

CARDIOVASCULAR STROKE PREDICTION USING MACHINE LEARNING TECHNIQUES

**A Project report submitted to
DEPARTMENT OF COMPUTER SCIENCE AND
ENGINEERING WITH ARTIFICIAL INTELLIGENCE**

In partial fulfillment of the requirements for the award of the Degree of

**BACHELOR OF TECHNOLOGY
IN
COMPUTER SCIENCE AND ENGINEERING
WITH ARTIFICIAL INTELLIGENCE**



By

CHIGURUPATI BHARGAVI	(Y20CAI013)
MAKKENA VASAVI	(Y20CAI032)
VALLALA BHARGAVI	(Y20CAI062)
P.K.V.SAI RAGHAVENDRA	(Y20CAI049)

Under the Guidance of

Dr.A.BALAJI , Ph.D

ASSOCIATE PROFESSOR

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
WITH ARTIFICIAL INTELLIGENCE**

**CHALAPATHI INSTITUTE OF ENGINEERING AND
TECHNOLOGY (AUTONOMOUS)**

**(Accredited by NAAC with 'A' grade, accredited by NBA,
Approved by A.I.C.T.E, Affiliated To Acharya Nagarjuna University)**

Guntur-522034

2023-2024

**CHALAPATHI INSTITUTE OF ENGINEERING
AND TECHNOLOGY (AUTONOMOUS)**

**(Accredited by NAAC with ‘A’ grade, accredited by NBA, Approved
by A.I.C.T.E Affiliated to Acharya Nagarjuna University)**

CHALAPATHI NAGAR, LAM, GUNTUR

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING WITH
ARTIFICIAL INTELLIGENCE**



CERTIFICATE

This is to certify that the project work entitled as “**Cardiovascular Stroke Prediction using Machine Learning Techniques**” submitted by **Ch.Bhargavi, M.Vasavi, V.Bhargavi and P. Kumara Venkata Sai Raghavendra** in partial fulfillment for the award of the Degree **BACHELOR OF TECHNOLOGY IN COMPUTER SCIENCE AND ENGINEERING WITH ARTIFICIAL INTELLIGENCE** is a record of bonafied work carried out under my guidance and supervision.

PROJECT GUIDE

Dr. A.Balaji, Ph.D.
Associate Professor

HEAD OF THE DEPARTMENT

Dr. V. N.Gopiraju, Ph.D.
CSAI-HOD

EXTERNAL EXAMINAR

DECLARATION

We hereby declare that the project entitled, "**CARDIOVASCULAR STROKE PREDICTION USING MACHINE LEARNING TECHNIQUES**" submitted in partial fulfillment of the requirements for the award of bachelor of technology in computer science and engineering with artificial intelligence. to Chalapathi Institute of Engineering and Technology (CIET), permanently affiliated to **ACHARYA NAGARJUNA UNIVERSITY (ANU), GUNTUR** in an authentic work and has not been submitted to university or institute for the award of the degree.

DATE:

PLACE:

SIGNATURE OF STUDENTS

CHIGURUPATI BHARGAVI	Y20CAI013
MAKKENA VASAVI	Y20CAI032
VALLALA BHARGAVI	Y20CAI062
P.K.V. SAI RAGHAVENDRA	Y20CAI049

ACKNOWLEDGEMENT

We express our sincere thanks to our beloved Chairman **Sri.Y.V. ANJANEYULU** for providing support and stimulating environment for developing the project.

We express deep sense of reverence and profound gratitude to **Dr. M. CHANDRA SEKHAR, Ph. D**, Principal for providing us the great support in carrying out the project.

It plunges us in exhilaration in taking privilege in expressing our heart-felt gratitude to **Dr. V. NAGA GOPI RAJU ,Ph. D**, HOD-CSAI for providing us every facility and for constant supervision.

We are thankful to our guide **Dr.A.BALAJI ,M.Tech, Ph. D**, Associate professor, Dept. of CSE 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 team mates for their valuable Co-operation.

By

CHIGURUPATI BHARGAVI	(Y20CAI013)
MAKKENA VASAVI	(Y20CAI032)
VALLALA BHARGAVI	(Y20CAI062)
P.K.V.SAI RAGHAVENDRA	(Y20CAI049)

ABSTRACT

As the human population increases, so is the chance of getting diseases. There are many illnesses globally, and one of the biggest problems faced by the hospital systems today is the lack of technology to know when the patients are ill. One such illness is Cardiovascular Disease or CVD. It refers to any heart disease, vascular disease, or blood vessel disease. According to WHO, more people die of CVD's worldwide than any other cause. It affects the low and middle-income countries more.

It is very hard for people living alone to contact the hospital when they are sick. Therefore, we have developed a model that can detect when a patient is ill and report back to the hospital. The system currently only identifies patients with heart disease and reports back to the hospital. We decided to go with heart disease identification because it is one of the most deadly diseases, and the risk of patients dying because of heart disease is high. Predicting whether a patient has heart disease or not is very clearly a classification problem. Therefore, we have used five models to classify. We take several factors such as blood sugar level, age, cholesterol level, and many more and give the outcome based on the input.

KEYWORDS:

Cardio Vascular Disease (CVD), Machine Learning, Deep Learning, WHO (World Health Organization).

LIST OF CONTENTS

S.NO	TABLE OF CONTENTS	PAGE NO
1	ACKNOWLEDGEMENT	IV
2	ABSTARCT	V
3	LIST OF FIGURES	VIII
4	ABBREVIATIONS	IX
5	CHAPTER-1: INTRODUCTION	1
	1. INTRODUCTION	2
	1.1 BACKGROUND	2
	1.2 PROBLEM DEFINITION	2
6	CHAPTER-2: LITERATURE SURVEY	3
	2.1 LITERATURE SURVEY	4
7	CHAPTER-3: SYSTEM ANALYSIS	5
	3. SYSTEM ALAYSIS	6
	3.1 EXISTING SYSTEM	6
	3.1.1 DISADVANTAGES	6
	3.2 PROPOSED SYSTEM	6
	3.2.1 ADVANTAGES	6
	3.3 SYSTEM REQUIREMENTS	7
	3.3.1 HARDWARE REQUIREMENTS	7
	3.3.2 SOFTWARE REQUIREMENTS	7
	3.4 SYSTEM STUDY	7
	3.4.1 FEASIBILITY STUDY	7
8	CHAPTER-4: SYSTEM ARCHITECTURE	10
	4.SYSTEM DESIGN	11
	4.1 SYSTEM ARCHITECTURE	11
	4.2 DATA FLOW DIAGRAMS	11
	4.3 UML DIAGRAMS	13
	4.3.1 USE CASE DIAGRAM	14
	4.3.2 CLASS DIAGRAM	15
	4.3.3 OBJECT DIAGRAM	16
	4.3.4 STATE DIAGRAM	17
	4.3.5 ACTIVITY DIAGRAM	18
	4.3.6 SEQUENCE DIAGRAM	19
	4.4 IMPLEMENTATION	19
	4.4.1 MODULES	19
	4.4.2 MODULES DEVELOPMENT	20

9	CHAPTER-5 SOFTWARE ENVIRONMENT	21
	5.1 PYTHON	22
	5.1.1 WHAT IS PYTHON?	23
	5.1.2 ADVANTAGES OF PYTHON	25
	5.1.3 ADVANTAGES OF PYTHON OVER OTHER LANGUAGES	26
	5.1.4 DISADVANTAGES OF PYTHON	27
	5.1.5 HISTORY OF PYTHON	28
	5.1.6 WHAT IS MACHINE LEARNING?	29
	5.1.7 CATEGORIES OF MACHINE LEARNING	29
	5.1.8 NEED FOR MACHINE LEARNING	30
	5.1.9 CHALLENGES IN MACHINE LEARNING	31
	5.1.10 APPLICATIONS OF MACHINE LEARNING	32
	5.2 HOW TO START LEARNING MACHINE LEARNING?	32
	5.2.1 HOW TO START LEARNING ML?	32
	5.2.2 LEARN VARIOUS CONCEPTS	34
	5.2.3 ADVANTAGES OF ML	36
	5.2.4 DISADVANTAGES OF ML	37
	5.3 PYTHON DEVELOPMENT STEPS	38
	5.3.1 HOW TO INSTALL PYTHON?	39
	5.3.2 MODULES USED IN PYTHON	40
	5.3.3 INSTALL PYTHON	43
10	CHAPTER-6: SYSTEM TESTING	51
	6.1 TYPES OF TESTING	52
	6.1.1 UNIT TESTING	52
	6.1.2 INTEGRATION TESTING	52
	6.1.3 FUNCTION TESTING	53
	6.1.4 SYSTEM TESTING	54
	6.1.5 WHITE BOX TESTING	54
	6.1.6 BLACK BOX TESTING	54
	6.2 TEST STRATEGY AND APPROACH	55
	6.3 SOURCE CODES	56
	6.3.1 PYTHON SOURCE CODE	56
	6.3.2 HTML SOURCE CODE	58
11	CHAPTER-7: SCREENSHOTS OF OUTPUT	64
	7.1 SCREENSHOTS	65
12	CHAPTER-8: CONCLUSION	68
	8.1 CONCLUSION	69
13	CHAPTER-9: REFERENCES	70
	9.1 REFERENECS	71
14	CHAPTER-10: FUTURE SCOPE	73
	10.1 FUTURE SCOPE	74

