

HEART DISEASE ANALYSIS USING MACHINE LEARNING ALGORITHMS

Submitted in partial fulfillment of the requirements for
the award of
Bachelor of Engineering degree in Computer Science and
Engineering

by

RAVURI SAI VIDYASAGAR (Reg. No. 37110633)



**DEPARTMENT OF COMPUTER SCIENCE AND
ENGINEERING SCHOOL OF COMPUTING**

SATHYABAMA

**INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)**

**Accredited with Grade
“A” by NAAC JEPPIAAR
NAGAR, RAJIV GANDHI
SALAI, CHENNAI – 600**

119

APRIL - 2021



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)

Accredited with "A" grade by NAAC
Jeppiaar Nagar, Rajiv Gandhi Salai, Chennai – 600 119

www.sathyabama.ac.in



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this project report is the bonafide work of **RAVURI SAI VIDYASAGAR (Reg. No. 37110633)** who carried out the project entitled
"HEART DISEASE ANALYSIS USING MACHINE LEARNING ALGORITHM'S"
under my
supervision from **August 2020** to **APRIL 2021**.

Internal Guide

Mrs. E. BRUMANCIA, M.E., (Ph.D.,)

Head of the Department

Dr.S.VIGNESHWARI, M.E.,Ph.D.,

Submitted for Viva voce Examination held on _____

Internal Examiner

External Examiner

DECLARATION

I **RAVURI SAI VIDYASAGAR** hereby declare that the Project Report entitled

**“HEART DISEASE ANALYSIS USING MACHINE
LEARNING**

ALGORITHM’S done by me under the guidance of **Mrs. E. Brumancia, M.E., (Ph.D.,)**

Department of Computer Science and Engineering at Sathyabama Institute of Science and Technology is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in Computer Science and Engineering.

DATE:

PLACE: CHENNAI

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management of SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T. Sasikala, M.E., Ph.D., Dean**, School of Computing, **Dr. S. Vigneswari, M.E., Ph.D., and Dr. L. Lakshmanan, M.E., Ph.D., Heads of the Department** of Computer Science and Engineering for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Mrs. E. Brumancia, M.E., (Ph.D.), Assistant Professor**, for her valuable guidance, suggestions, and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Computer Science and Engineering** who were helpful in many ways for the completion of the project.

ABSTRACT

Heart Failure (HEART FAILURE) is one of the regular infections that can prompt perilous circumstance and Heart Disease expectation is quite possibly the most confounded errands in clinical field. Consistently just about 26 million of patients are influencing with this sort of illness. From the heart specialist and specialist's perspective, it is mind boggling to foresee the cardiovascular breakdown on correct time. In the cutting-edge time, roughly one individual bite the dust each moment because of coronary illness. Information science assumes an essential part in handling immense measure of information in the field of medical care. As coronary illness forecast is a mind-boggling task, there is a need to robotize the expectation interaction to dodge chances related with it and alarm the patient well ahead of time. This paper utilizes coronary illness dataset accessible in UCI AI archive. The proposed work predicts the odds of Heart Disease and orders patient's danger level by actualizing diverse information mining strategies like Naive Bayes, Decision Tree, Logistic Regression and Random Forest. Along these lines, this paper presents a similar report by investigating the exhibition of various AI calculations. The preliminary outcomes confirm that Random Forest calculation has accomplished the most elevated exactness of 88.52% contrasted with other ML calculations executed.

TABLE OF CONTENTS

ABSTRACT	v
LIST OF FIGURES	Viii
LIST OF TABLES	Xi
LIST OF ABRIVATIONS	X

CHAPTER No.	TITLE	PAGE No.
1.	INTRODUCTION	1
	1.1 INTRODUCTION	1
	1.2 SYMPTOMS OF HEART DISEASE	2
	1.3 RESEARCH AND SIGNIFICANCE	3
2.	LITERATURE REVIEW	8
3.	AIM AND SCOPE	11
	3.1 AIM OF PROJECT	11
	3.2 OBJECTIVE	11
	3.3 SCOPE	12
	3.3.1 ADVANTAGES	12
	3.4 DATA SET DESCRIPTION	12
4.	ALGORITHM'S USED AND METHODOLOGY	15
	4.1 EXISTING SYSTEM	15
	4.2 HEART ATTACK PREDICTION USING INTELLEGENT TECHNIQUES.	15
	4.3 THE ARCHITECTURE OF DISEASE BY PREDICTION SYSTEM	17
	4.4 ALGORITHM'S USED	19
	4.4.1 LOGISTIC REGRESSION	19
	4.4.2 DECISION TREE	20
	4.4.3 KNN	20
	4.4.4 SVM(SUPPORT VECTOR MEACHINE)	22
	4.4.5 NAVIE BAYES	23
	4.4.6 RANDOM FOREST	24
	4.5REQUIREMENT SPECIFICATION	24

5.	RESULTS AND DISCUSSION	26
	5.1 RESULT	26
	5.2 PERFORMANCE ANALYSIS	32
6.	CONCLUSION AND FUTURE WORK	34
	6.1 CONCLUSION	34
	6.2 FUTURE WORK	35
	REFERENCES	36
	APPENDIX	38
	A. PLAGARISIM REPORT	38
	B. PAPER	39
	C. CODE	49

LIST OF FIGURES

FIGURE No.	FIGURE NAME	PAGE No.
4.1	K-NEAREST NEIGHBOUR CLASSIFICATION	21
4.2	SVM	22
5.1	DATA	27
5.2	SEX RATIO	27
5.3	HEART DISEASE FREQUENCY FOR AGES	27
5.4	HEART DISEASE FOR HEART RATE	28
5.5	LOGISTIC REGRESSION	28
5.6	KNN	29
5.7	NAVIE BAYES	29
5.8	DECISION TREE	30
5.9	RANDOM FOREST	30
5.10	ALL MODELS	31
5.11	CONFUSION MATRIX	31
5.2.1	HEART DISEASE PREDICTION APPLICATION	32

LIST OF TABLES

1.1	STAGES OF HEART FAILURE	5
2.1	DATASET DESCRIPTION	13

LIST OF ABBRIVATIONS

KNN	K- NEAREST NEIGHBOUR
SVM	SUPPORT VECTOR MEACHINE

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION

Heart is an important organ of the human body. It pumps blood to every part of our anatomy. If it fails to function correctly, then the brain and various other organs will stop working, and within few minutes, the person will die. Change in lifestyle, work related stress and bad food habits contribute to the increase in rate of several heart related diseases. Heart diseases have emerged as one of the most prominent cause of death all around the world. According to World Health Organisation, heart related diseases are responsible for the taking 17.7 million lives every year, 31% of all global deaths. In India too, heart related diseases have become the leading cause of mortality. Heart diseases have killed 1.7 million Indians in 2016, according to the 2016 Global Burden of Disease Report, released on September 15, 2017. Heart related diseases increase the spending on health care and reduce the productivity of an individual. Estimates made by the World Health Organisation (WHO), suggest that India have lost up to \$237 billion, from 2005-2015, due to heart related or Cardiovascular diseases. Thus, feasible and accurate prediction of heart related diseases is very important. Medical organisations, all around the world, collect data on various health related issues. These data can be exploited using various machine learning techniques to gain useful insights. But the data collected is very massive and, many a times, this data can be very noisy. These datasets, which are too overwhelming for human minds to comprehend, can be easily explored using various machine learning techniques. Thus, these algorithms have become very useful, in recent times, to predict the presence or absence of heart related diseases accurately.

The fundamental test in the present medical care is arrangement of best quality administrations and compelling exact analysis. Regardless of whether heart illnesses are found as the excellent wellspring of death on the planet lately,

they are likewise the ones that can be controlled and overseen successfully. The entire

exactness in administration of a sickness lies on the legitimate season of location of that illness. The proposed work makes an endeavor to distinguish these heart sicknesses at beginning phase to evade appalling outcomes. Records of huge arrangement of clinical information made by clinical specialists are accessible for dissecting and separating important information from it. Information mining procedures are the methods for extricating important and concealed data from the enormous measure of information accessible. Generally, the clinical data set comprises of discrete data. Subsequently, dynamic utilizing discrete information becomes mind boggling and intense errand. AI (ML) which is subfield of information mining handles enormous scope all around organized dataset productively. In the clinical field, AI can be utilized for determination, location, and expectation of different illnesses. The primary objective of this paper is to give a device to specialists to recognize coronary illness as beginning phase. This thus will assist with giving compelling treatment to patients and stay away from serious outcomes. ML assumes a vital part to recognize the concealed discrete examples and subsequently examine the given information. After investigation of information ML strategies help in coronary illness expectation and early determination. This paper presents execution examination of different ML methods, for example, Naive Bayes, Decision Tree, Logistic Regression and Random Forest for anticipating coronary illness at a beginning phase.

1.2 SYMPTOMS OF THE HEART DISEASE

Cardiovascular disease (CVD) or also known as heart disease include blood and heart of the human body. myocardial infarction (as a heart attack) is also a part of the CVD. Another type of Heart Disease is called Coronary Heart Disease (CHD), in this type of disease, a substance called Plaque develop in the coronary arteries. The development of plaque can block the vessel completely through the course of time.

