HEART DISEASE PREDICTION
USING
MACHINE LEARNING

A Project report submitted in partial fulfilment of the requirements for

the award of the degree of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

Submitted by

S. PAVAN KUMAR

Regd. No. 18811A0558

K. CHIRU CHAITANYA

Regd.No. 18811A0532

P. MANIKANTA

Regd.No. 18811A0551

G. ALEKHYA

Regd.No. 18811A0513

G. BHARGAVI

Regd.No. 18811A0522

Under the guidance of

Mr. V. TRINADH

Assistant Professor

Department of Computer Science and Engineering

AVANTHI INSTITUTE OF ENGINEERING & TECHNOLOGY

(*Approved by AICTE, Permanently Affiliated to JNTUK*)

(Accredited by NBA & NAAC and Recognized by UGC, New Delhi)

Tamaram, Makavarapalem Mandal, Visakhapatnam dist. (A.P) - 531113

2018-2022

ACKNOWLEDGEMENT

We would like to express our deep gratitude to our project guide **V. TRINADH** Assistant Professor, Department of Computer Science and Engineering, **AVANTHI INSTITUTE OF ENGINEERING & TECHNOLOGY** for guiding us through this project giving his valuable suggestions and helping us to overcome the difficulties faced during the design and coding stages of our project.

We are grateful to Head of the Department, **Dr. U. NANAJI**, MTech, PhD, Computer Science and Engineering, for providing us with the required facilities for the completion of the project work.

Our sincere thanks to **Dr C.P.V.N.J. MOHAN RAO**, M. Tech., PhD, Principal of **AVANTHI INSTITUTE OF ENGINEERING & TECHNOLOGY** for being a source inspiration and constantly encouraging us throughout the course to pursue new goals and ideas.

We express our thanks to Project Coordinator **P.V.S. PRABHAKAR**, for his continuous support and encouragement. We thank all **teaching faculty** of Department of CSE, whose suggestions during reviews helped us in accomplishment of our project.

We would like to thank our parents, friends, and classmates for their encouragement throughout our project period. At last, but not the least, we thank everyone for supporting us directly or indirectly in completing this project successfully.

PROJECT STUDENTS

SOOREDDY PAVAN KUMAR	18811A0558
KARANAM CHIRU CHAITANYA	18811A0532
POTHAMSETTI MANIKANTA	18811A0551
GOGINENI ALEKHYA	18811A0513
GURRAM BHARGAVI	18811A0522

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

AVANTHI INSTITUTE OF ENGINEERING & TECHNOLOGY

(*Approved by AICTE, Permanently Affiliated to JNTUK)*

(Accredited by NBA & NAAC and Recognized by UGC, New Delhi)

Tamaram, Makavarapalem Mandal, Visakhapatnam dist. (A.P) – 531113

CERTIFICATE

This is to certify that the project report entitled **“HEART DISEASE PREDICTION USING MACHINE LEARNING”** submitted by **S. Pavan Kumar (18811A0558), K. Chiru Chaitanya (18811A0532), P. Manikanta (18811A0551), G. Alekhya (18811A0513), G. Bhargavi (18811A0522)** in partial fulfilment of the requirements for the award of the degree of **Bachelor of Technology** in **Computer Science and Engineering** of **AVANTHI INSTITUTE OF ENGINEERING & TECHNOLOGY**, Makavarapalem is a record of bonafide work carried out under my guidance and supervision.

Project Guide **Head of the Department**

External Examiner

DECLARATION

We, **SOOREDDY PAVAN KUMAR, KARANAM CHIRUCHAITANYA, POTHAMSETTI MANIKANTA, GOGINENI ALEKHYA, GURRAM BHARGAVI** of final semester B.Tech., in the department of Computer Science and Engineering from AVANTHI INSTITUTE OF ENGINEERING & TECHNOLOGY, Visakhapatnam, hereby declare that the project work entitled **HEART DISEASE PREDICTION USING MACHINE LEARNING** is a bonafide work done by us in the year 2021-22 under the esteemed guidance of **Mr. V. Trinadh** and submitted for the partial fulfilment of the requirements for the award of **Bachelor of Technology in Computer Science and Engineering** from Jawaharlal Nehru Technological University Kakinada and has not been submitted to any other university for the award of any kind of degree.

S. PAVAN KUMAR

Regd. No. 18811A0558

K. CHIRU CHAITANYA	**P. MANIKANTA**
Regd. No. 18811A0532	**Regd. No. 18811A0551**
G. ALEKHYA	**G. BHARGAVI**
Regd. No. 18811A0513	**Regd. No. 18811A0522**

ABSTRACT

Machine Learning is used across many ranges around the world. The healthcare industry is no exclusion. Machine Learning can play an essential role in predicting presence/absence of locomotors disorders, heart diseases and more. Such information, if predicted well in advance, can provide important intuitions to doctors who can then adapt their diagnosis and dealing per patient basis. We work on predicting possible heart diseases in people using Machine Learning algorithms. In this project we perform the comparative analysis of classifiers like decision tree, Naïve Bayes, KNN, Logistic Regression, SVM and we propose an ensemble classifier which perform hybrid classification by taking strong and weak classifiers since it can have multiple number of samples for training and validating the data so we perform the analysis of existing classifier and proposed classifier like Random Forest which can give the better accuracy and predictive analysis.

Keywords: SVM; KNN; Naive Bayes; Decision Tree; Random Forest;

Logistic Regression; python programming; confusion matrix; correlation matrix.

INDEX

LIST OF FIGURES

LIST OF SCREENS

LIST OF TABLES

CHAPTER 1

INTRODUCTION

According to the World Health Organization, every year 12 million deaths occur worldwide due to heart disease. Heart disease is one of the biggest causes of morbidity and mortality among the population of the world. Prediction of cardiovascular disease is regarded as one of the most important subjects in the section of data analysis. The load of cardiovascular disease is rapidly increasing all over the world from the past few years. Many researches have been conducted in attempt to pinpoint the most influential factors of heart disease as well as accurately predict the overall risk. Heart Disease is even highlighted as a silent killer which leads to the death of the person without obvious symptoms. The early diagnosis of heart disease plays a vital role in making decisions on lifestyle changes in high-risk patients and in turn reduces the complications.

Machine learning proves to be effective in assisting in making decisions and predictions from the large quantity of data produced by the health care industry. This project aims to predict future heart disease by analyzing data of patients which classifies whether they have heart disease or not using machine-learning algorithm. Machine Learning techniques can be a boon in this regard. Even though heart disease can occur in different forms, there is a common set of core risk factors that influence whether someone will ultimately be at risk for heart disease or not. By collecting the data from various sources, classifying them under suitable headings & finally analyzing to extract the desired data we can say that this technique can be very well adapted to do the prediction of heart disease.

1.1 MOTIVATION FOR THE WORK

The main motivation of doing this research is to present a heart disease prediction model for the prediction of occurrence of heart disease. Further, this research work is aimed towards identifying the best classification algorithm for identifying the possibility of heart disease in a patient. This work is justified by performing a comparative study and analysis using three classification algorithms namely Naïve Bayes, Decision Tree, and Random Forest are used at different levels of evaluations. Although these are commonly used machine learning algorithms, the heart disease prediction is a vital task involving highest possible accuracy. Hence, the three algorithms are evaluated at numerous levels and types of evaluation strategies. This will provide researchers and medical practitioners to establish a better.

1.2 PROBLEM STATEMENT

The major challenge in heart disease is its detection. There are instruments available which can predict heart disease but either they are expensive or are not efficient to calculate chance of heart disease in human. Early detection of cardiac diseases can decrease the mortality rate and overall complications. However, it is not possible to monitor patients every day in all cases accurately and consultation of a patient for 24 hours by a doctor is not available since it requires more sapience, time and expertise. Since we have a good amount of data in today’s world, we can use various machine learning algorithms to analyze the data for hidden patterns. The hidden patterns can be used for health diagnosis in medicinal data.

CHAPTER 2
LITERATURE SURVEY

2.1 INTRODUCTION

With growing development in the field of medical science alongside machine learning various experiments and researches has been carried out in these recent years releasing the relevant significant papers.

Purushottam, et, al proposed a paper “Efficient Heart Disease Prediction System”
using hill climbing and decision tree algorithms. They used Cleveland dataset and preprocessing of data is performed before using classification algorithms. The Knowledge Extraction is done based on Evolutionary Learning (KEEL), an open-source data mining tool that fills the missing values in the data set. A decision tree follows top-down order. For each actual node selected by hill-climbing algorithm a node is selected by a test at each level. The parameters and their values used are confidence. Its minimum confidence value is 0.25. The accuracy of the system is about 86.7%.

Santhana Krishnan. J, et, al proposed a paper “Prediction of Heart Disease Using Machine Learning Algorithms” using decision tree and Naive Bayes algorithm for prediction of heart disease. In decision tree algorithm the tree is built using certain conditions which gives True or False decisions. The algorithms like SVM, KNN are results based on vertical or horizontal split conditions depends on dependent variables. But decision tree for a tree like structure having root node, leaves and branches base on the decision made in each of tree Decision tree also help in the understating the importance of the attributes in the dataset. They have also used Cleveland data set. Dataset splits in 70% training and 30% testing by using some methods. This algorithm gives 91% accuracy. The second algorithm is Naive Bayes, which is used for classification. It can handle complicated, nonlinear, dependent data so it is found suitable for heart disease dataset as this dataset is also complicated, dependent and nonlinear in nature. This algorithm gives an 87% accuracy.

Sonam Nikhar et al proposed paper “Prediction of Heart Disease Using Machine Learning Algorithms” their research gives point to point explanation of Naïve Bayes and decision tree classifier that are used especially in the prediction of Heart Disease. Some analysis has been led to think about the execution of prescient data mining strategy on the same dataset, and the result decided that Decision Tree has highest accuracy than Bayesian classifier.

Aditi Gavhane et al proposed a paper “Prediction of Heart Disease Using Machine Learning”, in which training and testing of dataset is performed by using neural network algorithm multi-layer perceptron. In this algorithm there will be one input layer and one output layer and one or more layers are hidden layers between these two input and output layers. Through hidden layers each input node is connected to output layer. This connection is assigned with some random weights. The other input is called bias which is assigned with weight based on requirement the connection between the nodes can be feedforwarded or feedback.

Avinash Golande et al, proposed “Heart Disease Prediction Using Effective Machine Learning Techniques” in which few data mining techniques are used that support the doctors to differentiate the heart disease. Usually utilized methodologies are k-nearest neighbors, Decision tree and Naïve Bayes. Other unique characterization-based strategies utilized are packing calculation, Part thickness, consecutive negligible streamlining and neural systems, straight Kernel self-arranging guide and SVM (Bolster Vector Machine).

Senthil Kumar Mohan et al, proposed “Effective Heart Disease Prediction Using Hybrid Machine Learning Techniques” in which their main objective is to improve exactness in cardiovascular problems. The algorithms used are KNN, LR, SVM, NN to produce an improved exhibition level with a precision level of 88.7% through the prediction model for heart disease with hybrid random forest with linear model (HRFLM).

2.2 EXISTING SYSTEM

Clinical decisions are often made based on doctor’s intuition and experience rather than on the knowledge rich data hidden in the database. This practice leads to unwanted biases, errors and excessive medical costs which affects the quality of service provided to patients. The National Patients Safety Foundation sites that 42% of medical patients feel they have had experienced a medical error or missed diagnosis.

DISADVANTAGES OF EXISTING SYSTEM

- ✓ Low Accuracy.
- ✓ Detection is not possible at an early stage.

2.3 PROPOSED SYSTEM

Machine Learning techniques can be a boon in this regard. Even though heart disease can occur in different forms, there is a common set of core risk factors that influence whether someone will ultimately be at risk for heart disease or not. The working of the system starts with the collection of data and selecting the important attributes. Then the required data is preprocessed into the required format. The data is then divided into two parts training and testing data. The algorithms are applied and the model is trained using the training data. The accuracy of the system is obtained by testing the system using the testing data. Random Forest has achieved an accuracy of 91% hence it is implemented in our system.

CHAPTER 3
REQUIREMENT ANALYSYS

3.1 INTRODUCTION

In the analysis phase, end user’s requirements are analyzed, and project goals get converted into the defined system functions that the organization intends to develop. The threeprimary activities involved in the analysis phase are as follows:

- ✓ Software Requirement Specification
- ✓ Creating content diagram
- ✓ Performing a detailed analysis by drawing a flowchart

Requirement Analysis will identify and consider the risks related to how the technology will be integrated into the standard operating procedures. Requirements Analysis will also collect the functional and system requirements of the business process, the user requirements, and the operational requirements. Hence this section deals with the software and hardware requirements with respect to our project.

3.2 SOFTWARE REQUIREMENT SPECIFICATION

Requirement Determination is the process by which an analyst gains the knowledge of the organization and applies it in selecting the right technology for an application.

A Software requirement specification (SRS) is a complete description of the behavior of the system to be developed. It includes a set of use cases that describe all the interactions the users will have with the software. Use case is also known as functional requirements. In addition to use cases, the SRS also contains non-functional (or supplementary requirements Non-functional requirements are requirements which impose constraints on the design and implementation (such as performance engineering requirements, quality standards, or design constraints).

3.2.1 FUNCTIONAL REQUIREMENTS

In software engineering, a functional requirement defines a function of a software system or its component. A function is described as a set of inputs, the behavior, and outputs. Functional requirements may be calculation, technical details, data manipulation and processing and other specific functionality that define what a system is supposed to accomplish. Behavioral requirements describing all the cases where the system uses the functional requirements are captured in use cases. Functional requirements are supported by non-functional requirements (also known as quality requirements), which impose constraints on the design or implementation such as performance requirements, reliability, and security. How a system implements functional requirements is detailed in the system design. In some cases, a requirements analyst generates use cases after gathering and validating a set of functional requirements. Often though, an analyst will begin by selecting a set of use cases, from which the analyst can derive the functional requirements that must be implemented to allow a user to perform each use case.

3.2.2 NON-FUNCTIONAL REQUIREMENTS

In systems engineering and requirements engineering a non-functional requirement is a requirement that specifies criteria that can be used to judge the operation of a system, rather than specific behaviors. This should be contrasted with functional requirements that define specific behavior or functions. In general, functional requirements define what a system is supposed to do whereas non-functional requirements are often called qualities of a system. Other terms for non-functional requirements are “constraints”, “quality goals” and “quality of service requirements”, and “non-behavioral requirements”. Quality that is non-functional requirements can be divided into two main categories.

3.2.3 SOFTWARE REQUIREMENTS

Software Requirements deal with defining software resource requirements and prerequisites that need to be installed on a computer to provide optimal functioning of an application. These requirements or pre-requisites are generally not included in the software installation package and need to be installed separately before the software is installed.

- ✓ **Operating System:** Windows 10
- ✓ **Software Used:** Anaconda Navigator, PyCharm IDLE, Jupyter Notebook.
- ✓ **Packages Used:** Flask, pandas, Matplotlib, Python, Sklearn.

3.2.4 HARDWARE REQUIREMENTS

The most common set of requirements defined by any operating system or software application is the physical computer resources also known as hardware. A hardware requirements list is often accompanied by a hardware compatibility list (HCL), especially in case of operating systems. An HCL lists tested, compatible hardware devices for a particular operating system or application. The following subsections discuss the various aspects of hardware requirements.

- ✓ **CPU type** **:** Intel i3 and above
- ✓ **RAM** **:** Minimum 4GB
- ✓ **Disk Space** **:** Minimum 5GB
- ✓ **System type** **:** 64-bit Operating system

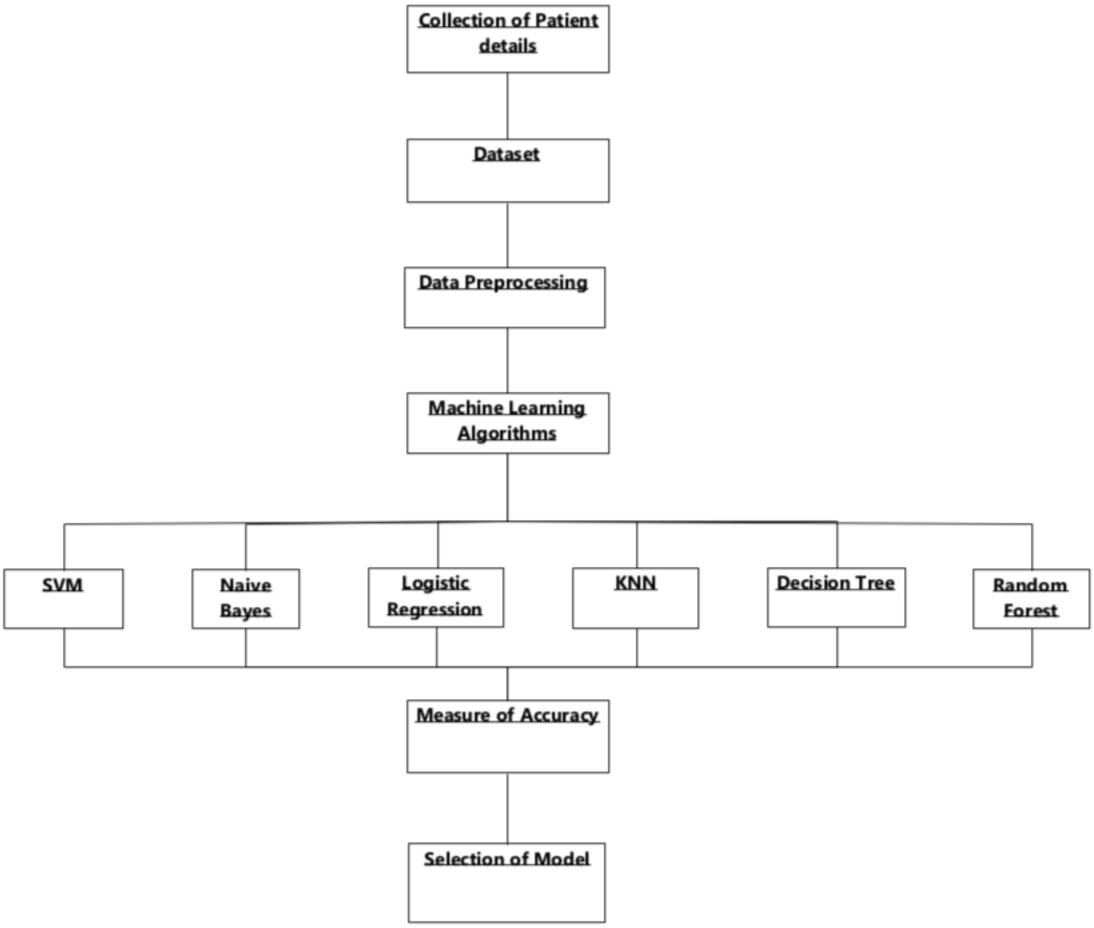
CHAPTER 4
WORKING OF SYSTEM

4.1 SYSTEM ARCHITECTURE

The system architecture gives an overview of the working of the system.

The working of this system is described as follows:

Dataset collection is collecting data which contains patient details. Attributes selection process selects the useful attributes for the prediction of heart disease. After identifying the available data resources, they are further selected, cleaned, made into the desired form. Different classification techniques as stated will be applied on preprocessed data to predict the accuracy of heart disease. Accuracy measure compares the accuracy of different classifiers.



4.2 DATA PRE-PROCESSING

The working of the system starts with the collection of data and selecting the important attributes. Then the required data is pre-processed into the required format. The data is then divided into two parts training and testing data. The algorithms are applied and the model is trained using the training data. The accuracy of the system is obtained by testing the system using the testing data. This system is implemented using the following modules.

1) Collection of Dataset

2) Selection of attributes

3) Data Pre-Processing

4) Balancing of Data

5) Disease Prediction

4.2.1 Collection of Dataset

Initially, we collect a dataset for our heart disease prediction system. After the collection of the dataset, we split the dataset into training data and testing data. The training dataset is used for prediction model learning and testing data is used for evaluating the prediction model. For this project, 70% of training data is used and 30% of data is used for testing. The dataset used for this project is Heart Disease UCI. The dataset consists of 76 attributes; out of which, 14 attributes are used for the system.

