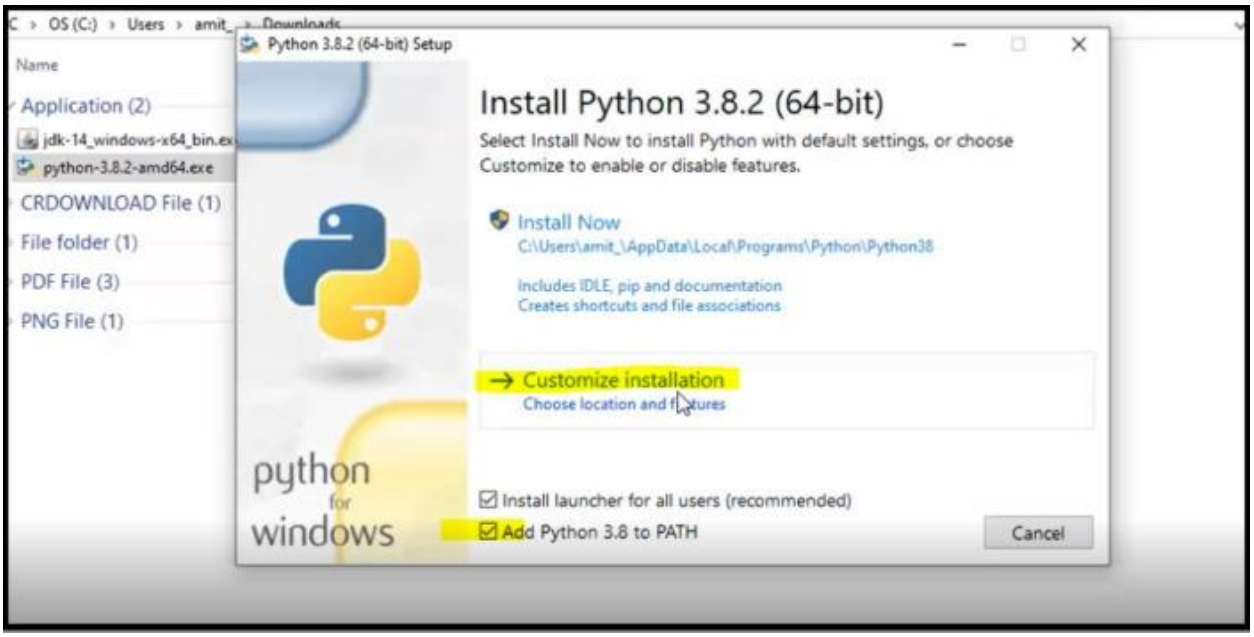
Here we will install Windows x86-64 web-based installer. Here your first part regarding which version of python is to be downloaded is completed. Now we move ahead with the second part in installing python i.e. Installation To download Windows32-bit python,you can select any one from the three options: Windowsx86 embeddable zip file, Windowsx86 executable installer or Windows x86 web-based installer. Note: To know the changes or updates that are made in the version you can click on the Release Note Option.

8.1.1 Installation of Python

Step1: Go to Download and Open the downloaded python version to carry out the installation process.



Step2: Before you click on Install Now, Make sure to put a tick on AddPython3.7 to PATH



Step3: Click on Install NOW After the installation is successful. Click on Close.

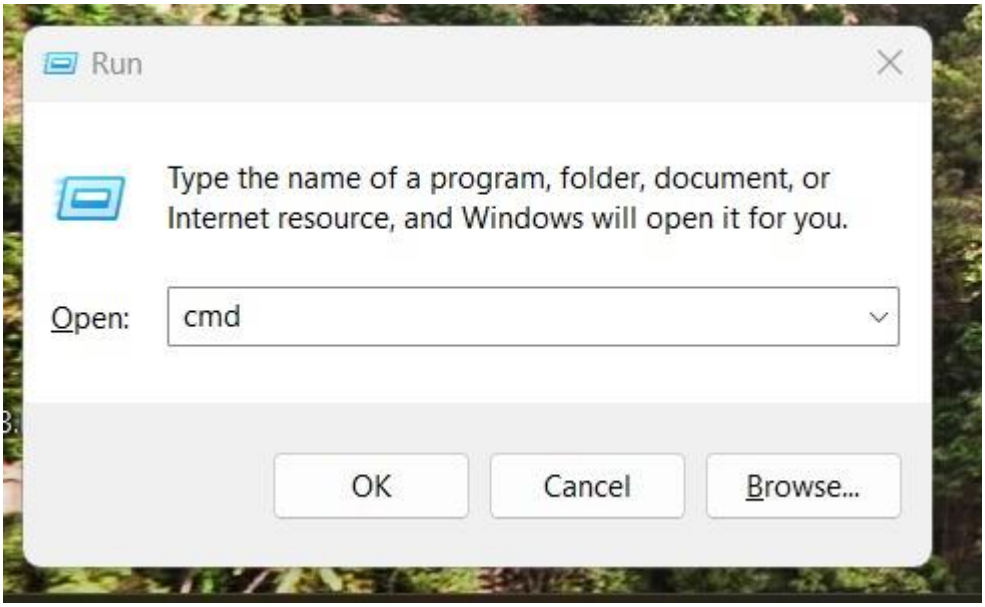
With these above three steps on python installation, you have successfully and correctly installed Python. Now is the time to verify the installation.

Note:The installation process might take acouple of minutes.

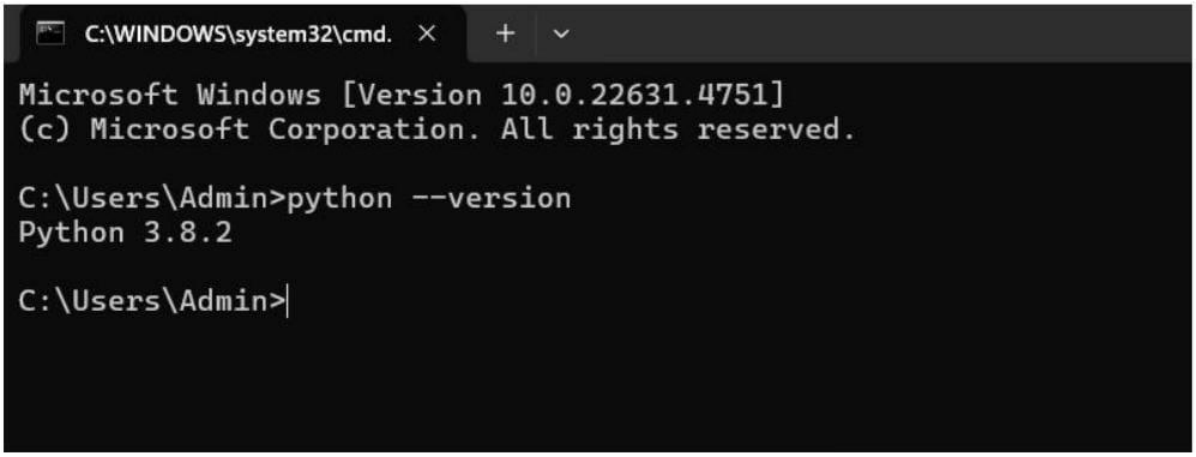
8.1.3 Verify the PythonInstallation

Step1: Click on Start

Step2:In the Windows RunCommand, type“cmd”



Step3: Open the Command promptoption.



Step4: Let us test whether the python is correctly installed. Type python-Vandpress Enter.

Step5: You will get the answer as3.7.4

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.

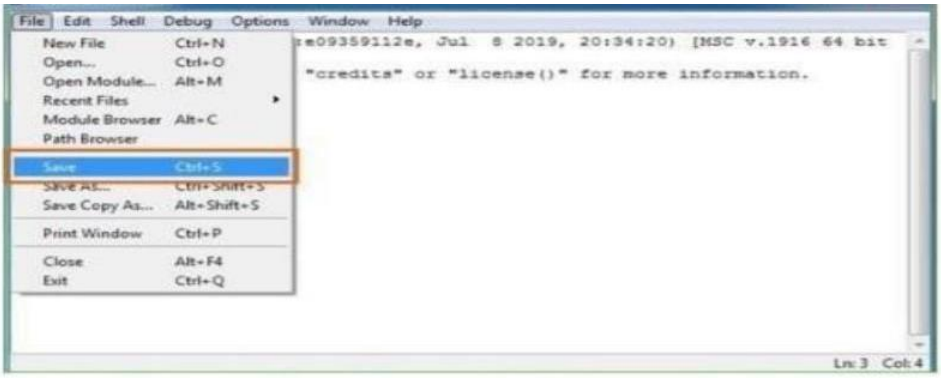
8.1.4 Check how the PythonIDLE works

Step 1: Click on Start

Step 2: In the Windows Run command, type "pythonidle"

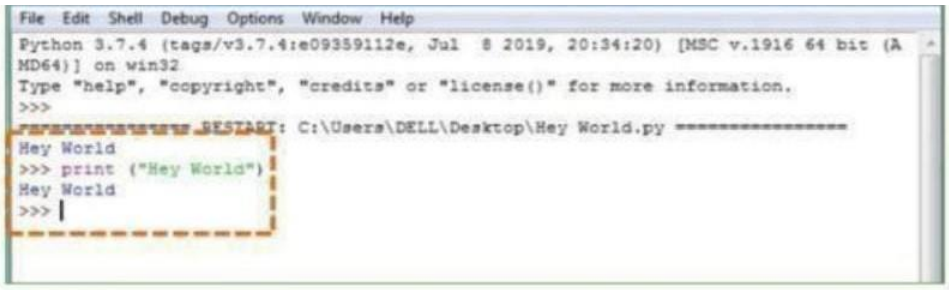
Step3: Click on IDLE(Python3.764-bit)and launch the program

Step4: To go ahead with working in IDLE you must first save the file.ClickonFile> Click on **Save**



Step5: Name the file and saveas type should be Pythonfiles. Clickon SAVE. Here I named the files as Hey World.

Step6: Now fore.g.**enterprint("HeyWorld")**and PressEnter.



You will see that the command given is launched. With this, we end our tutorial on how to install Python. You have learned how to download python for windows into your respective operating system.

Note: Unlike Java, Python doesn‟t needs emicolons at the end of the statements otherwise it won't work.

VS CODE

VS Code is a free source-code editor that supports debugging, syntax highlighting, and more. It's available for Linux, Windows, and Mac OS. VS Code is installed by default under C:\Users\(Username)\AppData\Local\Programs\Microsoft VS Code. You can also download a Zip archive, extract it, and run Code from there. Visual Studio Code is the most popular code editor and the IDEs provided by Microsoft for writing different programs and languages. It allows the users to develop new code bases for their applications and allow them to successfully optimize them. For its high popularity, individuals opt to Install Visual Studio Code on Windows over any other IDE. Installation of Windows Visual Studio Code is not a difficult matter.

Highlights on Visual Studio Code on Windows:

- VS Code is a very user-friendly code editor and it is supported on all the different types of OS.
- It has support for all the languages like C, C++, Java, Python, JavaScript, React, Node JS, etc.
- It is the most popular code editor in India.

