## A Project Report On

**“ PREDICTION OF HEART DISEASE USING MACHINE LEARNING”**

**Submitted in partial fulfilment of the requirements for the award of the Degree of**

**BACHELOR OF TECHNOLOGY**

**In**

**COMPUTER SCIENCE AND ENGINEERING**

Submitted by

**B. C. SWETHA (203N1A0503)**

**C.NIRANJAN REDDY (203N1A0512)**

### V.JYOTHI (203N1A0557)

**S.MADHURIKA BAI (203N1A0554)**

**Under the esteemed guidance of**

**Ms.K.RUBIYA ALAM M.Tech.,**

### Assistant Professor



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**P.V.K.K Institute of Technology**

**An ISO 9001-2015 Certified Institute**

(Approved by AICTE, New Delhi & Affiliated to JNTUA, Anantapuramu.) Sanapa Road, Rudrampeta, Anantapuramu-515001

**Andhra Pradesh 2023-2024**

P.V.K.K INSTITUTE OF TECHNOLOGY

**An ISO 9001-2015 Certified Institute**

**(Approved by AICTE, New Delhi & Affiliated to JNTUA, Anantapuramu) Sanapa road, Rudranpeta,Anantapuramu-515001**

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**



**CERTIFICATE**

This is to certify that project report entitled **“ PREDICTION OF HEART DISEASE USING MACHINE LEARNING”** is bonafide work done by

**B.C.SWETHA (203N1A0503)**

**C.NIRANJAN REDDY (203N1A0512)**

**V.JYOTHI (203N1A0557)**

**S.MADHURIKA BAI (203N1A0554)**

Under the supervision of, **Ms.K.RUBIYA ALAM** M.Tech.., Assistant Professor of CSE in the partial fulfilment of the requirements for the award of the degree of **BACHELOR OF TECHNOLOGY** with specialization in **COMPUTER SCIENCE AND ENGINEERING** in

**P.V.K.K Institute of Technology**, Sanapa Road, Rudrampeta, Anantapuramu during the academic year 2023-2024.

Signature of the Project Guide Signature of H.O.D

Ms.K.RUBIYA ALAM, M.Tech ., Dr. K. Bhargavi M.Tech, Ph.D

Assistant Professor Associate Professor & Head

Dept. of CSE Dept. of CSE

External Viva-voce Held on:

External Examiner

## DECLARATION

We hereby declare that the project work entitled “**PREDICTION OF HEART DISEASE USING MACHINE LEARNING**” is a

genuine work carried out by us under the guidance of **Ms.K.RUBIYA ALAM M.Tech., Assistant Professor, Department of Computer Science & Engineering**, in partial fulfillment for the award of the degree of **“B.Tech”** of **Jawaharlal Nehru Technological University Anantapur, Anantapuramu**. This has not been submitted in part of fulfillment towards any other degree.

B.C.SWETHA C.NIRANJAN REDDY

203N1A0503 203N1A0512

V.JYOTHI S.MADHURIKA BAI

203N1A0557 203N1A0554

### ACKNOWLEDGEMENT

At the outset, we thank our Honourable Founder, Correspondent & Secretary **Dr. Palle Raghunatha Reddy M.Sc,M. Phil,Ph.D.,** garu, Honourable Chairman **Dr. Palle Venkata Krishna Kishore Reddy M.Tech, MS (USA), AMIE...Ph.D.,** garu, and our Treasurer **Smt. P. Sindhura Reddy M.Tech.,** garu of Sri Balaji Educational Society for providing us with good faculty and all the required equipment in C.S.E labs and for their moral support throughout the course.

I wish to express our sense of gratitude to **Dr. B. RAMESH BABU B.Tech, M.S, Ph.D.,** garu Principal, **P.V.K.K INSTITUTE OF TECHNOLOGY**, for providing necessary facilities.

With a great sense of pleasure, we extend our gratitude to **Dr. K. BHARGAVI M.Tech, Ph.D,** garu, Head of the Dept., Department of Computer Science & Engineering for her co-operation and providing necessary help for completion of this project.

It is a great pleasure to express our heartful gratitude and thankfulness to our guide **Ms. K.RUBIYA ALAM M.Tech.,** garu Assistant Professor, Department of Computer Science & Engineering for her valuable guidance and helpful counsel rendered in the due course of the project.

Finally, we would like to thank one and all in the C.S.E department who helped us directly and indirectly along with our family members and friends for their cooperation to complete this project.