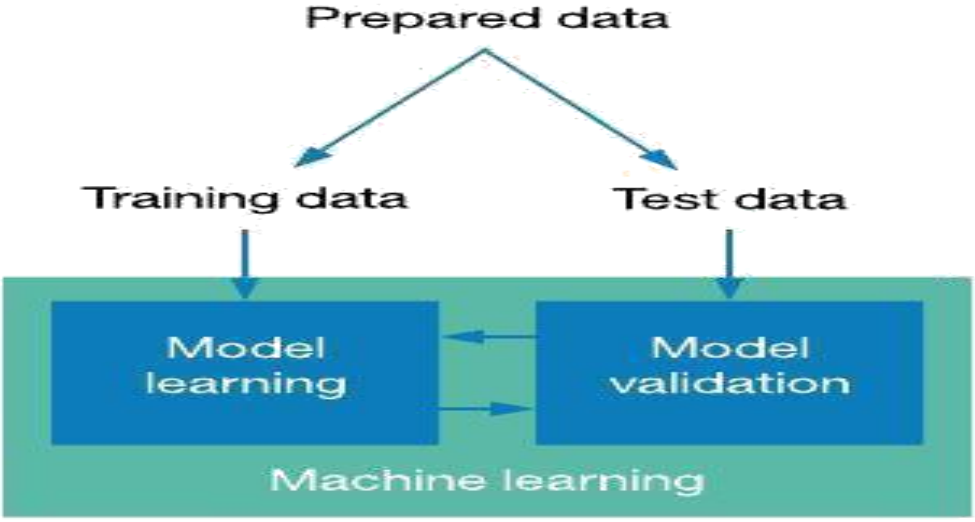
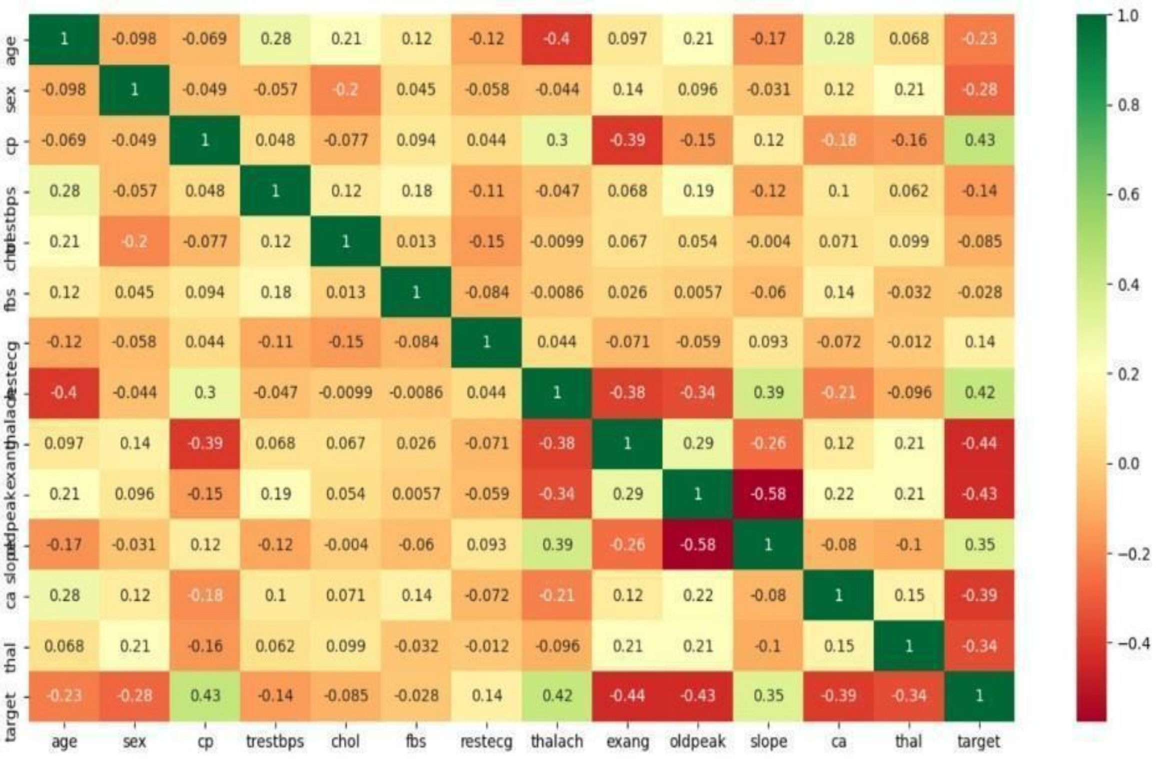


Fig: Collection of Data

4.2.2 Selection of attributes

Attribute or Feature selection includes the selection of appropriate attributes for the prediction system. This is used to increase the efficiency of the system. Various attributes of the patient like gender, chest pain type, fasting blood pressure, serum cholesterol, exang, etc. are selected for the prediction. The Correlation matrix is used for attribute selection for this model.



	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
age	1	-0.098	-0.069	0.28	0.21	0.12	-0.12	-0.4	0.097	0.21	-0.17	0.28	0.068	-0.23
sex	-0.098	1	-0.049	-0.057	-0.2	0.045	-0.058	-0.044	0.14	0.096	-0.031	0.12	0.21	-0.28
cp	-0.069	-0.049	1	0.048	-0.077	0.094	0.044	0.3	-0.39	-0.15	0.12	-0.18	-0.16	0.43
trestbps	0.28	-0.057	0.048	1	0.12	0.18	-0.11	-0.047	0.068	0.19	-0.12	0.1	0.062	-0.14
chol	0.21	-0.2	-0.077	0.12	1	0.013	-0.15	-0.0099	0.067	0.054	-0.004	0.071	0.099	-0.085
fbs	0.12	0.045	0.094	0.18	0.013	1	-0.084	-0.0086	0.026	0.0057	-0.06	0.14	-0.032	-0.028
restecg	-0.12	-0.058	0.044	-0.11	-0.15	-0.084	1	0.044	-0.071	-0.059	0.093	-0.072	-0.012	0.14
thalach	-0.4	-0.044	0.3	-0.047	-0.0099	-0.0086	0.044	1	-0.38	-0.34	0.39	-0.21	-0.096	0.42
exang	0.097	0.14	-0.39	0.068	0.067	0.026	-0.071	-0.38	1	0.29	-0.26	0.12	0.21	-0.44
oldpeak	0.21	0.096	-0.15	0.19	0.054	0.0057	-0.059	-0.34	0.29	1	-0.58	0.22	0.21	-0.43
slope	-0.17	-0.031	0.12	-0.12	-0.004	-0.06	0.093	0.39	-0.26	-0.58	1	-0.08	-0.1	0.35
ca	0.28	0.12	-0.18	0.1	0.071	0.14	-0.072	-0.21	0.12	0.22	-0.08	1	0.15	-0.39
thal	0.068	0.21	-0.16	0.062	0.099	-0.032	-0.012	-0.096	0.21	0.21	-0.1	0.15	1	-0.34
target	-0.23	-0.28	0.43	-0.14	-0.085	-0.028	0.14	0.42	-0.44	-0.43	0.35	-0.39	-0.34	1

Fig: Correlation matrix

4.2.3 Pre-processing of Data

Data pre-processing is an important step for the creation of a machine learning model. Initially, data may not be clean or in the required format for the model which can cause misleading outcomes. In pre-processing of data, we transform data into our required format. It is used to deal with noises, duplicates, and missing values of the dataset. Data pre-processing has the activities like importing datasets, splitting datasets, attribute scaling, etc. Pre-processing of data is required for improving the accuracy of the model.

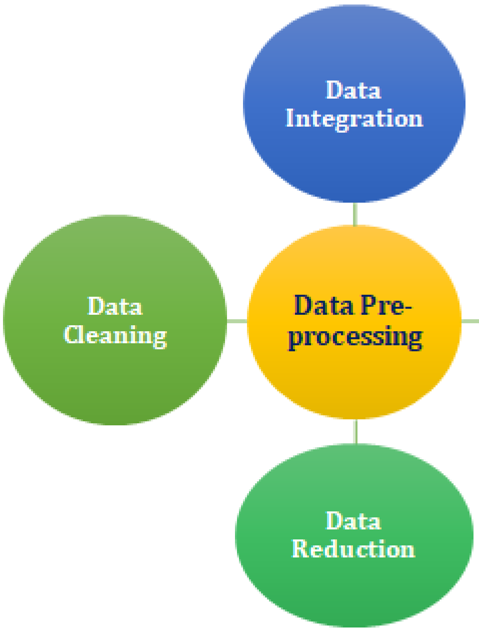


Fig: Data Pre-processing

4.2.4 Balancing of Data

Imbalanced datasets can be balanced in two ways. They are Under Sampling and Over Sampling.

(a) Under Sampling:

In Under Sampling, dataset balance is done by the reduction of the size of the sample class. This process is considered when the amount of data is adequate.

(b) Over Sampling:

In Over Sampling, dataset balance is done by increasing the size of the scarce samples.

This process is considered when the amount of data is inadequate.

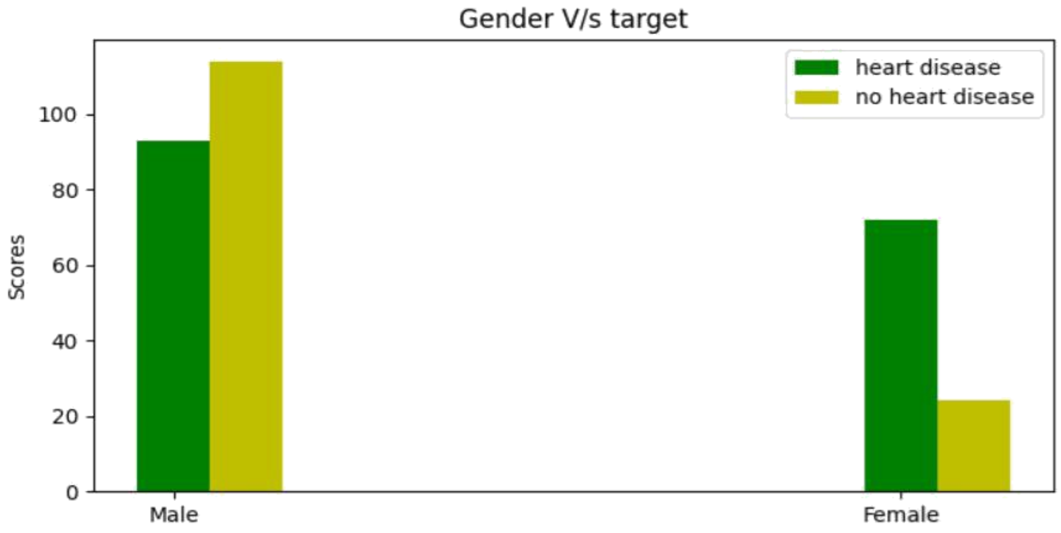


Fig: Data Balancing

4.2.5 Prediction of Disease

Various machine learning algorithms like SVM, Naive Bayes, Decision Tree, Random Forest, Logistic Regression are used for classification. Comparative analysis is performed among algorithms and the algorithm that gives the highest accuracy is used for heart disease prediction.

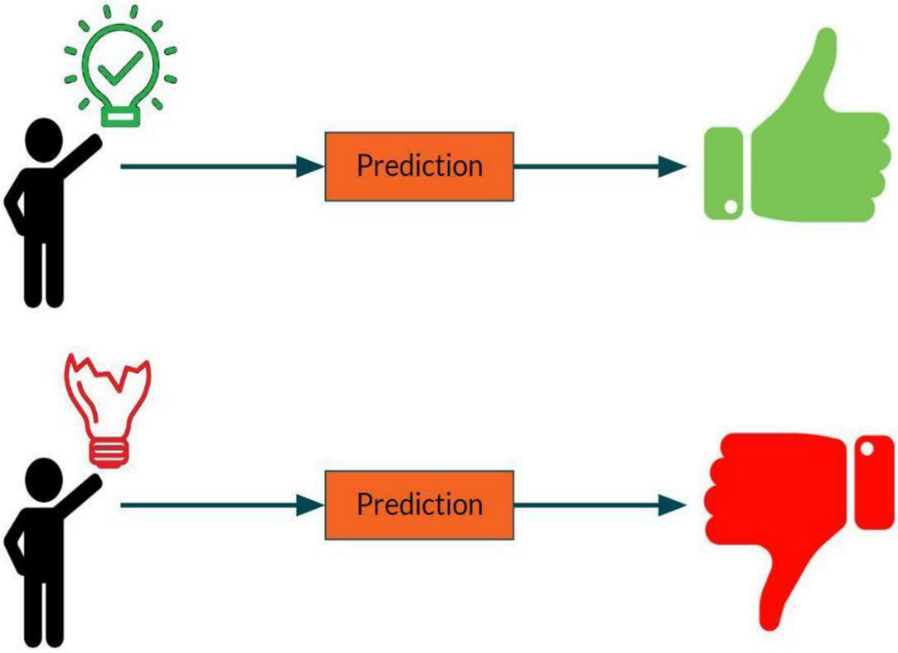
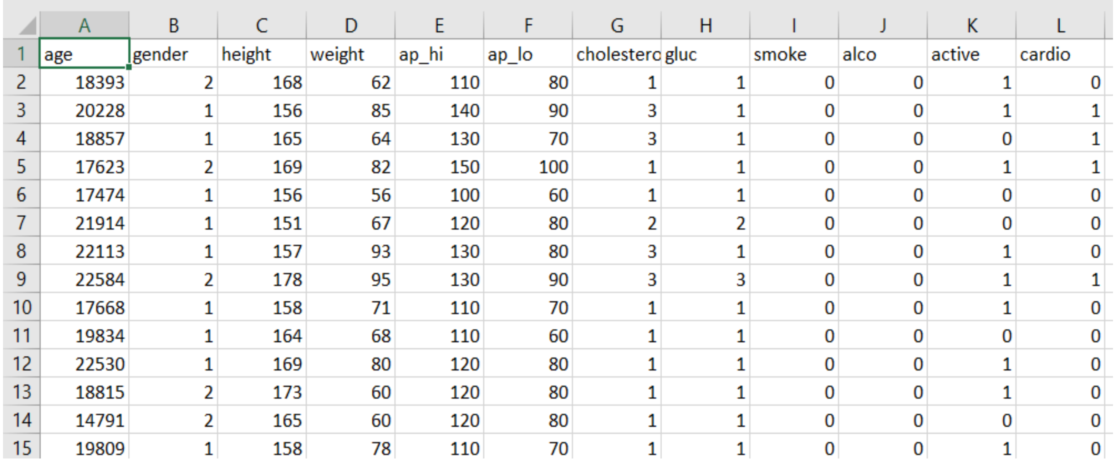


Fig: Prediction of Disease

4.3 DATASETS

Simple Prediction Dataset

- Source: https://www.kaggle.com/sulianova/cardiovascular-disease-dataset

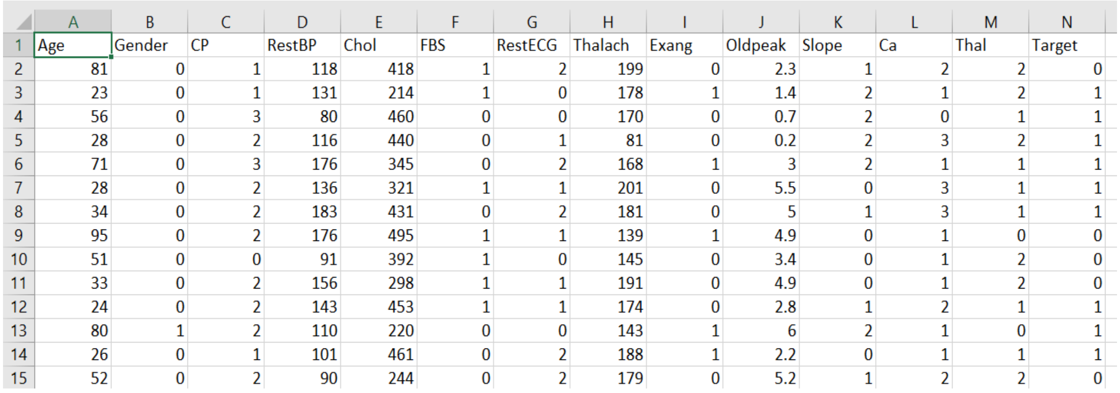


	A	B	C	D	E	F	G	H	I	J	K	L
1	age	gender	height	weight	ap_hi	ap_lo	cholesterc	gluc	smoke	alco	active	cardio
2	18393	2	168	62	110	80	1	1	0	0	1	0
3	20228	1	156	85	140	90	3	1	0	0	1	1
4	18857	1	165	64	130	70	3	1	0	0	0	1
5	17623	2	169	82	150	100	1	1	0	0	1	1
6	17474	1	156	56	100	60	1	1	0	0	0	0
7	21914	1	151	67	120	80	2	2	0	0	0	0
8	22113	1	157	93	130	80	3	1	0	0	1	0
9	22584	2	178	95	130	90	3	3	0	0	1	1
10	17668	1	158	71	110	70	1	1	0	0	1	0
11	19834	1	164	68	110	60	1	1	0	0	0	0
12	22530	1	169	80	120	80	1	1	0	0	1	0
13	18815	2	173	60	120	80	1	1	0	0	1	0
14	14791	2	165	60	120	80	1	1	0	0	0	0
15	19809	1	158	78	110	70	1	1	0	0	1	0

Advanced Prediction Dataset

- Of the 76 attributes available in the dataset,14 attributes are considered for the prediction of the output.

- Source: https://www.kaggle.com/datasets/johnsmith88/heart-disease-dataset



	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	Age	Gender	CP	RestBP	Chol	FBS	RestECG	Thalach	Exang	Oldpeak	Slope	Ca	Thal	Target
2	81	0	1	118	418	1	2	199	0	2.3	1	2	2	0
3	23	0	1	131	214	1	0	178	1	1.4	2	1	2	1
4	56	0	3	80	460	0	0	170	0	0.7	2	0	1	1
5	28	0	2	116	440	0	1	81	0	0.2	2	3	2	1
6	71	0	3	176	345	0	2	168	1	3	2	1	1	1
7	28	0	2	136	321	1	1	201	0	5.5	0	3	1	1
8	34	0	2	183	431	0	2	181	0	5	1	3	1	1
9	95	0	2	176	495	1	1	139	1	4.9	0	1	0	0
10	51	0	0	91	392	1	0	145	0	3.4	0	1	2	0
11	33	0	2	156	298	1	1	191	0	4.9	0	1	2	0
12	24	0	2	143	453	1	1	174	0	2.8	1	2	1	1
13	80	1	2	110	220	0	0	143	1	6	2	1	0	1
14	26	0	1	101	461	0	2	188	1	2.2	0	1	1	1
15	52	0	2	90	244	0	2	179	0	5.2	1	2	2	0

Fig: Dataset Attributes

Simple Prediction dataset attributes:

1) age – (in days)

2) gender – (value 1: Male,

   value 2: Female)

3) height – (in cm)

4) weight – (in kgs)

5) ap_hi – Systolic Blood Pressure

6) ap_lo – Diastolic Blood Pressure

7) cholesterol – (value 1: Normal,

   value 2: Above normal,

   value 3: Well above normal)

8) glucose – (value 1: Normal,

   value 2: Above normal,

   value 3: Well above normal)

9) smoke – is smoker? (value 0: No,

   value 1: Yes)

10) alco – is alcoholic? (value 0: No,

   value 1: Yes)

11) active – Physical Activity (value 0: No,

   value 1: Yes)

12) Cardio – (value 0: no disease,

   value 1: has disease)

Advanced Prediction dataset attributes:

1) Age – in years

2) Gender – (value 0: Female, Value 1: Male)

3) CP – Chest Pain Type (Value 0: Typical Angina,
   Value 1: Atypical Angina,
   Value 2: Non Anginal Pain,
   Value 3: Asymptomatic)

4) RestBP – resting blood pressure

5) Chol – serum cholesterol in mg/dl

6) FBS - Fasting Blood Sugar (value 0:< 120 mg/dl,
   value 1: > 120 mg/dl)

7) RestECG - resting electrocardiographic results (values - 0,1,2)

8) Thalach – maximum heart rate achieved

9) Exang – Exercised induced Angina (Value 0: No,
   Value 1: Yes)

10) old peak = ST depression induced by exercise relative to rest

11) Slope- the slope of the peak exercise ST segment (0 = up; 1 = flat; 2 = down)

12) Ca - number of major vessels (0-3) coloured by fluoroscopy

13) Thal - Thalassemia (0 = normal; 1 = fixed defect; 2 = reversable defect)

14) Target – (0 = no disease, 1 – has disease)

4.4 MACHINE LEARNING

In machine learning, classification refers to a predictive modelling problem where a class label is predicted for a given example of input data.

- **Supervised Learning**

Supervised learning is the type of machine learning in which machines are trained using well "labelled" training data, and on the basis of that data, machines predict the output. The labelled data means some input data is already tagged with the correct output.

In supervised learning, the training data provided to the machines work as the supervisor that teaches the machines to predict the output correctly. It applies the same concept as a student learns in the supervision of the teacher.

Supervised learning is a process of providing input data as well as correct output data to the machine learning model. The aim of a supervised learning algorithm is to find a mapping function to map the input variable(x) with the output variable(y).