The symptoms of the Heart Attack:

1. Chest Pain: The most common sign of a heart attack is chest pain. It mainly happens cause of the blockage of the coronary vessel of

the body due to the plaque.

2. Arms pain: The pain normally starts in the chest and move towards the arm mainly left arm.
3. Low in oxygen: Because of the plaque the level of oxygen drops in the body andcauses the dizziness and loss of balance.
4. tiredness: this cause for fatigues means simple chores become harder to do.
5. Excessive Sweating: Another common symptom is sweating.
6. Diabetics: In this, the patients have a heart rate of ~100 bpm and also occasionally having a heart rate of 130bpm.
7. Bradycardia: In this, the patient will have a slower heartbeat of 60 bpm.
8. Cerebrovascular Disease: The patient will have a high heart rate than normal usually of 200 bpm and higher than this can cause a Heart attack.
9. Shortness of breath: This often comes along with chest discomfort, but shortness of breath also can happen before chest discomfort.
10. Pain or discomfort in the jaw, neck, or back.
11. Pain or discomfort in one or both arms and shoulders
12. Hypertension: In this the patient's heart rate normally ranging from 100-200 bpm.
13. Other signs: Other possible signs include breaking out in a cold sweat, nausea or light-headedness.

1.3 RESEARCH AND SIGNIFICANCE

Current evidence indicates that heart failure is usually a progressive condition that begins with risk factors for cardiac dysfunction, proceeds to asymptomatic changes in cardiac structure and function, and then evolves into clinically overt HEART FAILURE, disability, and death. In recognition of the importance of this concept, a joint statement from the American Heart Association (AHA) and the American College of Cardiology (ACC) proposed a

novel HEART FAILURE model that classifies HEART FAILURE into stages: stage A, HEART FAILURE risk factors;

stage B, asymptomatic cardiac structural or functional abnormalities; stage C, symptomatic HEART FAILURE; and stage D, end-stage heart failure This model emphasizes progressive pathophysiology and underscores the importance of early detection and prevention of symptomatic HEART FAILURE.

The prevalence of the proposed HF stages in the community has not been determined. The prognostic implications of such a classification are unknown. Our objectives were to (1) estimate the prevalence of HF stages in a population-based cohort of 2029 adults aged ≥ 45 years, (2) provide neurohumoral validation of the staging model by measuring the association between B-type natriuretic peptide (BNP) concentration and HF stages, and (3) determine the prognostic significance of HF stages. In 2000, 90% of the 112 255 residents of Olmsted County, Minnesota, were white, 81% were urban, and 11% were ≥ 65 years of age. Since 1966, the Rochester Epidemiology Project has maintained an infrastructure for conducting population-based research, including a unified and indexed medical record system for inpatient and outpatient care.

HF Stage	AHA/ACC Guideline Description	As Operationally Defined in the Present Study
0		Without HF risk factors or abnormal ventricular structure or function; SAS class I
	Patients at high risk of developing HF because of the presence of a condition strongly associated with the development of HF. Such patients have no identified structural or functional abnormalities of the pericardium, myocardium, or cardiac valves and have never shown signs or symptoms of HF.	Hypertension, [*] diabetes mellitus, ^{*†} obesity, [*] coronary artery disease (excluding myocardial infarction) [†] without abnormal ventricular structure or function; SAS class I
B	Patients with structural heart disease that is strongly associated with the development of HF but without HF signs or symptoms	SAS class I with any of the following abnormalities: history of MI, [†] systolic dysfunction, ^{*†} echocardiographically determined LVH, ^{*†} LV enlargement, ^{*†} ECG-determined LVH, ^{*†} DD, [*] echocardiographic valvular

		heart disease, ^{*†} and LV regional wall- motion abnormality ^{*†}
--	--	--

C	Patients with current or prior symptoms of HF associated with underlying structural heart disease	LV structural or functional abnormality plus symptoms: stage C1 had SAS class II to III symptoms but did not fulfill Framingham HF criteria; stage C2 fulfilled Framingham criteria.
D	Patients with advanced structural heart disease and refractory symptoms of HF requiring specialized interventions	History of HF and SAS class IV functional status

TABLE 1: STAGES OF HEART FAILURE

Using the proposed HF stages as a template, stage 0 was defined as healthy and without HF risk factors. Stage A was defined by the presence of HF risk factors without cardiac structural or functional abnormality. Only risk factors that have been predictive of HF in longitudinal studies were used (hypertension, diabetes mellitus, obesity [defined as body mass index $\geq 30 \text{ kg/m}^2$], and coronary artery disease. Age >65 years and male sex are also well-established risk factors for HF. They are not modifiable, however, and their inclusion would (1) lead to an overestimate of the size of the pool of stage A subjects and (2) reduce the ability to measure the impact of age and sex on the association of HF stages and mortality. Therefore, age and sex were not used to assign stage A but were used to stratify and adjust multivariable analyses.

Stage B, asymptomatic ventricular dysfunction, was defined with clinical, ECG, and echocardiographic abnormalities shown to be predictive of incident HF: prior myocardial infarction, left ventricular hypertrophy (LVH), valvular heart disease, regional wall-motion abnormality, left ventricular (LV) enlargement, and

systolic and diastolic dysfunction. The asymptomatic state was defined as being SAS class I (able to exercise at ≥ 7 METS without fatigue or dyspnoea). To generate normative reference data for LVH, a healthy subset of 653 participants was derived by

excluding those with cardiac disease by medical record review, valvular heart disease, hypertension, or body mass index $\geq 30 \text{ kg/m}^2$ and those taking cardiac drugs. LVH was defined by ventricular mass/height >2 SDs above the mean in this disease-free subset, i.e., $>145.7 \text{ g/m}$ in men and $>125.2 \text{ g/m}$ in women.

CHAPTER 2

LITERATUR SURVEY

Senthil Kumar Mohan et Al, proposed Effective Heart Disease Prediction Using Hybrid Machine Learning Techniques in which system that goal is to finding basic incorporates by applying Machine Learning achieving improving the precision in the assumption for cardiovascular ailment. The assumption model is made with different mixes of features and a couple of known course of action systems. We produce an improved display level with an accuracy level of 88.7% through the forecast model for coronary illness with crossover random woodland with a direct model (HRFLM) they similarly instructed about Diverse information mining approaches and assumption methods, Suchas, KNN, LR, SVM, NN, and Vote have been genuinely acclaimed of late to recognize and anticipate coronary illness.

Sonam Nikhar et Al, has developed the paper named as Prediction of Heart Disease Using Machine Learning Algorithms by This investigation intends to give a point- by-point depiction of Naïve Bayes and decision tree classifier that are applied in ourassessment particularly in the forecast of Heart Disease. Some examination has been directed to consider the execution of judicious information mining procedure on the comparable dataset, and the outcome uncovers that Decision Tree beats over Bayesian grouping framework.

Aditi Gavhane et Al, Prediction of Heart Disease Using Machine Learning", In this paper proposed framework they utilized the neural organization calculation multi-layerperceptron (MLP) to prepare and test the dataset. In this calculation there will be numerous layers like one for input, second for yield and at least one layers are shrouded layers between these two information and yield layers. Every hub in info layer is associated with yield hubs through these concealed layers. This association is allocated for certain loads. There is another character input called inclination which is with weight b , which added to hub to adjust the perceptron. The association between the hubs can be feedforwarded

or input dependent on the necessity.

Abhay Kishore et Al, created Heart Attack Prediction Using Deep Learning in which This paper proposes a cardiovascular failure forecast framework utilizing Deep learning techniques, unequivocally Recurrent Neural System to foresee the likely possibilities of heart related contaminations of the patient. Repetitive Neural Network is a very earth shattering portrayal figuring that uses Deep Learning approach in Artificial Neural Network. The paper discusses in detail the critical modules of the structure close by the connected theory. The proposed model profound learning and information mining to give the exact results least bumbles. This paper gives an orientation and perspective for the headway of another sort of coronary episode expectation stage. Expectation stage.

Lakshmana Rao et Al, Machine Learning Techniques for Heart Disease Prediction in which the contributing components for coronary illness are more (circulatory strain, diabetes, flow smoker, elevated cholesterol, etc..). Thus, it is hard to recognize coronary illness. Various frameworks in information mining and neural frameworks have been used to find the reality of coronary illness among individuals. The possibility of CHD illness is stupefying, moreover, thusly, the infection should be managed carefully. Not doing early ID, may affect the heart or cause unexpected passing. The point of view of restorative science besides, information tunneling is utilized for finding different kinds of metabolic AI a system that makes the structure acquire from past data tests, models without being explicitly tweaked. AI makes reasoning reliant on chronicled data.