LIST OF FIGURES

S. No	Figure. No	CONTENT	Page. No
1	FIG: 4.1	SYSTEM ARCHITECTURE FOR CARDIO VASCULAR PREDICTION	11
2	FIG: 4.2	DATA FLOW DIAGRAM FOR CARDIO VASCULAR PREDICTION	11
3	FIG: 4.3	UNIFIED MODEL LANGUAGES	13
4	FIG: 4.3.1	USE CASE DIAGRAMS	14
5	FIG: 4.3.2	CLASS DIAGRAM	15
6	FIG: 4.3.3	OBJECT DIGARM	16
7	FIG: 4.3.4	STATE DIAGRAM	17
8	FIG: 4.3.5	ACTIVITY DIAGRAM	18
9	FIG: 4.3.6	SEQUENCE DIAGRAM	19

ABBREVIATIONS

CVD	CARDIO VASCULAR DISEASE
WHO	WORLD HEALTH ORGANIZATION
ML	MACHINE LEARNING
DL	DEEP LEARNING
LR	LOGISTIC REGRESSION
KNN	K-NEAREST NEIGHBOUR
SVM	SUPPORT VECTOR MACHINE
DT	DECISION TREE
RF	RANDOM FOREST

CHAPTER-1

INTRODUCTION

1. INTRODUCTION

1.1 BACKGROUND WORK

In a world grappling with escalating health challenges, particularly the pervasive threat of Cardiovascular Disease (CVD), our innovative model addresses a critical gap in patient care. By harnessing advanced technology to detect signs of heart disease, we empower individuals, especially those in underserved regions, to receive timely medical attention, potentially saving countless lives.

PROBLEM DEFINITION

The escalating global population amplifies the risk of diseases, with Cardiovascular Disease (CVD) standing as a leading cause of mortality, especially in low and middle-income countries. Hospital systems grapple with a lack of proactive technology to identify and respond promptly to patient ailments, particularly for those living alone. To address this, we've focused on developing a model specifically targeting heart disease detection, utilizing key health indicators to enable timely intervention and reduce the alarming mortality rates associated with CVD.

OBJECTIVE OF PROJECT

The primary objective of our project is to leverage advanced classification models to detect and predict Cardiovascular Disease (CVD) in individuals. By integrating crucial health parameters such as blood sugar levels, age, cholesterol, and more, our aim is to create a robust system that proactively identifies patients at risk. The ultimate goal is to provide a timely and automated alert mechanism to hospitals, facilitating prompt medical intervention and significantly reducing the mortality rate associated with heart diseases.

CHAPTER-2

LITERATURE SURVEY

2. LITERATURE SURVEY

Fahd Saleh Alotaibi (2019) has created a machine learning model comparing five various machine learning algorithms [1]. Author has deeply studied on the accuracy of Naive Bayes, LR, SVM, Decision Tree and Random Forest classification algorithms were compared. The highest accuracy was by the Decision Tree.

Theresa Princy. R, et al. (2016) have surveyed different models used for predicting heart disease. The classification techniques used by Theresa were Naive Bayes, Neural network, KNN) LR, Decision tree. The accuracy score was compared with all the models and the comparison was made efficiently [2].

Nagaraj M Lutimath, et al.,(2020) research on heart disease prediction using SVM and Naive Bayes classification technique. The way he went by the analysis are Sum of Squared Error, RMSE and Mean Absolute Error, it is established that the SVM emerged as the best classification technique in terms of accuracy.[3].

Anjan Nikhil Repaka, et al.,(2020) has proposed a machine learning model for predicting heart disease in [4] that uses techniques like Naive Bayesian for classification of the dataset and AES [8] algorithm for prediction of disease.

CHAPTER-3

SYSTEM ANALYSIS

3. SYSTEM ANALYSIS

3.1EXISTING SYSTEM:

Utilizing Support Vector Machines (SVM), our existing system excels in binary classification for Cardiovascular Disease (CVD) detection. Trained on diverse datasets, SVM effectively identifies patients at risk based on key health parameters. Continuous refinement ensures a robust and accurate model for timely intervention, crucial in combating heart diseases.

3.1.1Disadvantages

1. Support Vector Machines (SVM) can be computationally intensive.
2. Sensitive to parameter tuning
3. Less effective in handling large datasets.

3.1.1 Proposed system

The proposed system leverages Decision Trees (DT), Random Forest (RF), and Naive Bayes algorithms for Cardiovascular Disease (CVD) detection. This ensemble approach ensures enhanced accuracy, robustness, and versatility in handling diverse datasets, improving early identification and intervention.

3.2.1Advantages

1. The proposed system excels in accuracy with deep learning,
Accommodates complex data patterns,
2. Enhances early Cardiovascular Disease (CVD) detection.

3.3 SYSTEM REQUIREMENTS:

3.3.1 HARDWARE REQUIREMENTS:

- System : Pentium IV 2.4 GHz.
- Hard Disk : 40 GB.
- Floppy Drive : 1.44 Mb.
- Monitor : 15 VGA Color.
- Mouse : Logitech.
- Ram : 512 Mb.

3.3.2 SOFTWARE REQUIREMENTS:

- **Operating System:** Windows
- **Coding Language:** Python 3.7

3.4 SYSTEM STUDY

3.4.1 FEASIBILITY STUDY

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

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 all development of the system is limited. The expenditure must be justified. Thus they all 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

SOCIAL FEASIBILITY

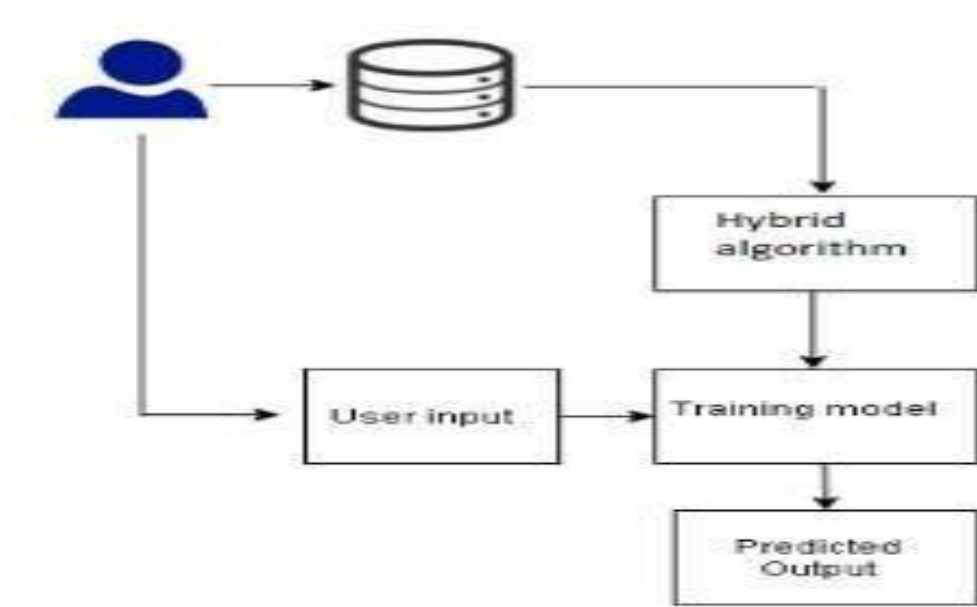
The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends 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 ARCHITECTURE

4. SYSTEM DESIGN

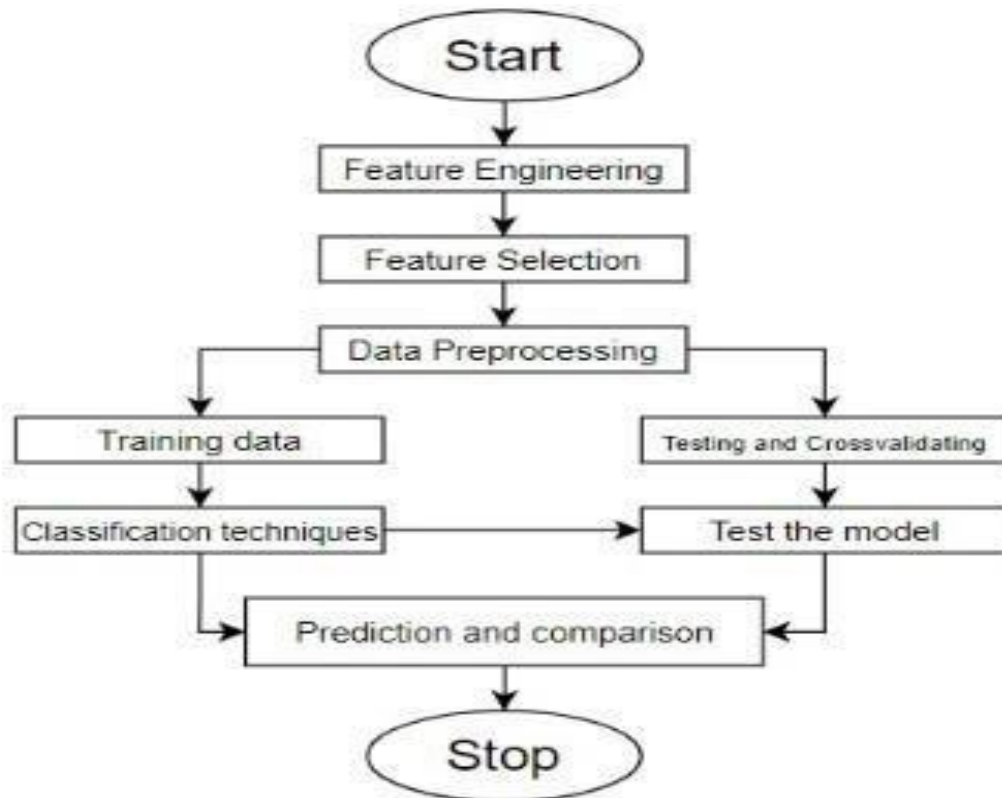
4.1 SYSTEM ARCHITECTURE:



4.2 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.