- **Unsupervised learning**

Unsupervised learning cannot be directly applied to a regression or classification problem because unlike supervised learning, we have the input data but no corresponding output data. The goal of unsupervised learning is to find the underlying structure of dataset, group that data according to similarities, and represent that dataset in a compressed format.

✓ Unsupervised learning is helpful for finding useful insights from the data.

✓ Unsupervised learning is much similar to how a human learns to think by their own experiences, which makes it closer to the real AI.

✓ Unsupervised learning works on unlabeled and uncategorized data which make unsupervised learning more important.

✓ In real-world, we do not always have input data with the corresponding output so to solve such cases, we need unsupervised learning.

- **Reinforcement learning**

Reinforcement learning is an area of Machine Learning. It is about taking suitable action to maximize reward in a particular situation. It is employed by various software and machines to find the best possible behavior or path it should take in a specific situation. Reinforcement learning differs from supervised learning in a way that in supervised learning the training data has the answer key with it so the model is trained with the correct answer itself whereas in reinforcement learning, there is no answer but the reinforcement agent decides what to do to perform the given task. In the absence of a training dataset, it is bound to learn from its experience.

4.5 ALGORITHMS

4.5.1 SUPPORT VECTOR MACHINE (SVM):

Support Vector Machine or SVM is one of the most popular Supervised Learning algorithms, which is used for Classification as well as Regression problems. However, primarily, it is used for Classification problems in Machine Learning.

The goal of the SVM algorithm is to create the best line or decision boundary that can segregate n-dimensional space into classes so that we can easily put the new data point in the correct category in the future. This best decision boundary is called a hyperplane. SVM chooses the extreme points/vectors that help in creating the hyperplane. These extreme cases are called support vectors, and hence the algorithm is termed as Support Vector Machine.

Support vector machines (SVMs) are powerful yet flexible supervised machine learning algorithms which are used both for classification and regression. But generally, they are used in classification problems. In the 1960s, SVMs were first introduced but later they got refined in 1990. SVMs have their unique way of implementation as compared to other machine learning algorithms. Lately, they are extremely popular because of their ability to handle multiple continuous and categorical variables.

The followings are important concepts in SVM -

Support Vectors - Data Points that are closest to the hyperplane are called support vectors. Separating line will be defined with the help of these data points.

Hyperplane - As we can see in the above diagram, it is a decision plane or space which is divided between a set of objects having different classes.

Margin - It may be defined as the gap between two lines on the closest data points of different classes. It can be calculated as the perpendicular distance from the line to the support vectors. Large margin is considered as a good margin and small margin is considered as a bad margin.

Types of SVM:

SVM can be of two types:

- **Linear SVM:** Linear SVM is used for linearly separable data, which means if a dataset can be classified into two classes by using a single straight line, then such data is termed as linearly separable data, and classifier is used called as Linear SVM classifier.

- **Non-linear SVM:** Non-Linear SVM is used for non-linearly separated data, which means if a dataset cannot be classified by using a straight line, then such data is termed as non-linear data and classifier used is called as Non-linear SVM classifier.

The objective of the support vector machine algorithm is to find a hyperplane in an N-dimensional space (N - the number of features) that distinctly classifies the data points.

The advantages of support vector machines are:

✓ Effective in high dimensional spaces.

✓ Still effective in cases where the number of dimensions is greater than the number of samples.

✓ Uses a subset of training points in the decision function (called support vectors), so it is also memory efficient.

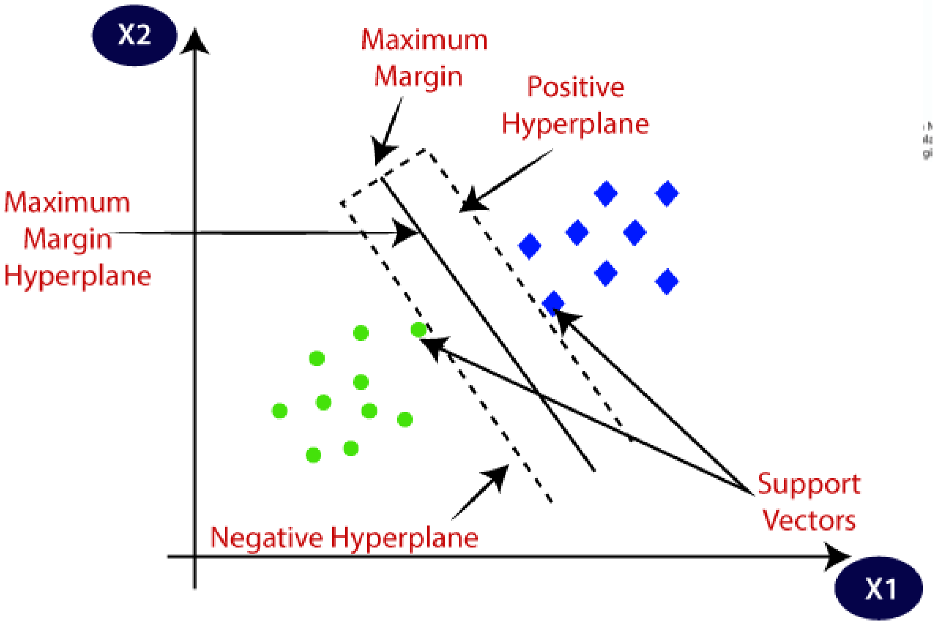


Fig: Support Vector Machine

✓ Versatile: different kernel functions can be specified for the decision function. Common kernels are provided, but it is also possible to specify custom kernels.

The disadvantages of support vector machines include:

✓ If the number of features is much greater than the number of samples, avoid over-fitting in choosing Kernel functions and regularization term is crucial.

✓ SVMs do not directly provide probability estimates, these are calculated using an expensive five-fold cross-validation.

4.5.2 NAÏVE BAYES ALGORITHM

Naive Bayes algorithm is a supervised learning algorithm, which is based on Bayes theorem and used for solving classification problems. It is mainly used in text classification that includes a high-dimensional training dataset.

Naive Bayes Classifier is one of the simple and most effective Classification algorithms which helps in building the fast machine learning models that can make quick predictions.

It is a probabilistic classifier, which means it predicts on the basis of the probability of an object. Some popular examples of Naïve Bayes Algorithm are spam filtration, Sentimental analysis, and classifying articles.

It is a classification technique based on Bayes’ Theorem with an assumption of independence among predictors. In simple terms, a Naive Bayes classifier assumes that the presence of a particular feature in a class is unrelated to the presence of any other feature.

The Naive Bayes model is easy to build and particularly useful for very large data sets. Along with simplicity, Naive Bayes is known to outperform even highly sophisticated classification methods.

The Naive Bayes algorithm is comprised of two words Naive and Bayes, Which can be described as:

- **Naive:** It is called Naive because it assumes that the occurrence of a certain feature is independent of the occurrence of other features. Such as if the fruit is identified on the basis of color, shape, and taste, then red, spherical, and sweet fruit is recognized as an apple. Hence each feature individually contributes to identify that it is an apple without depending on each other.

- **Bayes:** It is called Bayes because it depends on the principle of Bayes' Theorem.

Bayes’ theorem:

Bayes' theorem is also known as Bayes' Rule or Bayes' law, which is used to determine the probability of a hypothesis with prior knowledge. It depends on the conditional probability.

The formula for Bayes' theorem is given as:

$$\mathrm{P(A\,|\,B)} = \frac{P(B|A)P(A)}{P(B)}$$

Where,

P(A|B) is Posterior probability: Probability of hypothesis A on the observed event B.

P(B|A) is Likelihood probability: Probability of the evidence given that the probability of a hypothesis is true.

P(A) is Prior Probability: Probability of hypothesis before observing the evidence.

P(B) is Marginal Probability: Probability of Evidence.

Types of Naive Bayes model:

There are three types of Naive Bayes Model, which are given below:

- **Gaussian:** The Gaussian model assumes that features follow a normal distribution. This means if predictors take continuous values instead of discrete, then the model assumes that these values are sampled from the Gaussian distribution.

- **Multinomial:** The Multinomial Naïve Bayes classifier is used when the data is multinomial distributed. It is primarily used for document classification problems; it means a particular document belongs to which category such as Sports, Politics, education, etc. The classifier uses the frequency of words for the predictors.

- **Bernoulli:** The Bernoulli classifier works similar to the Multinomial classifier, but the predictor variables are the independent Booleans variables. Such as if a particular word is present or not in a document. This model is also famous for document classification tasks.

4.5.3 LOGISTIC REGRESSION ALGORITHM

Logistic regression is one of the most popular Machine Learning algorithms, which comes under the Supervised Learning technique. It is used for predicting the categorical dependent variable using a given set of independent variables.

Logistic regression predicts the output of a categorical dependent variable. Therefore, the outcome must be a categorical or discrete value. It can be either Yes or No, 0 or 1, true or False, etc. but instead of giving the exact value as 0 and 1, it gives the probabilistic values which lie between 0 and 1.

Logistic Regression is much similar to the Linear Regression except that how they are used. Linear Regression is used for solving Regression problems, whereas logistic regression is used for solving the classification problems.

In Logistic regression, instead of fitting a regression line, we fit an "S" shaped logistic function, which predicts two maximum values (0 or 1).

The curve from the logistic function indicates the likelihood of something such as whether the cells are cancerous or not, a mouse is obese or not based on its weight, etc.

Logistic Regression is a significant machine learning algorithm because it has the ability to provide probabilities and classify new data using continuous and discrete datasets.

Advantages:

Logistic Regression is one of the simplest machine learning algorithms and is easy to implement yet provides great training efficiency in some cases. Also due to these reasons, training a model with this algorithm doesn't require high computation power.

The predicted parameters (trained weights) give inference about the importance of each feature. The direction of association i.e., positive or negative is also given. So, we can use Logistic Regression to find out the relationship between the features.

This algorithm allows models to be updated easily to reflect new data, unlike Decision Tree or Support Vector Machine. The update can be done using stochastic gradient descent.

Logistic Regression outputs well-calibrated probabilities along with classification results. This is an advantage over models that only give the final classification as results. If a training example has a 95% probability for a class, and another has a 55% probability for the same class, we get an inference about which training examples are more accurate for the formulated problem.

Disadvantages:

Logistic Regression is a statistical analysis model that attempts to predict precise probabilistic outcomes based on independent features. On high dimensional datasets, this may lead to the model being over-fit on the training set, which means overstating the accuracy of predictions on the training set and thus the model may not be able to predict accurate results on the test set. This usually happens in the case when the model is trained on little training data with lots of features. So on high dimensional datasets, Regularization techniques should be considered to avoid over-fitting (but this makes the model complex). Very high regularization factors may even lead to the model being under-fit on the training data.

Nonlinear problems can't be solved with logistic regression since it has a linear decision surface. Linearly separable data is rarely found in real world scenarios. So the transformation of nonlinear features is required which can be done by increasing the number of features such that the data becomes linearly separable in higher dimensions.

Non-Linearly Separable Data:

It is difficult to capture complex relationships using logistic regression. More powerful and complex algorithms such as Neural Networks can easily outperform this algorithm.

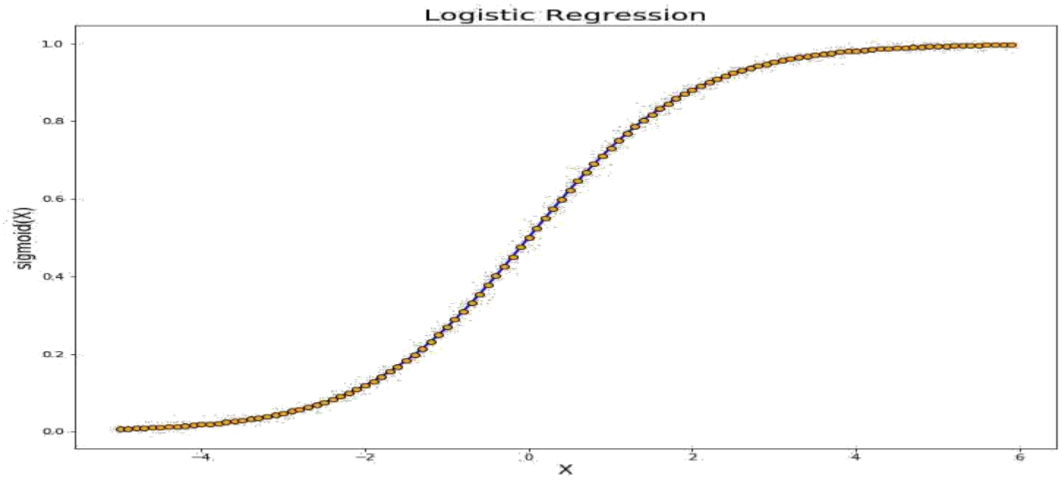


Fig: Logistic Regression

4.5.4 K - NEAREST NEIGHBORS ALGORITHM

- K-Nearest Neighbors is one of the simplest Machine Learning algorithms based on Supervised Learning technique.
- K-NN algorithm assumes the similarity between the new case/data and available cases and put the new case into the category that is most similar to the available categories.
- K-NN algorithm stores all the available data and classifies a new data point based on the similarity. This means when new data appears then it can be easily classified into a well suite category by using K- NN algorithm.
- K-NN algorithm can be used for Regression as well as for Classification but mostly it is used for the Classification problems.
- K-NN is a **non-parametric algorithm**, which means it does not make any assumption on underlying data.
- It is also called a **lazy learner algorithm** because it does not learn from the training set immediately instead it stores the dataset and at the time of classification, it performs an action on the dataset.
- KNN algorithm at the training phase just stores the dataset and when it gets new data, then it classifies that data into a category that is much similar to the new data.

The K-NN working can be explained on the basis of the below algorithm:

- **Step-1:** Select the number K of the neighbors
- **Step-2:** Calculate the Euclidean distance of **K number of neighbors**
- **Step-3:** Take the K nearest neighbors as per the calculated Euclidean distance.
- **Step-4:** Among these k neighbors, count the number of the data points in each category.
- **Step-5:** Assign the new data points to that category for which the number of the neighbors is maximum.
- **Step-6:** Our model is ready.

Advantages:

- ✓ It is simple to implement.
- ✓ It is robust to the noisy training data
- ✓ It can be more effective if the training data is large.

Disadvantages:

- ✓ Always needs to determine the value of K which may be complex some time.
- ✓ The computation cost is high because of calculating the distance between the data points for all the training samples.

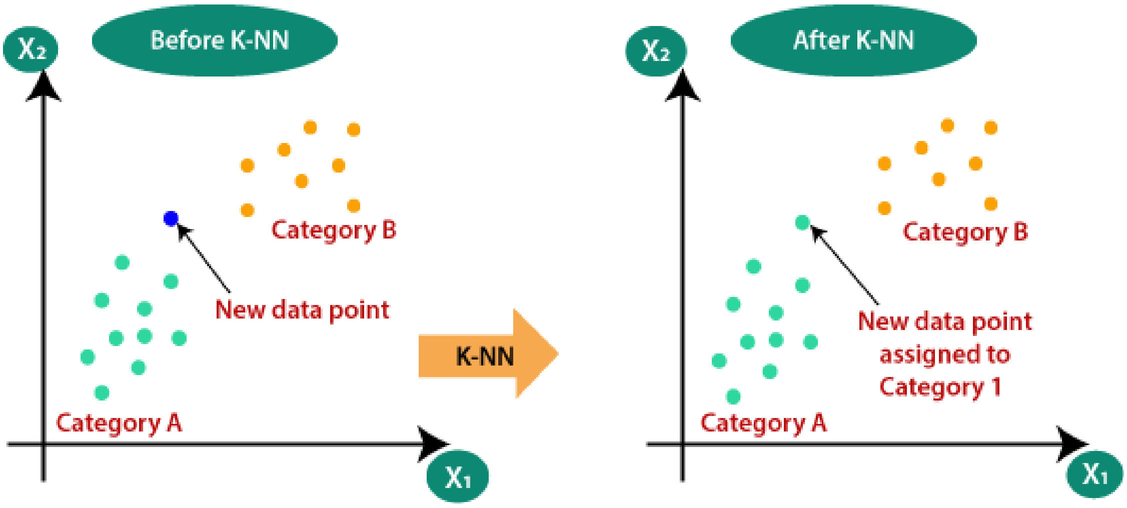


Fig: KNN Algorithm

4.5.5 DECISION TREE ALGORITHM

Decision Tree is a Supervised learning technique that can be used for both classification and regression problems, but mostly it is preferred for solving classification problems. It is a tree-structured classifier, where internal nodes represent the features of a dataset, branches represent the decision rules and each leaf node represents the outcome. In a Decision Tree, there are two nodes, which are the Decision Node and Leaf Node.

Decision nodes are used to make any decision and have multiple branches, whereas Leaf nodes are the output of those decisions and do not contain any further branches. The decisions or the test are performed on the basis of features of the given dataset. It is a graphical representation for getting all the possible solutions to a problem/decision based on given conditions. It is called a Decision Tree because, similar to a tree, it starts with the root node, which expands on further branches and constructs a tree-like structure. In order to build a tree, we use the CART algorithm, which stands for Classification and Regression Tree algorithm. A Decision Tree simply asks a question, and based on the answer (Yes/No), it further split the tree into subtrees.

The Decision Tree Algorithm belongs to the family of supervised machine learning algorithms. It can be used for both a classification problem as well as for a regression problem.

The goal of this algorithm is to create a model that predicts the value of a target variable, for which the decision tree uses the tree representation to solve the problem in which the leaf node corresponds to a class label and attributes are represented on the internal node of the tree.

There are various algorithms in Machine learning, so choosing the best algorithm for the given dataset and problem is the main point to remember while creating a machine learning model. Below are the two reasons for using the Decision Tree:

- Decision Trees usually mimic human thinking ability while making a decision, so it is easy to understand.
- The logic behind the decision tree can be easily understood because it shows a tree-like structure.

In Decision Tree the major challenge is to identify the attribute for the root node in each level. This process is known as attribute selection. We have two popular attribute selection measures:

1. Information Gain:

When we use a node in a Decision Tree to partition the training instances into smaller subsets, the entropy changes. Information gain is a measure of this change in entropy.

Entropy is the measure of uncertainty of a random variable, it characterizes the impurity of an arbitrary collection of examples.

The higher the entropy the more the information content.

2. Gini Index:

Gini Index is a metric to measure how often a randomly chosen element would be incorrectly identified. It means an attribute with lower Gini index should be preferred. Sklearn supports "Gini" criteria for Gini Index and by default, it takes "gini" value.

The most notable types of Decision Tree algorithms are: -

1. **IDichotomiser 3 (ID3):**

   This algorithm uses Information Gain to decide which attribute is to be used to classify the current subset of the data. For each level of the tree, information gain is calculated for the remaining data recursively.

2. **C4.5:** This algorithm is the successor of the ID3 algorithm. This algorithm uses either Information gain or Gain ratio to decide upon the classifying attribute. It is a direct improvement from the ID3 algorithm as it can handle both continuous and missing attribute values.

3. **Classification and Regression Tree (CART):** It is a dynamic learning algorithm which can produce a regression tree as well as a classification tree depending upon the dependent variable.

Working:

In a Decision Tree, for predicting the class of the given dataset, the algorithm starts from the root node of the tree. This algorithm compares the values of the root attribute with the record (real dataset) attribute and, based on the comparison, follows the branch and jumps to the next node.

For the next node, the algorithm again compares the attribute value with the other sub-nodes and moves further. It continues the process until it reaches the leaf node of the tree. The complete process can be better understood using the below algorithm:

- **Step-1:** Begin the tree with the root node, says S, which contains the complete dataset.
- **Step-2:** Find the best attribute in the dataset using Attribute Selection Measure (ASM).
- **Step-3:** Divide the S into subsets that contains possible values for the best attributes.
- **Step-4:** Generate the Decision Tree node, which contains the best attribute.
- **Step-5:** Recursively make new decision trees using the subsets of the dataset created in step -3. Continue this process until a stage is reached where you cannot further classify the nodes and call the final node as a leaf node.

4.5.6 RANDOM FOREST ALGORITHM

Random Forest is a supervised learning algorithm. It is an extension of machine learning classifiers which include the bagging to improve the performance of Decision Tree. It combines tree predictors, and trees are dependent on a random vector which is independently sampled. The distribution of all trees are the same. Random Forests splits nodes using the best among of a predictor subset that are randomly chosen from the node itself, instead of splitting nodes based on the variables. The time complexity of the worst case of learning with Random Forests is O(M(dnlogn)), where M is the number of growing trees, n is the number of instances, and d is the data dimension.

It can be used both for classification and regression. It is also the most flexible and easy to use algorithm. A forest consists of trees. It is said that the more trees it has, the more robust a forest is. Random Forests create Decision Trees on randomly selected data samples, get predictions from each tree and select the best solution by means of voting. It also provides a pretty good indicator of the feature importance.