Santhana Krishnan et Al, Prediction of coronary illness utilizing AI calculation This Paper predicts coronary illness for Male Patient utilizing Classification Techniques. The itemized data about Coronary Heart sicknesses like its Facts, Common Types, and Risk Factors has been clarified in this paper. The Data Mining device utilized is WEKA (Waikato Environment for Knowledge Analysis), a decent Data Mining Tool for Bioinformatics Fields. The every one of the three accessible Interface in WEKA is utilized here; Naive Bayes, Artificial Neural Networks and Decision Tree are Main Data Mining Techniques and

through this strategies coronary illness is anticipated in this System. The primary Methodology utilized for forecast is Decision Trees like CART, C4.5, CHAID, J48, ID3 Algorithms,

and Naive Bayes Techniques.

Avinash Golande et al., proposed Heart Disease Prediction Using Effective Machine Learning Techniques in which Specialists use a couple of information mining methodologies that are accessible to help the specialists or specialists recognize the coronary illness. Normally used procedure used are decision tree, k- nearest and Naïve Bayes. Other special portrayal-based procedures used are pressing count, Part thickness, successive irrelevant smoothing out and neural frameworks, straight Kernel selfarranging aide and SVM (Bolster Vector Machine). The accompanying territory clearly gives nuances of frameworks that were used in the assessment.

Monika Gandhi et. al., used Naïve Bayes, Decision Tree and neural network algorithms and analyzed the medical dataset. There are a huge number of features involved. So, there is need to reduce the number of features. This can be done by feature selection. On doing this, they say that time is reduced.

.

CHAPTER 3

AIM AND SCOPE

3.1 AIM OF PROJECT

The primary aim of this proposed system i.e., HEART DISEASE ANALYSIS USING MACHINE LEARNING ALGORITHM'S. The classification goal is to predict whether the patient has 10-year risk of future coronary heart disease (CHD). The dataset provides the patients' information. It includes over 4,000 records and 15 attributes. Each attribute is a potential risk factor. There are both demographic, behavioral, and medical risk factors. Here we are designing an web based user interface which helps to find the chances of getting heart disease.

3.2 OBJECTIVES

Developing a Web-based HDPS using ML method for doctors as well as for the hospitals. To study about HDs prediction using various algorithms and identify the important attributes of HD . To study and research on NB and DT algorithms for comparing accuracy. To evaluate and identify the best out of two algorithms. To build a HD Prediction Model by utilizing UCI Datasets and deploy the Model into a web app. Relatively few structures use the available clinical data for assumption purposes and whether they do, they are restricted by the gigantic number of affiliations. In existing framework, the information subtleties are gotten from the patient. At that point from the client inputs,utilizing ML procedures coronary illness is dissected. Presently, the got results are contrasted and the aftereffects of existing models inside a similar space and discovered to be improved. The information of coronary illness patients gathered from the UCI research facility is utilized to find designs with NN, DT, Support Vector machines SVM, and Naive Bayes. The outcomes are contrasted for execution and exactness and these calculations. The proposed mixture strategy returns aftereffects of 80% for F- measure, contending with the other existing techniques.

3.3 SCOPE

The Scope of work explained different AI calculations and ran after finding the best calculation by examining their highlights and work predicts coronary illness by investigating the previously mentioned four order calculations and does execution examination. Each calculation has given distinctive outcome in various circumstances. The goal of this examination is to successfully foresee if the patient experiences coronary illness. Further it is dissected just minimal precision is accomplished for prescient model of coronary illness and henceforth more mind- boggling models are expected to build the exactness of anticipating the early coronary illness. The wellbeing proficient enters the information esteems from the patient's wellbeing report.

The information is taken care of into model which predicts the likelihood of having coronary illness. Fig. 1 shows the whole cycle included. The finding of coronary illness is generally founded on signs, indications, and actual assessment of the patient. In this undertaking we will utilize a portion of the AI calculations like Logistic Regression, K-closestneighbors, Support Vector Machine, Decision Tree Classifier, Random woods. In proposed procedure came for early expectation of coronary illness with high exactness and least expense and intricacy.

3.3.1 ADVANTAGES

High execution and precision rate.

Increased precision for compelling coronary illness analysis.

Handles roughest(enormous) measure of information utilizing random woodlandcalculation and highlight determination.

Reduce the time intricacy of specialists.

Cost viable for patients.

3.4 DATA SET DESCRIPTION

This database contains 76 attributes, but all published experiments refer to using a subset of 14 of them. In particular, the Cleveland database is the only one

that has been used by ML researcher's date. The "goal" field refers to the presence of heart disease in the patient. It is integer valued from 0 (no presence) to 4. Experiments with the Cleveland database have concentrated on simply attempting to distinguish presence (values 1,2,3,4) from absence (value 0).

This dataset is created based on 303 cases of heart disease in the United States. The attributes are as follows:

Name	Definition	Data Type	Description
age	Age	string	Age of a patient. The age attribute only uses numbers.
sex	Gender	string	Gender of a patient: female or male.
cp	Chest pain type	string	Chest pain types, including typical, atypical, non-anginal, and asymptomatic
trestbps	Blood pressure	string	Blood pressure of a patient.
chol	cholesterol	string	Cholesterol of a patient.
fbs	Fasting blood sugar	string	True means that a patient's fasting blood sugar is greater than 120 mg/dl. False means that a patient's fasting blood sugar is equal to or less than 120 mg/dl.
restecgr	Resting electrocardiographic result	string	The resting electrocardiographic results include normal, having ST-T wave abnormality, and showing probable or definite left ventricular hypertrophy.

thalac h	Maximum heart rate	string	Maximum heart rate of a patient.
-------------	----------------------------	--------	----------------------------------

	achieved		
exang	Exercise induced angina	string	True means that a patient has exercise induced angina. False means that a patient does not have exercise induced angina.
oldpeak	ST depression induced by exercise relative to rest	string	ST depression of a patient.
slop	Slope of the peak exercise ST segment	string	Slopes of the peak exercise ST segment, including down, flat, and up.
ca	Number of major vessels coloured by fluoroscopy	string	Number of major vessels coloured by fluoroscopy
thal	Defect type	string	defect types, including norm, fix, and rev.
status	Heart disease status	string	Health means that a patient does not have heart disease. Sick means that a patient has heart disease.

TABLE 2 : DATASET DESCRIPTION

CHAPTER 4

ALGORITHMS USED & METHODOLOGY

4.1 EXISTING SYSTEM

Relatively few structures use the available clinical data for assumption purposes and whether or not they do, they are restricted by the gigantic number of affiliations. In existing framework, the information subtleties are gotten from the patient. At that point from the client inputs, utilizing ML procedures coronary illness is dissected. Presently, the got results are contrasted and the aftereffects of existing models inside a similar space and discovered to be improved. The information of coronary illness patients gathered from the UCI research facility is utilized to find designs with NN, DT, Support Vector machines SVM, and Naive Bayes. The outcomes are contrasted for execution and exactness and these calculations. The proposed mixture strategy returns aftereffects of 80% for F-measure, contending with the other existing techniques.

The performances of different machine learning predictive models for heart disease diagnosis on full and selected features were tested. Feature selection algorithms such as Relief, mRMR, and LASSO were used to select important features, and on these selected features, the performance of the classifiers was tested. The Cleveland heart disease dataset has been implemented in several studies [13] and is used in our study. The popular machine learning classifiers logistic regression, K-NN, ANN, SVM, DT, and NB were used in the system. The model's validation and performance evaluation metrics were computed. The methodology of the proposed system structured into five stages including (1) preprocessing of dataset, (2) feature selection, (3) cross-validation method, (4) machine learning classifiers, and (5) classifiers' performance evaluation methods.

4.2 HEART ATTACK PREDICTION USING INTELLIGENT TECHNIQUES

In this section the role of intelligent technique such as neural network for heart disease prediction is explained. Latha Parthiban and R. Subramanian presented Intelligent Heart Disease Prediction System using CANFIS and Genetic

Algorithm [4]. Adaptable based fuzzy inputs are adapted with a modular neural network to

rapidly and accurately approximate complex functions. The CANFIS model combined the neural network adaptive capabilities and the fuzzy logic quantitative approach then integrated with genetic algorithm to diagnosis the presence of the disease. Coactive neuro-fuzzy inference system model has good training performance and classification accuracies. Dataset of heart disease was obtained from UCI Machine Learning Repository .Coactive Neuro-fuzzy modeling was proposed as a dependable and robust method developed to identify a nonlinear relationship and mapping between the different attributes. Dangare et al. proposed Improved Study of Heart Disease Prediction System using Data Mining Classification Techniques [6]. Prediction System for heart disease used system contains huge amount of data, used to extract hidden information for making intelligent medical diagnosis. The main objective of this research was to build Intelligent Heart Disease Prediction System that gives diagnosis of heart disease using historical heart database. To develop the system, medical terms such as sex, blood pressure, and cholesterol like 13 input attributes are used. To get more appropriate results, two more attributes i.e. obesity and smoking, as attributes were considered as important attributes for heart disease. A Multi-layer Perceptron Neural Networks (MLPNN) that maps a set of input data onto a set of appropriate. It consists of 3 layers input layer, hidden layer & output layer. There is connection between each layer & weights are assigned to each connection. The primary function of neurons of input layer is to divide input into neurons in hidden layer. The dataset consists of total 573 records in heart disease ISSN: 2229-6956(ONLINE) ICTACT JOURNAL ON SOFT COMPUTING, JULY 2013, VOLUME: 03, ISSUE: 04 607 database.