### PROJECT ASSOCIATES

### B.C.SWETHA

**C. NIRANJANREDDY**

**V.JYOTHI**

**S .MADHURIKA BAI**

## 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 Heart 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.

## CONTENTS

1. [INTRODUCTION 1](#_TOC_250033)
   1. [Motivation 2](#_TOC_250032)
   2. [Problem Definition 2](#_TOC_250031)
   3. Objective of project 3
2. [SYSTEM ANALYSIS 4](#_TOC_250030)
   1. [Existing System 5](#_TOC_250029)
   2. [Limitations of Existing System 5](#_TOC_250028)
   3. [Proposed System 5](#_TOC_250027)
   4. [Advantages of proposed system 5](#_TOC_250026)
   5. [Review of Literature 6](#_TOC_250025)
3. SOFTWARE REQUIREMENT ANALYSIS AND SPECIFICATION 7
   1. [Hardware Requirements 8](#_TOC_250024)
   2. [Software Requirements 8](#_TOC_250023)
   3. [System Study 9](#_TOC_250022)
   4. [Software Environment 11](#_TOC_250021)
4. [SYSTEM DESIGN 37](#_TOC_250020)
   1. [System Architecture 38](#_TOC_250019)
   2. [Data Flow Diagram 38](#_TOC_250018)
   3. [UML Diagrams 40](#_TOC_250017)
   4. [Use Case Diagram 41](#_TOC_250016)
   5. [Class Diagram 42](#_TOC_250015)
   6. [Object Diagram 43](#_TOC_250014)
   7. [State Diagram 44](#_TOC_250013)
   8. [Activity Diagram 45](#_TOC_250012)
   9. [Sequence Diagram 46](#_TOC_250011)
   10. [Implementation 47](#_TOC_250010)
5. TESTING 48
   1. [Unit Testing 49](#_TOC_250009)
   2. [Integration Testing 49](#_TOC_250008)
   3. [Functional Testing 50](#_TOC_250007)
   4. [System Testing 50](#_TOC_250006)
   5. [White Box Testing 50](#_TOC_250005)
   6. [Black Box Testing 51](#_TOC_250004)
   7. [Test Cases 51](#_TOC_250003)
6. RESULTS 53
   1. [Screenshots 54](#_TOC_250002)
7. CONCLUSION 58
   1. [Conclusion 59](#_TOC_250001)
   2. [Future Enhancement 59](#_TOC_250000)
   3. **References**

**LIST OF FIGURES**

|  |  |  |  |
| --- | --- | --- | --- |
| **S.**  **No** | **Fig.No** | **Figure Name** | **Page No** |
| 1 | 3.4.1 | Random Forest | 23 |
| 2 | 3.4.2 | Decision Tree | 24 |
| 3 | 3.4.3 | Naïve Bayes | 25 |
| 4 | 3.4.4 | Python Webpage | 27 |
| 5 | 3.4.5 | Python File Link Image | 27 |
| 6 | 3.4.6 | Python Version File List | 28 |
| 7 | 3.4.7 | Python Version File Along With Operating System | 28 |
| 8 | 3.4.8 | Python Version File Along With Operating System | 29 |
| 9 | 3.4.9 | Python Application Installation | 29 |
| 10 | 3.4.10 | Python Installation Setup | 30 |
| 11 | 3.4.11 | Command Prompt Window | 31 |
| 12 | 3.4.12 | Command Prompt exe | 31 |
| 13 | 3.4.13 | Python IDLE Application | 32 |
| 14 | 3.4.14 | Python Shell | 32 |
| 15 | 4.1.1 | System Architecture | 38 |
| 16 | 4.2.1 | Data Flow Diagram | 39 |
| 17 | 4.4.1 | Use Case Diagram | 41 |
| 18 | 4.5.1 | Class Diagram | 42 |
| 19 | 4.6.1 | Objective Diagram | 43 |
| 20 | 4.7.1 | State Diagram | 44 |
| 21 | 4.8.1 | Activity Diagram | 45 |
| 22 | 4.9.1 | Sequence Diagram | 46 |
| 23 | 6.1.1 | Screenshot During The Execution Code | 54 |
| 24 | 6.1.2 | Screenshot Of Output Predicting The Cardiovascular Disease | 55 |
| 25 | 6.1.3 | Screenshot During The Execution Code | 56 |
| 26 | 6.1.4 | Screenshot Of Output Predicting The Cardiovascular Disease | 57 |

**LIST OF ABBREVIATIONS**

|  |  |  |
| --- | --- | --- |
| **S. No** | **Acronym** | **Abbreviation** |
| 1 | CVD | Cardiovascular Disease |
| 2 | SVM | Support Vector Machines |
| 3 | DT | Decision Tree |
| 4 | RFA | Random Forest Algorithm |
| 5 | HTML | Hypertext and Markup Language |
| 6 | CSS | Cascading Style Sheets |
| 7 | IDE | Integrated Development Environment |
| 8 | ML | Machine Learning |
| 9 | VGA | Video Graphics Array |