Random Forests have a variety of applications, such as recommendation engines, image classification and feature selection. It can be used to classify loyal loan applicants, identify fraudulent activity and predict diseases. It lies at the base of the Boruta algorithm, which selects important features in a dataset.

Random Forest is a popular machine learning algorithm that belongs to the supervised learning technique. It can be used for both Classification and Regression problems in ML. It is based on the concept of ensemble learning, which is a process of combining multiple classifiers to solve a complex problem and to improve the performance of the model.

As the name suggests, "Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset." Instead of relying on one decision tree, the random forest takes the prediction from each tree and based on the majority votes of predictions, and it predicts the final output.

The greater number of trees in the forest leads to higher accuracy and prevents the problem of overfitting.

Assumptions:

Since the random forest combines multiple trees to predict the class of the dataset, it is possible that some decision trees may predict the correct output, while others may not. But together, all the trees predict the correct output. Therefore, below are two assumptions for a better Random Forest classifier:

- There should be some actual values in the feature variable of the dataset so that the classifier can predict accurate results rather than a guessed result.
- The predictions from each tree must have very low correlations.

Algorithm:

- **Step-1:** Select random K data points from the training set.
- **Step-2:** Build the decision trees associated with the selected data points (Subsets).
- **Step-3:** Choose the number N for decision trees that you want to build.
- **Step-4:** Repeat Step 1 & 2.
- **Step-5:** For new data points, find the predictions of each decision tree, and assign the new data points to the category that wins the majority votes.

Advantages:

- ✓ Random Forest is capable of performing both Classification and Regression tasks.
- ✓ It is capable of handling large datasets with high dimensionality.
- ✓ It enhances the accuracy of the model and prevents the overfitting issue.

Disadvantages:

Although Random Forest can be used for both classification and regression tasks, it is not more suitable for Regression tasks.

CHAPTER 5

DESIGN

5.1 INTRODUCTION

Software Design is a process of planning the new or modified system. The design step produces the understanding and procedural details necessary for implementing the system recommended in the feasibility study. Analysis specifies what a new or modified system does. Design specifies how to accomplish the same. Design is essentially a bridge between requirement specification and the final solution satisfying the requirement. It is a blueprint or a solution for the system. The design step produces a data design, an architectural design and procedural design.

The design process for a software system has two levels. At first level the focus is on depending on which modules are needed for the system, the specification of these modules and how the modules should be interconnected. This is what is called system designing of top-level design. In the second level, the internal design of the modules, or how the specification shouldbe interconnected.

TOP LEVEL DESIGN

It is the first level of the design which produces the system design, which defines the components needed for the system, and how the components interact with each other. Its focus is on depending on which the modules are needed for the system; the specification of these modules should be interconnected.

LOGIC DESIGN

The logic design of this application shows the major features and how they are related to one another. The detailed specifications are drawn on the bases of user requirements. The outputs, inputs and relationship between the variables are designed in this phase.

INPUT DESIGN

The input design is the bridge between users and the information system. It specifies the manner in which the data enters the system for processing. It can ensure the reliability of the system and produces reports from accurate data, or it may result in the output of error information. While designing the following points have to be taken into consideration: Input Formats are designed as per user requirements.

- ✓ Interaction with the user is maintained in simple dialogues.

Appropriate fields are designed with validations there by allowing valid inputs

OUTPUT DESIGN

Each and every presented in the system is result-oriented. The most important feature of this application for users is the output. Efficient output design improves the usability and acceptability of the system and also helps in decision making. Thus, the following points are considered during the output design:

- ✓ What information to be present?
- ✓ How to display the results?

DATA DESIGN

Data design is the first of the three design activities that are conducted during software engineering. The Impact of data structure on program structure and procedural complexity causes data design to have a profound influence on software quality.

ARCHITECTURAL DESIGN

The architectural design defines the relationship among major structural components intoa procedural description of the software.

- ✓ The use cases are the functions that are to be performed in the module.
- ✓ An actor could be an end-user of the system or an external system.

SYSTEM DESIGN

System design is the process of defining the architecture, components, modules, interfaces, and data for a system to satisfy the specified requirements. System design could seeits application of systems theory to product development. There is an overlap with the disciplines of system analysis, system architecture and system engineering.

Grady Booch, James Rumbaugh, and Ivor Jacobson have collaborated to combine the best features of their individual object-oriented analysis and design methods into a unified methodthe unified modeling language, the version 1.0 for the Unified modeling was released in January 1997 the main parts of UML are based on the Brooch, OMT and OOSE methods.

The goals of UML are:

- ✓ To model systems using object-oriented concepts
- ✓ To establish an explicit coupling between conceptual as well as executable
- ✓ To address the issues of scale inherent in complex, mission critical system
- ✓ To create a modeling language usable by both humans and machines.

5.2 UML DIAGRAMS:

BUILDING BLOCKS OF UML:

The vocabulary of the UML encompasses three kinds of building blocks:

- ✓ **Things** are the abstractions that are first-class citizens in a model.
- ✓ **Relationships** tie these things together.
- ✓ **Diagrams** group interesting collections of things.

There are four kinds of things in the UML - Structural things, Behavioral things, Groupingthings, Annotational things.

STRUCTURAL THINGS

Structural things are the nouns of UML models. These are the mostly static parts of a model,representing elements that are either conceptual or physical. In all, there are seven kinds of structural things.

✓ A class is a description of a set of objects that share the same attributes, operations, relationships, and semantics. A class implements one or more interfaces. Graphically,a class is rendered as a rectangle, usually including its name, attributes, and operations.

✓ An interface is a collection of operations that specify a service of a class or component. An interface therefore describes the externally visible behavior of that element. An interface might represent the complete behavior of a class or component or only a partof that behavior.

✓ Collaboration defines an interaction and is a society of roles and other elements that work together to provide some cooperative behavior that's bigger than the sum of all the elements. Therefore, collaborations have structural, as well as behavioral, dimensions. Graphically, a collaboration is rendered as an ellipse with dashed lines, usually including only its name.

✓ Use case is a description of set of sequence of actions that a system performs that yields an observable result of value to a particular actor. A use case is used to structure the behavioral things in a model. Graphically, a use case is rendered as an ellipse with solidlines, usually including only its name.

✓ An active class is a class whose objects own one or more processes or threads and therefore can initiate control activity. An active class is just like a class except that its objects represent elements whose behavior is concurrent with other elements. Graphically, an active class is rendered just like a class, but with heavy lines, usually including its name, attributes, and operations.

✓ A component is a physical and replaceable part of a system that conforms to and provides the realization of a set of interfaces. Graphically, a component is rendered asa rectangle with tabs, usually including only its name.

✓ A node is a physical element that exists at run time and represents a computational resource, generally having at least some memory and, often, processing capability. A set of components may reside on a node and may also migrate from node to node. Graphically, a node is rendered as a cube, usually including only its name.

BEHAVIORAL THINGS

✓ Behavioral things are the dynamic parts of UML models. These are the verbs of a model, representing behavior over time and space. In all, there are two primary kinds of behavioral things.

✓ An interaction is a behavior that comprises a set of messages exchanged among a set of objects within a particular context to accomplish a specific purpose.

✓ A state machine is a behavior that specifies the sequences of states an object or an interaction goes through during its lifetime in response to events, together with its responses to those events. A state machine involves a number of other elements, including states, transitions (the flow from state to state), events (things that trigger a transition), and activities (the response to a transition). Graphically, a state is renderedas a rounded rectangle, usually including its name and its substrates, if any.

GROUPING THINGS

✓ Grouping things are the organizational parts of UML models. These are the boxes into which a model can be decomposed. In all, there is one primary kind of grouping thing, namely, packages.

✓ A package is a general-purpose mechanism for organizing elements into groups.

✓ Structural things, behavioral things, and even other grouping things may be placed in a package. Unlike components, a package is purely conceptual. Graphically, a package is rendered as a tabbed folder, usually including only itsname and, sometimes, its contents.

ANNOTATIONAL THINGS

✓ Annotational things are the explanatory parts of UML models. These are the comments you may apply to describe, illuminate, and remark about any element in a model. There is one primary kind of annotational thing, called a note.

✓ A note is simply a symbol for rendering constraints and comments attached to anelement or a collection of elements. Graphically, a note is rendered as a rectanglewith a dog-eared corner, together with a textual or graphical comment.

RELATIONSHIPS IN THE UML

There are four kinds of relationships in the UML – Dependency, Association, Generalization,Realization

These relationships are the basic relational building blocks of the UML. We use them towrite well-formed models.

- ✓ A **dependency** is a semantic relationship between two things in which a change toone thing may affect the semantics of the other thing (the dependent thing).
- ✓ An **association** is a structural relationship that describes a set of links, a link being a connection among objects.
- ✓ A **generalization** is a specialization/generalization relationship in which objects of the specialized element (the child) are substitutable for objects of the generalized element (the parent).
- ✓ A **realization** is a semantic relationship between classifiers, wherein one classifier specifies a contract that another classifier guarantees to carry out.

Some of the Diagrams that help for the Diagrammatic Approach for the Object-Oriented Software Engineering are - Class Diagram**,** Use Case Diagram, Sequence Diagram**,** Collaboration Diagram, State Chart Diagram. Using the above-mentioned diagrams, we can show the entire system regarding the working of the system or the flow of control and sequenceof flow, the state of the system and the activities involved in the system.

5.2.1 CLASS DIAGRAM

Class diagram is a static diagram. It represents the static view of an application. Class diagram is not only used for visualizing, describing, and documenting different aspects of a system but also for constructing executable code of the software application. Class diagram describes the attributes and operations of a class and al so the constraints imposed on the system. The class diagrams are widely used in the modeling of object-oriented systems because they are the only UML diagrams, which can be mapped directly with object-oriented languages. Class diagram shows a collection of classes, interfaces, associations, collaborations, and constraints. It is also known as a structural diagram. The Class diagram represents the conceptual structure of the application.

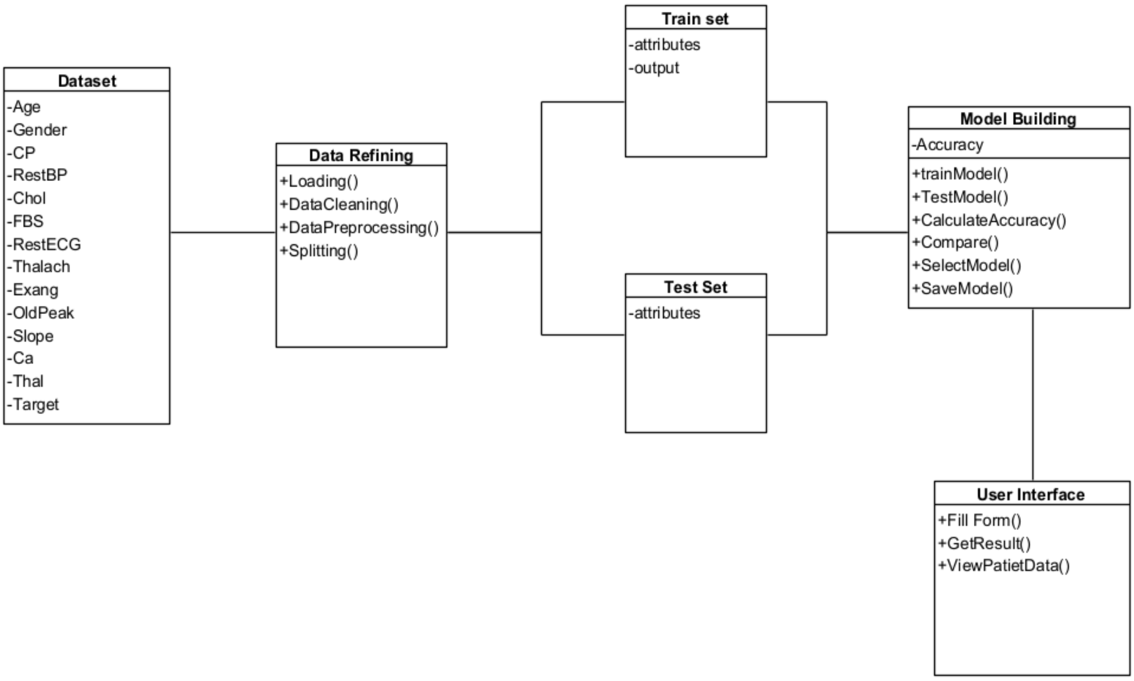


Fig: Class Diagram

5.2.2 USECASE DIAGRAM

A use case diagram is a dynamic or behavior diagram in UML. Use case diagrams model the functionality of a system using actors and use cases. It encapsulates the system's functionality by incorporating use cases, actors, and their relationships. It models the tasks, services, and functions required by a system/subsystem of an application.

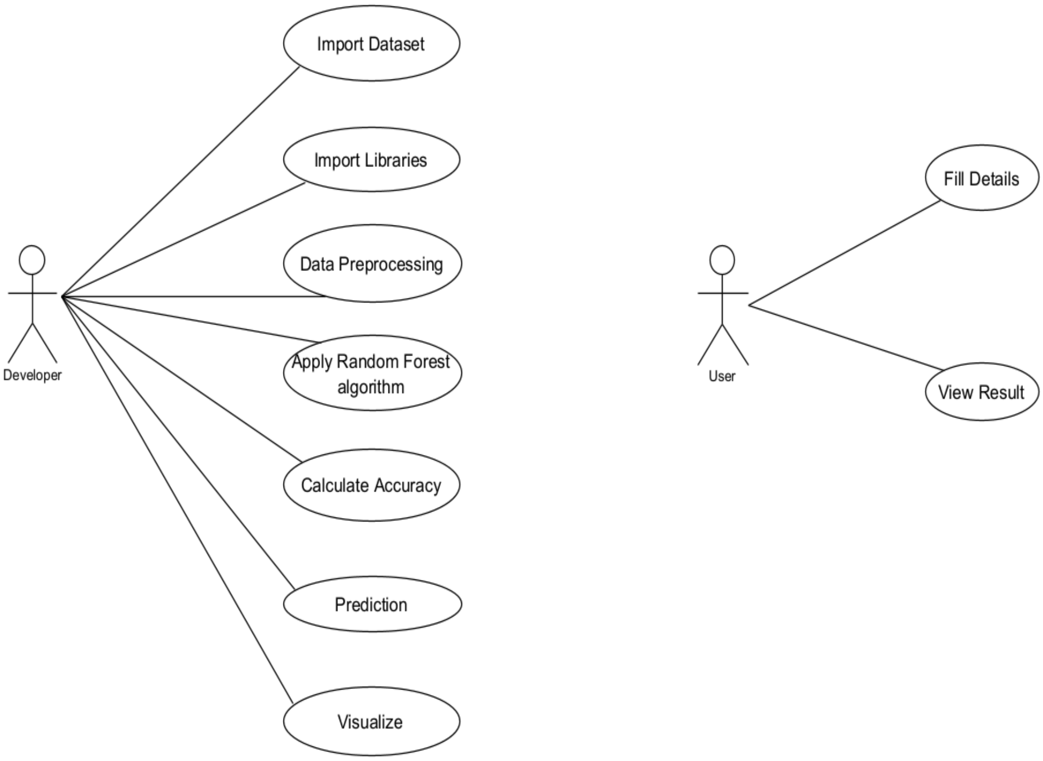


Fig: Usecase diagram

5.2.3 SEQUENCE DIAGRAM

Sequence diagrams describe interactions among classes in terms of an exchange of messages over time. They are also called event diagrams. A sequence diagram is a good way to visualize and validate various runtime scenarios. These can help to predict how a system will behave and to discover responsibilities a class may need to have in the process of modelling a new system.

The sequence diagram depicts the order or sequence in which the events take place in theapplication.

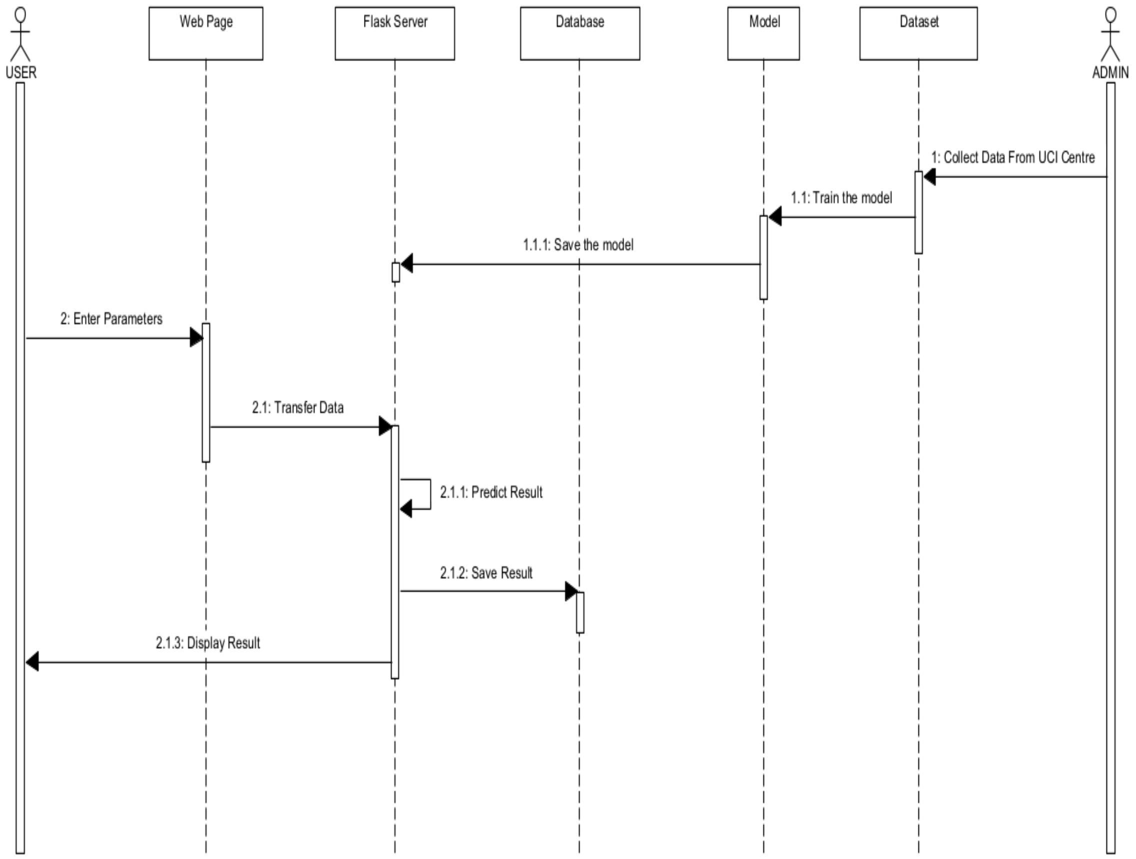


Fig: Sequence Diagram

5.2.4 ACTIVITY DIAGRAM

Activity Diagram captures actions and their results. They support and encourage parallel activities. These are important for business modeling and useful for concurrent programs. These can be used to describe use cases.

Classes: Models consist of **objects** that interact by sending each other **messages**. Think of an object as "alive." Objects have things they know (**attributes**) and things they can do (**behaviors** or **operations**). The values of an object's attributes determine its **state**.