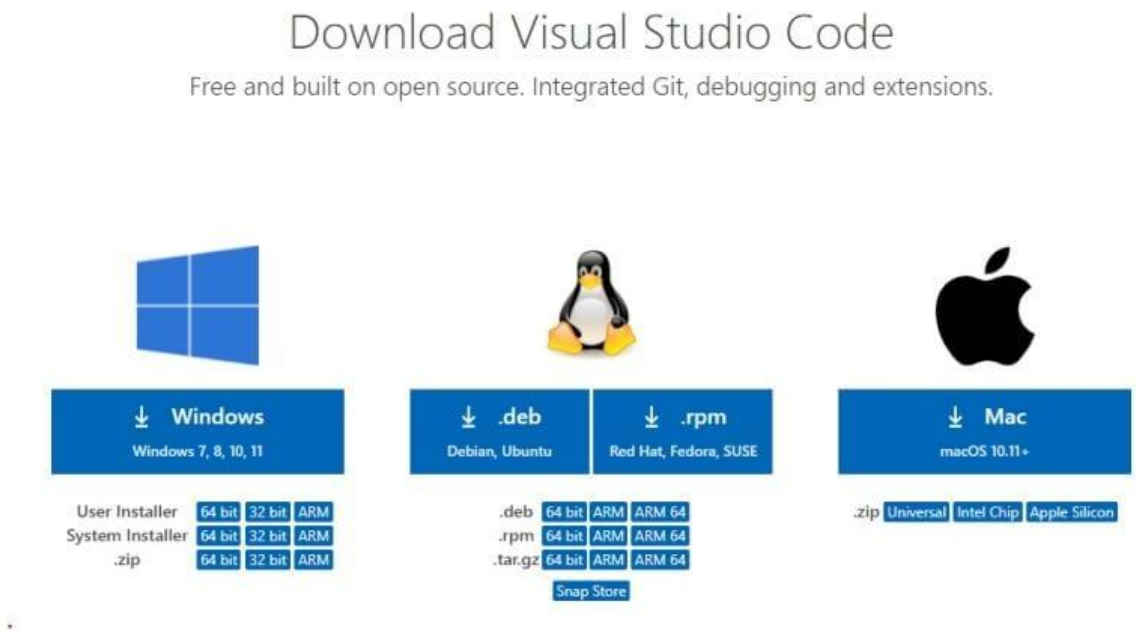
• It allows users with different types of in-app installed extensions.

• It allows the programmers to write the code with ease with the help of these extensions.

• Also, Visual Studio Code has a great vibrant software UI with amazing night mode features. • It suggests auto-complete code to the users which suggests the users complete their code with full ease.

Steps to Install Visual Studio Code on Windows

Step 1: Visit the Official Website of the Visual Studio Code using any web browser like Microsoft Edge, etc



Step 2: Press the "Download for Windows" button on the website to start the download of the Visual Studio Code Application.

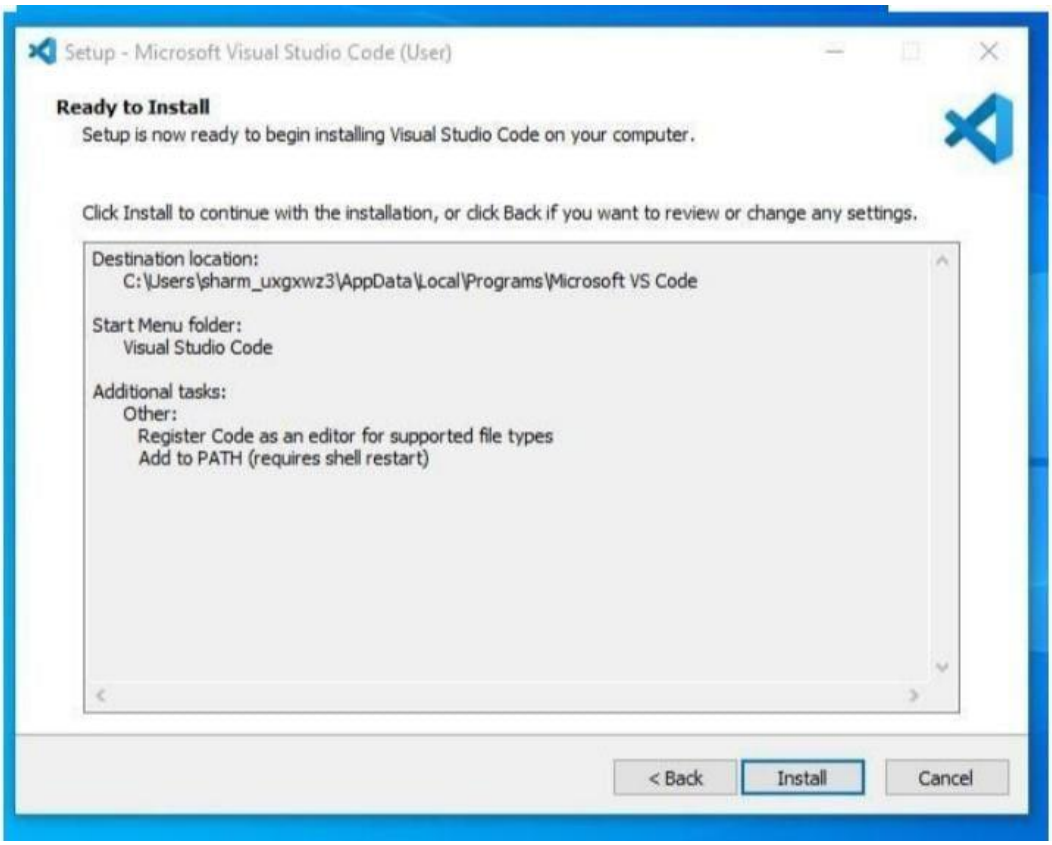
Step3: When the download finishes, then the Visual Studio Code Icon appears in the downloads folder.

Step4: Click on the Installer icon to start the installation process of the Visual Studio Code.

Step 5: After the Installer opens, it will ask you to accept the terms and conditions of the Visual Studio Code. Click on I accept the agreement and then click the Next button.

Step6: Choose the location data for running the Visual Studio Code. It will then ask you to browse the location. Then click on the Next button.

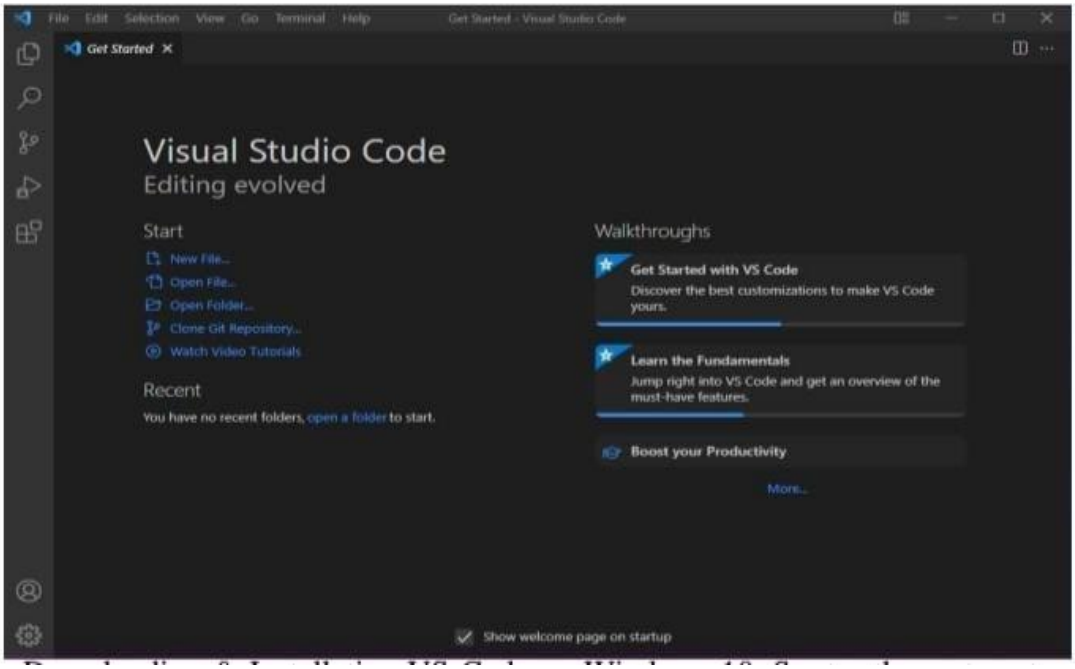
Step7: Then it will ask to begin the installation setup. Click on the Install button.



Step8: After clicking on Install, it will take about1minute to install the Visual Studio Code on your device.

STEP 9: After the Installation setup for Visual Studio Code is finished, it will show a window like this below. Tick the "Launch Visual Studio Code" checkbox and then click Next.

Step 10: After the previous step, the Visual Studio Code window opens successfully. Now you can create a new file in the Visual Studio Code window and choose a language of yours to begin your programming journey! So this is how we successfully installed Visual Studio Code on our Windows system. The steps mentioned in the above guideline can be used in any kind of Windows Browsers.



CHAPTER -9

SAMPLECODE

9.1 SAMPLE CODE ON PYTHON

User side views:

```
# Create your views here.
from django.shortcuts import render, HttpResponse
from django.contrib import messages
from .forms import UserRegistrationForm
from .models import UserRegistrationModel
from django.conf import settings



# Create your views here.
def UserRegisterActions(request):
    if request.method == 'POST':
        form = UserRegistrationForm(request.POST)
        if form.is_valid():
            print('Data is Valid')
            form.save()
            messages.success(request, 'You have been successfully registered')
            form = UserRegistrationForm()
            return render(request, 'UserRegistrations.html', {'form': form})
        else:
            messages.success(request, 'Email or Mobile Already Existed')
            print("Invalid form")
    else:
        form = UserRegistrationForm()
    return render(request, 'UserRegistrations.html', {'form': form})
```

```
def UserLoginCheck(request):
    if request.method == "POST":
        loginid = request.POST.get('loginid')
        pswd = request.POST.get('pswd')
        print("Login ID = ", loginid, ' Password = ', pswd)
        try:
            check = UserRegistrationModel.objects.get(loginid=loginid, password=pswd)
            status = check.status
            print('Status is = ', status)
            if status == "activated":
                request.session['id'] = check.id
                request.session['loggeduser'] = check.name
                request.session['loginid'] = loginid
                request.session['email'] = check.email
                print("User id At", check.id, status)
                return render(request, 'users/UserHomePage.html', {})
            else:
                messages.success(request, 'Your Account Not at activated')
                return render(request, 'UserLogin.html')
        except Exception as e:
            print('Exception is ', str(e))
            pass
        messages.success(request, 'Invalid Login id and password')
    return render(request, 'UserLogin.html', {})


def UserHome(request):
    return render(request, 'users/UserHomePage.html', {})


def DatasetView(request):
```

```
    path = settings.MEDIA_ROOT + "//" + 'DataSet.csv'
    import pandas as pd
    df = pd.read_csv(path, nrows=100,index_col=False)
    df.reset_index()
    df = df.to_html
    return render(request, 'users/viewdataset.html', {'data': df})

def preProcessData(request):
    from .utility.PreprocessedData import preProcessed_data_view
    data = preProcessed_data_view()
    return render(request, 'users/preproccessed_data.html', {'data': data})



def Model_Results(request):
    from .utility import PreprocessedData
    nb_report = PreprocessedData.build_naive_bayes()
    knn_report = PreprocessedData.build_knn()
    dt_report = PreprocessedData.build_decsionTree()
    rf_report = PreprocessedData.build_randomForest()
    svm_report = PreprocessedData.build_svm()
    mlp_report = PreprocessedData.build_mlp()
    return render(request, 'users/ml_reports.html', {'nb': nb_report,"knn":knn_report, 'dt': dt_report,
'rf': rf_report, 'svm': svm_report,'mlp':mlp_report})


def user_input_prediction(request):
    if request.method=='POST':
        from .utility import PreprocessedData
        joninfo  = request.POST.get('joninfo')
        result = PreprocessedData.predict_userInput(joninfo)
        print(request)
        return render(request, 'users/testform.html', {'result': result})
```

```
    else:
        return render(request,'users/testform.html',{})
```