4.1 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 ObjectManagement 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.

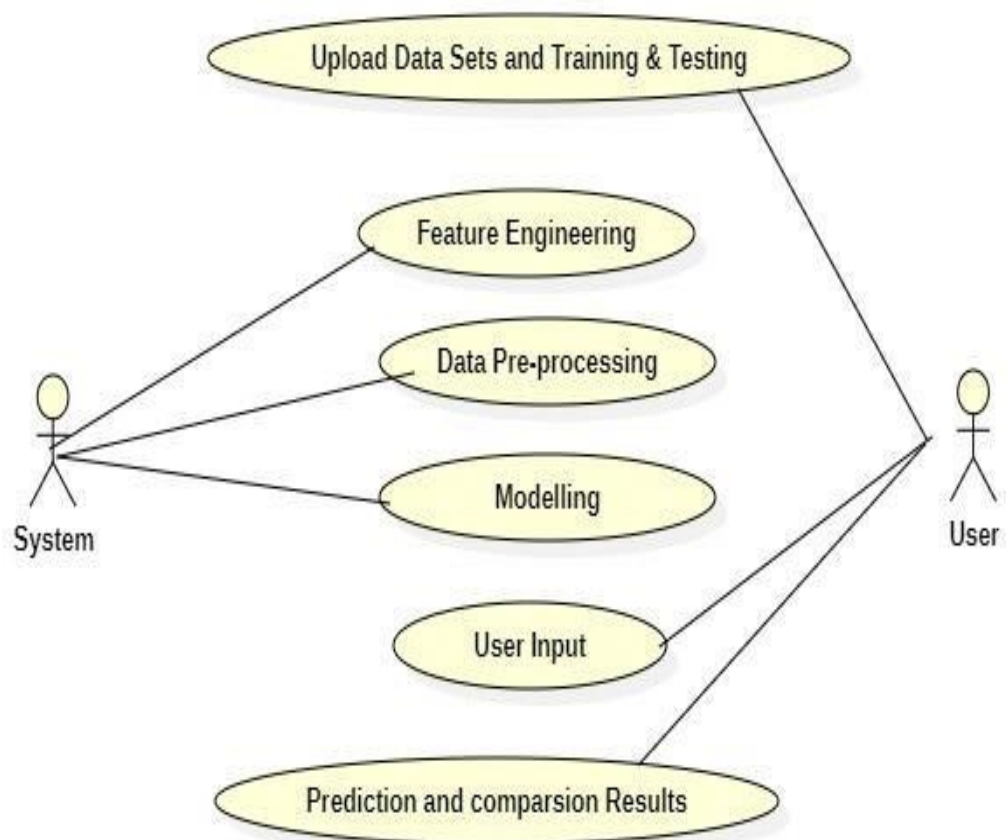
GOALS:

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

1. Provide users a ready-to-use, expressive visual modeling Language so thatthey 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 developmentprocess.

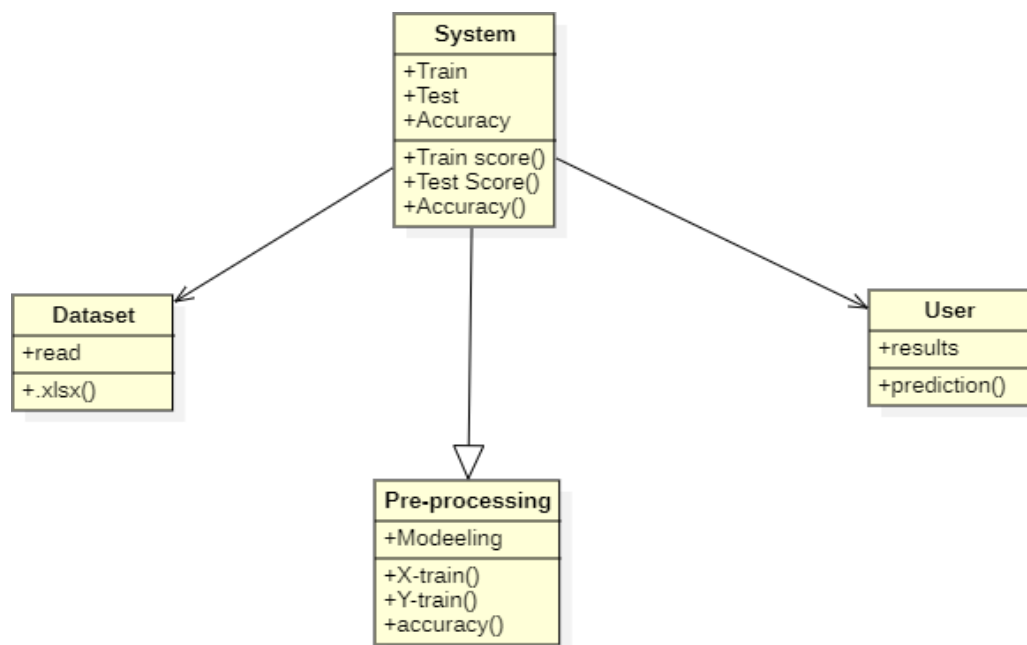
4.1.1 Use case diagram:

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.



4.1.2 Class diagram:

The class diagram is used to refine the use case diagram and define a detailed design of the system. The class diagram classifies the actors defined in the use case diagram into a set of interrelated classes. The relationship or association between the classes can be either an "is-a" or "has-a" relationship. Each class in the class diagram may be capable of providing certain functionalities. These functionalities provided by the class are termed "methods" of the class. Apart from this, each class may have certain "attributes" that uniquely identify the class.

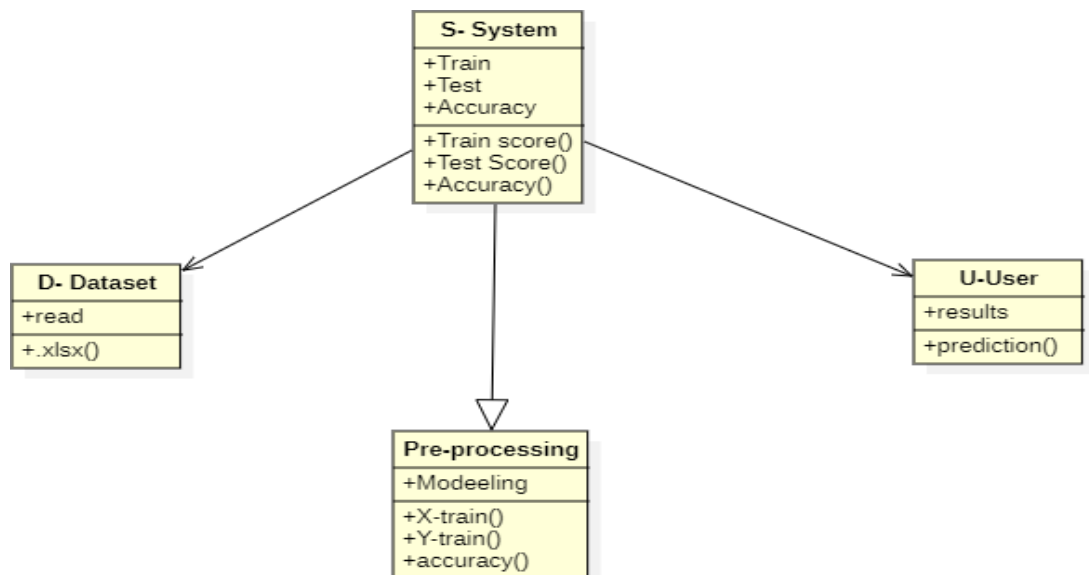


4.1.3 Object diagram:

The object diagram is a special kind of class diagram. An object is an instance of a class. This essentially means that an object represents the state of a class at a given point of time while

the system is running. The object diagram captures the state of different classes in the system and their relationships or associations at a given point of time.