Classes are the "blueprints" for objects. A class wraps attributes (data) and behaviors (methods or functions) into a single distinct entity. Objects are **instances** of classes

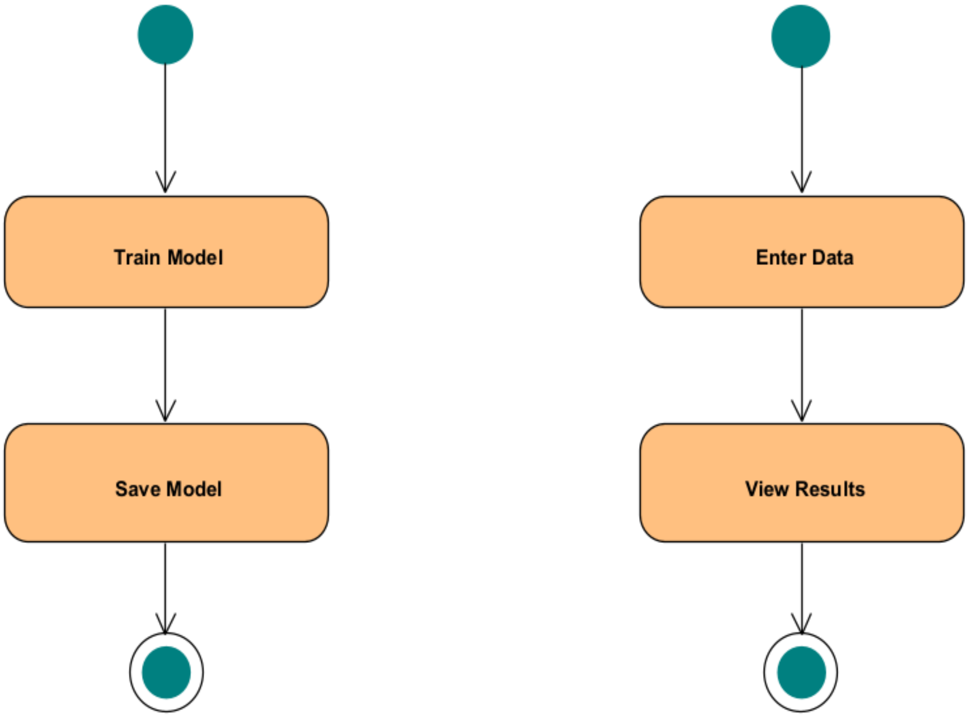


Fig: Activity Diagram

CHAPTER 6

IMPLEMENTATION AND RESULTS

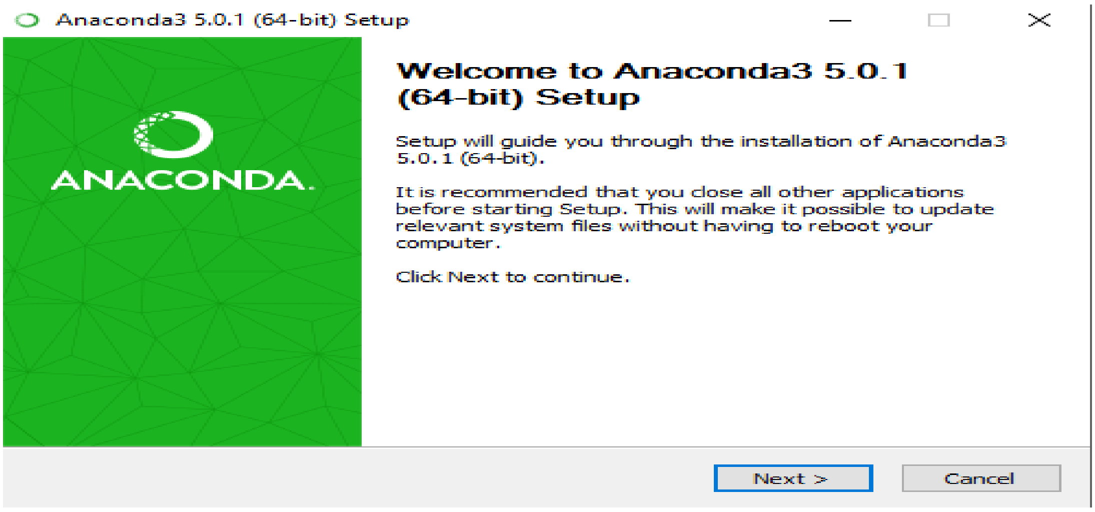
6.1 ANACONDA NAVIGATOR

Anaconda is an open-source distribution for python and R. It is used for data science, machine learning, deep learning, etc. With the availability of more than 300 libraries for data science, it becomes fairly optimal for any programmer to work on anaconda for data science.

Anaconda helps in simplified package management and deployment. Anaconda comes with a wide variety of tools to easily collect data from various sources using various machine learning and AI algorithms. It helps in getting an easily manageable environment setup whichcan deploy any project with the click of a single button.

Anaconda Navigator Installation:

1. Visit the Anaconda downloads page
   - ✓ **Go to the following link**: https://www.anaconda.com/products/individual
2. Select Windows where the three operating systems are listed –
   - ✓ Windows
   - ✓ MAC OS
   - ✓ LINUX
3. Download the most recent Python 3 release.
4. Open and run the installer
   - ✓ Once the download completes, open and run the ***.exe*** installer



Anaconda3 5.0.1 (64-bit) Setup

Welcome to Anaconda3 5.0.1 (64-bit) Setup

Setup will guide you through the installation of Anaconda3 5.0.1 (64-bit).

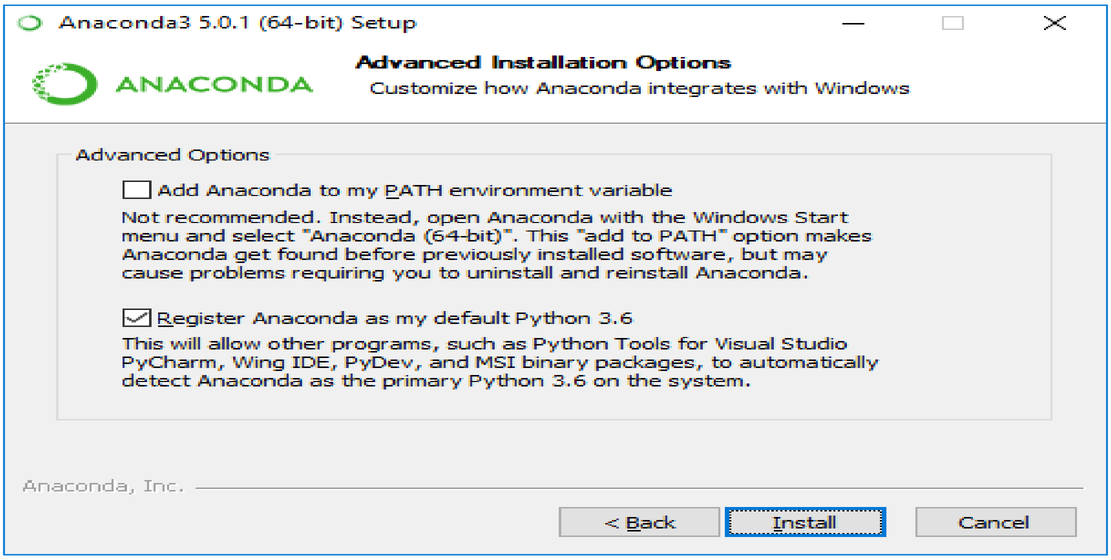
It is recommended that you close all other applications before starting Setup. This will make it possible to update relevant system files without having to reboot your computer.

Click Next to continue.

Screen 1: Anaconda Setup

5. Then Agree to the license.

6. At the Advanced Installation Options screen, I recommend that you do not check "AddAnaconda to my PATH environment variable".



Anaconda3 5.0.1 (64-bit) Setup

Advanced Installation Options

Customize how Anaconda integrates with Windows

Advanced Options

□ Add Anaconda to my PATH environment variable

Not recommended. Instead, open Anaconda with the Windows Start menu and select "Anaconda (64-bit)". This "add to PATH" option makes Anaconda get found before previously installed software, but may cause problems requiring you to uninstall and reinstall Anaconda.

☑ Register Anaconda as my default Python 3.6

This will allow other programs, such as Python Tools for Visual Studio PyCharm, Wing IDE, PyDev, and MSI binary packages, to automatically detect Anaconda as the primary Python 3.6 on the system.

Screen 2: Advanced Installation Options

7. The python programming along with the flask template rendering can be done in theAnaconda’s Jupyter Notebook by launching the jupyter notebook.

6.2 PyCharm

PyCharm is a hybrid platform developed by JetBrains as an IDE for Python. It is commonly used for Python application development. Some of the unicorn organizations such as Twitter, Facebook, Amazon, and Pinterest use PyCharm as their Python IDE!

It supports two versions: v2.x and v3.x.

We can run PyCharm on Windows, Linux, or Mac OS. Additionally, it contains modules and packages that help programmers develop software using Python in less time and with minimal effort. Further, it can also be customized according to the requirements of developers.

Features of PyCharm:

- ➢ Intelligent Code editor
- ➢ Code Navigation
- ➢ Refactoring
- ➢ Assistance for many other Web Technologies
- ➢ Support for popular Python Web Frameworks
- ➢ Assistance for Python Scientific Libraries

6.3 FRONT END

HTML stands for **H**yper**t**ext **M**arkup **L**anguage, and it is the most widely used language to write Web Pages.

✓ Hypertext refers to the way in which Web pages (HTML documents) are linked together. Thus, the link available on a webpage is called Hypertext.

✓ As its name suggests, HTML is a Mark-up Language which means you use HTML to simply "mark-up" a text document with tags that tell a Web browser how to structure it to display.

Originally, HTML was developed with the intent of defining the structure of documents like headings, paragraphs, lists, and so forth to facilitate the sharing of scientific information between researchers.

Now, HTML is being widely used to format web pages with the help of different tags availablein HTML language.

CSS (CASCADING STYLE SHEETS)

CSS is used to control the style of a web document in a simple and easy way. Cascading Style Sheets, fondly referred to as CSS, is a simple design language intended to simplify the process of making web pages presentable. CSS saves a lot of work. It can controlthe layout of multiple web pages all at once.

CSS can be added to HTML elements in 3 ways:

- ✓ **Inline** - by using the style attribute in HTML elements
- ✓ **Internal** - by using a <style> element in the <head> section
- ✓ **External** - by using an external CSS file

6.4 BACKEND

FLASK

Flask is a web application framework written in Python. It is developed by Armin Ronacher, who leads an international group of Python enthusiasts named Pocco. Flask is basedon the Werkzeug WSGI toolkit and Jinja2 template engine.

PYTHON

Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English words frequently whereas other languages use punctuation, and it has fewer syntactic constructions than other languages. Python was developed by Guido van Rossum in the late eighties and early nineties at the National Research Institute for Mathematics and Computer Science in the Netherlands. Python is now maintained by a core development team at the institute, although Guido van Rossum still holds a vital role in directing its progress.

SKLEARN

Scikit-learn is a free machine learning library for Python. It features various algorithms like support vector machine, random forests, and k-neighbours, and it also supports Python numerical and scientific libraries like NumPy and SciPy. The sklearn library contains lot of efficient machine learning tools including classification, regression, clustering, and dimensionality reduction. It should not be used for reading the data manipulating the data andsummarizing it. There are better libraries for that like NumPy, Pandas etc.

6.5 PERFORMANCE ANALYSIS

Accuracy- Accuracy is the ratio of the number of correct predictions to the total number of inputs in the dataset. It is expressed as:

Accuracy = (TP + TN) /(TP+FP+FN+TN)

Confusion Matrix- It gives us a matrix as output and gives the total performance of the system.

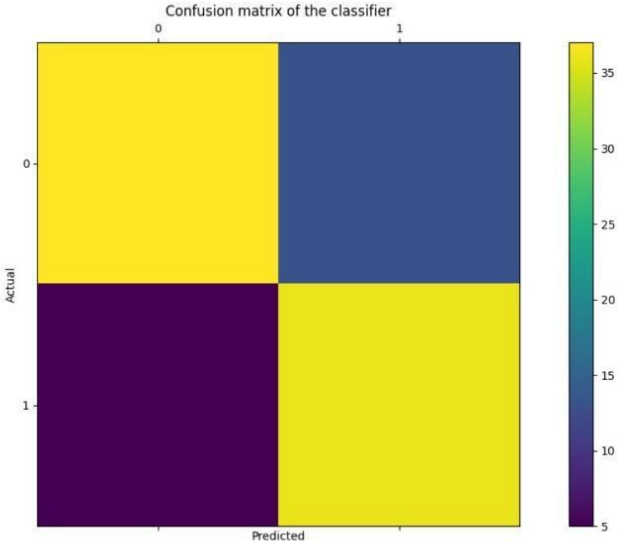


Fig: Confusion Matrix

Where

TP: True positive

FP: False Positive

FN: False Negative

TN: True Negative

6.6 CODE

The below code demonstrates the working of our project work.

Step 1: Data Refining

Step 2: Building Model

```
#Import libraries
import numpy as np
import pandas as pd
df1=pd.read_csv(r"C:\Users\spk09\Desktop\h2.csv")
#remove duplicates
df1.drop_duplicates()
#split the dataset
from sklearn.model_selection import train_test_split

predictors = df1.drop("Target",axis=1)
target = df1["Target"]
X_train,X_test,Y_train,Y_test =
train_test_split(predictors,target,test_size=0.3,random_state=0)
```

```
#compare models

#SVM

from sklearn.metrics import accuracy_score

from sklearn import svm

sv = svm.SVC(kernel='linear')

sv.fit(X_train, Y_train)

Y_pred_svm = sv.predict(X_test)

score_svm = round(accuracy_score(Y_pred_svm,Y_test)*100,2)

print("The accuracy score achieved using Linear SVM is: "+str(score_svm)+" %")

#Decision Tree

from sklearn.tree import DecisionTreeClassifier

dt = DecisionTreeClassifier()

dt.fit(X_train,Y_train)

Y_pred_dt = dt.predict(X_test)

score_dt = round(accuracy_score(Y_pred_dt,Y_test)*100,2)

print("The accuracy score achieved using Decision Tree is: "+str(score_dt)+" %")

#Logistic Regression

from sklearn.linear_model import LogisticRegression

lr = LogisticRegression(solver='lbfgs', max_iter=1000)

lr.fit(X_train,Y_train)

Y_pred_lr = lr.predict(X_test)

score_lr = round(accuracy_score(Y_pred_lr,Y_test)*100,2)
```

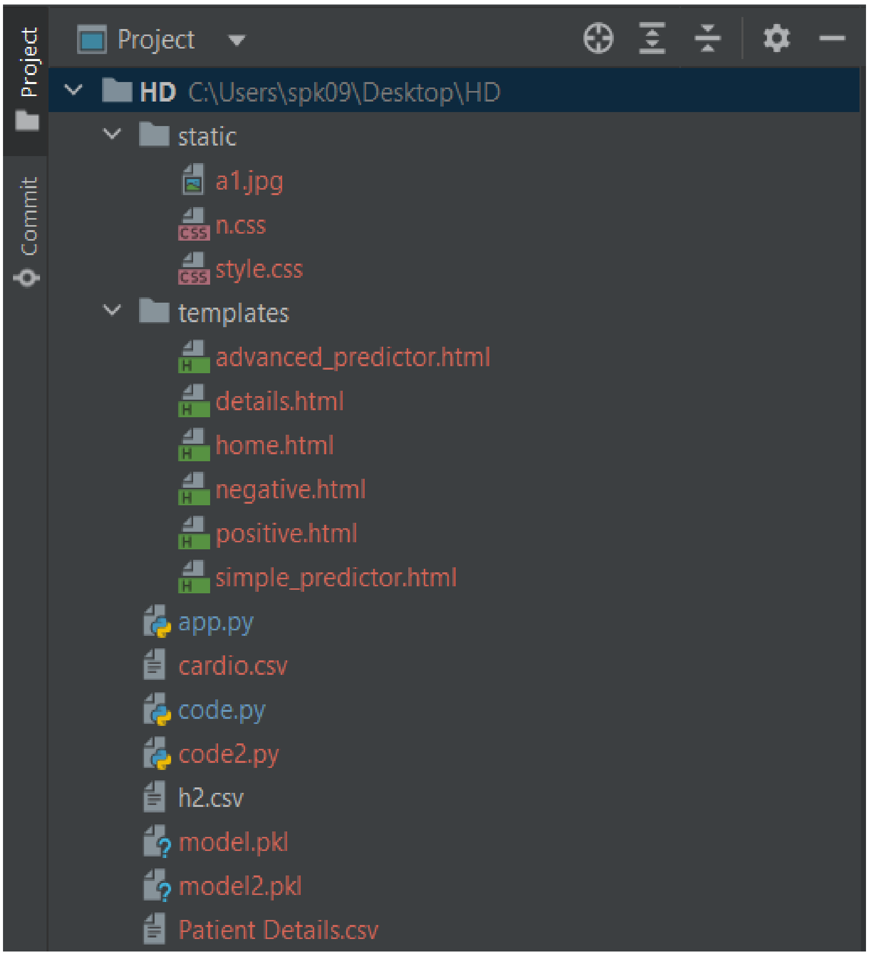
```
print("The accuracy score achieved using Logistic Regression is: "+str(score_lr)+" %")
#Naive Bayes
from sklearn.naive_bayes import GaussianNB
nb = GaussianNB()
nb.fit(X_train,Y_train)
Y_pred_nb = nb.predict(X_test)
score_nb = round(accuracy_score(Y_pred_nb,Y_test)*100,2)
print("The accuracy score achieved using Naive Bayes is: "+str(score_nb)+" %")
#KNN
from sklearn.neighbors import KNeighborsClassifier
knn = KNeighborsClassifier(n_neighbors=7)
knn.fit(X_train,Y_train)
Y_pred_knn=knn.predict(X_test)
score_knn = round(accuracy_score(Y_pred_knn,Y_test)*100,2)
print("The accuracy score achieved using KNN is: "+str(score_knn)+" %")
#Random Forest
from sklearn.ensemble import RandomForestClassifier
rf = RandomForestClassifier()
rf.fit(X_train,Y_train)
Y_pred_rf = rf.predict(X_test)
score_rf = round(accuracy_score(Y_pred_rf,Y_test)*100,2)
print("The accuracy score achieved using Random Forest is: "+str(score_rf)+" %")
```

COMPARE MODELS

After performing the machine learning approach for training and testing we find that accuracy of the Random Forest is better compared to other algorithms. Accuracy is calculated with the support of the confusion matrix of each algorithm, here the number count of TP, TN, FP, FN is given and using the equation of accuracy, value has been calculated and it is concluded that Random Forest is best with 91% accuracy and the comparison is shown below.

```
The accuracy score achieved using Linear SVM is: 83.73 %
The accuracy score achieved using Decision Tree is: 84.85 %
The accuracy score achieved using Logistic Regression is: 83.44 %
The accuracy score achieved using Naive Bayes is: 84.65 %
The accuracy score achieved using KNN is: 62.9 %
The accuracy score achieved using Random Forest is: 91.29 %
```

ORGANIZATION OF PROJECT FOLDER IN PYCHARM



```
HD C:\Users\spk09\Desktop\HD
  static
    a1.jpg
    n.css
    style.css
  templates
    advanced_predictor.html
    details.html
    home.html
    negative.html
    positive.html
    simple_predictor.html
  app.py
  cardio.csv
  code.py
  code2.py
  h2.csv
  model.pkl
  model2.pkl
  Patient Details.csv
```

Step 3: Save Model

#Save model for Simple Prediction as model2.pkl

code2.py

```
import numpy as np
import pandas as pd
df1=pd.read_csv("cardio.csv")
from sklearn.model_selection import train_test_split

predictors = df1.drop("cardio",axis=1)
target = df1["cardio"]

X_train,X_test,Y_train,Y_test =
train_test_split(predictors,target,test_size=0.3,random_state=0)
from sklearn.metrics import accuracy_score
from sklearn.ensemble import RandomForestClassifier

rf = RandomForestClassifier()
rf.fit(X_train,Y_train)
Y_pred_rf = rf.predict(X_test)
score_rf = round(accuracy_score(Y_pred_rf,Y_test)*100,2)


import pickle
with open('model2.pkl','wb') as files:
   pickle.dump(rf,files)
```

#Save model for Advanced Prediction as model.pkl

Code.py

```
import numpy as np
import pandas as pd
df1=pd.read_csv("h2.csv")
from sklearn.model_selection import train_test_split

predictors = df1.drop("Target",axis=1)
target = df1["Target"]

X_train,X_test,Y_train,Y_test =
train_test_split(predictors,target,test_size=0.3,random_state=0)
```

```
from sklearn.metrics import accuracy_score
from sklearn.ensemble import RandomForestClassifier

rf = RandomForestClassifier()
rf.fit(X_train,Y_train)
Y_pred_rf = rf.predict(X_test)
score_rf = round(accuracy_score(Y_pred_rf,Y_test)*100,2)


import pickle
with open('model.pkl','wb') as files:
    pickle.dump(rf,files)
```