# CHAPTER-1 INTRODUCTION

## INTRODUCTION

Now a days, heart disease prediction has been a major concept in recent world that is impacting the society towards health. The main concept is to identify the age group and heart rate using the Random forest algorithm. Our project tells how the heart rate and condition is estimated based on the inputs such as blood pressure and many more being provided by the user to a system. This is being much better way when it comes with others algorithms the implementation of RFA gives the better experience and provide accurate result. This helps in early prediction of the disease and is used in many ways, where as it is being provided with the input, in order to find the heart rate based on the health condition.

### MOTIVATION:

In a world grappling with escalating health challenges, particularly the pervasive threat of Heart Disease or 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 Heart Disease or 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.

### 1.2 OBJECTIVE OF PROJECT:

The primary objective of our project is to leverage advanced classification models to detect and predict Heart Disease or 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 SYSTEM ANALYSIS

## SYSTEM ANALYSIS

### EXISTING SYSTEM:

Utilizing Support Vector Machines (SVM), our existing system excels in binary classification for Heart Disease or 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.

### LIMITATIONS OF EXISTING SYSTEM:

1. Support Vector Machines (SVM) can be computationally intensive.
2. Prediction of cardiovascular disease results is not accurate.
3. Cannot handle enormous datasets for patient records.

### PROPOSED SYSTEM:

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

### ADVANTAGES OF PROPOSED SYSTEM:

1. Increased accuracy for effective heart disease diagnosis.
2. Reduce the time complexity of doctors.
3. Cost effective for patients.

### REVIEW OF LITERATURE:

Fahd Saleh Alotaibi (2019) has created a machine learning model comparing five various machine learning algorithms [1]. Author has deeply studied on the 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, ea tl.,(2020) has proposed a machine learning model for predicting heart disease in [4] that uses techniques like Navive Bayesian for classification of the dataset and AES [8] algorithm for prediction of disease.

# CHAPTER-3

**SYSTEM REQUIREMENT ANALYSIS & SPECIFICATION**

## 3. SOFTWARE REQUIREMENT SPECIFICATION

### HARDWARE REQUIREMENTS:

* + - System : Pentium IV 2.4 GHz
    - RAM : 2 GB (min)
    - Hard Disk : 40 GB
    - Key Board : Standard Windows Keyboard
    - Mouse : Two or Three Button Mouse
    - Monitor : 15 VGA Colour

### SOFTWARE REQUIREMENTS:

* + - Operating system : Windows 7
    - Coding Language : Python 3.7.4
    - Front**-**End : HTML,CSS
    - Back**-**End : Flask,Jupyter notebook

### SYSTEM STUDY:

#### 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

1. ECONOMICAL FEASIBILITY
2. TECHNICAL FEASIBILITY
3. 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.

### SOFTWARE ENVIRONMENT :

**Python:** Below are some facts about Python.

Python is currently the most widely used multi-purpose, high-level programming language. Python allows programming in Object-Oriented and Procedural paradigms. Python programs generally are smaller than other programming languages like Java.Programmers have to type relatively less and indentation requirement of the language, makes them readable all the time.Python language is being used by almost all tech-giant companies like – Google, Amazon, Facebook, Instagram, Dropbox, Uber… etc.

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

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

#### Advantages of Python:

Let’s see how Python dominates over other languages.

#### Extensive Libraries:

Python downloads with an extensive library and it contain code for various purpose like regular expression browsers, Threading, database, CGI, email, image manipulation and more. So, we don’t have to write the complete code for that manually.

#### 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.

#### 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.

#### Free and Open-Source:

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

#### 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 is’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.

#### 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.

#### 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

#### 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.

1. **Python is for Everyone:**

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

#### Disadvantages of Python:

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

#### Speed Limitations:

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

#### 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 Bryton is that it isn’t that secure.

#### 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**.

#### 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.

#### Simple:

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

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

#### History of Python:

What do the alphabet and the programming language Python have in common? Right, both start with ABC. If we are talking about ABC in the Python context, it's clear that the programming language ABC is meant. ABC is a general-purpose programming language and programming environment, which had been developed in the Netherlands, Amsterdam, at the CWI (Centrum Wickenden &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 Centumvir Wickenden 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."