The total records are divided into two data sets one is used for training consists of 303 records & another for testing consists of 270 records. Initially dataset contained some fields, in which some value in the records was missing. These were identified and replaced with most appropriate values using Replace Missing Values filter. The Replace Missing Values filter scans all records & replaces missing values with mean mode method known as Data Preprocessing. After preprocessing the data, data mining classification techniques such as

Neural Networks, is used for classification. Many problems in business, science, industry, and medicine can be treated as classification problems. Owing to the wide range of applicability of ANN and their ability to learn complex and nonlinear relationships including noisy or less

precise information, neural networks are well suited to solve problems in biomedical engineering. So here use for the neural network technique is classification of medical dataset 14 attributes by considering the single and multilayerneural network models [19]. Olatubosun Olabode et al. [22] to classify the Cerebrovascular disease by using artificial neural network with back propagation error method. The Multi- layer perceptrons artificial neural networks with back-propagation error method were feed- forward nets with one or more layers of nodes between the input and output nodes. These additional layers contain hidden units or nodes that were not directly connected to both the input and output nodes. The neural network was trained using back propagation algorithm with sigmoid function on one hidden layer with the 16 input attributes. Predictive models were used in variety of domains for the diagnosis. Dataset for this work were collected 100 records (60 males and 40 females) from federal medical fields. The input values obtained from the records of the forms the input variables in the input layer with 16 nodes. The neural network weights were initialized randomly. This work range of the weights was between [-0.5 and 0.5] and the learning rate was set between 0.1 and 0.9. The training, validation, generalization accuracy was measured.

4.3 THE ARCHITECTURE OF DISEASE PREDICTION SYSTEM

The architecture of Disease Prediction System. The architecture of DPS includes multiple following fields:

Input :

We are taking input from the user of the disease prediction system as a symptoms list.

Get Data:

In this field, the user will provide data about their symptoms.

Data Acquisition and Processing:

In this field, the input is provided for processing. Data acquisition and processing perform two operations, first is the acquiring the

data and then second is the processing of the data and extracting information based on that acquired data.

Get Symptoms of the Body:

In this field symptoms of the body are gathered and analyzed. So that this information can be used by the algorithm to predict the possible diseases. Get Functional Problem of Body In this field, functional problems of the body that is associated with the symptoms are gathered. So that it is analyzed to get the possible disease.

Dataset Disease (symptoms, functions)

In this field, we have a predefined dataset of diseases that involves symptoms and functions that are caused by the disease. This dataset is further used to match the data that has been obtained from the user and if matched properly then the system will suggest the possible diseases.

Train_Data ()

In this field training of the system is performed. Our disease prediction system is trained using the SVM (support vector machine)

, Logistic Regression, Naive Bayes , Random Forest, Decision Tree , KNN(K- nearest Neighbor) algorithm Here we are using the SVM algorithm to solve a problem related to regression.

Prev_Disease (Dataset Disease)

In this field Dataset of the diseases is provided as parameter and processing are performed based on this dataset.

Predict_Multilinear_Regression (symptoms, function)

In this field, the prediction is performed using the MLR algorithm. In MLR, multiple independent variables are used to perform the prediction of the disease. Symptoms and their functions in the user's body are involved in the prediction.

Possible Disease (symptoms, function)

In this field symptoms and functions are passed as a parameter and possible diseases are calculated based on these parameters. Data Processing This field contains the above five data processing

fields and is the main part of our disease prediction system. It has all the necessary fields for processing the data.

Output:

After Data Acquisition and Processing, possible diseases are generated as output.

Max Possible Disease :

This field contains the maximum possible disease as output.

Min Possible Disease :

This field contains the minimum possible diseases as output.

4.4 ALGORITHM'S USED:

4.4.1 LOGISTIC REGRESSION

Logistic Regression is a supervised learning that computes the probabilities for classification problems with two outcomes. It can also be extended to predict several classes. In Logistic Regression model, we apply the sigmoid function, which is

$$S(z)=1/1+e^{-z}$$

This function successfully maps any number into the value between 0 and 1 and we can regard this value as the probability of predicting classes. For example, we have two classes and they are presence of heart disease and absence of disease. If we set the threshold as 0.5, applying the sigmoid function gives us a value of 0.7, which means the man has the 70% probability of having heart disease so we will predict that he has heart Disease.

Logistic regression uses an equation as the representation, very much like linear regression. Input values (x) are combined linearly using weights or coefficient values (referred to as the Greek capital letter Beta) to predict an output value (y). A key difference from linear regression is that the output value being modeled is a binary values (0 or 1) rather than a numeric value.

Below is an example logistic regression equation:

$$y = e^{(b_0 + b_1 x)} / (1 + e^{(b_0 + b_1 x)})$$

Where y is the predicted output, b_0 is the bias or intercept term and b_1 is the coefficient for the single input value (x). Each column in your input data has an associated b coefficient (a constant real value) that must be learned from your training data. The actual representation of the model that you would store in memory or in a file are the coefficients in the equation (the beta value or b 's).

4.4.2 DECISION TREE

A Decision tree is a decision support tool that uses a tree like graph or model of decision and their possible consequences together with chance event, resource cost, outcome and utility. This algorithm based on flow chart like tree structure in which each internal node represent a “test” on an attribute and each branch represent the outcome of the test, then each leaf node represents a class label (The decision will take after complete all the attribute). It is a popular classification algorithm and the paths from root to leaf represent classification rule. This algorithm includes Quinlan's ID3, C4.5, C5, and Breiman et al.'s CART. The main objective of partition algorithm is to find a variable- threshold pair that maximizes the homogeneity of the resulting two or more subgroups of sample. In most of the case generally used mathematical algorithm for split up process that Entropy based information gain (used in ID3, C4.5, C5), Gini index (used in CART), and Chi-squared test (used in CHAID).

Decision tree is an of administered learning calculation. This strategy is for the most part utilized in characterization issues. It performs easily with consistent and absolute credits. This calculation partitions the populace into at least two comparable sets dependent on the main indicators. Decision Tree calculation, first figures the entropy of every single property. At that point the dataset is part with the assistance of the factors or indicators with greatest data gain or least entropy. These two stages are performed recursively with the leftover ascribes.

4.4.3 KNN(K-Nearest Neighbor)

A nonparametric procedure for design order which is famously known

the K- Nearest Neighbor rule[13]. K-Nearest Neighbor procedure is quite possibly the most rudimentary yet exceptionally successful arrangement methods. It makes no suspensions about the information and is for the most part be utilized for grouping assignments when there is less or no earlier information about the information appropriation. This calculation includes finding the k closest information focuses in the preparation set to the information point for which an objective worth is inaccessible and doing out the normal estimation of the discovered information focuses to it .

K-Nearest neighbor (KNN) is a simple, lazy and nonparametric classifier. KNN is preferred when all the features are continuous. KNN is also called as case-based reasoning and has been used in many applications like pattern recognition, statistical estimation. Classification is obtained by identifying the nearest neighbor to determine the class of an unknown sample. KNN is preferred over other classification algorithms due to its high convergence speed and simplicity [17].

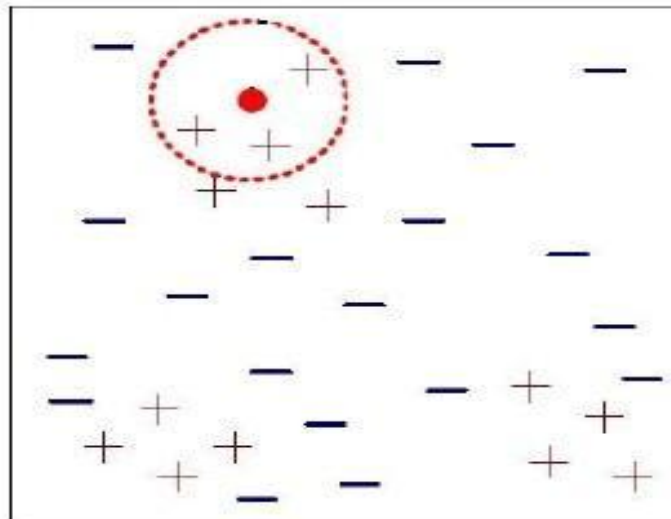


Figure 1. K-Nearest neighbor classification.

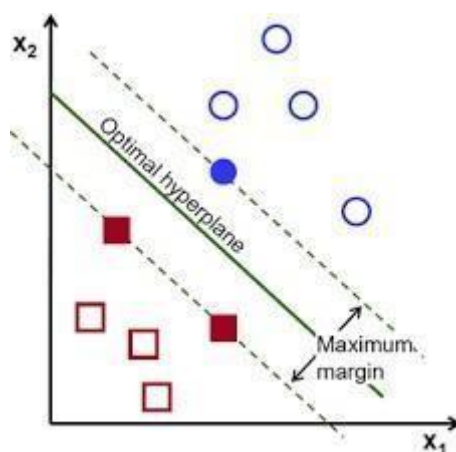
FIG.No 4.1 K-Nearest neighbour Classification

- 1) Find the k number of instances in the dataset that is closest to instance S
- 2) These k number of instances then vote to determine the class of instance S . The Accuracy of KNN depends on distance metric and K value.