base.html:

```
{%load static%}
<!DOCTYPE html>
<html>
<head>

    <!-- /.website title -->
    <title>Clouds html5 Multipurpose Landing Page for Apps</title>
    <meta name="viewport" content="width=device-width, initial-scale=1, maximum-scale=1, user-scalable=no">

    <!-- CSS Files -->
    <link href="{%static 'css/bootstrap.min.css'%}" rel="stylesheet" media="screen">
    <link href="{%static 'css/font-awesome.min.css'%}" rel="stylesheet">
    <link href="{%static 'fonts/icon-7-stroke/css/pe-icon-7-stroke.css'%}" rel="stylesheet">
    <link href="{%static 'css/animate.css'%}" rel="stylesheet" media="screen">
    <link href="{%static 'css/owl.theme.css'%}" rel="stylesheet">
    <link href="{%static 'css/owl.carousel.css'%}" rel="stylesheet">

    <link href="{%static 'css/styles.css'%}" rel="stylesheet" media="screen">

    <!-- Google Fonts -->
    <link href='http://fonts.googleapis.com/css?family=Open+Sans:400,300,600,700' rel='stylesheet' type='text/css'>
    <link href='http://fonts.googleapis.com/css?family=Alegreya+Sans:100,300,400,700' rel='stylesheet' type='text/css'>
```

```
    <!-- Font Awesome -->
    <link  href="http://maxcdn.bootstrapcdn.com/font-awesome/4.2.0/css/font-awesome.min.css" rel="stylesheet">
</head>

<body data-spy="scroll" data-target="#navbar-scroll">

<div id="top"></div>

<!-- NAVIGATION -->
<div id="menu">
    <nav class="navbar-wrapper navbar-default" role="navigation">
        <div class="container">
            <div class="navbar-header">
                <button type="button" class="navbar-toggle" data-toggle="collapse" data-target=".navbar-themers">
                    <span class="sr-only">Toggle navigation</span>
                    <span class="icon-bar"></span>
                    <span class="icon-bar"></span>
                    <span class="icon-bar"></span>
                </button>
                <a class="navbar-brand site-name" href="#top" style="COlor:WHITE"><h2>Fake Job Posting</h2></a>
            </div>

            <div id="navbar-scroll" class="collapse navbar-collapse navbar-themers navbar-right">
                <ul class="nav navbar-nav">
                    <li><a href="{%url 'index'%}">Home</a></li>
                    <li><a href="{%url 'UserLogin'%}">ML Users</a></li>
                    <li><a href="{%url 'AdminLogin'%}">Admin</a></li>
                    <li><a href="{%url 'UserRegister'%}">Registrations</a></li>
```

```
            </ul>
          </div>
        </div>
    </nav>
</div>

{%block contents%}
{%endblock%}
<!-- /.footer -->
<footer id="footer">
    <div class="container">
        <div class="col-sm-4 col-sm-offset-4">
            <!-- /.social links -->

            <div class="text-center wow fadeInUp" style="font-size: 14px;">Copyright Alex Corporations Template by <a
                    href="#">Alex Hales</a></div>
            <a href="#" class="scrollToTop"><i class="fa fa-arrow-circle-o-up"></i></a>
        </div>
    </div>
</footer>

<!-- /.javascript files -->
<script src="{%static 'js/jquery.js'%}"></script>
<script src="{%static 'js/bootstrap.min.js'%}"></script>
<script src="{%static 'js/custom.js'%}"></script>
<script src="{%static 'js/jquery.sticky.js'%}"></script>
<script src="{%static 'js/wow.min.js'%}"></script>
<script src="{%static 'js/owl.carousel.min.js'%}"></script>
<script src="{%static 'js/ekko-lightbox-min.js'%}"></script>
<script type="text/javascript">
```

```
    $(document).delegate('*[data-toggle="lightbox"]', 'click', function (event) {
        event.preventDefault();
        $(this).ekkoLightbox();
    });
</script>
<script>
    new WOW().init();
</script>
</body>
</html>
```

Index.html:

```
{%extends 'base.html'%}
{%load static%}
{%block contents%}
<!-- /.parallax full screen background image -->
<div class="fullscreen landing parallax banner" style="background-image:url('{%static 'images/bg.jpg'%}');"
     data-img-width="2000" data-img-height="1325" data-diff="100">

    <div class="overlay">
        <div class="container">
            <div class="row">


                <div class="col-md-6">

                    <!-- /.main title -->
                    <h1 class="wow fadeInLeft">
                        Common types of Job Scam
                    </h1>
```

```
            <!-- /.header paragraph -->
            <div class="landing-text wow fadeInLeft">
              <p>
                Fraudsters who want to gain other people's personal information like insurance details, bank
                details, income tax details, date of birth, national id create fake job advertisements.
                Advance fee scams occur when frauds ask for money showing reasons like admin charges,
                information security checking cost, management cost etc. Sometimes fraudsters act them-
                selves as employers and ask people about passport details, bank statements, driving license
                etc. as pre-employment check. Illegal money mulling scams occur when they convince students
                to pay money into their accounts and then transfer it back.

              </p>
            </div>

          </div>

          <!-- /.phone image -->
          <div class="col-md-6">
            <img src="{%static 'images/header-phone.png'%}" alt="phone"
              class="header-phone img-responsive wow fadeInRight">
          </div>
        </div>
      </div>
    </div>
</div>
```

```
<!-- /.feature section -->
<div id="feature">
   <div class="container">
      <div class="row">
         <div class="col-md-10 col-md-offset-1 col-sm-12 text-center feature-title">

            <!-- /.feature title -->
            <h2>Related Works</h2>
            <p>
               Many researches occurred to predict if a job post is real or
               fake. A good number of research works are to check online
               fraud job advertiser. Vidros [1] et al. identified job scammers
               as fake online job advertiser. They found statistics about many
               real and renowned companies and enterprises who produced
               fake job advertisements or vacancy posts with ill-motive. They
               experimented on EMSCAD dataset using several classification
               algorithms like naive bayes classifier, random forest classifier,Zero R, One R etc.
Random Forest
               Classifier showed the
               best performance on the dataset with 89.5% classification
               accuracy
            </p>
         </div>
      </div>

   </div>
</div>

{%endblock%}
```

Admin side views:

```
from django.shortcuts import render, HttpResponse
```

```
from django.contrib import messages
from users.models import UserRegistrationModel

# Create your views here.
def AdminLoginCheck(request):
    if request.method == 'POST':
        usrid = request.POST.get('loginid')
        pswd = request.POST.get('pswd')
        print("User ID is = ", usrid)
        if usrid == 'admin' and pswd == 'admin':
            return render(request, 'admins/AdminHome.html')

        else:
            messages.success(request, 'Please Check Your Login Details')
    return render(request, 'AdminLogin.html', {})


def AdminHome(request):
    return render(request, 'admins/AdminHome.html')


def RegisterUsersView(request):
    data = UserRegistrationModel.objects.all()
    return render(request,'admins/viewregisterusers.html',{'data':data})


def ActivaUsers(request):
    if request.method == 'GET':
        id = request.GET.get('uid')
        status = 'activated'
        print("PID = ", id, status)
```

```
    UserRegistrationModel.objects.filter(id=id).update(status=status)
    data = UserRegistrationModel.objects.all()
    return render(request,'admins/viewregisterusers.html',{'data':data})


def AdminCartResults(request):
    from users.utility.ProcessCart import start_process_cart
    rslt_dict = start_process_cart()
    return render(request, "admins/admincartresults.html", rslt_dict)


def AdminGBDTResults(request):
    from users.utility.ProcessCart import start_process_gbdt
    rslt_dict = start_process_gbdt()
    return render(request, "admins/admingbdtresults.html", rslt_dict)


def classification_report(request):
    from users.utility import PreprocessedData
    nb_report = PreprocessedData.build_naive_bayes()
    knn_report = PreprocessedData.build_knn()
    dt_report = PreprocessedData.build_decsionTree()
    rf_report = PreprocessedData.build_randomForest()
    svm_report = PreprocessedData.build_svm()
    mlp_report = PreprocessedData.build_mlp()
    return render(request, 'admins/reports.html',
                  {'nb': nb_report, "knn": knn_report, 'dt': dt_report, 'rf': rf_report, 'svm': svm_report,
                   'mlp': mlp_report})
```

CHAPTER-10

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.

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.

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.

CHAPTER -11

TEST CASES

S.no	Test Case	Excepted Result	Result	Remarks(IF Fails)
1	User Register	If User registration successfully.	Pass	If already user email exist then it fails.
2	User Login	If Username and password is correct then it will getting valid page.	Pass	Un Register Users will not logged in.
3	Random forest and svm	The request will be accepted by the random forest and svm	Pass	The request will be accepted by the random forest and svm otherwise its failed
4	Decision Tree and multilayer perceptron	The request will be accepted by the Decision Tree and multilayer perceptron	Pass	The request will be accepted by the Decision Tree and multilayer perceptron otherwise its failed
5	Naive Bayes and k-nearest neighbour	The request will be accepted by the Naive Bayes and k-nearest neighbour	Pass	The request will be accepted by the Naive Bayes and k-nearest neighbour otherwise its failed

6	View dataset by user	Data set will be displayed by the user	Pass	Results not true failed
7	User classification	Display reviews with true results	Pass	Results not true failed
8	Calculate accuracy macro avg and weighted avg	macro avg and weighted avg calculated	Pass	macro avg and weighted avg not displayed failed
9	Admin login	Admin can login with his login credential. If success he get his home page	Pass	Invalid login details will not allowed here
10	Admin can activate the register users	Admin can activate the register user id	Pass	If user id not found then it won't login.

CHAPTER -12

IMPLEMENTATION 12.1 IMPLEMENTATION PROCESS

MODULES:

- User
- Admin
- Data Preprocessing
- Machine Learning

MODULES DESCRIPTION:

User:

The User can register the first. While registering he required a valid user email and mobile for further communications. Once the user register then admin can activate the user. Once admin activated the user then user can login into our system. User can upload the dataset based on our dataset column matched. For algorithm execution data must be in float format. Here we took Employment Scam Aegean Dataset (EMSCAD) containing 18000 sample dataset. User can also add the new data for existing dataset based on our Django application. User can click the Classification in the web page so that the data calculated Accuracy and macro avg, weighted avg based on the algorithms. User can display the ml results. user can also display the prediction results.