#### Machine Learning:

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

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

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

#### Categories Of Machine Leaning:

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

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

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

#### Need for Machine Learning:

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

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

#### Challenges on Machine Learning:

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

* **Quality of data** − Having good-quality data for ML algorithms is one of the biggest challenges. Use of low-quality data leads to the problems related to data preprocessing and feature extraction.
* **Time-Consuming task** − Another challenge faced by ML models is the consumption of time especially for data acquisition, feature extraction and retrieval.
* **Lack of specialist persons** − As ML technology is still in its infancy stage, availability of expert resources is a tough job.
* **No clear objective for formulating business problems** − Having no clear objective and well-defined goal for business problems is another key challenge for ML because this technology is not that mature yet.
* **Issue of overfitting & underfitting** − If the model is overfitting or underfitting, it cannot be represented well for the problem.
* **Curse of dimensionality** − Another challenge ML model faces is too many features of data points. This can be a real hindrance.
* **Difficulty in deployment** − Complexity of the ML model makes it quite difficult to be deployed in real life.

#### Applications on Machine Learning:

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

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

#### How to start learning 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.

#### 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 math’s 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.

#### 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.

#### Learn Python :

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

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

#### Step 2 – Learn Various ML Concepts:

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

#### Terminologies of Machine Learning:

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

#### 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 unlabeled 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 unlabeled data like Unsupervised Learning with a small amount of labeled data. Using labeled data vastly increases the learning accuracy and is also more cost-effective than Supervised Learning.
  + **Reinforcement Learning –** This involves learning optimal actions through trial and error. So, the next action is decided by learning behaviors that are based on the current state and that will maximize the reward in the future.

#### Advantages of Machine learning:

1. **Easily identifies trends and patterns:**

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

#### 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 software’s; they learn to filter new threats as they are recognized. ML is also good at recognizing spam.

#### Continuous Improvement :

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

#### 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.

#### Wide Applications:

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

#### Disadvantages of Machine Learning:

1. **Data Acquisition:**

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

#### 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.

#### 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.

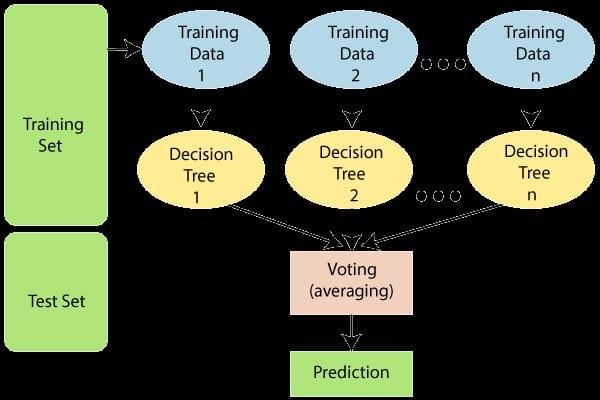
#### High error-susceptibility:

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

#### Machine learning algorithms:

**Random Forest:**

Random forest is a commonly-used machine learning algorithm, trademarked by Leo Breiman and Adele Cutler, that combines the output of multiple decision trees to reach a single result. Its ease of use and flexibility have fueled its adoption, as it handles both classification and regression problems.

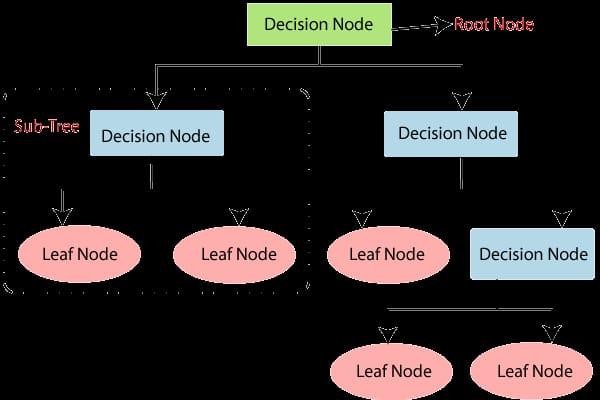


#### Fig:3.4.1 Random Forest

**Decision Tree:**

Decision Tree algorithm belongs to the family of supervised learning algorithms. Unlike other supervised learning algorithms, the decision tree algorithm can be used for solving regression and classification problems too.

The goal of using a Decision Tree is to create a training model that can use to predict the class or value of the target variable by learning simple decision rules inferred from prior data(training data).

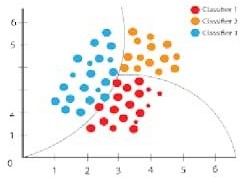


#### Fig:3.4.2 Decision Tree

**Naïve Bayes Algorithm:**

The Naïve Bayes classifier is a supervised machine learning algorithm that is used for classification tasks such as text classification. They use principles of probability to perform classification tasks.