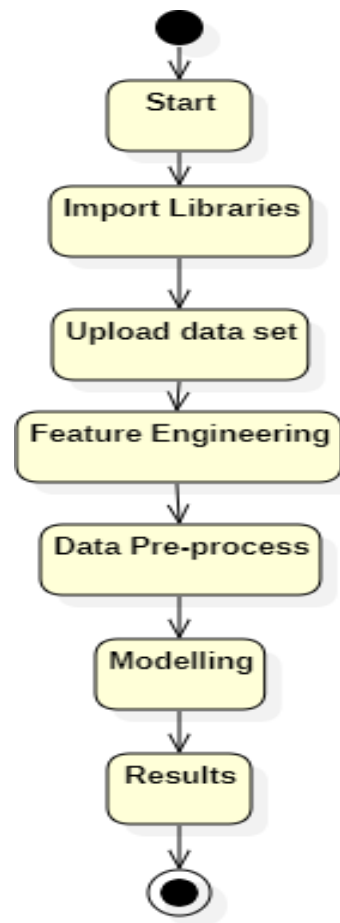
Object diagram



4.1.4 State diagram:

A state diagram, as the name suggests, represents the different states that objects in the system undergo during their life cycle. Objects in the system change states in response to events. In addition to this, a state diagram also captures the transition of the object's state from an initial state to a final state in response to events affecting the system.

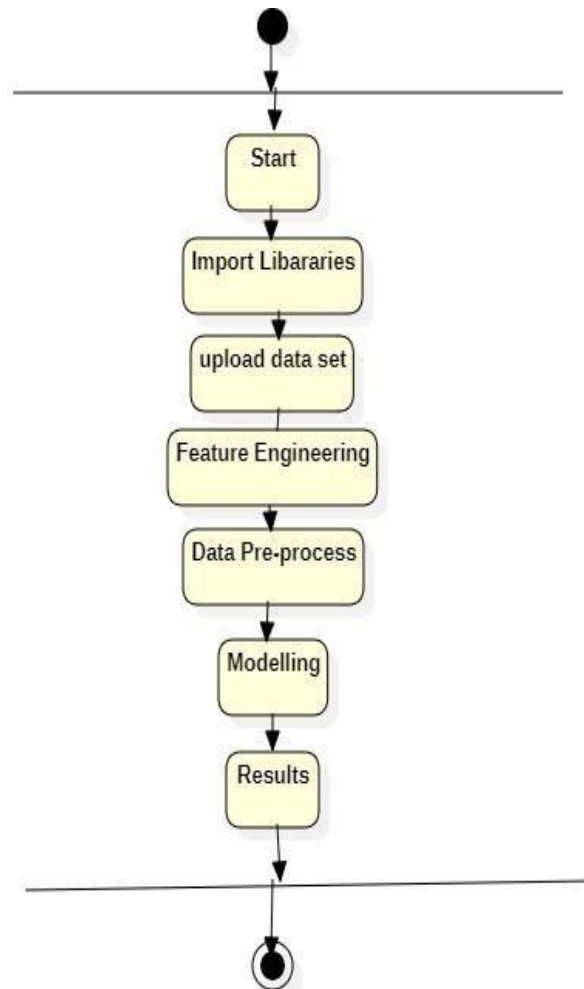
State diagram:



4.1.5 Activity diagram:

The process flows in the system are captured in the activity diagram. Similar to a state diagram, an activity diagram also consists of activities, actions, transitions, initial and final states, and guard conditions.

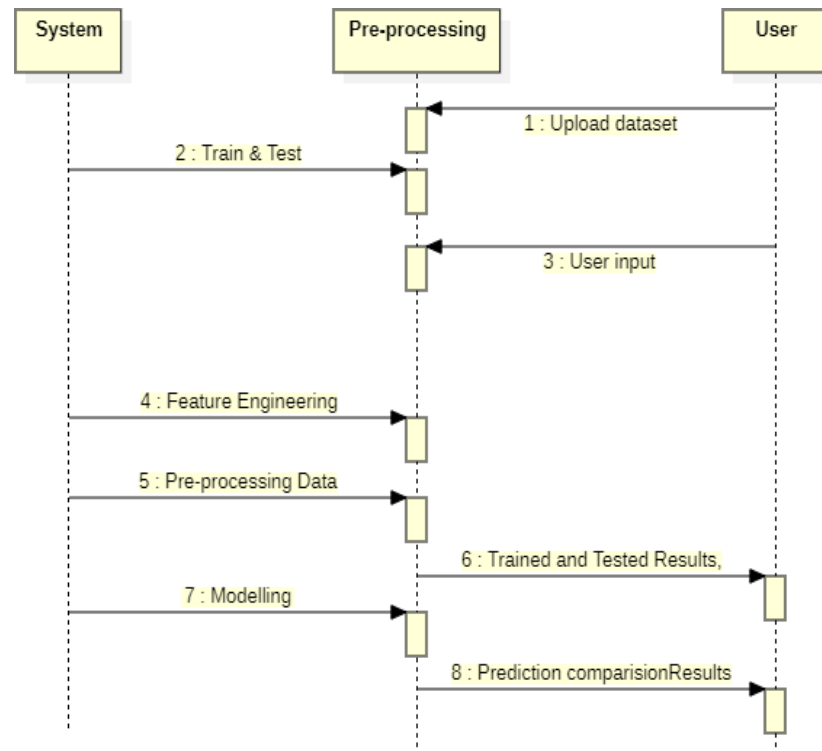
Activity diagram:



4.1.6 Sequence diagram:

A sequence diagram represents the interaction between different objects in the system. The important aspect of a sequence diagram is that it is time-ordered. This means that the exact sequence of the interactions between the objects is represented step by step. Different objects in the sequence diagram interact with each other by passing "messages".

Sequence diagram:



4.2 IMPLEMENTATION:

4.2.1 MODULES:

Data Collection: Assemble a comprehensive dataset, encompassing diverse health indicators such as age, blood sugar, and cholesterol, sourced from healthcare institutions, public repositories, and wearables for a holistic representation.

Data Preprocessing: Cleanse and standardize the dataset, managing missing values and outliers, ensuring a robust foundation for subsequent analysis and modeling.

Feature Selection: Identify and prioritize relevant features, optimizing the dataset for model efficiency and accuracy, laying the groundwork for effective predictive analytics.

4.2.2 Model Development:

The Modeling Module employs Decision Trees (DT), Random Forest (RF), and Naive Bayes (NB) algorithms to predict Cardiovascular Disease. DT captures intricate patterns, RF mitigates overfitting, and NB efficiently handles diverse features. This ensemble approach ensures robust and accurate predictions, contributing to effective early detection of CVD.

CHAPTER-5

SOFTWARE ENVIRONMENT

5. SOFTWARE ENVIRONMENT

5.1 Python

5.1.1 What is Python:-

Below are some facts about Python:

Python is currently the most widely used multi-purpose, high-level programming language. Python allows programming in Object-Oriented and Procedural paradigms.

Python programs generally are smaller than other programming languages like Java.

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

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

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

- Machine Learning
- GUI Applications (like Kivy, Tkinter, PyQt etc.)
- Web frameworks like Django (used by YouTube, Instagram, Dropbox)
- Image processing (like Opencv, Pillow)
- Web scraping (like Scrapy, BeautifulSoup, Selenium)
- Test frameworks
- Multimedia

5.1.2 Advantages of Python:-

Let's see how Python dominates over other languages.

1. Extensive Libraries

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

2. Extensible

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

3. Embeddable

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

4. Improved Productivity

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

5. IOT Opportunities

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

6. Simple and Easy

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

7. Readable

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

8. Object-Oriented

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

9. Free and Open-Source

Like we said earlier, Python is **freely available**. But not only can you **download Python** for free, but you can also download its source code, make changes to it, and even distribute it. It downloads with an extensive collection of libraries to help you with your tasks.

10. Portable

When you code your project in a language like C++, you may need to make some changes to it if you want to run it on another platform. But it isn't the same with Python. Here, you

need to **code only once**, and you can run it anywhere. This is called **Write Once Run Anywhere (WORA)**. However, you need to be careful enough not to include any system- dependent features.

11. Interpreted

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

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

5.1.3 Advantages of Python Over Other Languages

1. Less Coding

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

2. Affordable

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

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

3. Python is for Everyone

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

5.1.4 Disadvantages of Python:-

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

1. Speed Limitations

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

2. Weak in Mobile Computing and Browsers

While it serves as an excellent server-side language, Python is much rarely seen on the **client-side**. Besides that, it is rarely ever used to implement smartphone-based applications. One such application is called **Carbonnelle**. The reason it is not so famous despite the existence of Brython is that it isn't that secure.

3. Design Restrictions

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

4. Underdeveloped Database Access Layers

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

5. Simple

No, we're not kidding. Python's simplicity can indeed be a problem. Take my example. I don't do Java, I'm more of a Python person. To me, its syntax is so simple that the verbosity of Java code seems unnecessary. This was all about the Advantages and Disadvantages of Python Programming Language.