Various ways of measuring the distance between two instances are cosine, Euclidian distance. To evaluate the new unknown sample, KNN computes its K nearest neighbors and assign a class by majority voting.

4.4.4 SVM (Support Vector Machine)

Support Vector Machine is an extremely popular supervised machine learning technique (having a pre-defined target variable) which can be used as a classifier as well as a predictor. For classification, it finds a hyper-plane in the feature space that differentiates between the classes. An SVM model represents the training data points as points in the feature space, mapped in such a way that points belonging to separate classes are segregated by a margin as wide as possible. The test data points are then mapped into that same space and are classified based on which side of the margin they fall. Fig. 1: Support Vector Machine Shan Xu et al. have used SVM to achieve an accuracy of 98.9% in People's Hospital dataset [5]. In [9], SVM performs the best with 85.7655% of correctly classified instance and in [10] SVM is used with boosting technique to give an accuracy of 84.81%. Houda Mezrigui et al. have used SVM to attain a f-measure value of 93.5617 [11]. In [12] SVM classifies the pixel variation with an accuracy of 92.1%



helping to identify the affected region accurately

Fig.No 4.2 : SVM

4.4.5 NAVIE BAYES

A Naive Bayes' classifier may be a term addressing a simple probabilistic classification supported applying Bayes' theorem. In easy terms, a Naïve Bayes classifier assumes that the presence (or absence) of a specific feature of a category is unrelated to the presence (or absence) of the other feature. As an example, a fruit could also be thought of to be an apple if it's red, round, and regarding 4" in diameter. Even supposing these options rely on the existence of the opposite options, a Naive Bayes' classifier considers all of those properties to independently contribute to the likelihood that this fruit is an apple.

Given training data X , posterior probability of a hypothesis H , $P(H|X)$, follows the Bayes theorem $P(H|X)=P(X|H)P(H)/P(X)$ (1.1) Algorithm The Naive Bayes algorithm is based on Bayesian theorem as given by equation (1.1) Steps in algorithm are as follows:

1. Each data sample is represented by an n dimensional feature vector, $X = (x_1, x_2, \dots, x_n)$, depicting n measurements made on the sample from n attributes, respectively A_1, A_2, A_n
2. Suppose that there are m classes, $C_1, C_2, \dots, C_m$. Given an unknown data sample, X (i.e., having no class label), the classifier will predict that X belongs to the class having the highest posterior probability, conditioned if and only if: $P(C_i|X) > P(C_j|X)$ for all $1 < j < m$ and $j \neq i$ Thus we maximize $P(C_i|X)$. The class C_i for which $P(C_i|X)$ is maximized is called the maximum posteriori hypothesis. By Bayes theorem,
3. As $P(X)$ is constant for all classes, only $P(X|C_i)P(C_i)$ need be maximized. If the class prior probabilities are not known, then it is commonly assumed that the classes are equally likely, i.e. $P(C_1) = P(C_2) = \dots = P(C_m)$, and we would therefore maximize $P(X|C_i)$. Otherwise, we maximize $P(X|C_i)P(C_i)$. Note that the class prior probabilities may be estimated by $P(C_i) = s_i/s$, where S_i is the number of training samples of class C_i , and s is the total number of training samples. on X . That is, the naive probability assigns an unknown

sample X to the class Ci

4.4.6 RANDOM FOREST-

Random Forest algorithms are used for classification as well as regression. It creates a tree for the data and makes prediction based on that. Random Forest algorithm can be used on large datasets and can produce the same result even when large sets record values are missing. The generated samples from the decision tree can be saved so that it can be used on other data. In random forest there are two stages, firstly create a random forest then make a prediction using a random forest classifier created in the first stage.

This ensemble classifier builds several decision trees and incorporates them to get the best result. For tree learning, it mainly applies bootstrap aggregating or bagging. For a given data, $X = \{x_1, x_2, x_3, \dots, x_n\}$ with responses $Y = \{y_1, y_2, y_3, \dots, y_n\}$ which repeats the bagging from $b = 1$ to B

4.5 REQUIREMENT SPECIFICATION

This proposed software runs effectively on a computing system that has the minimum requirements. Undertaking all the equipment necessities are not satisfied but rather exist in their systems administration between the customer's machines already. So, the main need is to introduce appropriate equipment for the product. The requirements are split into two categories, namely:

4.5.1 Software Requirements

The basic software requirements to run the program are:

- i. Microsoft Windows XP, Windows 7, Windows 10
- ii. Python code editor
- iii. Jupyter notebook.
- iv. Anaconda prompt
- v. Streamlit

4.5.2 Hardware Requirements

The basic hardware required to run the program are:

- i. Hard disk of 20 GB
- ii. System memory (RAM) of 512 MB
- iii. I3 processor-based computer or higher.

CHAPTER 5

RESULTS AND DISCUSSION, PERFORMANCE ANALYSIS

5.1 RESULTS

The results obtained by applying Random Forest, Decision Tree, Naive Bayes and Logistic Regression are shown in this Section. The metrics used to carry out performance analysis of the algorithm are Accuracy score, Precision (P), Recall

(R) and F- measure. Precision (mentioned in equation (2)) metric provides the measure of positive analysis that is correct. Recall [mentioned in equation (3)] defines the measure of actual positives that are correct.

F-measure [mentioned in equation (4)] tests accuracy. Precision = $(TP) / (TP + FP)$

$$(2) \text{ Recall} = (TP) / (TP + FN)$$

$$(3) \text{ F-Measure} = (2 * \text{Precision} * \text{Recall}) / (\text{Precision} + \text{Recall})$$

- TP True positive: the patient has the disease and the test is positive.
- FP False positive: the patient does not have the disease but the test is positive.
- TN True negative: the patient does not have the disease and the test is negative.
- FN False negative: the patient has the disease but the test is negative.

In the experiment the pre-processed dataset is used to carry out the experiments and the above-mentioned algorithms are explored and applied. The above mentioned performance metrics are obtained using the confusion matrix. Confusion Matrix describes the performance of the model. The confusion matrix obtained by the proposed model for different algorithms is shown below in Table 2. The accuracy score obtained for Random Forest, Decision Tree, Logistic Regression and Naive Bayes classification techniques

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
0	63	1	3	145	233	1	0	150	0	2.30	0	0	1	1
1	37	1	2	130	250	0	1	187	0	3.50	0	0	2	1
2	41	0	1	130	204	0	0	172	0	1.40	2	0	2	1
3	56	1	1	120	236	0	1	178	0	0.80	2	0	2	1
4	57	0	0	120	354	0	1	163	1	0.60	2	0	2	1

FIG.NO 5.1 DATA

SEX RATIO:

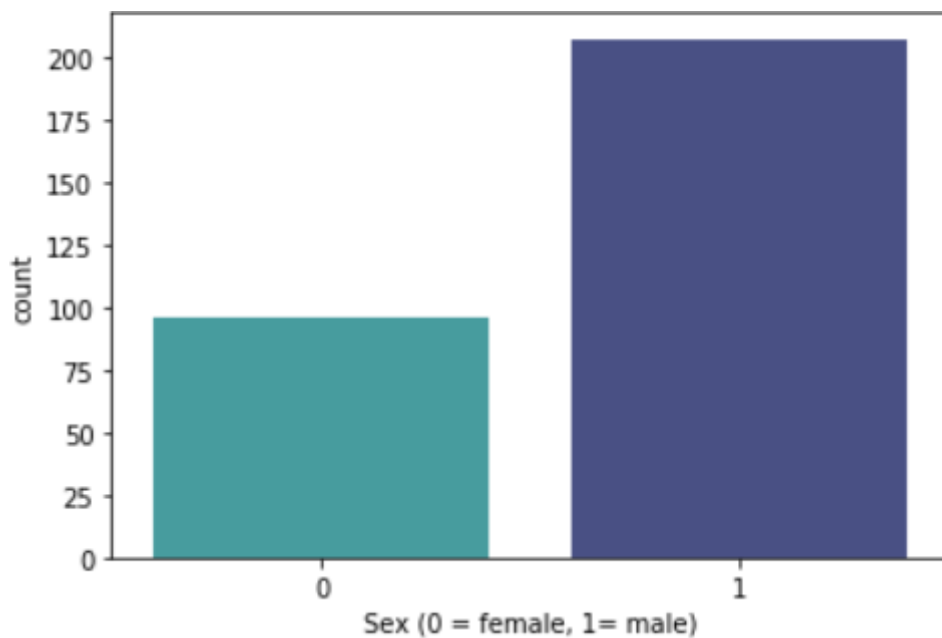


FIG .NO 5.2 SEX RATIO

Percentage of Female Patients: 31.68% Percentage of Male Patients: 68.32%

HEART DISEASE FREQUENCY FOR AGES

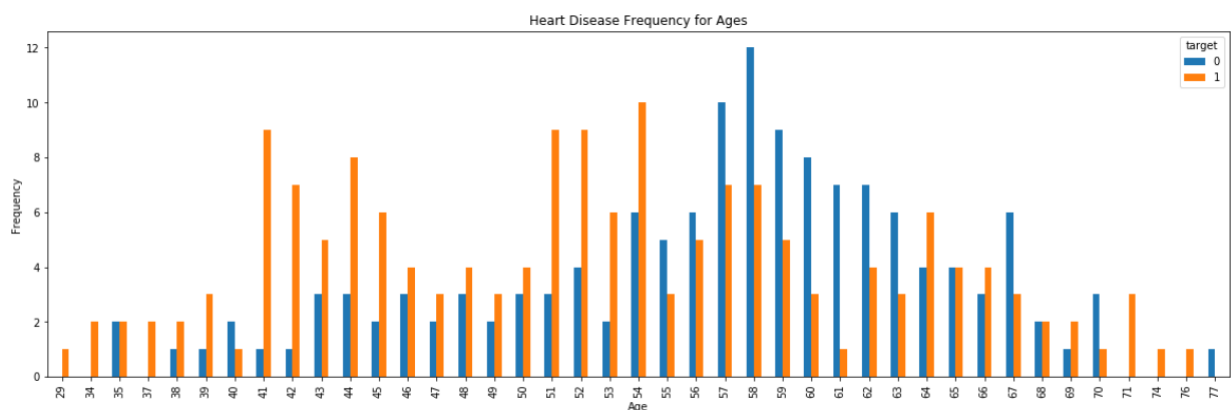


FIG.NO 5.3 HEART DISEASE FREQUENCY FOR AGES
HEART DISEASE FOR HEART RATE

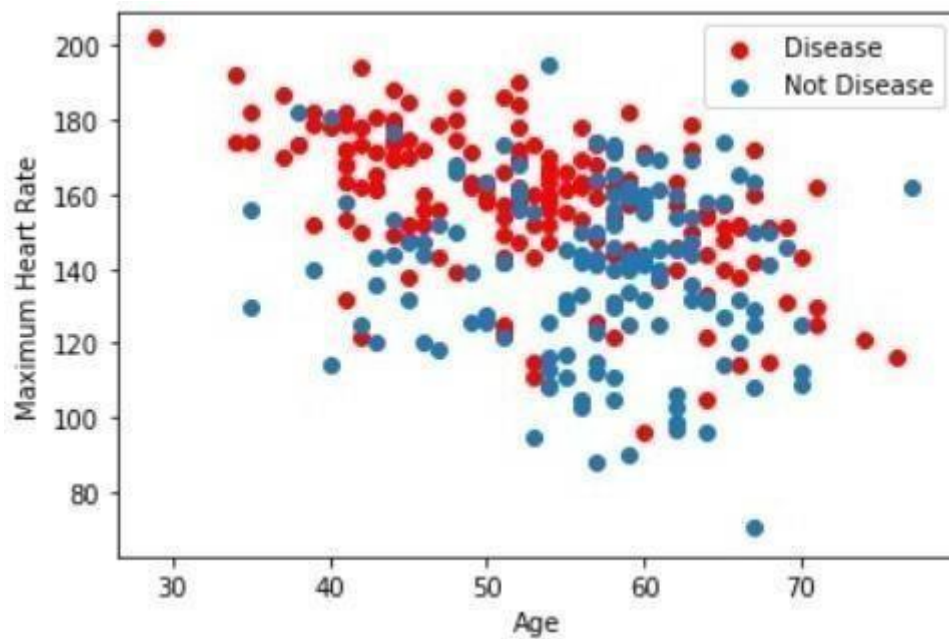
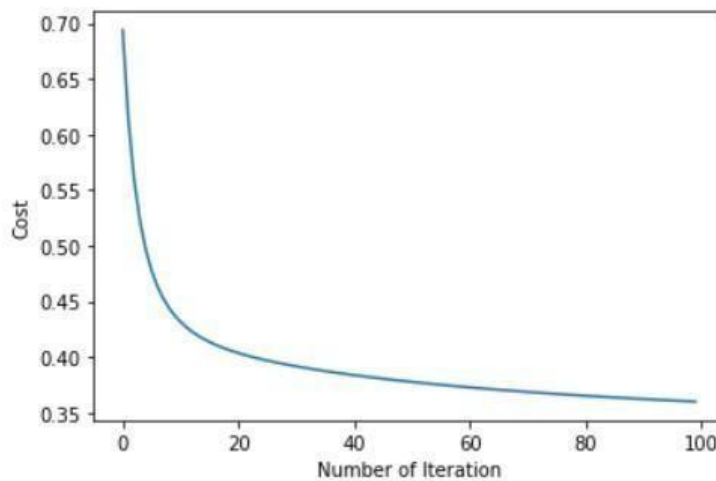


FIG.NO 5.4 HEART DISEASE FOR HEART RATE
LOGISTIC REGRESSION ANALYSIS

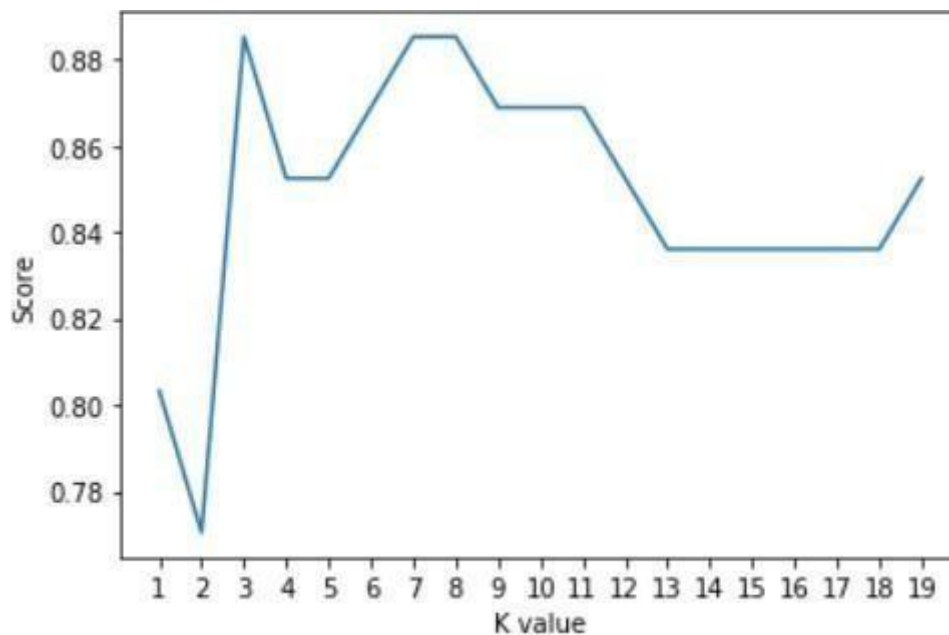
iteration: 100
cost: 0.35977361236645333



Manuel Test Accuracy: 86.89%

FIG.NO 5.5 LOGISTIC REGRESSION

KNN ACCURACY



Maximum KNN Score is 88.52%

FIG.NO 5.6 KNN

NAVIE BAYES ACCURACY

```
1 from sklearn.naive_bayes import GaussianNB
2 nb = GaussianNB()
3 nb.fit(x_train.T, y_train.T)
4
5 acc = nb.score(x_test.T, y_test.T)*100
6 accuracies['Naive Bayes'] = acc
7 print("Accuracy of Naive Bayes: {:.2f}%".format(acc))
```

Accuracy of Naive Bayes: 86.89%

Accuracy of Naive Bayes: 86.89%

FIG.NO 5.7 NAVIE BAYES

DECISION TREE ACCURACY

```
1 from sklearn.tree import DecisionTreeClassifier
2 dtc = DecisionTreeClassifier()
3 dtc.fit(x_train.T, y_train.T)
4
5 acc = dtc.score(x_test.T, y_test.T)*100
6 accuracies['Decision Tree'] = acc
7 print("Decision Tree Test Accuracy {:.2f}%".format(acc))
```