Naïve Bayes algorithm is used for classification problems. It is highly used in text classification. In text classification tasks, data contains high dimension (as each word represent one feature in the data). It is used in spam filtering, sentiment detection, rating classification etc. The advantage of using naïve Bayes is its speed. It is fast and making prediction is easy with high dimension of data.



#### Fig:3.4.3 Naïve Bayes

**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.

#### Installation Process of Python:

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 operating system and based processor, you must download the python version. My system type is a **Windows 64- bit operating system**. So, the steps below are to install python version 3.7.4 on Windows 7 device or to install Python 3. [Download the Python Cheatsheet here**.**](https://myelearninghub.com/python-cheat-sheet/)The steps on how to install Python on Windows 10, 8 and 7 are **divided into 4 parts** to help understand better.

Download the Correct version into the system:

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



#### Fig: 3.4.4 Python webpage

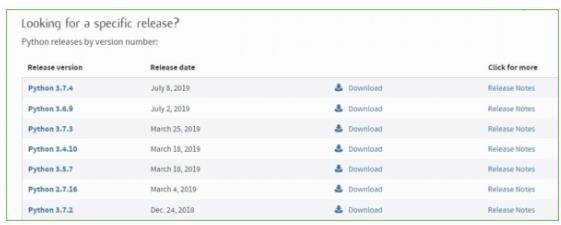
Now, check for the version 3.7.4 for your operating system.

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



#### Fig: 3.4.5 Python file link image

**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



**Fig: 3.4.6 Python version file list 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.

#### Fig: 3.4.7 Python version file along with operating system

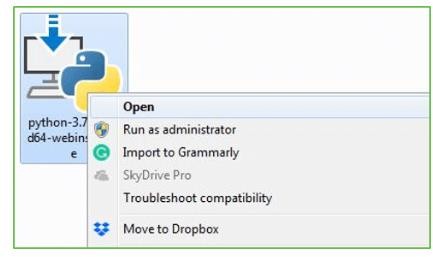
* To download Windows 32-bit python, you can select any one from the three options: Windows x86 embeddable zip file, Windows x86 executable installer or Windows x86 web-based installer.
* To download Windows 64-bit python, you can select any one from the three options: Windows x86-64 embeddable zip file, Windows x86-64 executable installer or Windows x86-64 web-based installer.Here we will install Windows x86-64 web-based installer. Here your first part regarding which version of python is to be downloaded is completed. Now we move ahead with the second part in installing python i.e., Installation



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

#### Installation of Python:

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



#### Fig: 3.4.8 Python version file along with operating system

**Step 2:** Before you click on Install Now, make sure to put a tick on Add to Python path



#### Fig: 3.4.9 Python application installation

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



#### Fig: 3.4.10 Python installation setup

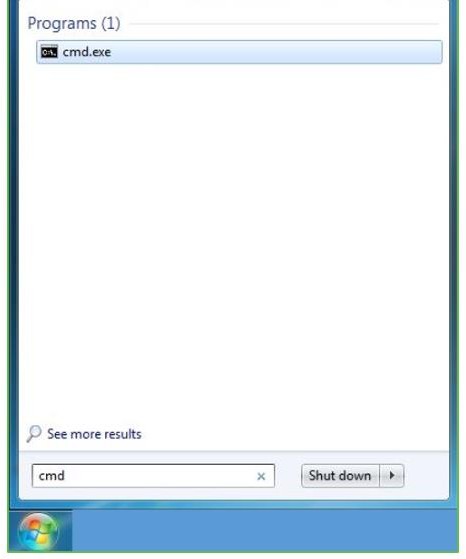
With these above three steps on python installation, you have successfully and correctly installed Python. Now is the time to verify the installation.

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

Verify the Python Installation

**Step 1:** Click on Start

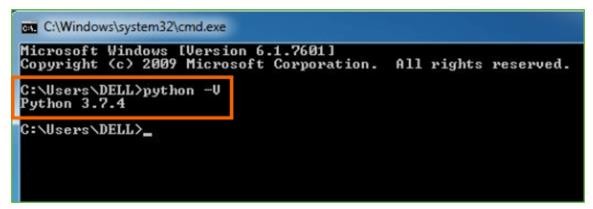
**Step 2:** In the Windows Run Command, type “cmd”.