5.1.5 History of Python: -

What do the alphabet and the programming language Python have in common? Right, both start with ABC. If we are talking about ABC in the Python context, it's clear that the programming language ABC is meant. ABC is a general-purpose programming language and programming environment, which had been developed in the Netherlands, Amsterdam, at the CWI (Centrum Wiskunde & Informatica). The greatest achievement of ABC was to influence the design of Python. Python was conceptualized in the late 1980s. Guido van Rossum worked that time in a project at the CWI, called

Amoeba, a distributed operating system In an interview with Bill Venners¹, Guido van Rossum said: "In the early 1980s, I worked as an implementer on a team building a language called ABC at Centrum voor Wiskunde en Informatica (CWI). I don't know how well people know ABC's influence on Python. I try to mention ABC's influence because I'm indebted to everything I learned during that project and to the people who worked on it." Later on in the same Interview, Guido van Rossum continued: "I remembered all my experience and some of my frustration with ABC. I decided to try to design a simple scripting language that possessed some of ABC's better properties, but without its problems. So I started typing. I created a simple virtual machine, a simple parser, and a simple runtime. I made my own version of the various ABC parts that I liked. I created a basic syntax, used indentation for statement grouping instead of curly braces or begin-end blocks, and developed a small number of powerful data types: a hash table (or dictionary, as we call it), a list, strings, and numbers."

5.1.6 What is Machine Learning: -

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

Fundamentally, machine learning involves building mathematical models to help understand data. "Learning" enters the fray when we give these models *tunable parameters* that can be adapted to observed data; in this way the program can be considered to be "learning" from the data. Once these models have been fit to previously seen data, they can be used to predict and understand aspects of newly observed data. I'll leave to the reader the more philosophical.

machine learning is essential to using these tools effectively, and so we will start with some broad categorizations of the types of approaches we'll discuss here.

5.1.7 Categories Of Machine Learning:-

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

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

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

5.1.8 Need for Machine Learning

Human beings, at this moment, are the most intelligent and advanced species on earth because they can think, evaluate and solve complex problems. On the other side, AI is still in its initial stage and haven't surpassed human intelligence in many aspects. Then the

question is that what is the need to make machine learn? The most suitable reason for doing this is, “to make decisions, based on data, with efficiency and scale”.

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

5.1.9 Challenges in Machines Learning :-

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

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

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

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

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

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

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

5.1.10 Applications of Machines Learning:-

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

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

5.2 How to Start Learning Machine Learning?

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

And that was the beginning of Machine Learning! In modern times, Machine Learning is one of the most popular (if not the most!) career choices. According to Indeed, Machine Learning Engineer Is The Best Job of 2019 with a 344% growth and an average base salary of **\$146,085** per year.

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

5.2.1 How to start learning ML?

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

Step 1 – Understand the Prerequisites

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

(a) Learn Linear Algebra and Multivariate Calculus

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

(b) Learn Statistics

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

(c) Learn Python

Some people prefer to skip Linear Algebra, Multivariate Calculus and Statistics and learn them as they go along with trial and error. But the one thing that you absolutely cannot skip is Python! While there are other languages you can use for Machine Learning like R, Scala, etc. Python is currently the most popular language for ML. In fact, there are many Python libraries that are specifically useful for Artificial Intelligence and Machine Learning such as Keras, TensorFlow, Scikit-learn, etc. So if you want to learn ML, it's best if you learn Python! You can do that using various online resources and courses such as **Fork Python** available Free on GeeksforGeeks.

5.2.2 Step 2 – Learn Various ML Concepts

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

(a) Terminologies of Machine Learning

- **Model** – A model is a specific representation learned from data by applying some machinelearning algorithm. A model is also called a hypothesis.

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

(b) Types of Machine Learning

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

5.2.3 Advantages of Machine learning :-

1. Easily identifies trends and patterns -

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

2. No human intervention needed (automation)

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

3. Continuous Improvement

As **ML algorithms** gain experience, they keep improving in accuracy and efficiency. This lets them make better decisions. Say you need to make a weather forecast model. As the amount of data you have keeps growing, your algorithms learn to make more accurate predictions faster.

4. Handling multi-dimensional and multi-variety data

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

5. Wide Applications

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

5.2.4 Disadvantages of Machine Learning :-

1. Data Acquisition

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

2. Time and Resources

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

3. Interpretation of Results

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

4. High error-susceptibility

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

5.3 Python Development Steps : -

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

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

Purpose:-

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

5.3.1 Python

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

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

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

Python also acknowledges that speed of development is important. Readable and terse code is part of this, and so is access to powerful constructs that avoid tedious repetition of code. Maintainability also ties into this may be an all but useless metric, but it does say something about how much code you have to scan, read and/or understand to troubleshoot problems or tweak behaviors. This speed of development, the ease with which a programmer of other languages can pick up basic Python skills and the huge standard library is key to another area where Python excels. All its tools have been quick to implement, saved a lot of time, and

several of them have later been patched and updated by people with no Python background - without breaking.

5.3.2 Modules Used in Project:-

Tensorflow:

TensorFlow is a free and open-source software library for dataflow and differentiable programming across a range of tasks. It is a symbolic math library, and is also used for machine learning applications such as neural networks. It is used for both research and production at Google.

TensorFlow was developed by the Google Brain team for internal Google use. It was released under the Apache 2.0 open-source license on November 9, 2015.

Numpy:

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

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

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

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

Pandas:

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

Matplotlib:

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

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

Scikit – learn:

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

Python

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

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

- Python is Interpreted – Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.
- Python is Interactive – you can actually sit at a Python prompt and interact with the interpreter directly to write your programs. Python also acknowledges that speed of development is important. Readable and terse code is part of this, and so is access to powerful constructs that avoid tedious repetition of code. Maintainability also ties into this may be an all but useless metric, but it does say something about how much code you have to scan, read and/or understand to troubleshoot problems or tweak behaviors. This speed of development, the ease with which a programmer of other languages can pick up basic Python skills and the huge standard library is key to another area where Python excels. All its tools have been quick to implement, saved a lot of time, and several of them have later been patched and updated by people with no Python background - without breaking.

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

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

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

How to Install Python on Windows and Mac:

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

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

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

Download the Correct version into the system

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



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

Looking for a specific release?

Python releases by version number:

Release version	Release date		Click for more
Python 3.7.4	July 8, 2019	 Download	Release Notes
Python 3.6.9	July 2, 2019	 Download	Release Notes
Python 3.7.3	March 25, 2019	 Download	Release Notes
Python 3.4.10	March 18, 2019	 Download	Release Notes
Python 3.5.7	March 18, 2019	 Download	Release Notes
Python 2.7.16	March 4, 2019	 Download	Release Notes
Python 3.7.2	Dec. 24, 2018	 Download	Release Notes

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

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

Files

Version	Operating System	Description	MD5 Sum	File Size	GPG
Gzipped source tarball	Source release		68111671e5b2db4aef7b9ab01b0f9be	23017663	50G
XZ compressed source tarball	Source release		d33e4aa66057051c2eca45ee3604803	17131432	50G
macOS 64-bit/32-bit installer	Mac OS X	for Mac OS X 10.6 and later	6428b4fa7583da71a40c2ba0ee08e6	34898416	50G
macOS 64-bit installer	Mac OS X	for OS X 10.9 and later	56d075c38217a45773b95e4a936b241f	28082845	50G
Windows help file	Windows		063999573a2c06b2ac56cade6b4f7cd2	8131761	50G
Windows x86-64 embeddable zip file	Windows	for AMD64/EM64T/x64	9b00c3cfd89ec08abed318aa40728a2	7504701	50G
Windows x86-64 executable installer	Windows	for AMD64/EM64T/x64	a702b4b6a076de9db3041a183e563a00	26680368	50G
Windows x86-64 web-based installer	Windows	for AMD64/EM64T/x64	28c31c080b6d73a8e53a3bd311b4bd2	1362904	50G
Windows x86 embeddable zip file	Windows		9fab08138841879fda94133174139d0	6741626	50G
Windows x86 executable installer	Windows		37cc0c2942a54446a3d6451476294780	25603848	50G
Windows x86 web-based installer	Windows		1b670cfa5d117d82c30983ea371d87c	1324608	50G

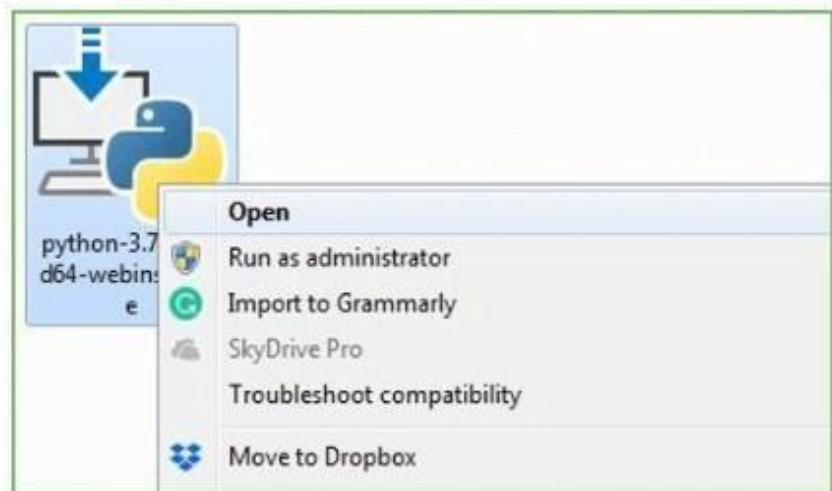
- To download Windows 32-bit python, you can select any one from the three options: Windows x86 embeddable zip file, Windows x86 executable installer or Windows x86 web-based installer.
- To download Windows 64-bit python, you can select any one from the three options: Windowsx86-64 embeddable zip file, Windows x86-64 executable installer or Windows x86-64 web- based installer.