Decision Tree Test Accuracy 77.05%

Test Accuracy of Decision Tree Algorithm: 78.69%

FIG.NO 5.8 DECISION

TREE RANDOM FROEST ACCURACY

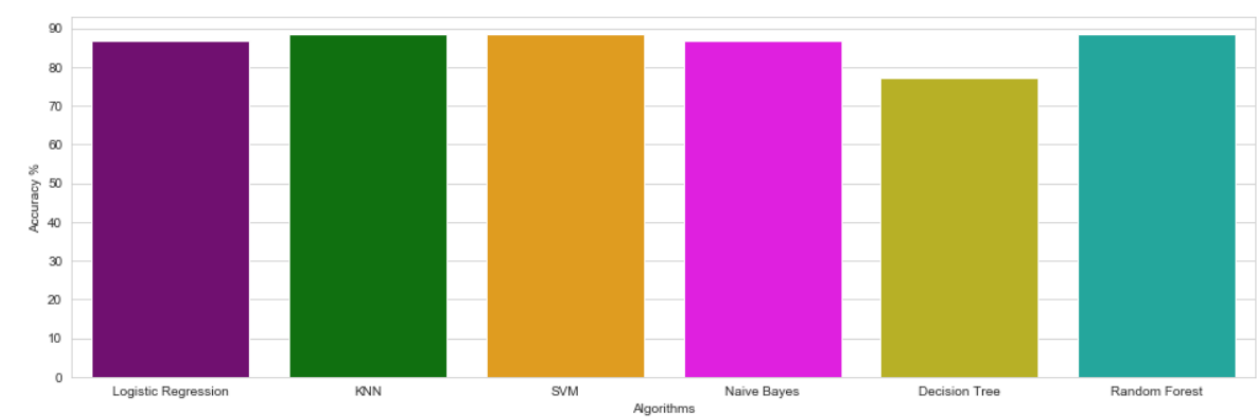
```
1 # Random Forest Classification
2 from sklearn.ensemble import RandomForestClassifier
3 rf = RandomForestClassifier(n_estimators = 1000, random_state = 1)
4 rf.fit(x_train.T, y_train.T)
5
6 acc = rf.score(x_test.T, y_test.T)*100
7 accuracies['Random Forest'] = acc
8 print("Random Forest Algorithm Accuracy Score : {:.2f}%".format(acc))
```

Random Forest Algorithm Accuracy Score : 88.52%

Test Accuracy of Random Forest: 88.52%

FIG.NO 5.9 RANDOM FOREST

COMPARING ALL MODELS



Our models work fine but best of them are KNN and Random Forest with 88.52% of accuracy. Let's look their confusion matrixes.

FIG.N0 5.10 ALL MODEL COMPARISION
CONFUSION MATRIX

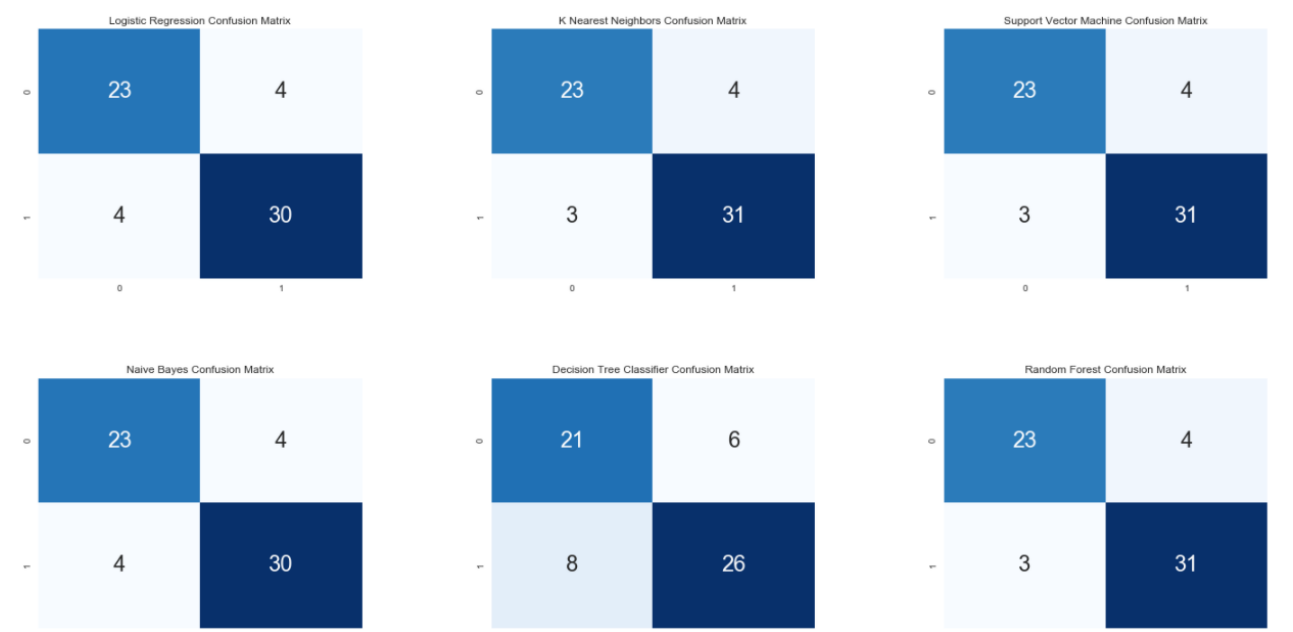
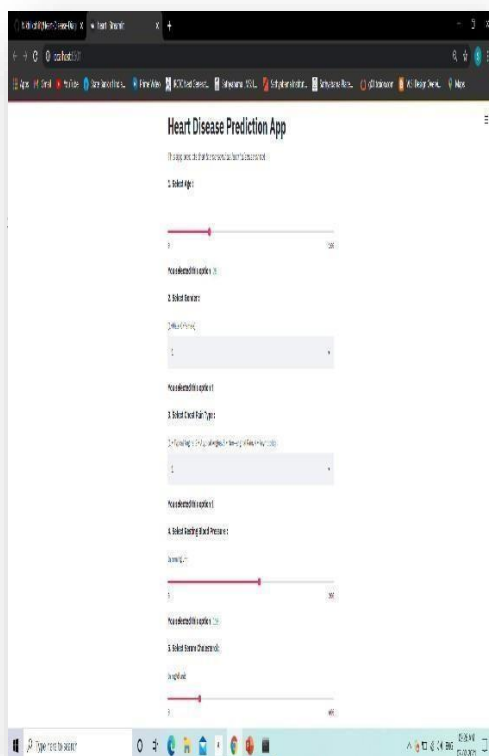


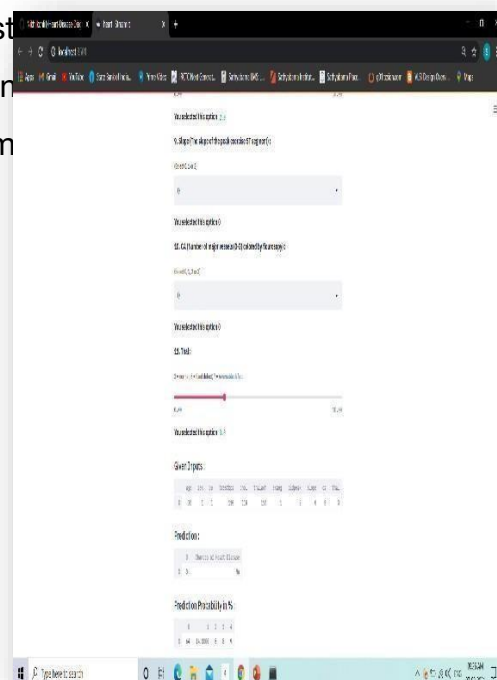
FIG NO 5.11 CONFUSION MATRIX

5.2 PERFORMANCE ANALYSIS

The test was done on an openly accessible data set for coronary illness. The dataset contains a sum of 303 records that were separated into two sets, preparing set (40%) and testing set (60%). An information mining apparatus named Weka 3.6.11 was utilized for the analysis. Furthermore, multilayer perceptron neural organization (MLPNN) with backpropagation (BP) was utilized as the preparation calculation. Sending of model in a web interface for easy-to-understand expectation. Table 1 introduced the correlation of the analysis' outcomes directed in2 this investigation with the past work. Generally speaking, each classifier has shown great execution in this examination as contrast withthe past work. As per the table, the exactness of the decision tree model was the most elevated between all models, while the exhibition of the Naïve Bayes has shown the least precision in this investigation.



HEART DISEASE PREDICTION APPLICATION



with the past work delineated in the third segment of Table 1, the exploration applied the trial utilizing five calculations [7]. That exploration utilized the equivalent dataset (UCI) with highlight determination approach. The precision of our model has improved the presentation of the classifiers. For instance, Naïve Bayes precision expanded 3%, Logistic Regression and SVM upgraded 5%, Decision Tree improved the exactness proportion 11%, and finally the Random Forest AI classifier expanded 8%. Nonetheless, in past work, the investigation likewise utilized a similar stage, which is Rapid Miner. However, precision execution enlarged in our work may be a result of utilizing 10- overlap cross approval, while the past work utilized 5-crease cross approval approach. More emphasis during the learning stage can assist with creating more precise outcomes. Another conceivable explanation for the positive advancement in the precision is the size of the dataset, which has been enhanced in this examination as talked about in information outline area. It features that the huge size of the dataset can make positive effect on classifier precision as it upgrades the learning cycle.