#### Fig: 3.4.11 commend prompt window

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

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



**Fig: 3.4.12 commend prompt exe 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

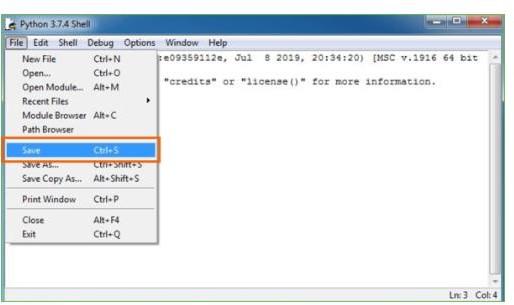
**Step 1:** Click on Start

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

**Fig: 3.4.13 Python IDLE application 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



**Fig: 3.4.14 Python shell**

**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**

#### 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 lists, dict, str and others

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

* Print is now a function
* Views and iterators instead of lists
* The rules for ordering comparisons have been simplified. 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, 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" behavior.
* Text Vs. Data Instead of Unicode Vs. 8-bit

#### Purpose:

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

#### Python:

Python is an interpreted high-level programming language for general-purpose programming. Created by Guido van Rossum and first released in 1991, Python has



a design philosophy that emphasizes code readability, notably using significant whitespace. Python features a dynamic type system and automatic memory management. It supports multiple programming paradigms, including object- oriented, imperative, functional and procedural, and has a large and comprehensive standard library.

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

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

#### Modules Used in Project:

**Tensor flow:**

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

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

#### NumPy:

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

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

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

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

#### Pandas:

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

#### Matplotlib:

Matplotlib is a Python 2D plotting library which produces publication quality figures in a variety of hardcopy formats and interactive environments across platforms. Matplotlib can be used in Python scripts, the Python and [I Python](http://ipython.org/) shells, the Jupiter Notebook, web application servers, and four graphical user interface toolkits. Matplotlib tries to make easy things easy and hard things possible. You can generate plots, histograms, power spectra, bar charts, error charts, scatter plots, etc.,

with just a few lines of code. For examples, see the [sample plots](https://matplotlib.org/tutorials/introductory/sample_plots.html) and [thumbnail](https://matplotlib.org/gallery/index.html) [gallery.](https://matplotlib.org/gallery/index.html)

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

#### Scikit – learn:

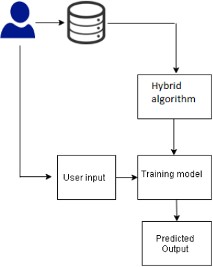
Scikit-learn provides a range of supervised and unsupervised learning algorithms via a consistent interface in Python. It is licensed under a permissive simplified BSD license and is distributed under many Linux distributions, encouraging academic and commercial use.

# CHAPTER-4

**SYSTEM DESIGN**

## 4.SYSTEM DESIGN

### SYSTEM ARCHITECTURE:



#### Fig: 4.1.1 SYSTEM ARCHITECTURE

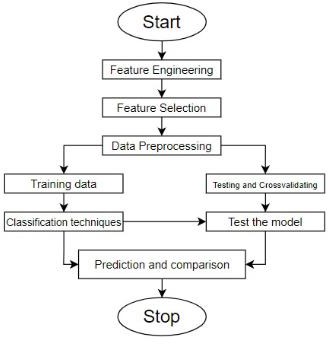
The major goal is to use an automated medical diagnosis tool to forecast cardiac disease using machine learning. The best classification approach for prediction is the hybrid model, which is what we are using. For processing the algorithm, I’ve picked the Cleveland dataset from Kaggle. Around 14 attributes and 303 instances are included in the dataset. I categorize it using a binary classification system, where “0” denotes the lack of heart illness and “1” indicates the existence of heart disease.

### 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.

* + 1. 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.
    2. 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.



**Fig: 4.2.1 DATA FLOW DIAGRAM**

### 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.

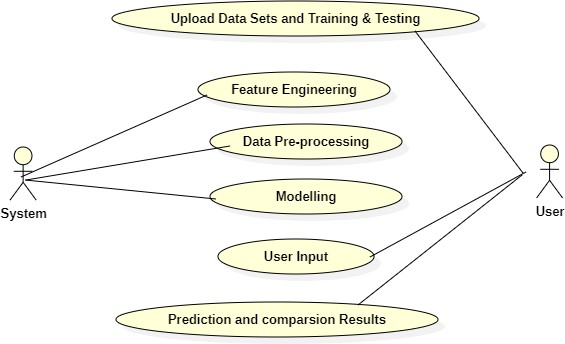
### GOALS:

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:

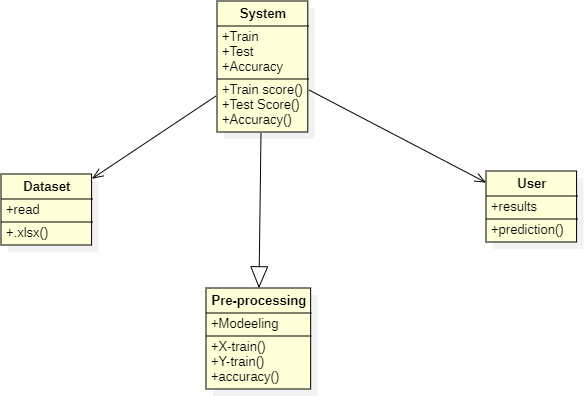
A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies 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.