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

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

Installation of Python

Step 1: Go to Download and Open the downloaded python version to carry out the installationprocess.



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



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



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

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

Verify the Python Installation

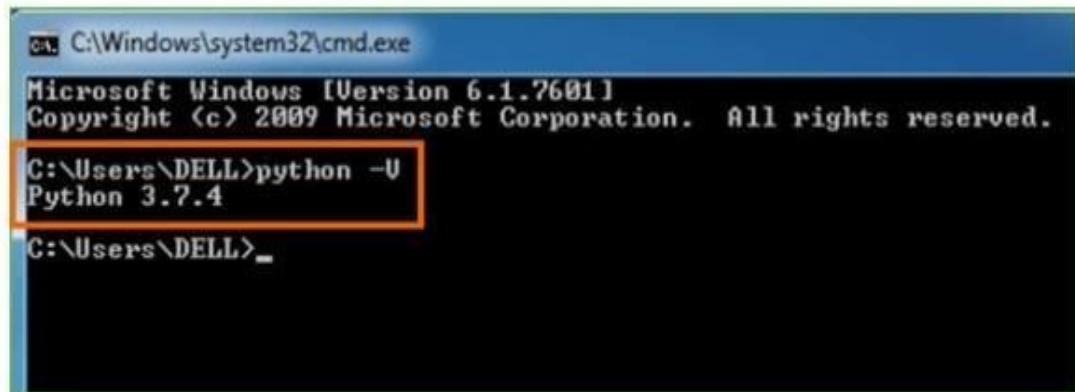
Step 1: Click on Start

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



Step 3: Open the Command prompt option.

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



```
CA. C:\Windows\system32\cmd.exe
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.
C:\Users\DELL>python -U
Python 3.7.4
C:\Users\DELL>_
```

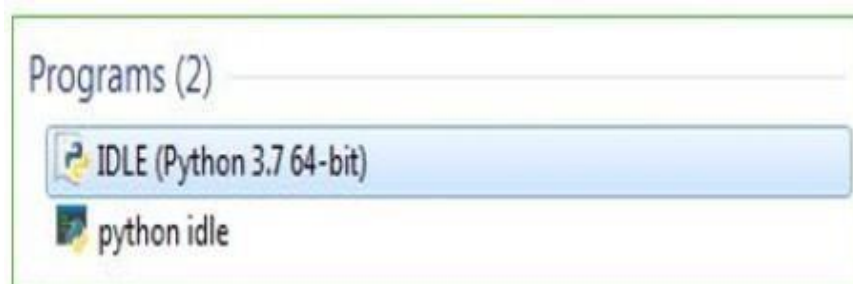
Step 5: You will get the answer as 3.7.4

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

Check how the Python IDLE works

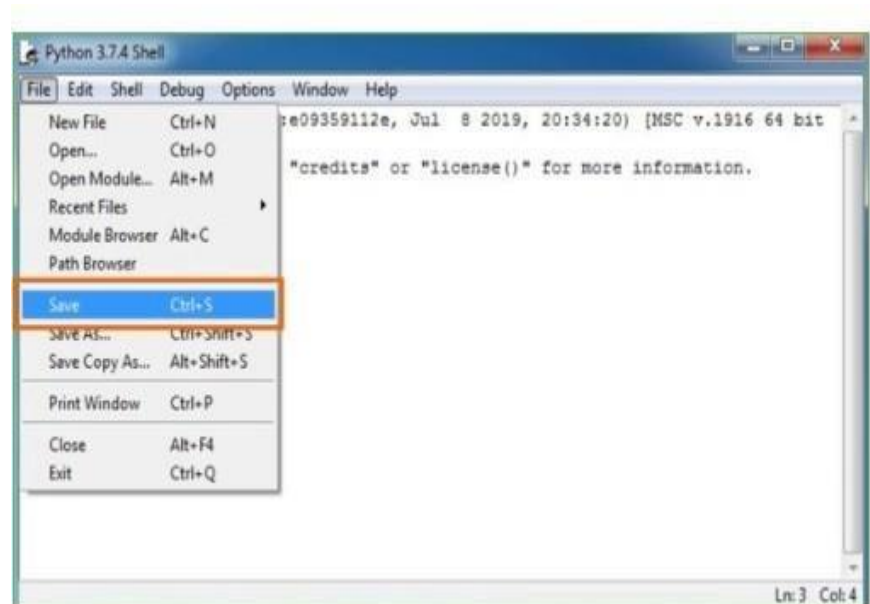
Step 1: Click on Start

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



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

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



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

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

CHAPTER-6

SYSTEM TESTING

6.SYSTEM TESTING

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.

6.1 TYPES OF TESTS

6.1.1 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.

6.1.2 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 satisfactory, as shown by successful 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.

6.1.3 Functional testing

Functional tests provide systematic demonstrations that the functions tested are available as specified by the business and alter 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 function 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, with its systematic coverage pertaining to identify Business process is also the flows; data fields, predefined processes, and successive processes will must be considered for testing. Before functional testing is almost very complete, additional tests are identified and the effective value of the current tests is determined.

6.1.4 System Testing

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.

6.1.5 White Box Testing

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

6.1.6 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.

6.2 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 requirements.

6.3 SOURCE CODES

6.3.2 PYTHON CODES

```
from flask import Flask, render_template, url_for, flash, redirect
import joblib
from flask import request
import numpy as np

app = Flask(__name__, template_folder='templates')

@app.route("/")
def index():
    return render_template("index.html")

def ValuePredictor(to_predict_list, size):
    to_predict = np.array(to_predict_list).reshape(1,size)
    if(size==7):
        loaded_model =
            joblib.load(r'C:\Users\HOME\PycharmProjects\CCVDPUML\.venv\Cardi
            ovascularDiseasePrediction\ModelFile\model.pkl')
        result = loaded_model.predict(to_predict)
    return result[0]

@app.route('/predict', methods = ["POST"])
def predict():
    if request.method == "POST":
        to_predict_list = request.form.to_dict()
        to_predict_list = list(to_predict_list.values())
        to_predict_list = list(map(float, to_predict_list))

        if(len(to_predict_list)==7):
            result = ValuePredictor(to_predict_list,7)

            if(int(result)==1):
                prediction = "You have heart condition, Consult the doctor immediately"
            else:
                prediction = "You are safe. You have no dangerous symptoms !!! :-)"
    return(render_template("prediction_result.html", prediction_text=prediction))
```

```
if __name__ == "__main__":  
    # Use below for local flask deployment  
    #app.run(debug=True)  
  
    #Use below for AWS EC2 deployment  
    app.run(host='0.0.0.0',port=8080)
```

6.3.2 HTML CODE

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>CardioVascular Disease Prediction</title>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <!--
=====
=====-->
  <link rel="icon" type="image/png" href="static/images/icons/heart-icon.ico"/>
  <!--
=====
=====-->
  <link rel="stylesheet" type="text/css"
    href="static/vendor/bootstrap/css/bootstrap.min.css">
  <!--
=====
=====-->
  <link rel="stylesheet" type="text/css" href="static/fonts/font-awesome-
    4.7.0/css/font-awesome.min.css">
  <!--
=====
=====-->
  <link rel="stylesheet" type="text/css" href="static/vendor/animate/animate.css">
  <!--
=====
=====-->
  <link rel="stylesheet" type="text/css" href="static/vendor/css-
    hamburgers/hamburgers.min.css">
  <!--
=====
=====-->
  <link rel="stylesheet" type="text/css" href="static/vendor/select2/select2.min.css">
  <!--
=====
=====-->
  <link rel="stylesheet" type="text/css" href="static/css/util.css">
  <link rel="stylesheet" type="text/css" href="static/css/main.css">
```

```
<!--  
=====
```

```
=====-->  
  
<style>  
::-webkit-input-placeholder {  
    color: red;  
}  
  
:-ms-input-placeholder {  
    color: red;  
}  
  
::placeholder {  
    color: red;  
}  
<!------->  
a:link {  
    color: magenta;  
}  
  
a:visited {  
    color: red;  
}  
  
a:hover {  
    color: blue;  
}  
  
a:active {  
    color: blue;  
}  
  
.bg-contact3{  
    background-color: black;  
    color: black;  
}
```

```

<!------->
.contact3-form-title{
color:black;
font-size: 30px;
font-family: Arial, Helvetica, sans-serif;
font-variant: normal;
}
<!------->