Step 4: Build Front end HTML pages using CSS

style.css

```
* {
   margin: 0;
   padding: 0;
   font-family: 'sans-serif';
}

.banner {
    width: 100%;
    height: 100vh;
   background-image : linear-gradient(rgba(0, 0, 0, 0.75), rgba(0, 0, 0, 0.75)), 
url(a1.jpg);
   background-size: cover;
   background-position: center;
}

.navbar {
    width: 85%;
    margin: auto;
    display: flex;
    align-items: center;
    padding: 25px 0;
    justify-content: space-between;
}
.navbar ul li {
    list-style: none;
    display: inline-block;
    margin: 0 20px;
    position: relative;
}
.navbar ul li a {
    text-decoration: none;
    color: #fff;
    text-transform: uppercase;
}
.navbar ul li::after {
    content: ";
    height: 3px;
    width: 0;
    background: #009688;
    position: absolute;
    left: 0;
    bottom: -10px;
    transition: 0.5s;
}
.navbar ul li:hover::after {
    width: 100%;
}
.content {
   width: 100%;
   position: absolute;
   top: 50%;
   transform: translateY(-50%);
   text-align: center;
   color: #fff;
}
.content h1{
    font-size: 70px;
    margin-top: 80px;
}
.content h2{
    font-size: 80px;
    margin-top: 50px;
}
.content p{
    margin: 20px auto;
    font-weight: 100;
    line-height: 25px;
```

```
}
button{
   width: 200px;
   padding: 15px 0;
   text-align: center;
   margin: 20px 10px;
   border-radius: 25px;
   font-weight: bold;
   border: 2px solid #009688;
   background: transparent;
   color: #fff;
   cursor: pointer;
   position: relative;
   overflow: hidden;
}
span{
   background: #009688;
   height: 100%;
   width: 0;
   border-radius: 25px;
   position: absolute;
   left: 0;
   bottom: 0;
   z-index: -1;
   transition: 0.5s;
}
button:hover span{
   width:100%;
}
button:hover{
   border:none;
}
legend {
      color: black;
      background: none;
     }
.container {
   padding-top: 10px;
}
```

n.css

```
body {
  margin: 0;
  padding: 0;
  font-family: 'sans-serif';
}

.banner {
    width: 100%;
    height: 100vh;
  background-image : linear-gradient(rgba(0, 0, 0, 0.75), rgba(0, 0, 0, 0.75)),
url(a1.jpg);
  background-size: cover;
  background-position: center;
}
.navbar {
    width: 95%;
    margin: auto;
    display: flex;
    align-items: center;
    padding: 25px 0;
    justify-content: space-between;
}
.navbar ul li {
    list-style: none;
    display: inline-block;
    margin: 0 20px;
    position: relative;
}
.navbar ul li a {
    font-family:    'sans-serif';
    text-decoration: none;
    color: #fff;
    text-transform: uppercase;
}
.navbar ul li::after {
    content: '';
    height: 3px;
    width: 0;
    background: #009688;
    position: absolute;
    left: 0;
    bottom: -10px;
    transition: 0.5s;
}
.navbar ul li:hover::after {
    width: 100%;
}
```

home.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Heart Disease Predictor</title>
    <link rel="stylesheet" type="text/css" href="{{ url_for('static', filename='style.css')
}}">
</head>
<body>
    <div class="banner">
        <div class="navbar">
            <ul>
                <li><a href="p_det">Patient Details</a></li>
            </ul>
        </div>
        <div class="content">
            <p>Welcome To</p>
            <h1>Heart Disease Predictor</h1>
            <p>developed by students of <br>Avanthi Institute of Engineering &
Technology</p>
            <div>
                <a href=simple>
                <button type="button"><span></span>Simple</button>
                </a>
                <a href=advanced>
                <button type="button"><span></span>Advanced</button>
                </a>
            </div>
        </div>
    </div>
</body>
</html>
```

simple_predictor.html

```
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <title>Heart Disease Predictor</title>
   <link rel="stylesheet" type="text/css" href="{{ url_for('static', filename='style.css')
}}">
  <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css"
integrity="sha384-
JcKb8q3iqJ61gNV9KGb8thSsNjpSL0n8PARn9HuZOnIxN0hoP+VmmDGMN5t9UJ0
Z" crossorigin="anonymous">
     <script src="https://code.jquery.com/jquery-3.5.1.slim.min.js" integrity="sha384-
DfXdz2htPH0lsSSs5nCTpuj/zy4C+OGpamoFVy38MVBnE+IbbVYUew+OrCXaRkfj"
crossorigin="anonymous"></script>
     <script src="https://cdn.jsdelivr.net/npm/popper.js@1.16.1/dist/umd/popper.min.js"
integrity="sha384-
9/reFTGAW83EW2RDu2S0VKaIzap3H66lZH81PoYlFhbGU+6BZp6G7niu735Sk7lN"
crossorigin="anonymous"></script>
     <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"
integrity="sha384-
B4gt1jrGC7Jh4AgTPSdUtOBvfO8shuf57BaghqFfPlYxofvL8/KUEfYiJOMMV+rV"
crossorigin="anonymous"></script>

</head>
<body>
     <script src="https://code.jquery.com/jquery-3.5.1.slim.min.js" integrity="sha384-
DfXdz2htPH0lsSSs5nCTpuj/zy4C+OGpamoFVy38MVBnE+IbbVYUew+OrCXaRkfj"
crossorigin="anonymous"></script>
     <script src="https://cdn.jsdelivr.net/npm/popper.js@1.16.1/dist/umd/popper.min.js"
integrity="sha384-
9/reFTGAW83EW2RDu2S0VKaIzap3H66lZH81PoYlFhbGU+6BZp6G7niu735Sk7lN"
crossorigin="anonymous"></script>
     <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"
integrity="sha384-
B4gt1jrGC7Jh4AgTPSdUtOBvfO8shuf57BaghqFfPlYxofvL8/KUEfYiJOMMV+rV"
crossorigin="anonymous"></script>

   <div class="banner">
     <div class="navbar">
          <ul>
```

```
                <li><a href="h">Home</a></li>
                <li><a href="p_det">Patient Details</a></li>
            </ul>
        </div>
      <div class="container">
         <br>
         <!--Form-->
         <form action = "{{url_for('simple')}}" method ="POST" >
            <fieldset>

              <div class="card card-body" >
                 <div class="form-group  row">
                    <div class="col-sm">
                     <label for="name">Name</label>
                     <input type="text" class="form-control" id="name" name="name"
required>
                    </div>
                  <div class="col-sm">
                     <label for="age">Age</label>
                     <input type="number" class="form-control" id="age" name="age"
required>
                  </div>
                  <div class="col-sm">
                     <label for="gender">Gender</label>
                     <select class="form-control" id="gender"  name="gender" required>
                       <option disabled selected value> -- Select an Option -- </option>
                       <option value = "1">Female</option>
                       <option value = "2">Male</option>
                     </select>
                   </div>
                 </div>
                 <br>
                 <div class="form-group  row">
                    <div class="col-sm">
                     <label for="height">Height (in cm)</label>
                     <input type="number" class="form-control" id="height" name="height"
required>
                    </div>
                    <div class="col-sm">
                     <label for="weight">Weight (in kgs)</label>
                     <input type="number" class="form-control" id="weight" name="weight"
required>
                    </div>
                    <div class="col-sm">
                     <label for="sys_bp">Systolic Blood Pressure</label>
                     <input type="number" class="form-control" id="sys_bp" name="sys_bp"
required>
                    </div>
                 </div>
                 <br>
                 <div class="form-group  row">
                    <div class="col-sm">
                     <label for="dia_bp">Diastolic Blood Pressure</label>
                     <input type="number" class="form-control" id="dia_bp" name="dia_bp"
required>
                    </div>
                    <div class="col-sm">
                     <label for="cholesterol">Cholesterol</label>
                     <select class="form-control" id="cholesterol" name = "cholesterol"
required>
                       <option disabled selected value> -- Select an Option -- </option>
                       <option value = "1">Normal</option>
                       <option value = "2">Above normal</option>
                       <option value = "3">Well above normal</option>
                     </select>
                    </div>
                    <div class="col-sm">
                     <label for="glucose">Glucose</label>
                     <select class="form-control" id="glucose" name = "glucose" required>
                       <option disabled selected value> -- Select an Option -- </option>
                       <option value = "1">Normal</option>
                       <option value = "2">Above normal</option>
                       <option value = "3">Well above normal</option>
                     </select>
                    </div>
                 </div>
                 <br>
                 <div class="form-group row">
                     <div class="col-sm">
                        <label for="smoke">Smoking</label>
                        <select class="form-control" id="smoke" name="smoke" required>
                           <option disabled selected value> -- Select an Option -- </option>
                           <option value = "0">No</option>
                           <option value = "1">Yes</option>
                        </select>
                     </div>
                     <div class="col-sm">
                       <label for="alcohol">Alcohol</label>
                       <select class="form-control" id="alcohol" name="alcohol" required>
                         <option disabled selected value> -- Select an Option -- </option>
                         <option value = "0">No</option>
                         <option value = "1">Yes</option>
                       </select>
                     </div>
                      <div class="col-sm">
                       <label for="exer">Physical Activity</label>
                       <select class="form-control" id="exer" name="exer" required>
                         <option disabled selected value> -- Select an Option -- </option>
                         <option value = "0">No</option>
                         <option value = "1">Yes</option>
                       </select>
                     </div>
                 </div>
                 <br>
                 <div class="form-group">
                  <input class="btn btn-primary" type="submit" value="Result">
                 </div>

              </div>
           </fieldset>
          </form>
            </div>
     </div>
</body>
</html>
```

advanced_predictor.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Heart Disease Predictor</title>
    <link rel="stylesheet" type="text/css" href="{{ url_for('static', filename='style.css')
}}">
   <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css"
integrity="sha384-
```

```
JcKb8q3iqJ61gNV9KGb8thSsNjpSL0n8PARn9HuZOnIxN0hoP+VmmDGMN5t9UJ0
Z" crossorigin="anonymous">
    <script src="https://code.jquery.com/jquery-3.5.1.slim.min.js" integrity="sha384-
DfXdz2htPH0lsSSs5nCTpuj/zy4C+OGpamoFVy38MVBnE+IbbVYUew+OrCXaRkfj"
crossorigin="anonymous"></script>
    <script src="https://cdn.jsdelivr.net/npm/popper.js@1.16.1/dist/umd/popper.min.js"
integrity="sha384-
9/reFTGAW83EW2RDu2S0VKaIzap3H66lZH81PoYlFhbGU+6BZp6G7niu735Sk7lN"
crossorigin="anonymous"></script>
    <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"
integrity="sha384-
B4gt1jrGC7Jh4AgTPSdUtOBvfO8shuf57BaghqFfPlYxofvL8/KUEfYiJOMMV+rV"
crossorigin="anonymous"></script>

</head>
<body>
    <script src="https://code.jquery.com/jquery-3.5.1.slim.min.js" integrity="sha384-
DfXdz2htPH0lsSSs5nCTpuj/zy4C+OGpamoFVy38MVBnE+IbbVYUew+OrCXaRkfj"
crossorigin="anonymous"></script>
    <script src="https://cdn.jsdelivr.net/npm/popper.js@1.16.1/dist/umd/popper.min.js"
integrity="sha384-
9/reFTGAW83EW2RDu2S0VKaIzap3H66lZH81PoYlFhbGU+6BZp6G7niu735Sk7lN"
crossorigin="anonymous"></script>
    <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"
integrity="sha384-
B4gt1jrGC7Jh4AgTPSdUtOBvfO8shuf57BaghqFfPlYxofvL8/KUEfYiJOMMV+rV"
crossorigin="anonymous"></script>


   <div class="banner">
    <div class="navbar">
        <ul>
            <li><a href="h">Home</a></li>
            <li><a href="p_det">Patient Details</a></li>
        </ul>
      </div>
    <div class="container">
      <br>
      <!--Form-->
      <form action = "{{url_for('advanced')}}" method ="POST" >
        <fieldset>
          <div class="card card-body" >
            <div class="form-group  row">
            <div class="col-sm-3">
             <label for="name">Name</label>
             <input type="text" class="form-control" id="name" name="name"
required>
            </div>
           <div class="col-sm-3">
             <label for="age">Age</label>
             <input type="number" class="form-control" id="age" name="age"
required>
           </div>
           <div class="col-sm-3">
             <label for="gender">Gender</label>
             <select class="form-control" id="gender"  name="gender" required>
               <option disabled selected value> -- Select an Option -- </option>
               <option value = "0">Female</option>
               <option value = "1">Male</option>
             </select>
           </div>
          </div>
          <br>
          <div class="form-group  row">
            <div class="col-sm">
             <label for="cp">Chest Pain Type</label>
             <select class="form-control" id="cp" name = "cp" required>
               <option disabled selected value> -- Select an Option -- </option>
               <option value = "0">Typical Angina</option>
               <option value = "1">Atypical Angina</option>
               <option value = "2">Non-anginal Pain</option>
               <option value = "3">Asymptomatic</option>
             </select>
            </div>
            <div class="col-sm">
             <label for="trestbps">Resting Blood Pressure in mm Hg</label>
             <input type="number" class="form-control" id="trestbps" name="trestbps"
required>
            </div>
            <div class="col-sm">
             <label for="chol">Serum Cholesterol in mg/dl</label>
             <input type="number" class="form-control" id="chol" name="chol"
required>
            </div>
            <div class="col-sm">
             <label for="fbs">Fasting Blood Sugar > 120 mg/dl</label>
             <select class="form-control" id="fbs" name="fbs" required>
               <option disabled selected value> -- Select an Option -- </option>
               <option value = "0">No</option>
               <option value = "1">Yes</option>
             </select>
           </div>
          </div>

          <br>
          <div class="form-group row">
             <div class="col-sm">
                <label for="restecg">Resting ECG Results </label>
                <select class="form-control" id="restecg" name="restecg" required>
                  <option disabled selected value> -- Select an Option -- </option>
                  <option value = "0">Normal</option>
                  <option value = "1">Having ST-T wave abnormality  </option>
                  <option value = "2"> Probable or definite left ventricular hypertrophy </option>
                </select>
               </div>
               <div class="col-sm">
                <label for="thalach">Maximum Heart Rate(Thalach)</label>
                <input type="number" class="form-control" id="thalach" name="thalach" required>
               </div>
               <div class="col-sm">
                <label for="exang">Exercise Induced Angina </label>
                <select class="form-control" id="exang" name="exang" required>
                  <option disabled selected value> -- Select an Option -- </option>
                  <option value = "0">No</option>
                  <option value = "1">Yes</option>
                </select>
               </div>
                <div class="col-sm">
                <label for="oldpeak">ST Depression Induced(Oldpeak)</label>
                <input type="number" step="any" class="form-control" id="oldpeak" name="oldpeak" required>
               </div>
            </div>
            <br>
            <div class="form-group  row">
            <div class="col-sm">
             <label for="slope">Slope of the Peak Exercise ST Segment </label>
```

```
                     <select class="form-control" id="slope" name="slope" required>
                       <option disabled selected value> -- Select an Option -- </option>
                       <option value = "0">Up</option>
                       <option value = "1">Flat</option>
                       <option value = "2">Down</option>
                     </select>
                    </div>
                    <div class="col-sm">
                     <label for="ca">Number of Vessels Colored by Flourosopy</label>
                     <select class="form-control" id="ca" name = "ca" required>
                       <option disabled selected value> -- Select an Option -- </option>
                       <option value = "0">0</option>
                       <option value = "1">1</option>
                       <option value = "2">2</option>
                       <option value = "3">3</option>
                     </select>
                    </div>
                    <div class="col-sm">
                     <label for="thal">Thalassemia</label>
                     <select class="form-control" id="thal" name = "thal" required>
                       <option disabled selected value> -- Select an Option -- </option>
                       <option value = "0">Normal</option>
                       <option value = "1">Fixed defect</option>
                       <option value = "2">Reversable defect</option>
                     </select>
                    </div>
                 </div>
                 <br>
                 <div class="form-group">
                  <input class="btn btn-primary" type="submit" value="Result">
                 </div>

              </div>
          </fieldset>
         </form>
            </div>
    </div>
</body>
</html>
```

positive.html

```
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <title>Heart Disease Predictor</title>
   <link rel="stylesheet" type="text/css" href="{{ url_for('static', filename='style.css')
}}">

</head>
<body>
   <div class="banner">
      <div class="navbar">
         <ul>
            <li><a href="h">Home</a></li>
            <li><a href="p_det">Patient Details</a></li>
         </ul>
      </div>
      <div class="content">
         <p>Dear {{person}}</p>
         <h2>I'm very sorry !!</h2>
         <p>You are likely to get a heart disease.</p>
         <div>
            <a href=simple>
            <button type="button"><span></span>Simple Predictor</button>
            </a>
            <a href=advanced>
            <button type="button"><span></span>Advanced Predictor</button>
            </a>
         </div>
      </div>
   </div>
</body>
</html>
```

negative.html

```
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <title>Heart Disease Predictor</title>
   <link rel="stylesheet" type="text/css" href="{{ url_for('static', filename='style.css')
}}">
</head>
<body>
    <div class="banner">
        <div class="navbar">
            <ul>
                <li><a href="h">Home</a></li>
                <li><a href="p_det">Patient Details</a></li>
            </ul>
        </div>
        <div class="content">
            <p>Dear {{person}}</p>
            <h2>Congratulations</h2>
            <p>You are Healthy.</p>
            <div>
                <a href=simple>
                <button type="button"><span></span>Simple Predictor</button>
                </a>
                <a href=advanced>
                <button type="button"><span></span>Advanced Predictor</button>
                </a>
            </div>
        </div>
    </div>
</body>
</html>
```

details.html

```
<!DOCTYPE html>

<html>
  <head>

    <link rel="stylesheet" type="text/css" href="{{ url_for('static', filename='n.css') }}">
    <!-- Favicon -->
    <!--link rel="shortcut icon" href="{{url_for('static',
filename='images/favicon.ico')}}"-->

    <!-- JQuery -->
    <script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
```

```
    <!-- Bootstrap -->
    <link rel="stylesheet" type="text/css"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css">
    <script type = "text/javascript"
src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.9/umd/popper.min.js"></scrip
t>
    <script type = "text/javascript"
src="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.min.js"></script>
    <script type = "text/javascript"
src="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.bundle.min.js"></s
cript>

    <!-- Datatable -->
    <link rel="stylesheet" type="text/css"
href="https://cdn.datatables.net/1.10.20/css/jquery.dataTables.min.css">
    <link rel="stylesheet" type="text/css"
href="https://cdn.datatables.net/responsive/2.2.3/css/responsive.dataTables.min.css">
    <script
src="https://cdnjs.cloudflare.com/ajax/libs/moment.js/2.11.2/moment.min.js"></script>
    <script type = "text/javascript"
src="https://cdn.datatables.net/1.10.20/js/jquery.dataTables.min.js"></script>
    <script type = "text/javascript"
src="https://cdn.datatables.net/responsive/2.2.3/js/dataTables.responsive.min.js"></scri
pt>
    <script type = "text/javascript"  src="https://cdn.datatables.net/plug-
ins/1.10.15/dataRender/datetime.js"></script>
  </head>
  <body>

<div class="banner">
  <div class="navbar">
            <ul>
                <li><a href="h">Home</a></li>
                <li><a href="simple">Simple Predictor</a></li>
                <li><a href="advanced">Advanced Predictor</a></li>
            </ul>
        </div>
    <div class="card">
      <div class="card-body">
        <h1>Patient Data</h1>
        <div class="mt-4">
            <table id="proxies" class="display table nowrap responsive" style="width:
100%">
```

```
              <thead>
                <tr>
                  {% for header in results[0].keys() %}
                    <th>{{header}}</th>
                  {% endfor %}
                </tr>
              </thead>
              <tbody>
                {% for row in results %}
                  <tr>
                    {% for index in range(0, len(fieldnames)) %}
                      <td>{{row[fieldnames[index]]}}</td>
                    {% endfor %}
                  </tr>
                {% endfor %}
              </tbody>
            </table>
</div>
        </div>
      </div>
    </div>
  </body>
  <script type="text/javascript">
    $('#proxies').DataTable();
  </script>

</html>
```