**Fig: 4.4.1 Use case diagram**

### CLASS DIAGRAM:

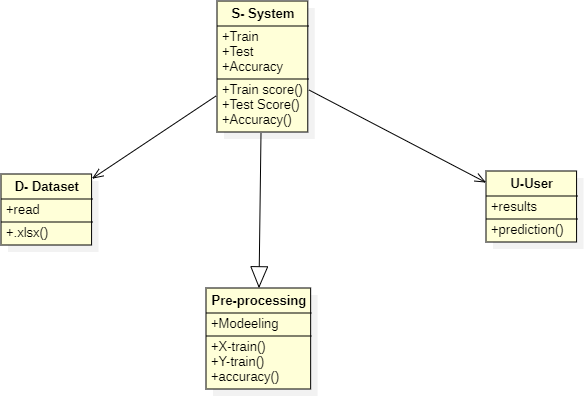
The class diagram is used to refine the use case diagram and define a detailed design of the system. The class diagram classifies the actors defined in the use case diagram into a set of interrelated classes. The relationship or association between the classes can be either an "is-a" or "has-a" relationship. 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.



**Fig: 4.5.1 Class diagram**

### OBJECT DIAGRAM:

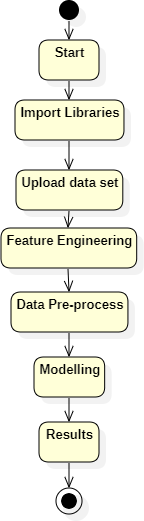
The object diagram is a special kind of class diagram. An object is an instance of a class. This essentially means that an object represents the state of a class at a given point of time while the system is running.



**Fig: 4.6.1 Object diagram**

### STATE DIAGRAM:

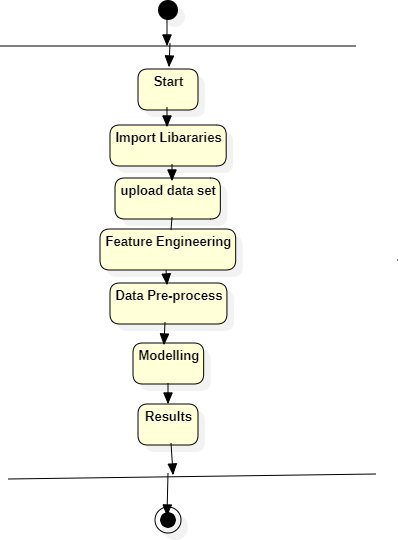
A state diagram, as the name suggests, represents the different states that objects in the system undergo during their life cycle. Objects in the system change states in response to events. In addition to this, a state diagram also captures the transition of the object's state from an initial state to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



**Fig: 4.7.1 State diagram**

### ACTIVITY DIAGRAM:

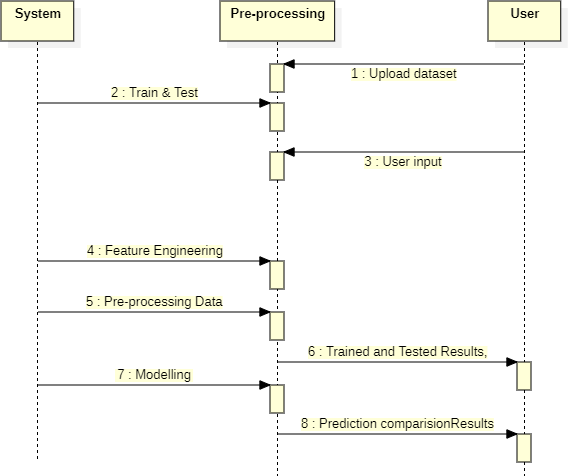
The process flows in the system are captured in the activity diagram. Similar to a state diagram, an activity diagram also consists of activities, actions, transitions, initial and final states, and guard conditions.



**Fig: 4.8.1 Activity Diagram**

### 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".



**Fig: 4.9.1 Sequence diagram**

### IMPLEMENTATION:

**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.

### 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 SYSTEM TESTING

## 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.

### 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 TESTING:

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 : 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 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.