</style>
</head>
<body>
  <div class="bg-contact3" style="background-image: url('static/images/ecg.jpg');">
    <div class="container-contact3">
      <div class="wrap-contact3">
        <form action="{{ url_for('predict')}}" method="post" class="contact3-form
        validate-form">
          <span class="contact3-form-title">
            AI CardioVascular Disease Prediction<br>
            <p>By : Yasodha Solutions</p>
            <p>Email - yasodhasolutions@gmail.com</p>
          </span>

          <div class="form-group">
            <label for="cp">Type of Chest Pain</label>
            <select class="form-control" id="cp" name = "cp" required="required"
            placeholder="Chest pain type">
              <option value="1">Typical Angina</option>
              <option value="2">Atypical Angina</option>
              <option value="3">Non-Anginal Pain</option>
              <option value="4">Asymptomatic</option>
            </select>
          </div>

          <div class="form-group">
            <!-- TODO : Year to accept only year date picker -->
            <label for="trestbps">Resting Blood Pressure (in mm Hg)</label>

```

```

        <input class="form-control" type="number" id="trestbps"
name="trestbps" required="required" placeholder="Resting Blood Pressure">
    </div>

```

```

    <div class="form-group">
        <label for="chol">Serum Cholesterol in mg/dl</label>
        <input class="form-control" type="number" step="any" id="chol"
name="chol" required="required" placeholder="Serum Cholesterol">
    </div>

```

```

    <div class="form-group">
        <label for="fbs">Fasting Blood Sugar</label>
        <select class="form-control" id="fbs" name = "fbs" required="required"
placeholder="Fasting Blood Sugar">
            <option value="0">Fasting Blood Sugar < 120 mg/dl</option>
            <option value="1">Fasting Blood Sugar > 120 mg/dl</option>
        </select>
    </div>

```

```

    <div class="form-group">
        <label for="restecg">Resting Electro-Cardiographic Result</label>
        <select class="form-control" type="number" id="restecg"
name="restecg" required="required" placeholder="Resting Electro-
Cardiographic Result">
            <option value="0">Normal</option>
            <option value="1">Having ST-T wave abnormality</option>
            <option value="2">Showing probable or definite left ventricular
hypertrophy</option>
        </select>
    </div>

```

```

    <div class="form-group">
        <label for="thalach">Maximum Heart Rate Achieved</label>
        <input class="form-control" id="thalach" name = "thalach"
required="required" >
    </div>

```

```

    <div class="form-group">

```

```

        <label for="exang">Exercise Induced Angina</label>
        <select class="form-control" id="exang" name = "exang"
required="required" >
            <option value="1">Yes</option>
            <option value="0">No</option>
        </select>
    </div>

    <div class="container-contact3-form-btn">
        <button type="submit" class="contact3-form-btn">
            Predict
        </button>
    </div>
</form>

</div>
</div>
</div>

<!--
=====
=====-->
<script src="vendor/jquery/jquery-3.2.1.min.js"></script>
<!--
=====
=====-->
<script src="vendor/bootstrap/js/popper.js"></script>
<script src="vendor/bootstrap/js/bootstrap.min.js"></script>
<!--
=====
=====-->
<script src="vendor/select2/select2.min.js"></script>
<script>
    $(".selection-2").select2({
        minimumResultsForSearch: 20,
        dropdownParent: $('#dropDownSelect1')
    });
</script>

```