Admin:

Admin can login with his login details. Admin can activate the registered users. Once he activate then only the user can login into our system. Admin can view the overall data in the browser. Admin can click the Results in the web page so calculated Accuracy and macro avg, weighted avg based on the algorithms is displayed. All algorithms execution complete then admin can see the overall accuracy in web page. And also display the classification results.

Data Preprocessing:

They worked on this dataset in three steps- data pre-processing, feature selection and fraud detection using classifier. In the preprocessing step, they removed noise and html tags from the data so that the general text pattern remained preserved. They applied feature selection technique to reduce the number of attributes effectively and efficiently. Support Vector Machine was used for feature selection and ensemble classifier using random forest was used to detect fake job posts

from the test data. Random forest classifier seemed a tree structured classifier which worked as ensemble classifier with the help of majority voting technique. This classifier showed 97.4% classification accuracy to detect fake job posts.

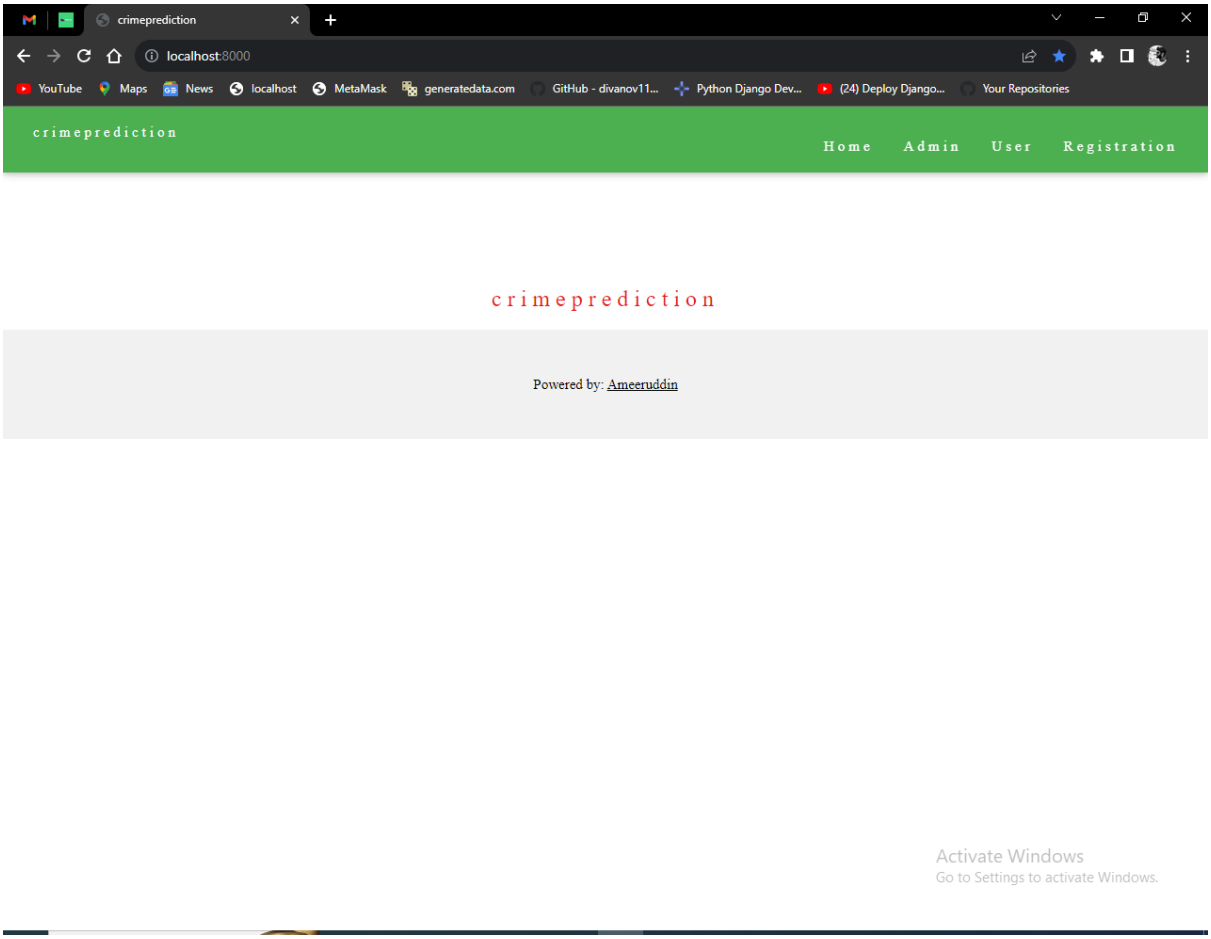
Machine learning:

This paper proposed to use different data mining techniques and classification algorithm like KNN, decision tree, support vector machine, naïve bayes classifier, random forest classifier, multilayer perceptron and deep neural network to predict a job post if it is real or fraudulent. The Accuracy and macro avg weighted avg of the classifiers was calculated and displayed in my results. The classifier which bags up the highest accuracy could be determined as the best classifier.