### 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.

### TEST CASES:

#### 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.

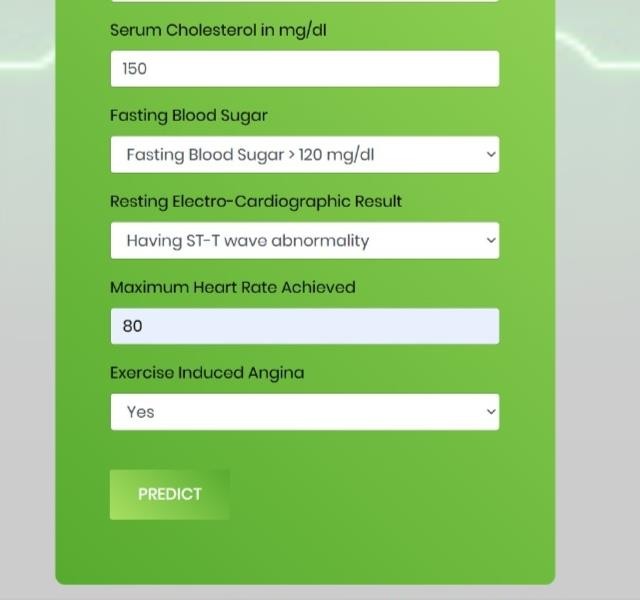
### Acceptance Testing:

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-6 RESULTS**

### 6.1 SCREEN SHOTS:



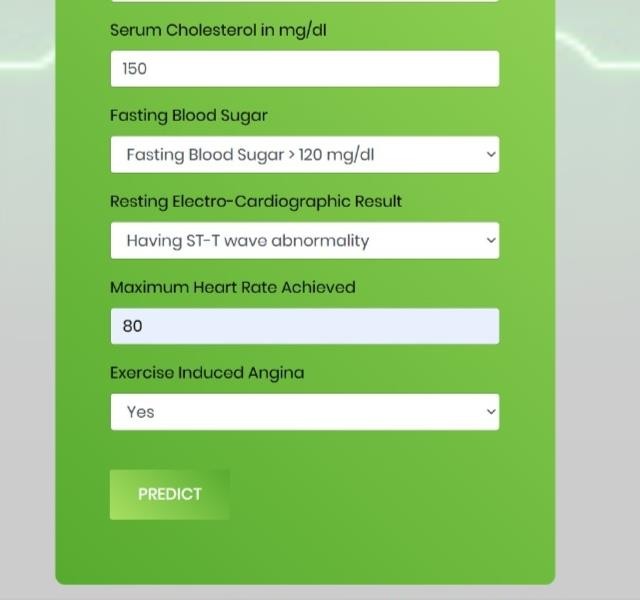
#### Fig:6.1.1 Screenshot during the execution code

In this page the user will give the inputs(like cholesterol,blood pressure,sugar, etc..) to the system.



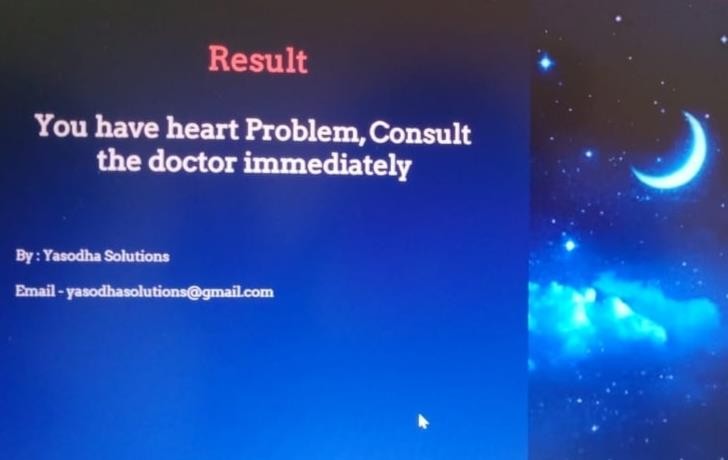
#### Fig:6.1.2 Screenshot of output predicting the cardiovascular disease

Once we give inputs to the system,the page will display like this.It display like the patient is safe or not.



#### Fig:6.1.3 Screenshot during the execution of code

In this page the user will give the inputs(like cholesterol,blood pressure,sugar, etc..) to the system.



#### Fig:6.1.4 Screenshot of output predicting the cardiovascular disease

Once we give inputs to the system,the page will display like this.It display like the patient is safe or not.

**CHAPTER-7 CONCLUSION**

## 7.CONCLUSION &FUTURE ENHANCEMENT

### 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%.

### FUTURE ENHANCEMENT:

In future, we can implement multiple feature selection technique that extracts optimal feature to develop models and design an application using real-time data from hospitals that will help the doctors recognise heart disease.

### REFERENCES:

1. Fahd Saleh Alotaibi, ˆaImplementation of Machine Learning Model to Predict Heart Failure Diseaseˆa 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 data mining 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 data mining 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 Research and. 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.”.