Step 5: Deploy through flask

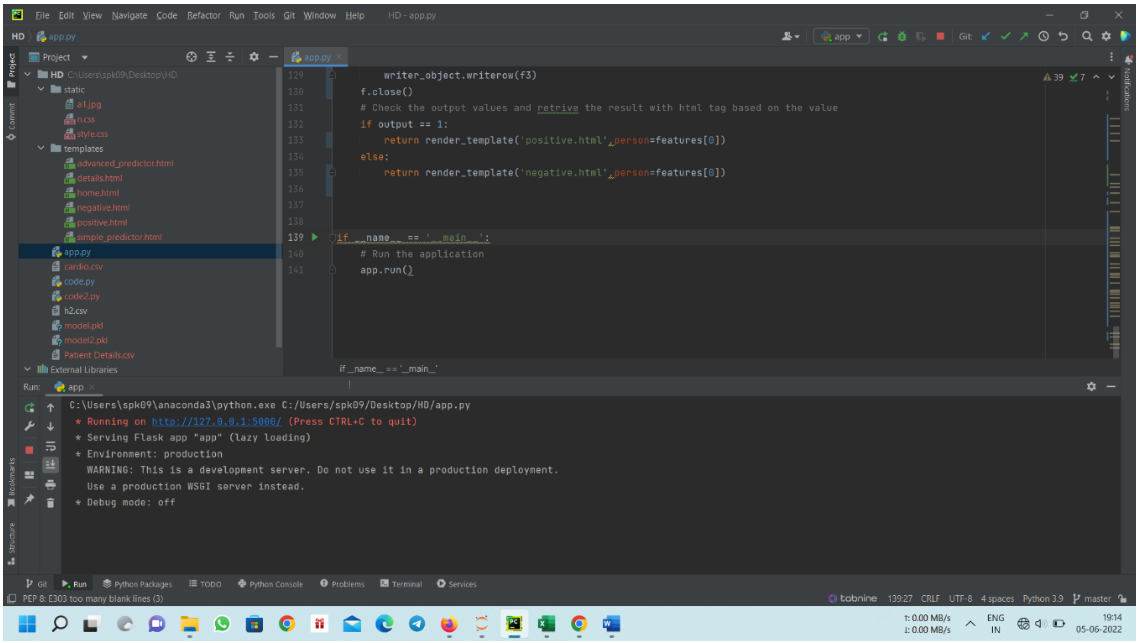
app.py

```
import numpy as np
import pickle
from flask import Flask, request, render_template
import csv

# Load ML model
model = pickle.load(open('model.pkl', 'rb'))
model2 = pickle.load(open('model2.pkl', 'rb'))
# Create application
app = Flask(__name__)
```

```
# Bind home function to URL
@app.route('/')
def home():
    return render_template('home.html')

@app.route('/h')
def h():
    return render_template('home.html')

@app.route('/simple')
def simple():
    return render_template('simple_predictor.html')

@app.route('/advanced')
def advanced():
    return render_template('advanced_predictor.html')

@app.route('/p_det')
def details():
    results = []
    with open('Patient Details.csv') as csvfile:
        reader = list(csv.DictReader(csvfile))

    for row in reader:
        results.append(dict(row))

    fieldnames = [key for key in results[0].keys()]

    return render_template('details.html', results=results, fieldnames=fieldnames, len=len)

@app.route('/simple', methods=['POST'])
def detect():
    features = [i for i in request.form.values()]
    f2 = [int(i) for i in features[1:]]
    f2[0]*=365
    array_features = np.asarray(f2)
    prediction = model2.predict(array_features.reshape(1, -1))
    output = prediction[0]
    if output == 1:
        return render_template('positive.html',person=features[0])
    else:
        return render_template('negative.html',person=features[0])
```

```
# Bind predict function to URL
@app.route('/advanced', methods=['POST'])
def predict():
    # Put all form entries values in a list
    features = [i for i in request.form.values()]
    # Convert features to array
    f2=[float(i) for i in features[1:]]
    array_features = np.asarray(f2)
    # Predict features
    prediction = model.predict(array_features.reshape(1, -1))
    output = prediction[0]
    f3=[]
    # Convert to user format to store in database
    # get date
    from datetime import datetime
    now = datetime.now()
    dt = now.strftime("%d/%m/%Y %H:%M:%S").split()
    f3.append(dt[0])
    f3.append(dt[1])
    f3.append(features[0]) # get name
    f3.append(features[1]) # get age
    if features[2]=='1':
        f3.append('Male')
    else:
        f3.append('Female')     # get gender
    if features[3]=='0':
        f3.append('Typical Angina')
    elif features[3]=='1':
        f3.append('Atypical Angina')
    elif features[3]=='2':
        f3.append('Non-Anginal Pain')
    else:
        f3.append('Asymptomatic')   # get cp type
    f3.append(features[4]) # get RestBP
    f3.append(features[5]) # get Chol
    if features[6]=='1':
        f3.append('Yes')
    else:
        f3.append('No')
    if features[7]=='0':
        f3.append('Normal')
    elif features[7]=='1':
        f3.append('Having ST-T wave abnormality')
    else:
        f3.append('Probable or definite left ventricular hypertrophy') # get RestECG
    f3.append(features[8]) # get Thalach
    if features[9]=='1':
        f3.append('Yes')
    else:
        f3.append('No') # get exang
    if output == 1:
        f3.append('Positive')
    else:
        f3.append('Negative') # get result
    f3.append(features[10]) # get oldpeak
    if features[11]=='0':
        f3.append('Up')
    elif features[11]=='1':
        f3.append('Flat')
    else:
        f3.append('Down') # get slope
    f3.append(features[12]) # get ca
    if features[13]=='0':
        f3.append('Normal')
    elif features[13]=='1':
        f3.append('Fixed Defect')
    else:
        f3.append('Reversable Defect') #get thal
    from csv import writer
    # Save Patient Data
    with open("Patient Details.csv", 'a', newline='') as f:
        writer_object = writer(f)
        writer_object.writerow(f3)
    f.close()
    # Check the output values and retrive the result with html tag based on the value
    if output == 1:
        return render_template('positive.html',person=features[0])
    else:
        return render_template('negative.html',person=features[0])


if __name__ == '__main__':
    # Run the application
    app.run()
```

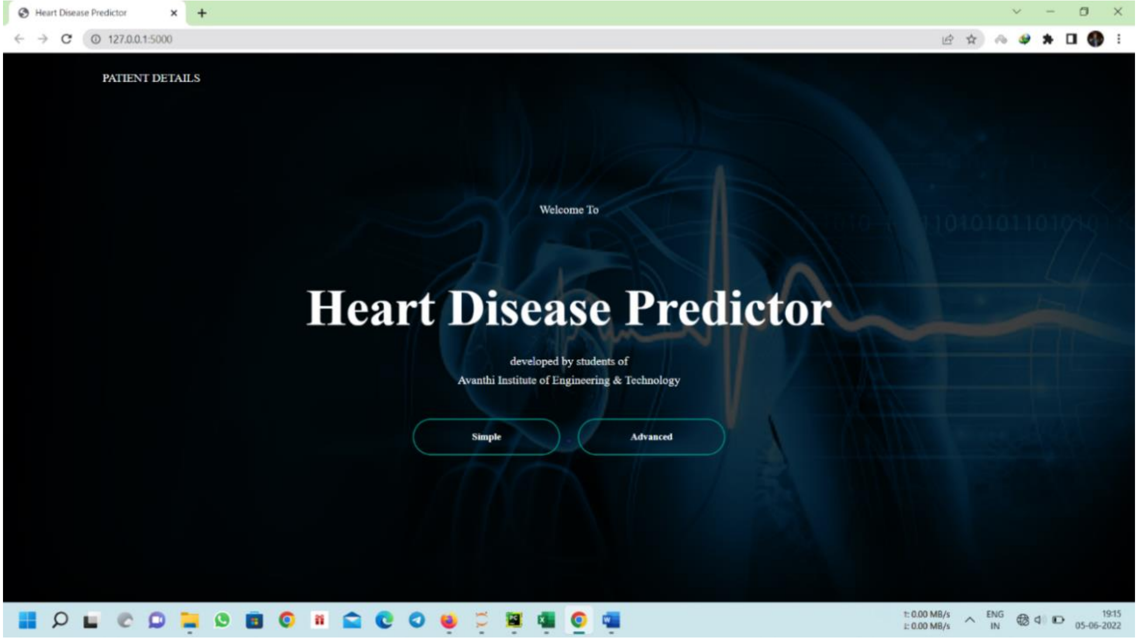
6.7 OUTPUT SCREENS


File Edit View Navigate Code Refactor Run Tools Git Window Help HD - app.py
HD
app.py
Project
HD C:\Users\spk09\Desktop\HD
static
a1.jpg
n.css
style.css
templates
advanced_predictor.html
details.html
home.html
negative.html
positive.html
simple_predictor.html
app.py
cardio.csv
code.py
code2.py
h2.csv
model.pkl
model2.pkl
Patient Details.csv
External Libraries
Run: app
Git Run Python Packages TODO Python Console Problems Terminal Services
PEP 8: E303 too many blank lines (3)
tabnine 139:27 CRLF UTF-8 4 spaces Python 3.9 master
19:14
05-06-2022

```
        writer_object.writerow(f3)
    f.close()
    # Check the output values and retrive the result with html tag based on the value
    if output == 1:
        return render_template('positive.html',person=features[0])
    else:
        return render_template('negative.html',person=features[0])



if __name__ == '__main__':
    # Run the application
    app.run()
```

```
C:\Users\spk09\anaconda3\python.exe C:/Users/spk09/Desktop/HD/app.py
 * Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
 * Serving Flask app "app" (lazy loading)
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deployment.
   Use a production WSGI server instead.
 * Debug mode: off
```

Screen 3: Executing the code in PyCharm



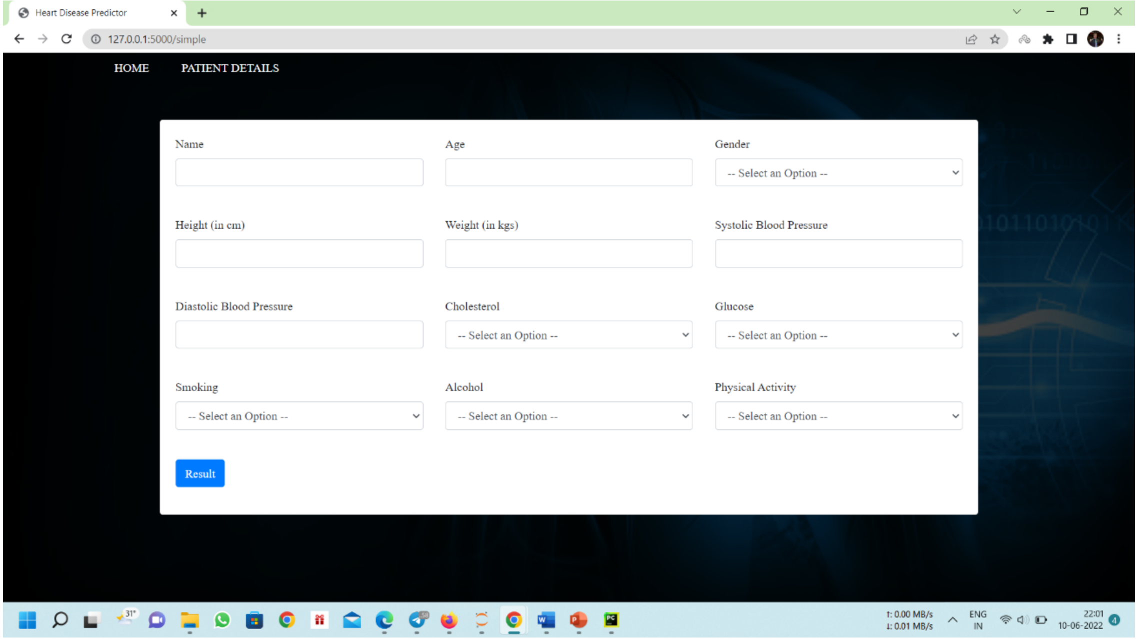
PATIENT DETAILS

Welcome To

Heart Disease Predictor

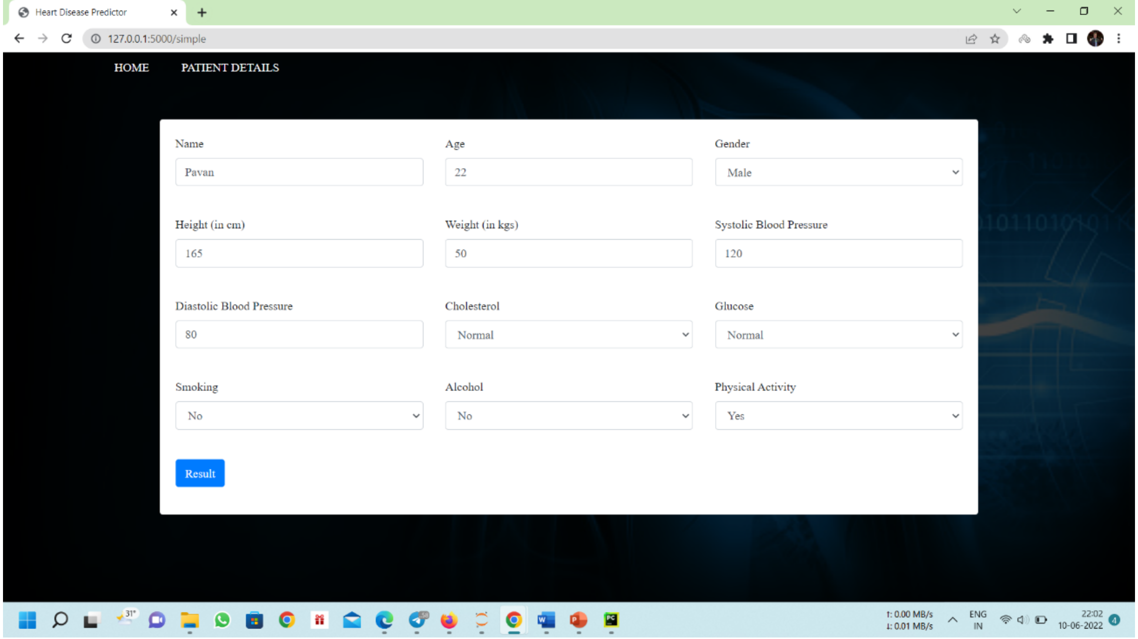
developed by students of
Avanthi Institute of Engineering & Technology

Screen 4: Home Page



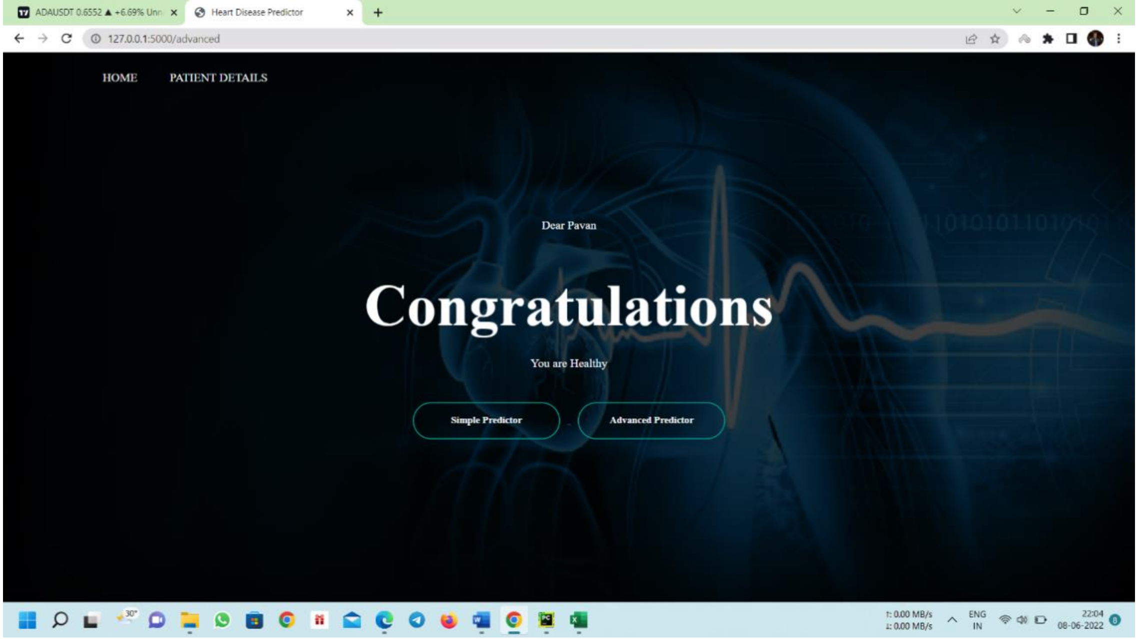
Name	Age	Gender
		-- Select an Option --
Height (in cm)	Weight (in kgs)	Systolic Blood Pressure
Diastolic Blood Pressure	Cholesterol	Glucose
	-- Select an Option --	-- Select an Option --
Smoking	Alcohol	Physical Activity
-- Select an Option --	-- Select an Option --	-- Select an Option --

Screen 5: Simple Predictor



Name	Age	Gender
Pavan	22	Male
Height (in cm)	Weight (in kgs)	Systolic Blood Pressure
165	50	120
Diastolic Blood Pressure	Cholesterol	Glucose
80	Normal	Normal
Smoking	Alcohol	Physical Activity
No	No	Yes

Screen 6: Fill form

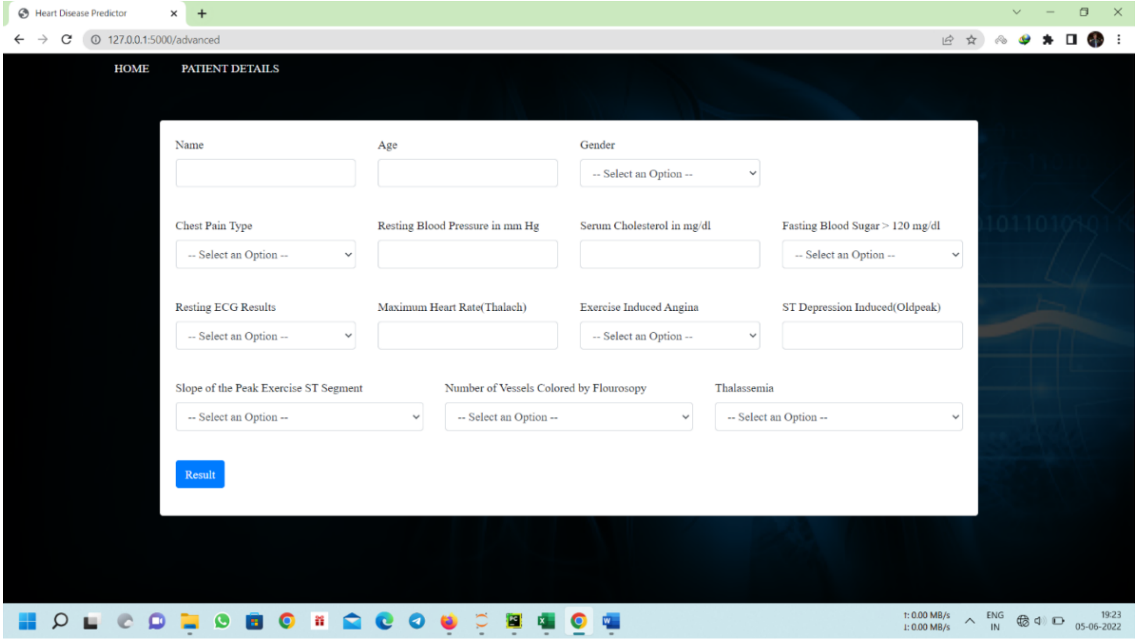


Dear Pavan

Congratulations

You are Healthy

Screen 7: Result – no disease



Name

Age

Gender: -- Select an Option --

Chest Pain Type: -- Select an Option --

Resting Blood Pressure in mm Hg

Serum Cholesterol in mg/dl

Fasting Blood Sugar > 120 mg/dl: -- Select an Option --

Resting ECG Results: -- Select an Option --

Maximum Heart Rate(Thalach)

Exercise Induced Angina: -- Select an Option --

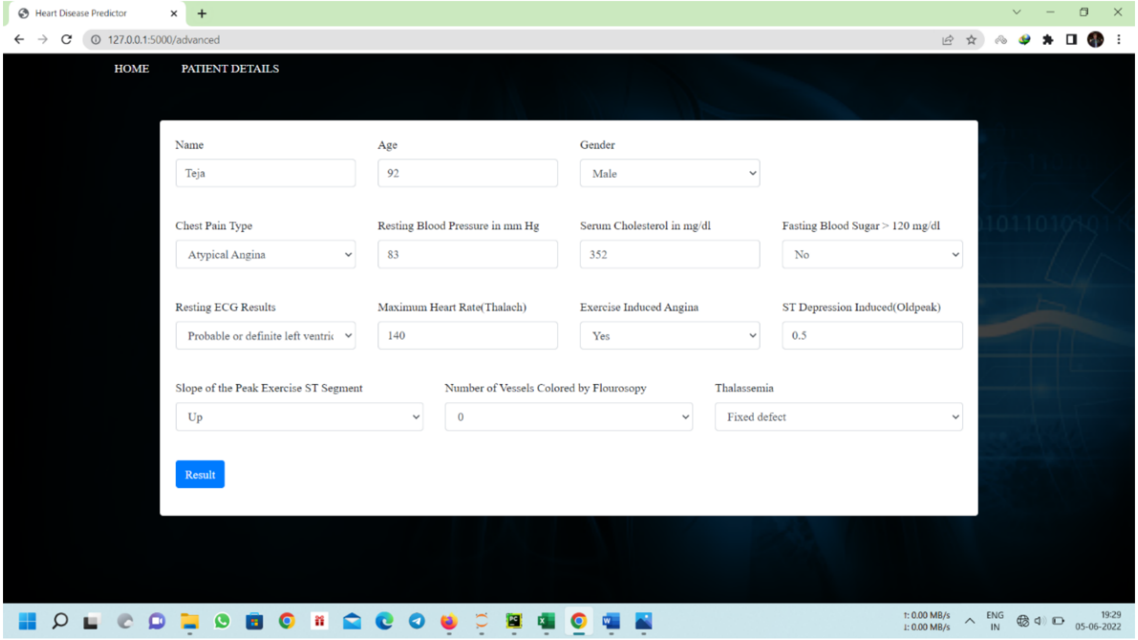
ST Depression Induced(Oldpeak)