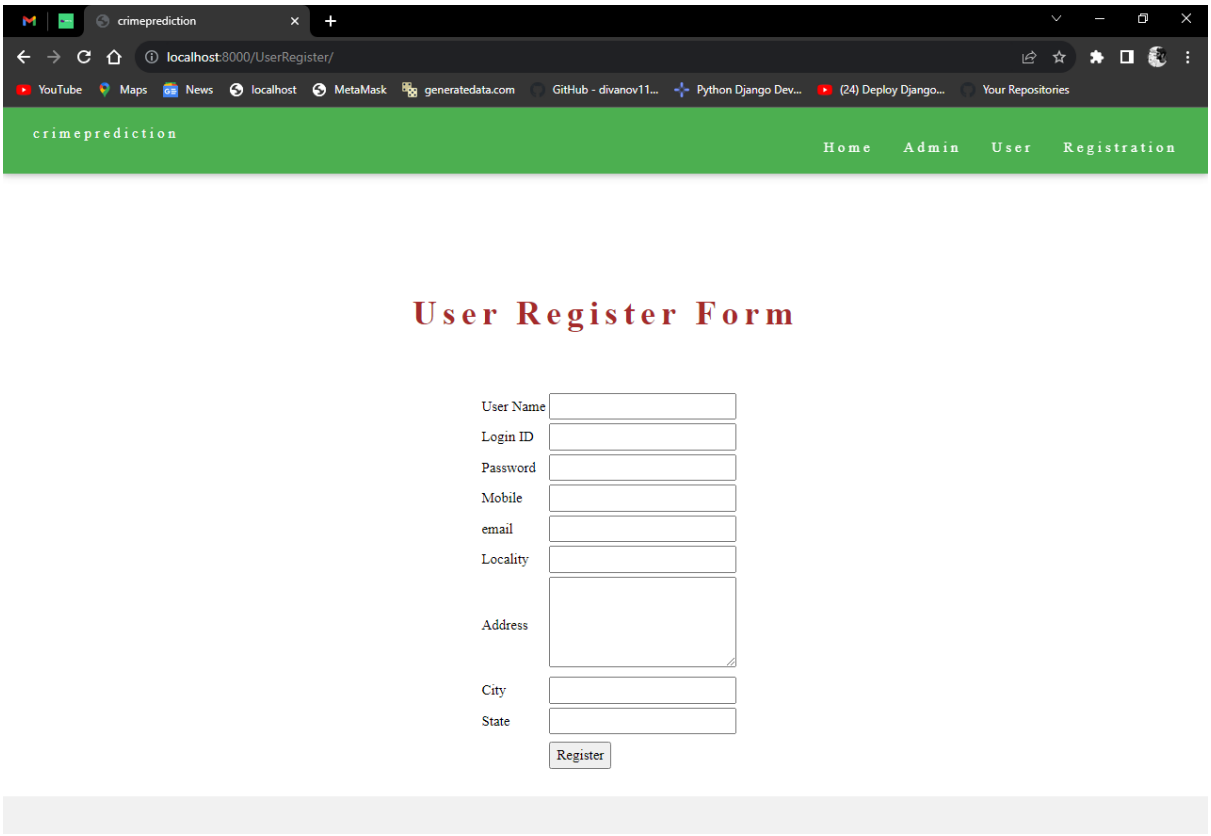
CHAPTER -13

SCREEN SHOTS

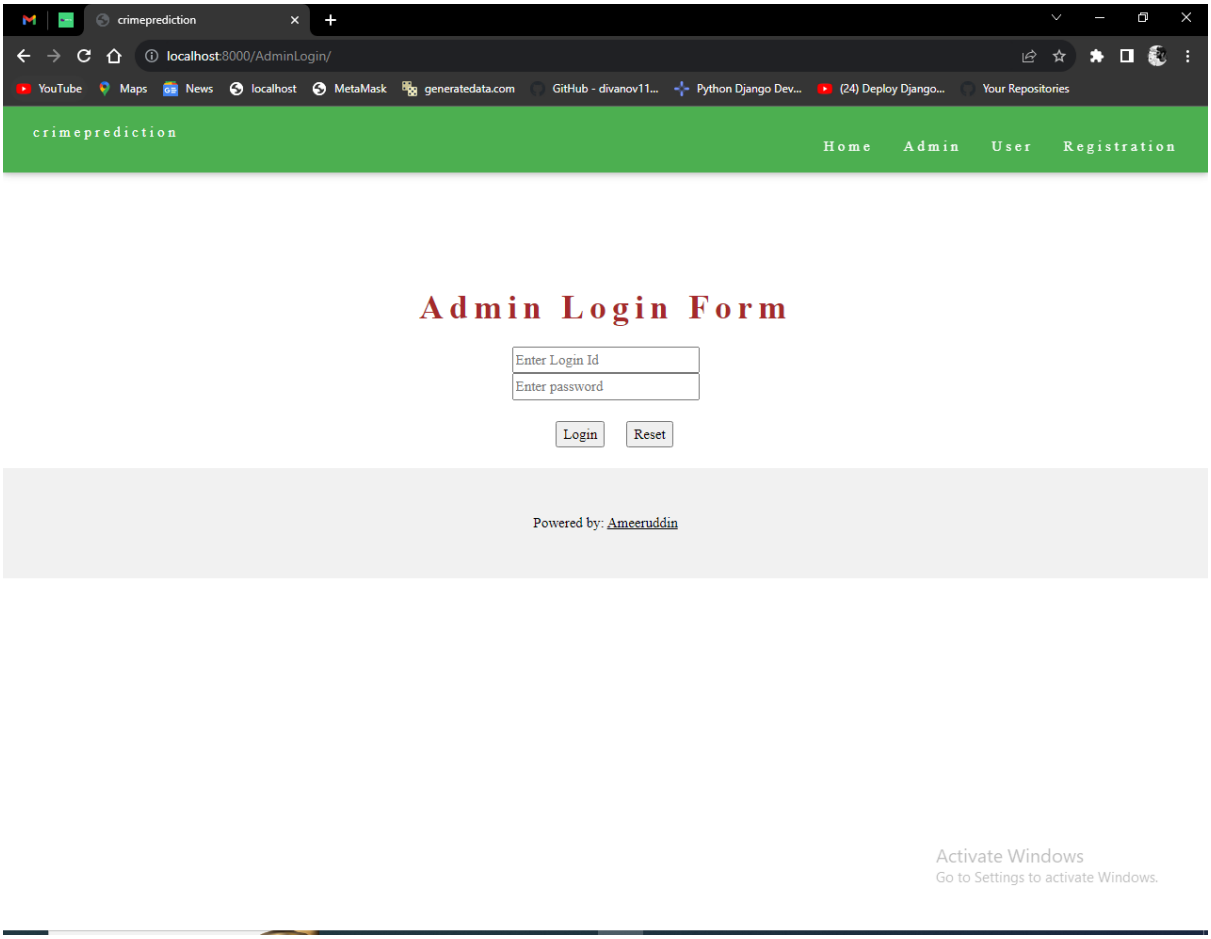
Home Page



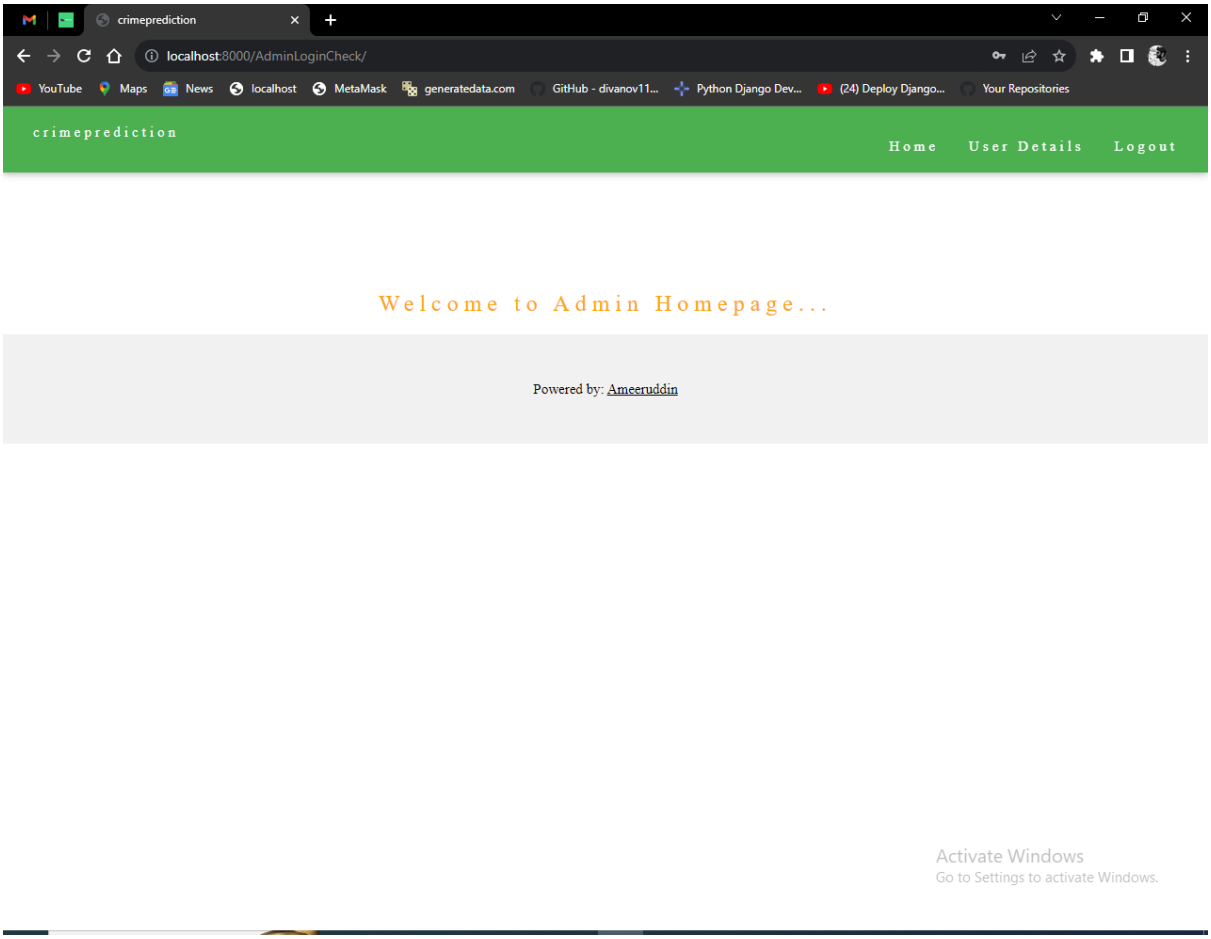
User Registration



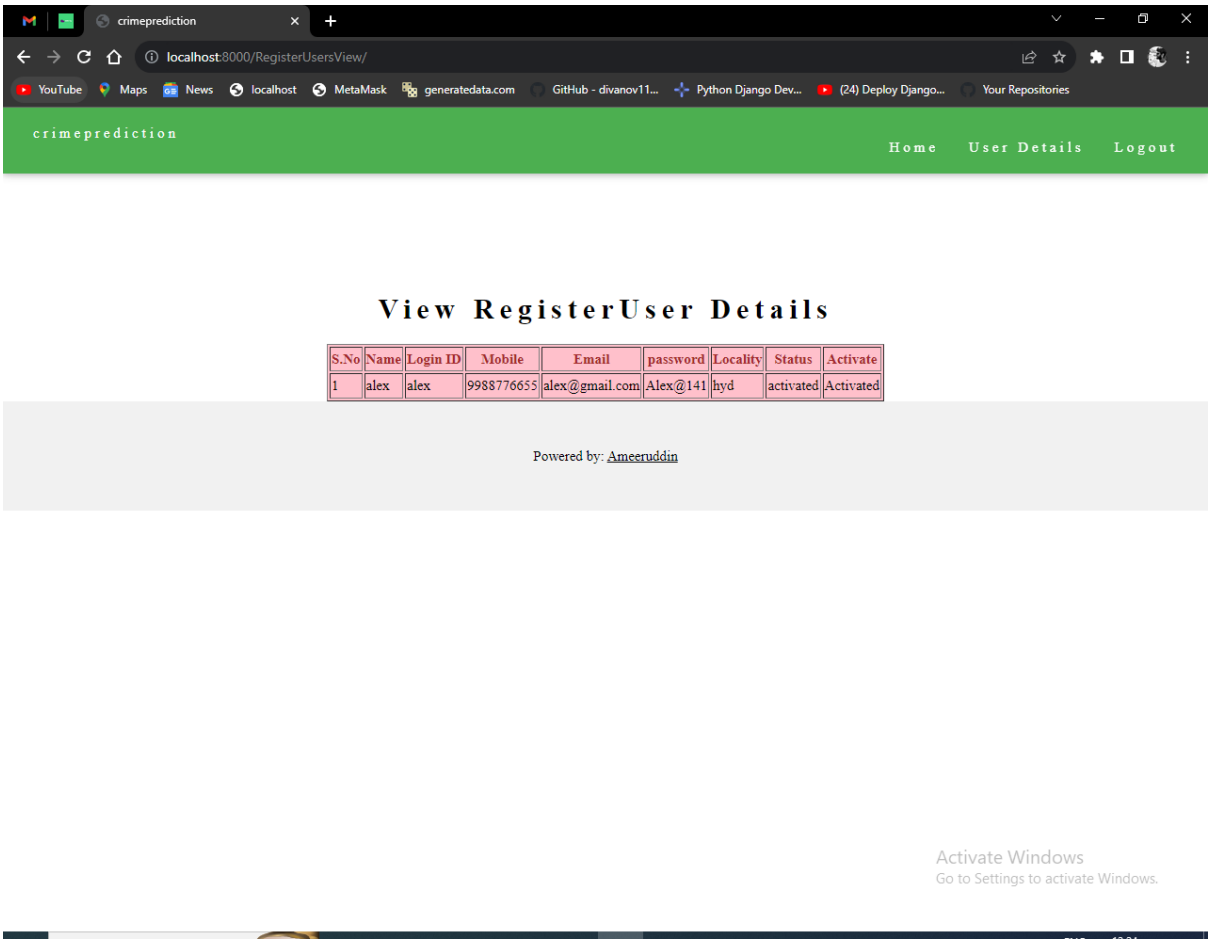
Admin Login


crimeprediction
localhost:8000/AdminLogin/
YouTube
Maps
News
localhost
MetaMask
generatedata.com
GitHub - divanov11...
Python Django Dev...
(24) Deploy Django...
Your Repositories
crimeprediction
Home
Admin
User
Registration
Admin Login Form
Enter Login Id
Enter password
Login
Reset
Powered by: Ameeruddin
Activate Windows
Go to Settings to activate Windows.