```

<!--
=====
=====-->

<script src="js/main.js"></script>

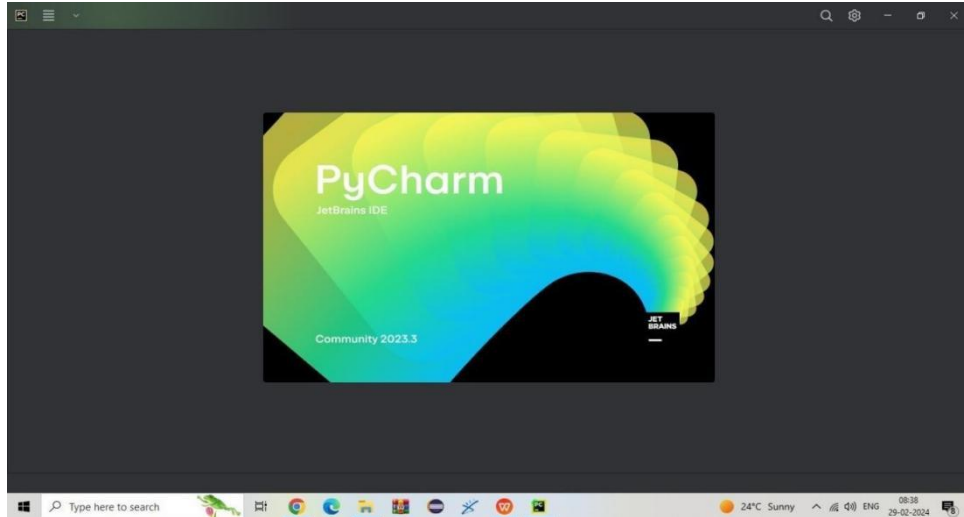
<!-- Global site tag (gtag.js) - Google Analytics -->
<script async src="https://www.googletagmanager.com/gtag/js?id=UA-23581568-
13"></script>
<script>
  window.dataLayer = window.dataLayer || [];
  function gtag(){dataLayer.push(arguments);}
  gtag('js', new Date());
  gtag('config', 'UA-23581568-13');
</script>
</body>
</html>

```


CHAPTER-7

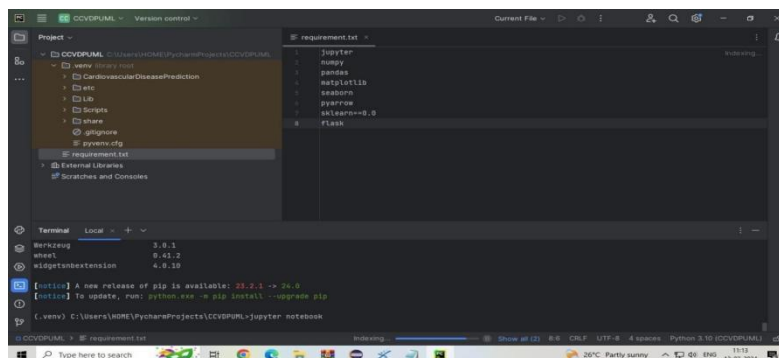
SCARENSHOTS OF OUTPUT

FIG-7.1:



- The **fig.7.1** shows the IDLE that we have used for the project “PYCHARM”.
- PyCharm is an integrated development surroundings (IDE) used for programming in Python. It is advanced by means of the Czech organization Jet Brains and is to be had for Windows, mac OS, and Linux .We have used pycharm community version.It consists of multiple benefits like smart code completion ,code inspection, debugging, testing, version control etc .

FIG-7.2



- The **fig-7.2** shows that the required modules for the

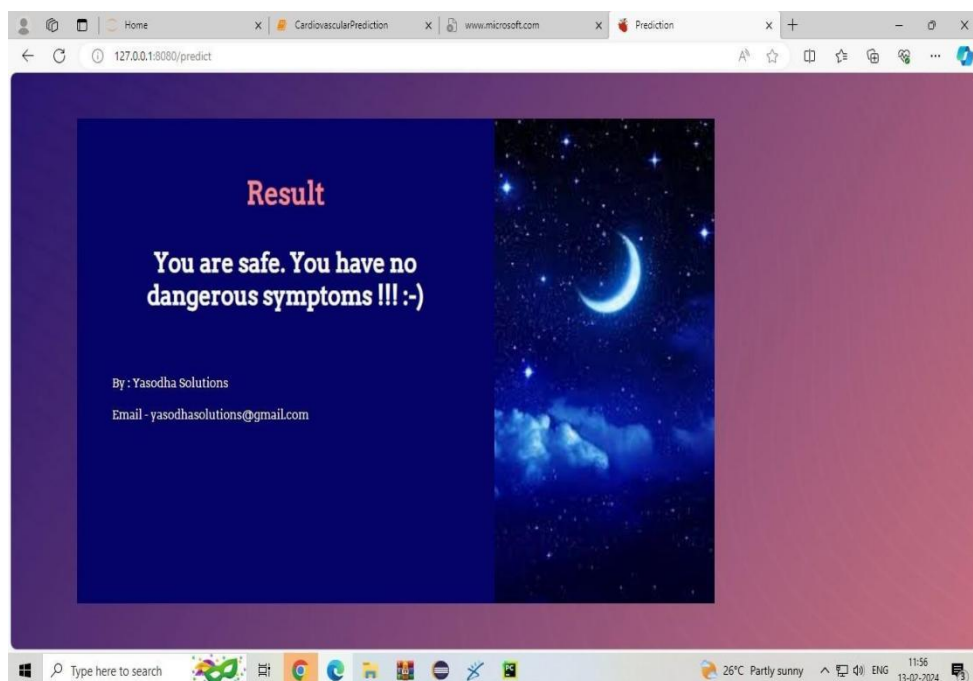
project and installation of the modules.

Output -1: The below figures shows how the application asks user to enter numerical values of given attributes. Based on the provided input by user it shows output whether the person has a heart disease or not and suggests them to consult doctor for treatment. Here, the output was safe means the person has no disease.

The screenshot shows a web browser window with the URL `127.0.0.1:8080`. The page is titled "DISEASE PREDICTION" and is attributed to "YASODHA SOLUTIONS" with the email "YASODHASOLUTIONS@GMAIL.COM". The interface features a green background with a white ECG line on the left and a green heart on the right. The central form contains the following fields and values:

- Type of Chest Pain: Typical Angina (dropdown menu)
- Resting Blood Pressure (in mm Hg): Resting Blood Pressure (text input)
- Serum Cholesterol in mg/dl: Serum Cholesterol (text input)
- Fasting Blood Sugar: Fasting Blood Sugar < 120 mg/dl (dropdown menu)
- Resting Electro-Cardiographic Result: Normal (dropdown menu)
- Maximum Heart Rate Achieved: (text input, currently empty)

The Windows taskbar at the bottom shows the date as 13-02-2024 and the time as 11:54.

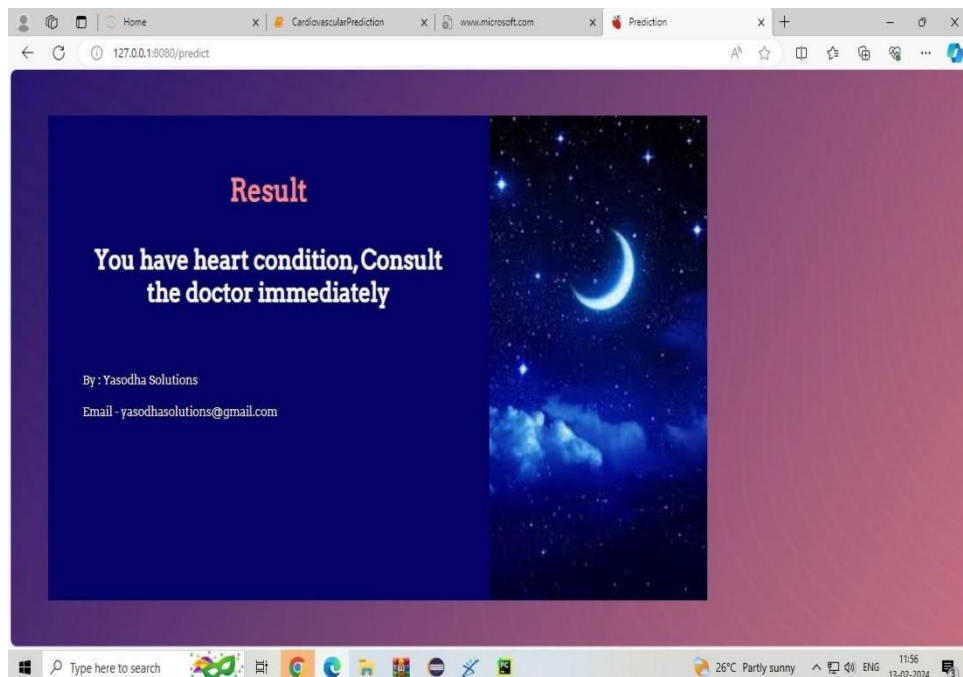


Output-2: The below pictures shows that , how the application measures whether the person has a heart disease or not based on some attributes like type of chest pain, blood pressure, cholesterol, blood sugar etc. Here, the output was that, person has a heart disease.

The screenshot shows a web browser window with the URL `127.0.0.1:8080`. The application interface has a green central panel with input fields for various medical attributes. On the left, there is a faint ECG waveform, and on the right, a faint heart icon. The input fields are as follows:

- Type of Chest Pain:
- Resting Blood Pressure (in mm Hg):
- Serum Cholesterol in mg/dl:
- Fasting Blood Sugar:
- Resting Electro-Cardiographic Result:
- Maximum Heart Rate Achieved:
- Exercise Induced Angina:

The Windows taskbar at the bottom shows the date and time as 11:56 on 13-02-2024.



CHAPTER-8

CONCLUSION

8. CONCLUSION

8.1 Conclusion

An increasing death rate due to heart disease is a cause of concern for every citizen. Also, the increase in population that decreases the efficiency of hospitals in giving timely treatment. Therefore this calls out for an immediate solution. It used machine learning models such as RF, NB, DT etc. It is possible to detect when a patient has a heart-related problem. To reduce the over-fitting of the models, we created synthetic data. We analysed the dataset completely, cleaned the data, and generated a new feature, BMI, in order to improve our model's performance. The Gradient Boosting Classifier performed best in terms of train score 80.29% in training accuracy score, and the ML algorithm works best in terms of the test score, i.e. 72.68%.

CHAPTER-9

REFERENCE

REFERENCES

- [1] Fahd Saleh Alotaibi, "A Implementation of Machine Learning Model to Predict Heart Failure Disease" International Journal of Advanced Computer Science and Applications(IJACSA), 10(6), 2019.
<http://dx.doi.org/10.14569/IJACSA.2019.0100637>
- [2] J. Thomas and R. T. Princy, "Human heart disease prediction system using datamining techniques," 2016 International Conference on Circuit, Power and Computing Technologies (ICCPCT), 2016, pp. 1-5, doi: 10.1109/ICCPCT.2016.7530265.
- [3] J. Thomas and R. T. Princy, "Human heart disease prediction system using datamining techniques," 2016 International Conference on Circuit, Power and Computing Technologies (ICCPCT), 2016, pp. 1-5, doi: 10.1109/ICCPCT.2016.7530265.
- [4] Rajdhan, Apurb Agarwal, Avi Sai, Milan Ghuli, Poonam. (2020). Heart Disease Prediction using Machine Learning. International Journal of Engineering and research . V9.
10.17577/IJERTV9IS040614.
- [5] Suthaharan, S. (2016). Support vector machine. In Machine learning models and algorithms for big data classification (pp. 207-235). Springer, Boston, MA.
- [6] Jiang, L., Cai, Z., Wang, D., Jiang, S. (2007, August). Survey of improving k- nearest-neighbor for classification. In Fourth international conference on fuzzy systems and knowledge discovery (FSKD 2007) (Vol. 1, pp. 679-683). IEEE.
- [7] Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., ... Liu, T. Y. (2017). Lightgbm: A highly efficient gradient boosting decision tree. Advances in neural information processing systems, 30, 3146-3154.
- [8] Selent, D. (2010). Advanced encryption standard. Rivier Academic Journal, 6(2), 1-14.

- [9] Yegnanarayana, B. (2009). Artificial neural networks. PHI Learning Pvt. Ltd..
- [10] Amin-Naji, M., Aghagolzadeh, A., Ezoji, M. (2019). CNNs hard voting for multi-focus image fusion. Journal of Ambient Intelligence and Humanized Computing, 1-21.
- [11] Nuttall, F. Q. (2015). Body mass index: obesity, BMI, and health: a critical review. Nutrition today, 50(3), 117.
- [12] Zheng, A., Casari, A. (2018). Feature engineering for machine learning: principles and techniques for data scientists. ” O’Reilly Media, Inc.”.
- [13] Cai, J., Luo, J., Wang, S., Yang, S. (2018). Feature selection in machine learning: A new perspective. Neurocomputing, 300, 70-79.
- [14] Wan, X. (2019, June). Influence of feature scaling on convergence of gradient iterative algorithm. In Journal of Physics: Conference Series (Vol. 1213, No. 3, p. 032021). IOP Publishing.
- Kearns, C. E., Schmidt, L. A., Glantz, S. A. (2016). Sugar industry and coronary heart disease research: a historical analysis of internal industry documents.

CHAPTER-10

FUTURE SCOPE

10.1 Future Scope:

Future enhancements for cardiovascular prediction using machine learning techniques are poised to revolutionize preventive healthcare. Advanced machine learning algorithms, including deep learning models such as convolutional neural networks (CNNs) and recurrent neural networks (RNNs), will enable more accurate and personalized risk assessment by analyzing diverse biomedical data types, including genomic, clinical, and imaging data.

Integration of wearable devices and continuous monitoring technologies will provide real-time data streams, facilitating early detection of cardiovascular abnormalities and enabling timely interventions. Moreover, the development of interpretable machine learning models will enhance clinicians' understanding of the underlying factors contributing to cardiovascular risk, thereby enabling more informed decision-making. Collaborative efforts among healthcare providers, researchers, and technology developers will foster the creation of comprehensive predictive models capable of capturing the complex interplay of genetic, environmental, and lifestyle factors influencing cardiovascular health.

Furthermore, the adoption of federated learning and privacy-preserving techniques will ensure patient data confidentiality while leveraging distributed datasets for model training. Embracing a holistic approach to healthcare, future enhancements will prioritize the integration of socioeconomic and behavioral data, empowering clinicians to address underlying disparities and tailor interventions to individuals' unique needs. Overall, the future of cardiovascular prediction using machine learning holds immense promise in enabling proactive, personalized healthcare interventions that mitigate the burden of cardiovascular diseases worldwide.