Slope of the Peak Exercise ST Segment: -- Select an Option --

Number of Vessels Colored by Flourosopy: -- Select an Option --

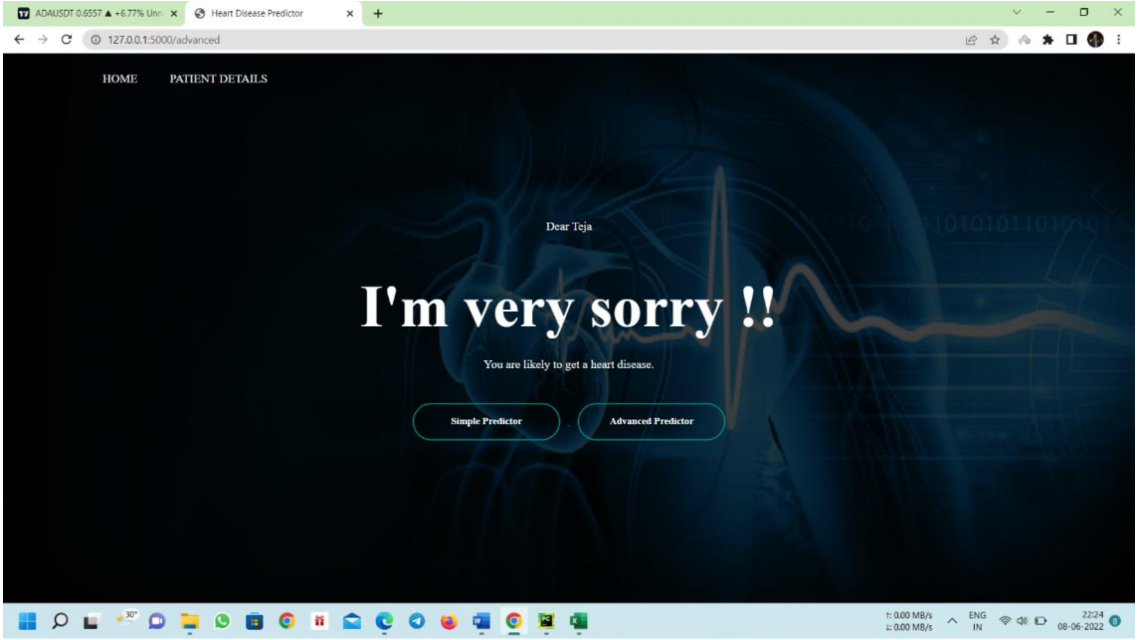
Thalassemia: -- Select an Option --

Screen 8: Advanced Predictor



Name Teja	Age 92	Gender Male	
Chest Pain Type Atypical Angina	Resting Blood Pressure in mm Hg 83	Serum Cholesterol in mg/dl 352	Fasting Blood Sugar > 120 mg/dl No
Resting ECG Results Probable or definite left ventric	Maximum Heart Rate(Thalach) 140	Exercise Induced Angina Yes	ST Depression Induced(Oldpeak) 0.5
Slope of the Peak Exercise ST Segment Up	Number of Vessels Colored by Flourosopy 0	Thalassemia Fixed defect	

Screen 9: Fill form

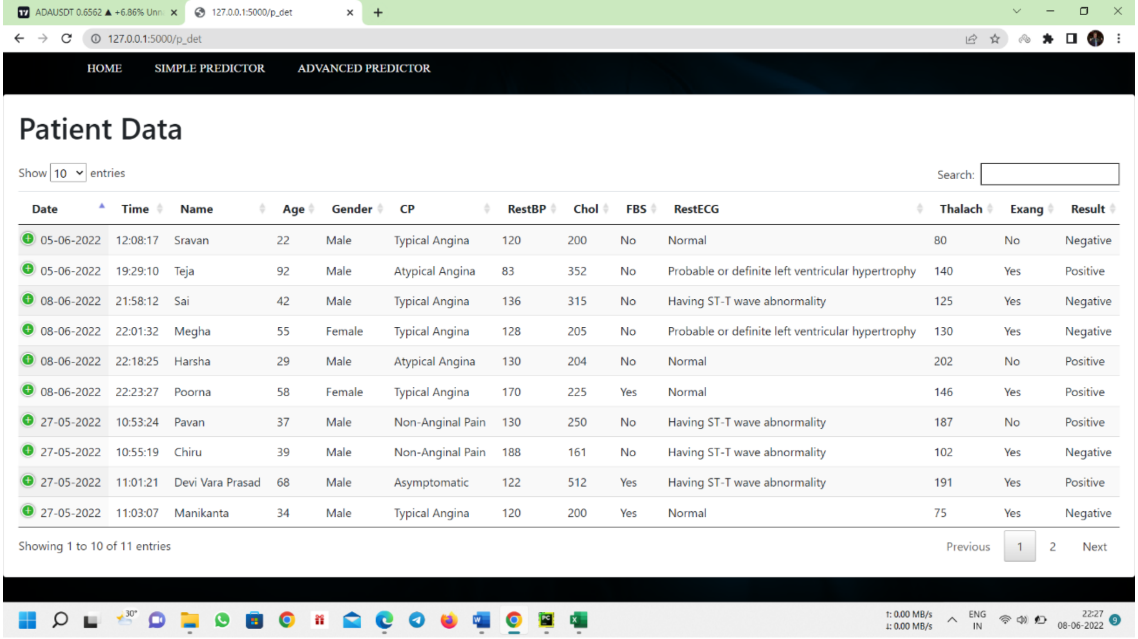


Dear Teja

I'm very sorry !!

You are likely to get a heart disease.

Screen 10: Result – Has Disease



Patient Data

Show 10 entries

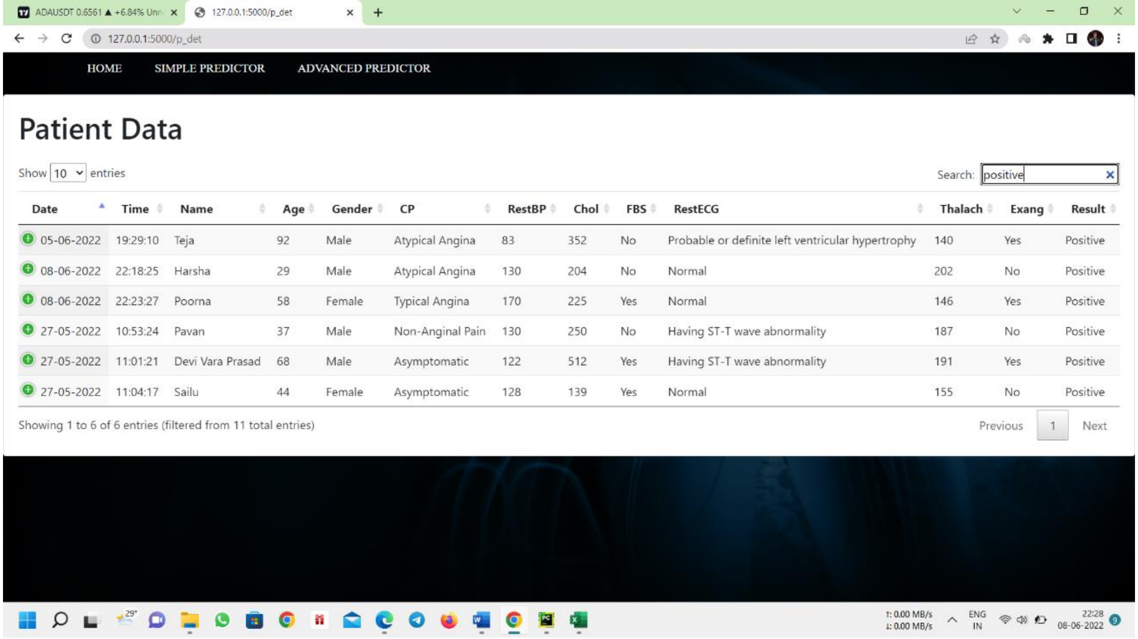
Search:

Date	Time	Name	Age	Gender	CP	RestBP	Chol	FBS	RestECG	Thalach	Exang	Result
05-06-2022	12:08:17	Sravan	22	Male	Typical Angina	120	200	No	Normal	80	No	Negative
05-06-2022	19:29:10	Teja	92	Male	Atypical Angina	83	352	No	Probable or definite left ventricular hypertrophy	140	Yes	Positive
08-06-2022	21:58:12	Sai	42	Male	Typical Angina	136	315	No	Having ST-T wave abnormality	125	Yes	Negative
08-06-2022	22:01:32	Megha	55	Female	Typical Angina	128	205	No	Probable or definite left ventricular hypertrophy	130	Yes	Negative
08-06-2022	22:18:25	Harsha	29	Male	Atypical Angina	130	204	No	Normal	202	No	Positive
08-06-2022	22:23:27	Poorna	58	Female	Typical Angina	170	225	Yes	Normal	146	Yes	Positive
27-05-2022	10:53:24	Pavan	37	Male	Non-Anginal Pain	130	250	No	Having ST-T wave abnormality	187	No	Positive
27-05-2022	10:55:19	Chiru	39	Male	Non-Anginal Pain	188	161	No	Having ST-T wave abnormality	102	Yes	Negative
27-05-2022	11:01:21	Devi Vara Prasad	68	Male	Asymptomatic	122	512	Yes	Having ST-T wave abnormality	191	Yes	Positive
27-05-2022	11:03:07	Manikanta	34	Male	Typical Angina	120	200	Yes	Normal	75	Yes	Negative

Showing 1 to 10 of 11 entries

Previous 1 2 Next

Screen 11: Patient details



Patient Data

Show 10 entries

Search: positive

Date	Time	Name	Age	Gender	CP	RestBP	Chol	FBS	RestECG	Thalach	Exang	Result
05-06-2022	19:29:10	Teja	92	Male	Atypical Angina	83	352	No	Probable or definite left ventricular hypertrophy	140	Yes	Positive
08-06-2022	22:18:25	Harsha	29	Male	Atypical Angina	130	204	No	Normal	202	No	Positive
08-06-2022	22:23:27	Poorna	58	Female	Typical Angina	170	225	Yes	Normal	146	Yes	Positive
27-05-2022	10:53:24	Pavan	37	Male	Non-Anginal Pain	130	250	No	Having ST-T wave abnormality	187	No	Positive
27-05-2022	11:01:21	Devi Vara Prasad	68	Male	Asymptomatic	122	512	Yes	Having ST-T wave abnormality	191	Yes	Positive
27-05-2022	11:04:17	Sailu	44	Female	Asymptomatic	128	139	Yes	Normal	155	No	Positive

Showing 1 to 6 of 6 entries (filtered from 11 total entries)

Previous 1 Next

Screen 12: Searching patients

CHAPTER 7

TESTING AND VALIDATION

7.1 INTRODUCTION

7.1.1 SCOPE

A primary purpose for testing is to detect software failures so that defects may be uncovered and corrected. This is a non-trivial pursuit. Testing cannot establish that a product functions properly under all conditions but can only establish that it does not function properlyunder specific conditions. The scope of software testing often includes examination of code as well as execution of that code in various environments and conditions as well as examining theaspects of code: does it do what it is supposed to do and do what it needs to do. In the current culture of software development, a testing organization may be separate from the development team. There are various roles for testing team members. Information derived from software testing may be used to correct the process by which software is developed.

7.1.2 DEFECTS AND FAILURES

Not all software defects are caused by coding errors. One common source of expensive defects is caused by requirements gaps, e.g., unrecognized requirements that result in errors ofomission by the program designer. A common source of requirements gaps is non-functional requirements such as testability, scalability, maintainability, usability, performance, and security.

Software faults occur through the following processes. A programmer makes an error (mistake), which results in a defect (fault, bug) in the software source code. If this defect is executed, in certain situations the system will produce wrong results, causing a failure. Not all defects will necessarily result in failures. For example, defects in dead code will never result in failures. A defect can turn into a failure when the environment is changed. Examples of these changes in environment include the software being run on a new hardware platform, alterations in source data or interacting with different software. A single defect may result in awide range of failure symptoms.

7.1.3 COMPATIBILITY

A frequent cause of software failure is compatibility with another application, a new operating system, or, increasingly, web browser version. In the case of lack of backward compatibility, this can occur because the programmers have only considered coding their programs for, or testing the software upon, "the *latest* version of" this-or-that operating system.The unintended consequence of this fact is that: their latest work might not be fully compatiblewith earlier mixtures of software/hardware, or it might not be fully compatible with another important operating system. In any case, these differences, whatever they might be, may have resulted in software failures, as witnessed by some significant population of computer users.

7.1.4 INPUT COMBINATIONS & PRECONDITIONS

A very fundamental problem with software testing is that testing under *all* combinations of inputs and preconditions is not feasible, even with a simple product. This means that the number of defects in a software product can be very large and defects that occur infrequently are difficult to find in testing. More significantly, nonfunctional dimensions of quality (how itis supposed to *be* versus what it is supposed to *do*) can be highly subjective; something that constitutes sufficient value to one person may be intolerable to another.

7.1.5 STATIC Vs. DYNAMIC TESTING

There are many approaches to software testing. Reviews, walkthroughs or inspections are considered as static testing, whereas actually executing programmed code with a given setof test cases is referred to as dynamic testing. The former can be, omitted, whereas the latter takes place when programs begin to be used for the first time - which is normally, considered the beginning of the testing stage. This may actually begin before the program is 100% complete in order to test particular sections of code. For example, Spread sheet programs are, by their very nature, tested to a large extent "on the fly" during the build process as the result of some calculation or text manipulation is shown interactively immediately after each formulais entered.

7.1.6 SOFTWARE VERIFICATION & VALIDATION

Software testing is used in association with verification and validation:

Verification: Have we built the software right (i.e., does it match the specification?)?

- ✓ It is process based.

Validation: Have we built the right software (i.e., is this what the customer wants?)?

- ✓ It is product based.

7.2 DESIGN OF TESTCASES AND SCENARIO

Test case is an object for execution for other modules in the architecture does not represent any interaction by itself. A test case is a set of sequential steps to execute a test operating on a set of predefined inputs to produce certain expected outputs.

There are two types of test cases: manual and automated

.A manual test case is executed manually while an automated test case is executed using automation. In system testing, test data should cover the possible values of each parameter based on the requirements. Since testing every value is impractical, a few values should be chosen from each equivalence class. An equivalence class is a set of values that should all be treated the same. Ideally, test cases that check error conditions are written separately from thefunctional test cases and should have steps to verify the error messages and logs. Realistically, if functional test cases are not yet written, it is ok for testers to check for error conditions when performing normal functional test cases. It should be clear which test data if any is expected totrigger errors.

Testing Scenario	**Expected Result**	**Actual Result**	**Test Result**
Entering the parameters	Positive	Positive	Pass
Entering the parameters	Negative	Negative	Pass
Entering the parameters	Positive	Positive	Pass
Entering the parameters	Negative	Positive	Fail

Table 1: Test Results

7.3 VALIDATION TESTING

At the culmination of the integration testing, the software was completely assembled as a package, interfacing errors have been uncovered and final series of software validation testing began. Here we test the system function in a manner that can be reasonably expected by customer, the system was tested against system requirement specification. Different unusual inputs that the user may use where assumed and the outputs were verified for such unprecedented inputs. This test is performed to validate the software. In this the entire software will be created and will test all the components of the software together. Validation testing ensures that the product actually meets the client's needs.

Testing Objectives

- ✓ Testing is a process of executing a program with the intent of finding an error.
- ✓ A good test case is one that has a high probability of finding an as-yet- undiscovered error.
- ✓ A successful test is that uncovers an as-yet-undiscovered error.

Test Levels - The test strategy describes the test level to be performed. There are primarilythree levels of testing.

- ✓ Unit Testing
- ✓ Integration Testing
- ✓ System Testing

7.3.1 UNIT TESTING

Unit testing is done on individual modules as they are completed and become executable. It is confined only to the designer's requirements. Each module can be tested using the following two strategies.

BLACK BOX TESTING

In this strategy some test cases are generated as input conditions that fully execute all functional requirements for the program. In this testing only the output is checked for correctness. The logical flow of the data is not checked.

Equivalence Class Partitioning

Equivalence Partitioning Method is also known as Equivalence class partitioning (ECP). It is a software testing technique or black box testing that divides input domain into classes ofdata, and with the help of these classes of data, test cases can be derived. An ideal test case identifies class of error that might require many arbitrary test cases to be executed before general error is observed.

In equivalence portioning, equivalence classes are evaluated for given input conditions. Whenever any input is given, then type of input condition is checked, then for this input conditions, Equivalence class represents or describes set of valid or invalid states.

WHITE BOX TESTING

In this test cases are generated on the logic of each module by drawing flow graphs ofthat module and logical decisions are tested on all the cases.

7.3.2 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 satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems thatarise from the combination of components.

7.3.3 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.

CHAPTER 8

CONCLUSION AND FUTURE SCOPE

Heart diseases are a major killer in India and throughout the world, application of promising technology like machine learning to the initial prediction of heart diseases will have a profound impact on society. The early prognosis of heart disease can aid in making decisions on lifestyle changes in high-risk patients and in turn reduce the complications, which can be a great milestone in the field of medicine. The number of people facing heart diseases is on a raise each year. This prompts for its early diagnosis and treatment. The utilization of suitable technology support in this regard can prove to be highly beneficial to the medical fraternity and patients. In this paper, the six different machine learning algorithms used to measure the performance are SVM, Decision Tree, Random Forest, Naïve Bayes, Logistic Regression, KNN applied on the dataset.

The expected attributes leading to heart disease in patients are available in the dataset which contains 76 features and 14 important features that are useful to evaluate the system are selected among them. If all the features taken into the consideration, then the efficiency of the system the author gets is less. To increase efficiency, attribute selection is done. In this n features have to be selected for evaluating the model which gives more accuracy. The correlation of some features in the dataset is almost equal and so they are removed. If all the attributes present in the dataset are taken into account, then the efficiency decreases considerably.

All the six machine learning methods accuracies are compared based on which one prediction model is generated. Hence, the aim is to use various evaluation metrics like confusion matrix, accuracy, precision, recall, and f1-score which predicts the disease efficiently. Comparing all six the Random Forest classifier gives the highest accuracy of 91%.

In the future, we can get more patient data where we can see other important features which can give more accurate results. And in the upcoming days, new machine learning algorithms may be developed thorough which we can get higher accuracy.

REFERENCES

[1] Soni J, Ansari U, Sharma D & Soni S (2011). Predictive data mining for medical diagnosis: an overview of heart disease prediction. International Journal of Computer Applications, 17(8), 43-8

[2] Dangare C S & Apte S S (2012). Improved study of heart disease prediction system using data mining classification techniques. International Journal of Computer Applications, 47(10), 44-8.

[3] Ordonez C (2006). Association rule discovery with the train and test approach for heart disease prediction. IEEE Transactions on Information Technology in Biomedicine, 10(2), 334-43.

[4] Shinde R, Arjun S, Patil P & Waghmare J (2015). An intelligent heart disease prediction system using k-means clustering and Naïve Bayes algorithm. International Journal of Computer Science and Information Technologies, 6(1), 637-9.

[5] Bashir S, Qamar U & Javed M Y (2014, November). An ensemble-based decision support framework for intelligent heart disease diagnosis. In International Conference on Information Society (i-Society 2014) (pp. 259-64). IEEE. ICCRDA 2020 IOP Conf. Series: Materials Science and Engineering 1022 (2021) 012072 IOP Publishing doi:10.1088/1757-899X/1022/1/012072 9

[6] Jee S H, Jang Y, Oh D J, Oh B H, Lee S H, Park S W & Yun Y D (2014). A coronary heart disease prediction model: the Korean Heart Study. BMJ open, 4(5), e005025.

[7] Ganna A, Magnusson P K, Pedersen N L, de Faire U, Reilly M, Ärnlöv J & Ingelsson E (2013). Multilocus genetic risk scores for coronary heart disease prediction. Arteriosclerosis, thrombosis, and vascular biology, 33(9), 2267-72.

[8] Jabbar M A, Deekshatulu B L & Chandra P (2013, March). Heart disease prediction using lazy associative classification. In 2013 International Mutli-Conference on Automation, Computing,Communication, Control and Compressed Sensing (iMac4s) (pp. 40- 6). IEEE.

[9] Brown N, Young T, Gray D, Skene A M & Hampton J R (1997). Inpatient deaths from acute myocardial infarction, 1982-92: analysis of data in the Nottingham heart attack register. BMJ, 315(7101), 159-64.

[10] Folsom A R, Prineas R J, Kaye S A & Soler J T (1989). Body fat distribution and self-reported prevalence of hypertension, heart attack, and other heart disease in older women. International journal of epidemiologyy, 18(2), 361-7.