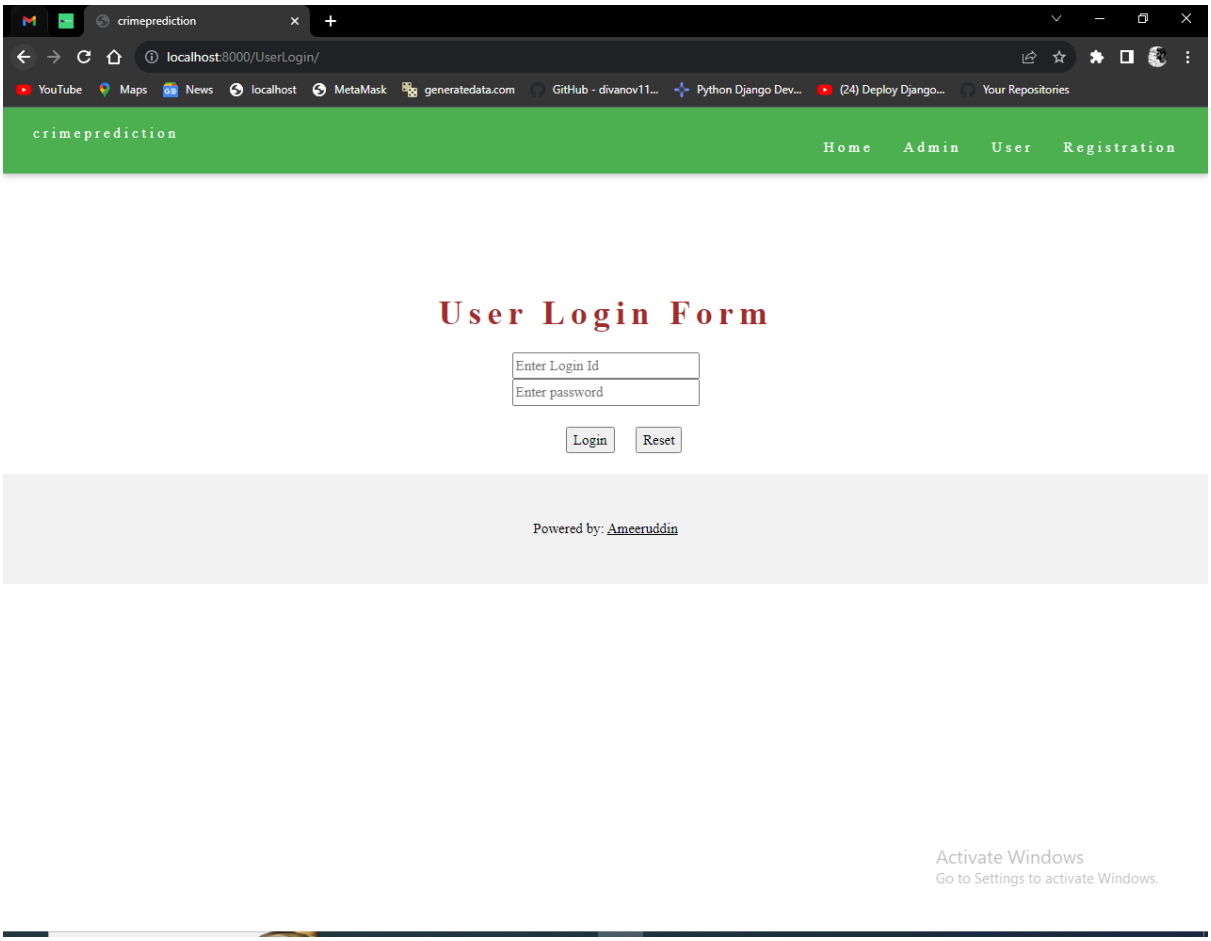
Admin Home


crimeprediction
localhost:8000/AdminLoginCheck/
YouTube
Maps
News
localhost
MetaMask
generatedata.com
GitHub - divanov11...
Python Django Dev...
(24) Deploy Django...
Your Repositories
crimeprediction
Home
User Details
Logout
Welcome to Admin Homepage...
Powered by: Ameeruddin
Activate Windows
Go to Settings to activate Windows.

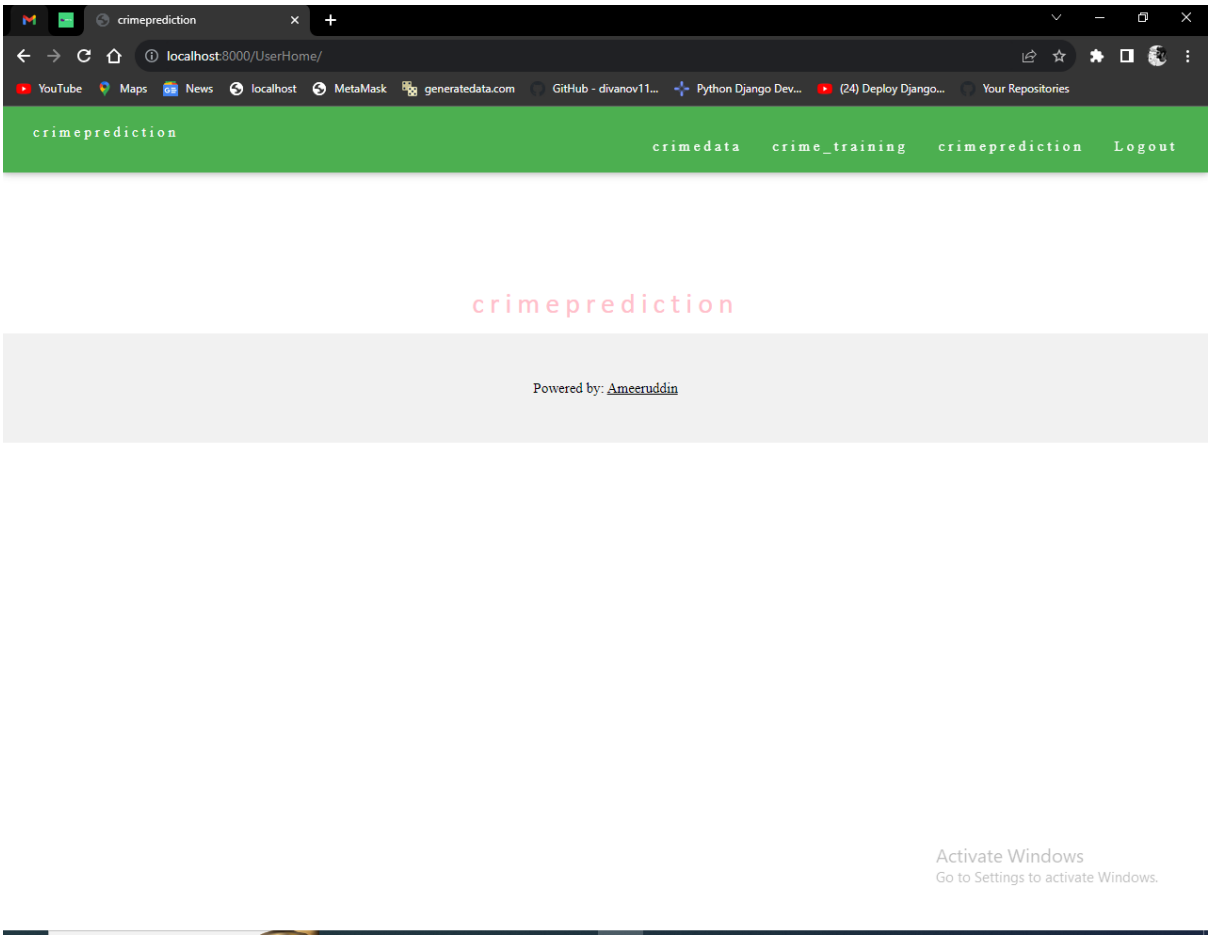
Admin Activate Users



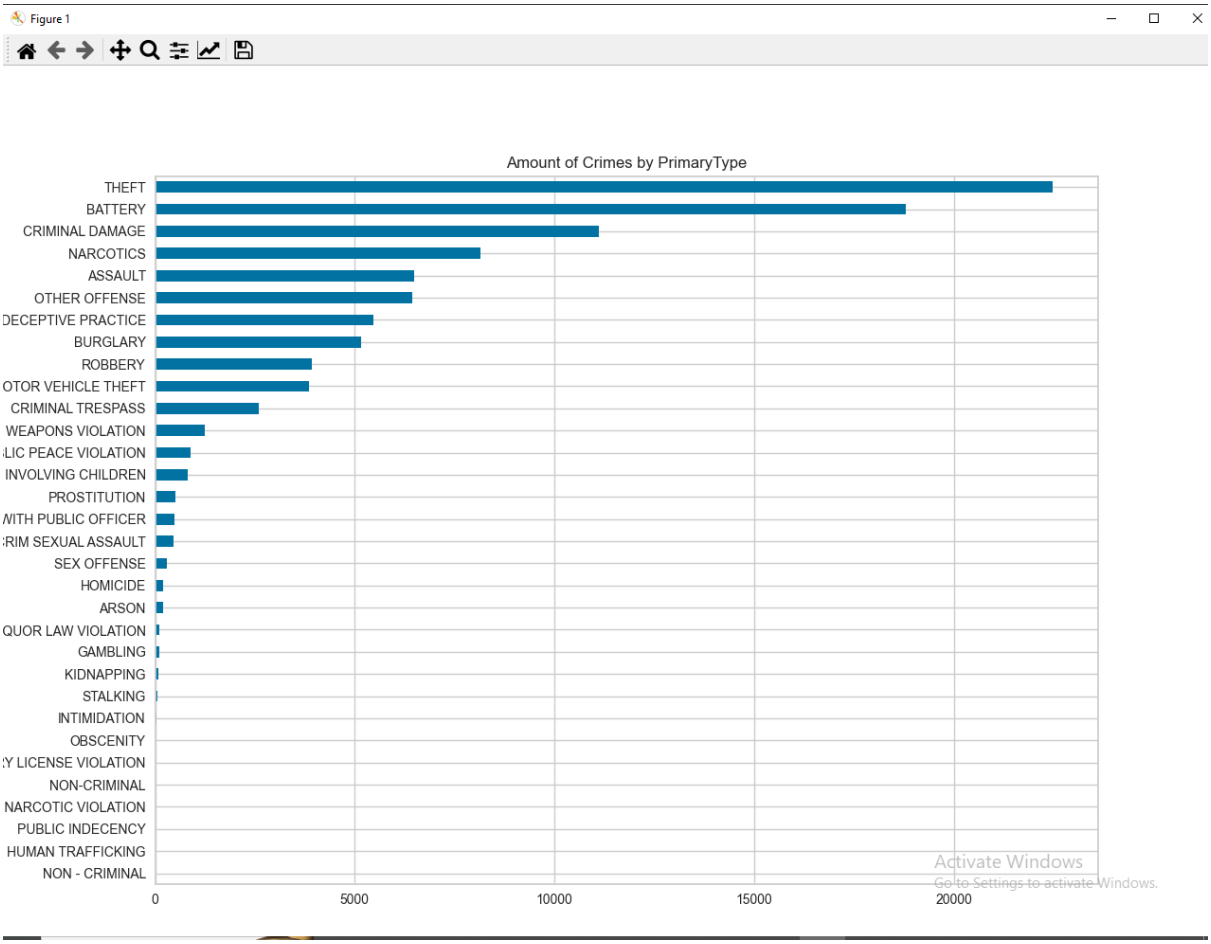
User Login


crimeprediction
localhost:8000/UserLogin/
YouTube
Maps
News
localhost
MetaMask
generatedata.com
GitHub - divanov11...
Python Django Dev...
(24) Deploy Django...
Your Repositories
crimeprediction
Home
Admin
User
Registration
User Login Form
Enter Login Id
Enter password
Login
Reset
Powered by: Ameeruddin
Activate Windows
Go to Settings to activate Windows.

User Home


crimeprediction
localhost:8000/UserHome/
YouTube
Maps
News
localhost
MetaMask
generatedata.com
GitHub - divanov11...
Python Django Dev...
(24) Deploy Django...
Your Repositories
crimeprediction
crimedata
crime_training
crimeprediction
Logout
crimeprediction
Powered by: Ameeruddin
Activate Windows
Go to Settings to activate Windows.

User Training



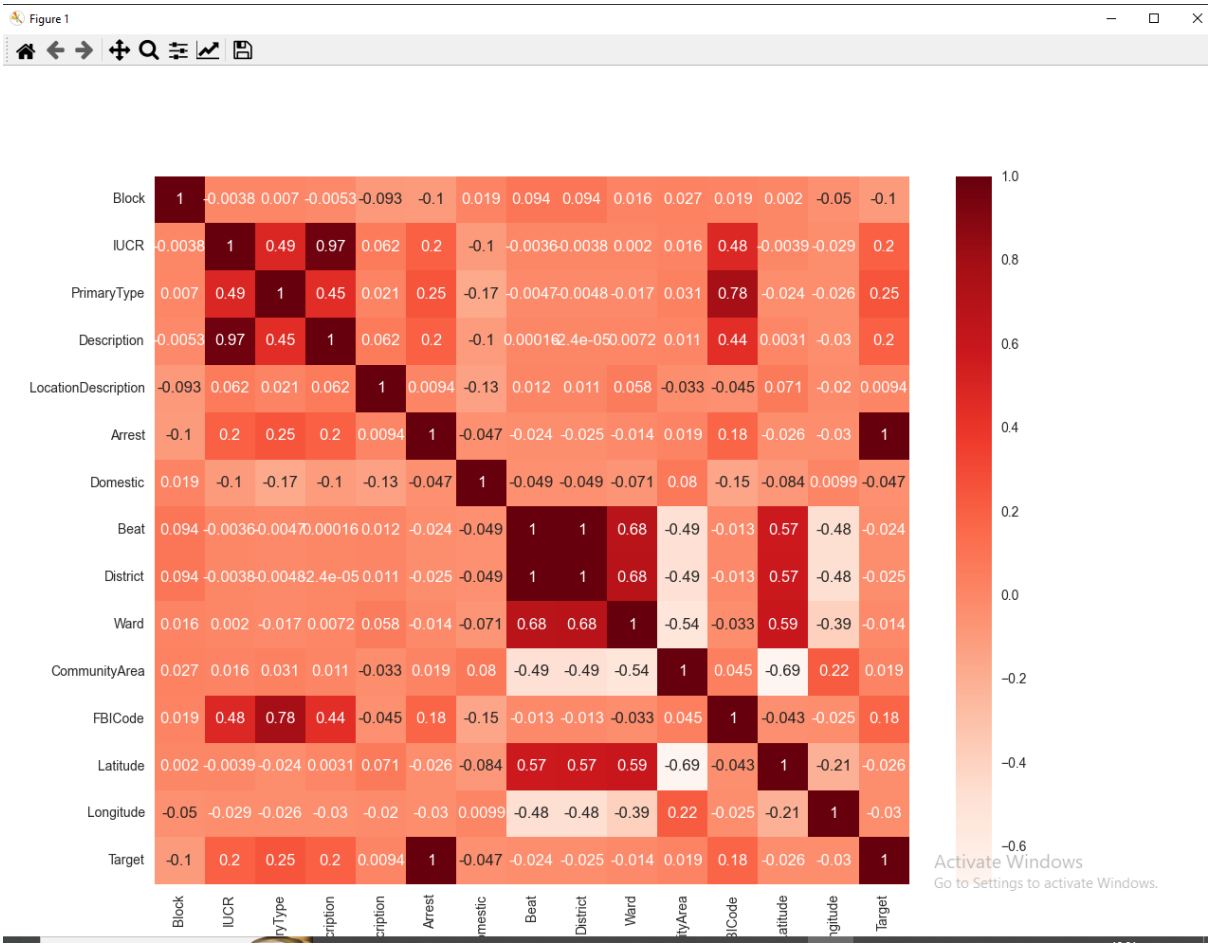
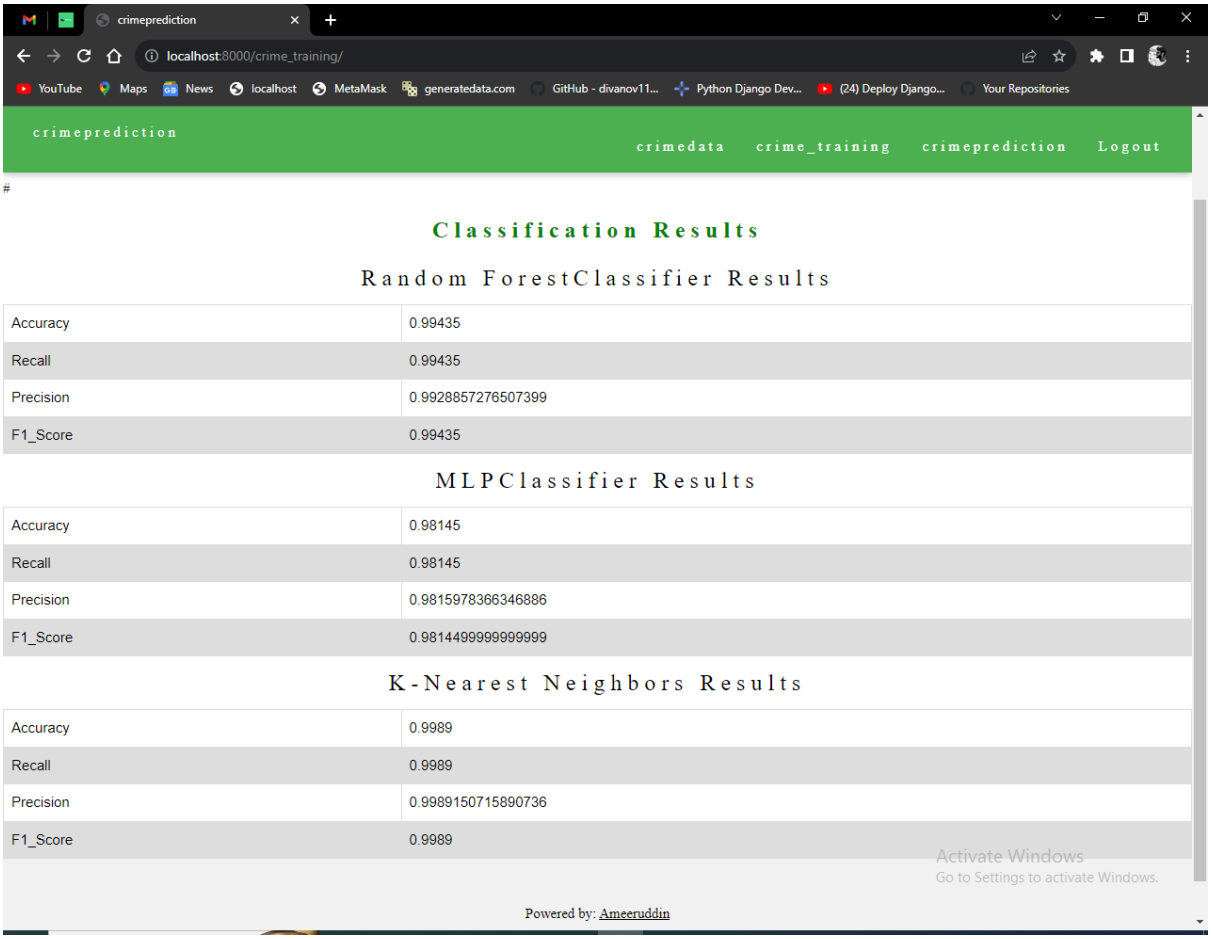
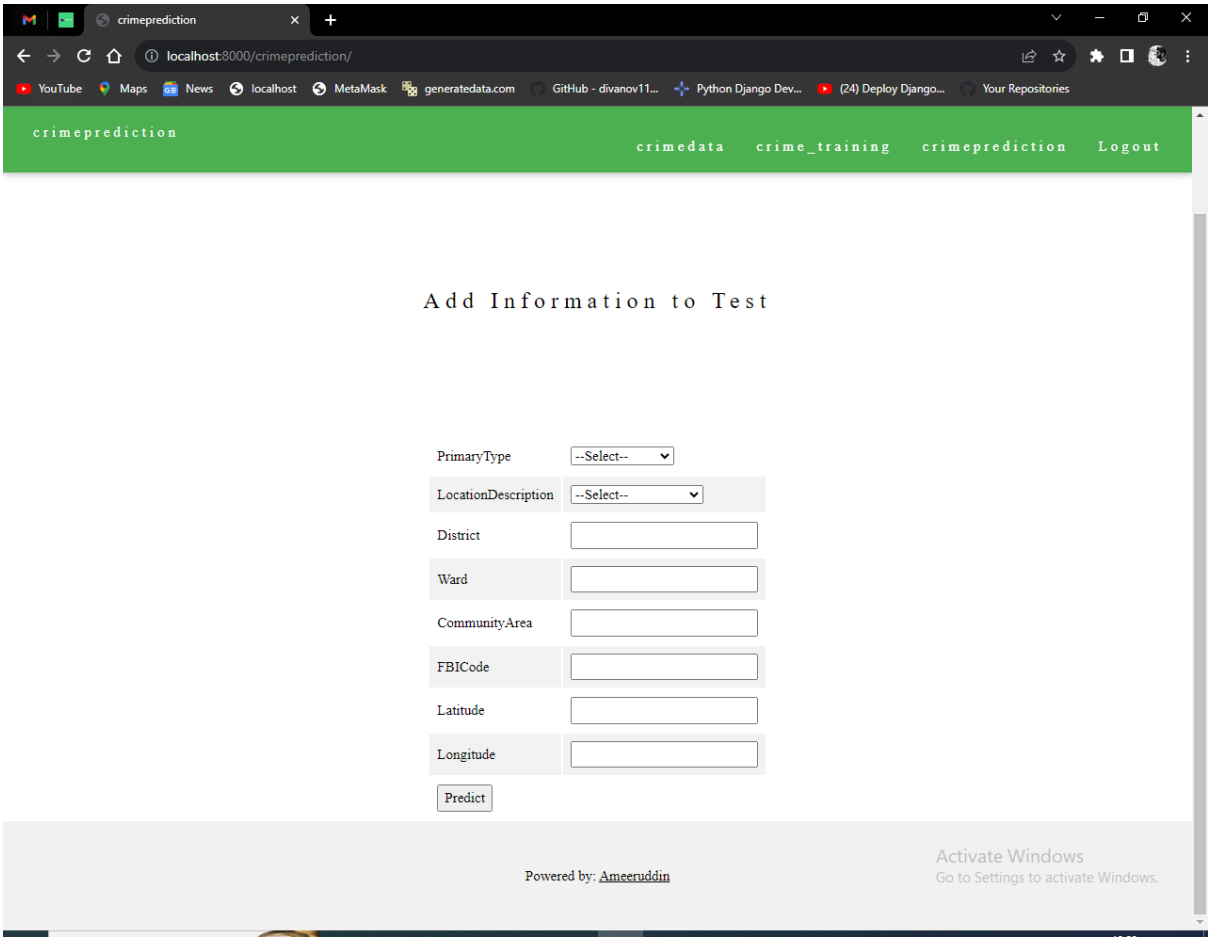


Figure 1
Block 1 -0.0038 0.007 -0.0053 -0.093 -0.1 0.019 0.094 0.094 0.016 0.027 0.019 0.002 -0.05 -0.1
IUCR -0.0038 1 0.49 0.97 0.062 0.2 -0.1 -0.0036 -0.0038 0.002 0.016 0.48 -0.0039 -0.029 0.2
PrimaryType 0.007 0.49 1 0.45 0.021 0.25 -0.17 -0.0047 -0.0048 -0.017 0.031 0.78 -0.024 -0.026 0.25
Description -0.0053 0.97 0.45 1 0.062 0.2 -0.1 0.00016 2.4e-05 0.0072 0.011 0.44 0.0031 -0.03 0.2
LocationDescription -0.093 0.062 0.021 0.062 1 0.0094 -0.13 0.012 0.011 0.058 -0.033 -0.045 0.071 -0.02 0.0094
Arrest -0.1 0.2 0.25 0.2 0.0094 1 -0.047 -0.024 -0.025 -0.014 0.019 0.18 -0.026 -0.03 1
Domestic 0.019 -0.1 -0.17 -0.1 -0.13 -0.047 1 -0.049 -0.049 -0.071 0.08 -0.15 -0.084 0.0099 -0.047
Beat 0.094 -0.0036 -0.0047 0.00016 0.012 -0.024 -0.049 1 1 0.68 -0.49 -0.013 0.57 -0.48 -0.024
District 0.094 -0.0038 -0.0048 2.4e-05 0.011 -0.025 -0.049 1 1 0.68 -0.49 -0.013 0.57 -0.48 -0.025
Ward 0.016 0.002 -0.017 0.0072 0.058 -0.014 -0.071 0.68 0.68 1 -0.54 -0.033 0.59 -0.39 -0.014
CommunityArea 0.027 0.016 0.031 0.011 -0.033 0.019 0.08 -0.49 -0.49 -0.54 1 0.045 -0.69 0.22 0.019
FBICode 0.019 0.48 0.78 0.44 -0.045 0.18 -0.15 -0.013 -0.013 -0.033 0.045 1 -0.043 -0.025 0.18
Latitude 0.002 -0.0039 -0.024 0.0031 0.071 -0.026 -0.084 0.57 0.57 0.59 -0.69 -0.043 1 -0.21 -0.026
Longitude -0.05 -0.029 -0.026 -0.03 -0.02 -0.03 0.0099 -0.48 -0.48 -0.39 0.22 -0.025 -0.21 1 -0.03
Target -0.1 0.2 0.25 0.2 0.0094 1 -0.047 -0.024 -0.025 -0.014 0.019 0.18 -0.026 -0.03 1
Block IUCR ryType cription cription Arrest mestic Beat District Ward hyArea BICode atitude ngitude Target
1.0 0.8 0.6 0.4 0.2 0.0 −0.2 −0.4 −0.6
Activate Windows
Go to Settings to activate Windows.

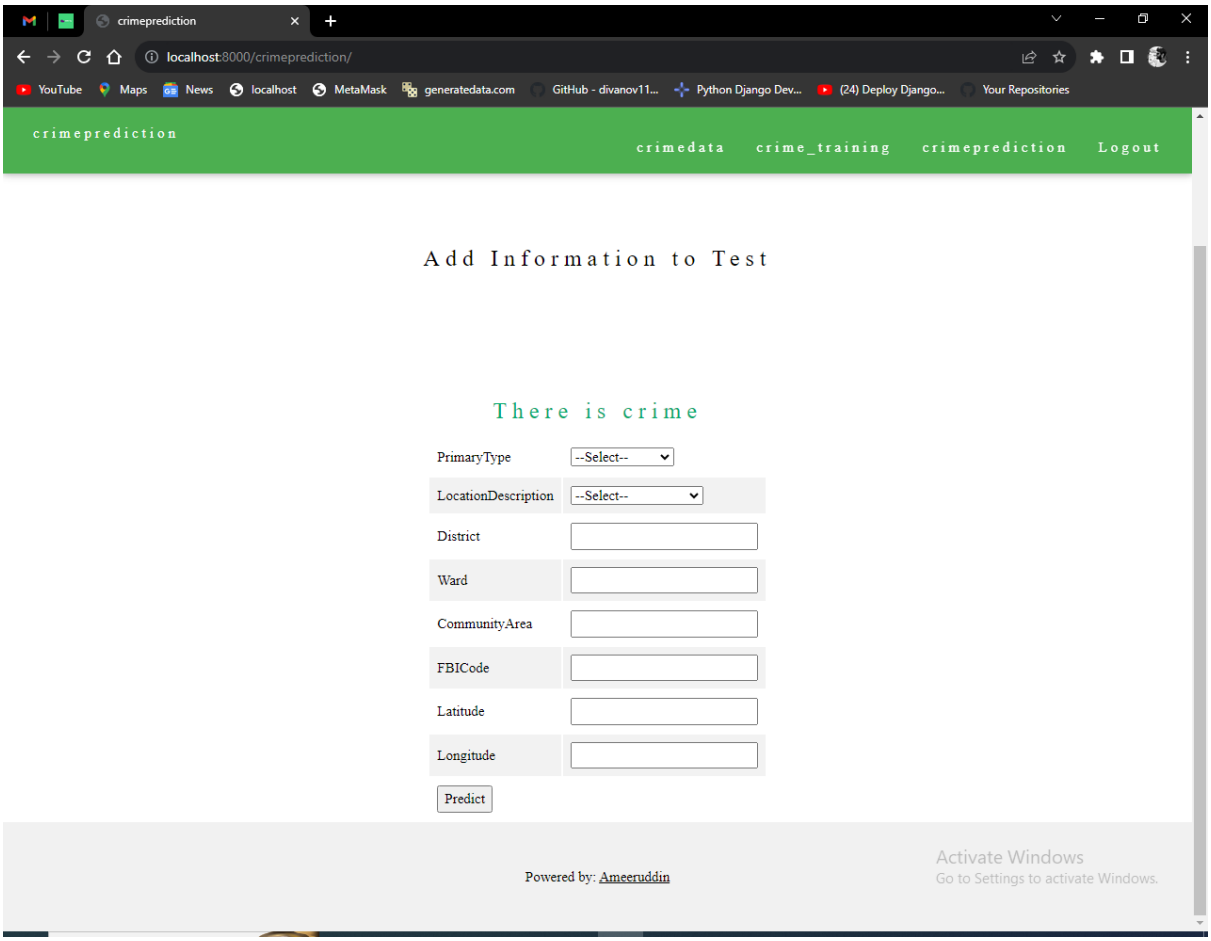
Training Results



Crime Prediction Form



Crime Prediction Result



CHAPTER -14

CONCLUSION & FUTURE ENHANCEMENT

In this work, we have introduced a crime prediction and evaluation framework for machine learning algorithms of network edge. We collected data from 2012 to 2019 to analyze and evaluate our forecast. We used machine learning approaches to anticipate crime events, which can make a significant contribution to improving city public safety, which is a big problem in many cities across the world. It was fascinating to see how pre-processing, and transformation may affect the model's output, particularly when breaking the day into many time periods. Due to the provenance of the data, this solution was created for a specific city in the country. However, if equivalent data is made accessible, the technique may be applied to other cities. Based on the training set input for the four algorithms, we find the Decision tree method to be extremely successful and accurate in predicting crime data. The Decision Stump algorithm's poor performance could be attributed to a certain amount of randomness in the various crimes and associated features (shows a low correlation coefficient among the four algorithms); the KNN's branches are more rigid and only give accurate results if the test set follows the pattern modelled.

Some general ideas for potential future enhancements in the domain of crime prediction using edge-assisted machine learning frameworks. These suggestions are based on trends and advancements in the field up to September 2021:

1. **Data Fusion and Integration**: Enhance the framework by incorporating diverse data sources beyond traditional crime data, such as social media activity, weather patterns, traffic data, and demographic information. Integrating these sources can provide a more comprehensive understanding of crime patterns and potential risk factors.
2. **Real-Time Prediction and Response**: Focus on making the prediction framework more responsive by utilizing edge computing to process and analyze data in real-time. This can enable law enforcement agencies to respond quickly to emerging crime trends.
3. **Privacy-Preserving Techniques**: Consider implementing privacy-preserving techniques when handling sensitive data. Differential privacy, federated learning, and homomorphic encryption can help protect individuals' privacy while still extracting meaningful insights.
4. **Multi-Modal Data Analysis**: Explore the integration of different data modalities, such as images, videos, and audio, to capture a broader range of contextual information for crime prediction and evaluation.
5. **Interpretable AI Models**: Develop models that provide explanations for their predictions, making it easier for law enforcement and decision-makers to understand the factors contributing to predictions and ensuring transparency in the decision-making process.
6. **Adaptive Learning Algorithms**: Implement machine learning algorithms that can adapt and self-improve over time as new data becomes available. This could involve techniques such as online learning or reinforcement learning.

7. **Human-Centric Design**: Involve law enforcement professionals and other stakeholders in the design process to ensure the framework aligns with their needs and is practical for real-world implementation.
8. **Spatiotemporal Analysis**: Enhance the prediction framework by incorporating spatiotemporal analysis techniques, allowing for the identification of crime hotspots and trends that evolve over time.
9. **Collaborative Edge Networks**: Create a collaborative network of edge devices that can share insights and predictions across different locations, improving the overall accuracy of the system.
10. **Long-Term Impact Assessment**: Develop methods to assess the long-term impact of the implemented framework on crime prevention and reduction, and continuously refine the algorithms based on these assessments.
11. **Ethical Considerations**: Address potential biases and fairness issues in the predictive models to ensure that the framework is equitable and doesn't disproportionately target certain groups or areas.
12. **Scalability and Resource Efficiency**: Optimize the edge-assisted framework to be resource-efficient and scalable, especially when dealing with a large volume of data from various sources.

It's important to note that the enhancements you choose should be guided by a deep understanding of the problem domain, the limitations of the existing work, and the potential benefits for law enforcement agencies and communities. Collaboration with experts in machine learning, crime prevention, and law enforcement can provide valuable insights for refining and implementing these enhancements effectively.

CHAPTER-15

REFERENCES:

[1] "68% of the world population projected to live in urban areas by 2050, says UN — UN DESA — United Nations Department of Economic and Social Affairs." https://www.un.org/development/desa/en/news/population/2018- revision-of-world-urbanization-prospects.html (accessed Oct. 14, 2021).

[2] Y. Wu, W. Zhang, J. Shen, Z. Mo, and Y. Peng, "Smart city with Chinese characteristics against the background of big data: Idea, action and risk," Journal of Cleaner Production, vol. 173, pp. 60–66, Feb. 2018.

[3] M. Kowsher, A. Tahabilder, and S. A. Murad, "Impact-learning: A robust machine learning algorithm," in ACM International Conference Proceeding Series, pp. 9–13, Jul. 2020.

[4] Priyanka and D. Kumar, "Decision tree classifier: A detailed survey," International Journal of Information and Decision Sciences, vol. 12, no. 3, pp. 246–269, 2020.

[5] L. Jiang, Z. Cai, D. Wang, and S. Jiang, "Survey of improving Knearest-neighbour for classification," Proceedings - Fourth International Conference on Fuzzy Systems and Knowledge Discovery, FSKD 2007, vol. 1, pp. 679–683, 2007.

[6] L. McClendon and N. Meghanathan, "Using Machine Learning Algorithms to Analyse Crime Data," Machine Learning and Applications: An International Journal, vol. 2, no. 1, pp. 1–12, Mar. 2015.

[7] Fonseca, Luis, F. C. Pinto, and S. Sargento. "An Application for Risk of Crime Prediction Using Machine Learning." International Journal of Computer and Systems Engineering 15.2, pp. 166-174, 2021.

[8] E. Ahishakiye, D. Taremwa, E. O. Omulo, and I. Niyonzima, "Crime Prediction Using Decision Tree (J48) Classification Algorithm," 2017. [Online]. Available: www.ijcit.com188

[9] S. K. Senthil Kumar, G. Adarsh, J. Shashank, and A. Sameer, "CRIME PREDICTION AND ANALYSIS USING MACHINE LEARNING", [Online]. Available: http://ijte.uk/

[10] S. Kim, P. Joshi, P. S. Kalsi, and P. Taheri, "Crime Analysis Through Machine Learning," in 2018 IEEE 9th Annual Information Technology, Electronics and Mobile Communication Conference, IEMCON 2018, pp. 415–420, Jan. 2019