CHAPTER 6 SUMMARY AND CONCLUSIONS

6.1 CONCLUSION

With the increasing number of deaths due to heart diseases, it has become mandatory to develop a system to predict heart diseases effectively and accurately. The motivation for the study was to find the most efficient ML algorithm for detection of heart diseases. This study compares the accuracy score of Decision Tree, Logistic Regression, Random Forest and Naive Bayes algorithms for predicting heart disease using UCI machine learning repository dataset. The result of this study indicates that the Random Forest algorithm is the most efficient algorithm with accuracy score of 90.16% for prediction of heart disease. In future the work can be enhanced by developing a web application based on the Random Forest algorithm as well as using a larger dataset as compared to the one used in this analysis which will help to provide better results and help health professionals in predicting the heart disease effectively and efficiently.

Eventually, we are able to achieve all the proposed objectives of this project. We developed a web application of HDPS that has all features and functionalities what we planned. The research of this project is successfully met in the system. This system resolves the real-environment problem. It is successfully predicted the presence of HD in patient. It is also store and manage the prediction report of heart patient by doctor account. Admin user can handle create doctor account, mange doctor account and view the report of the patient. The overall system can solve the problem statement of the project and make a novel tool to predict an HD at hospital. We were able to research or investigate our topic by conducting literature review as domain and technical research. At domain research, we achieved 13 important attributes of HD that every researcher had done their research to predict the HD. we also achieved an appropriate two algorithm to provide high accuracy in heart disease dataset of the heart patient such as DT and NB and finally NB provides 88% accuracy among other algorithms. But the research of dataset of heart may failed due to

distributed result of HD dataset. That's why we change the dataset from Kaggle. At technical and other research, we were successfully meet finding to final project. Programming language, database, methodology is suitable to conduct best

performance to accomplish this project. In our research of the topic, we were difficult to find details about attributes of the HD. The improper attributes name, their impact, and their sub-attributes were less information in journal paper and dataset description. Due to lack of spare time of doctor, we were unable to collect more data about HD. lack of time constraint, we were unable to explore other features to include the system and explore different heart diseasetype prediction

6.2 FUTURE OF WORK

Future enhance of the HDPS is to predict a specific HD type such Heart attracts, CVD, CAD, etc. the potential of the HDPS in a different area are hospital, Clinic, smartphone, smart wear, hospital/police emergency system and integrate with fitness mobile application. We will integrate this model in hospital and clinic system to predict heartdisease. We will implement this HDP Model into smart wears to detect essential attributesof HD and suggest to the precaution of HD. we will also apply this model into a mobile app to easily test ourselves HD. we will integrate smart wear to the hospital and police emergency system to save the life of the patient at the emergency condition.

REFERENCE

- [1] Abhay Kishore¹, Ajay Kumar², Karan Singh³, Maninder Punia⁴, Yogita Hambir⁵,” HeartAttack Prediction Using Deep Learning”, International Research Journal of Engineering and Technology (IRJET), Volume: 05 Issue: 04 | Apr-2018.
- [2] Aditi Gavhane, Gouthami Kokkula, Isha Pandya, Prof. Kailas Devadkar (PhD),” Prediction of Heart Disease Using Machine Learning”, Proceedings of the 2nd International conference on Electronics, Communication and Aerospace Technology (ICECA 2018).IEEE Conference Record # 42487; IEEE Xplore ISBN:978-1- 5386-09651
- [3] Avinash Golande, Pavan Kumar T,” Heart Disease Prediction Using Effective Machine Learning Techniques”, International Journal of Recent Technology and Engineering (IJRTE) ISSN: 2277-3878, Volume-8, Issue-1S4, June 2019.
- [4] A.Lakshmanarao, Y.Swathi, P.Sri Sai Sundareswar,” Machine Learning Techniques For Heart Disease Prediction”, International Journal Of Scientific & Technology Research Volume 8, Issue 11, November 2019.
- [5] Mr.Santhana Krishnan.J, Dr.Geetha.S,” Prediction of Heart Disease Using Machine Learning Algorithms”,2019 1st International Conference on Innovations in Information and Communication Technology(ICIICT), doi:10.1109/ICIICT1.2019.8741465.
- [6] M. S. Amin, Y. K. Chiam, K. D. Varathan,“Identification of significant features and data mining techniques in predicting heart disease,” Telematics Inform., vol. 36, pp. 82–93, Mar.2019.
- [7] V. Manikantan and S. Latha, “Predicting the analysis of heart disease symptoms using medicinal data mining methods”, International Journal of Advanced Computer Theory and Engineering, vol. 2, pp.46-51, 2013.
- [8] V.V. Ramalingam, Ayantan Dandapath, M Karthik Raja,” Heart disease prediction using machine learning techniques: a survey”, International Journal of Engineering & Technology, 7 (2.8) (2018) 684-687.
- [9] Senthilkumar Mohan, Chandrasegar Thirumalai, Gautam Srivastava —Effective Heart Disease Prediction Using Hybrid Machine Learning TechniquesII, Digital Object Identifier 10.1109/ACCESS.2019.2923707, IEEE Access, VOLUME

7, 2019 S.P. Bingulac, —On the Compatibility of Adaptive Controllers,II Proc. Fourth Ann. Allerton Conf. Circuits andSystems Theory, pp. 8-16, 1994. (Conference proceedings)

- [10] Sonam Nikhar, A.M. Karandikar” Prediction of Heart Disease Using Machine Learning Algorithms” International Journal of Advanced Engineering, Management and Science (IJAEMS) Infogain Publication,[Vol-2, Issue-6, June-2016].I.S. Jacobs and C.P. Bean, “Fine particles, thin films and exchange anisotropy,” in Magnetism, vol. III, G.T. Rado and H. Suhl, Eds. New York: Academic, 1963, pp. 271-350.
- [11] S. M. S. Shah, S. Batool, I. Khan, M. U. Ashraf, S. H. Abbas, and S. A. Hussain,“Feature extraction through parallel probabilistic principal component analysis for heart disease diagnosis,” Phys. A, Stat. Mech. Appl.,vol. 482, pp. 796– 807,2017. doi:10.1016/j.physa.2017.04.113.

APPENDIX

A. Plagiarism Report

ORIGINALITY REPORT

18%

SIMILARITY INDEX

12%

INTERNET SOURCES

13%

PUBLICATIONS

9%

STUDENT PAPERS

B. Journal Paper

HEART DISEASE ANALYSIS USING MACHINE LEARNING ALGORITHM'S

RAVURI SAI VIDYASAGAR [1], RAVI RAGHUNANDAN CHOWDARY [2] , BRUMANCIA E[3]

[1][2] UG Student, Dept. of CSE, Sathyabama Institute of Science and Technology, Chennai, India

[3] Associate professor, Dept. of CSE, Sathyabama Institute of Science and Technology, Chennai, India

ABSTRACT— *In the ebb and flow time, Heart Failure (HF) is one of the regular infections that can prompt perilous circumstance and Heart Disease expectation is quite possibly the most confounded errands in clinical field. Consistently just about 26 million of patients are influencing with this sort of illness. From the heart specialist and specialist's perspective, it is mind boggling to foresee the cardiovascular breakdown on correct time. In the cutting-edge time, roughly one individual bite the dust each moment because of coronary illness. Information science assumes an essential part in handling immense measure of information in the field of medical care. As coronary illness forecast is a mind-boggling task, there is a need to robotize the expectation interaction to dodge chances related with it and alarm the patient well ahead of time. This paper utilizes coronary illness dataset accessible in UCI AI archive. The proposed work predicts the odds of Heart Disease and orders patient's danger level by actualizing diverse information mining strategies like Naive Bayes, Decision Tree, Logistic Regression and Random Forest. Along these lines, this paper presents a similar report by investigating the exhibition of various AI calculations. The preliminary outcomes confirm that Random Forest calculation has accomplished the most elevated exactness of 88.52% contrasted with other ML calculations executed.*

Keywords— *Decision Tree, Naive Bayes, Logistic Regression, Random Forest, Heart Disease Prediction*

INTRODUCTION

The fundamental driver of heart stroke is because of blockage in courses. It has numerous different names, for example, cardiovascular infection and blood vessel hypertension [1]. Roughly, there are just about 26 million individuals around the planet influencing with coronary illness [2]. The concern point is, this proportion is required to increment quickly in coming years, if precautionary measures are not taken proficiently [3]. Aside from making way of life solid and diet

control, the ideal time diagnosing and extensive examination are other fundamental variables, which can eventually save the lives [4]. Subsequently, this paper has made a little stride towards saving the

existences of HF patients and portrays an approach to improve the exhibition of diagnosing the patients on the bases of their clinical history. More often than not patients goes for a few tests, which can overburden them with extra proactive tasks, time, and without a doubt extra monetary charges [5]. As past examinations recommended the regular purposes for coronary illness can be undesirable food, tobacco, unreasonable sugar, overweight or additional muscle versus fat [3], [6]. Though the normal manifestations can be torment in arms and chest [7]. Recognizably, these reasons are autonomous from one another; appropriate investigation on this sort of dataset can improve the way toward diagnosing and can help the heart specialists too. Already, various investigates utilized number of procedures to improve the HF finding interaction, for example, Extreme Learning Machine [8], coronary illness order [9], and AI classifiers [1].

The fundamental test in the present medical care is arrangement of best quality administrations and compelling exact analysis [1]. Regardless of whether heart illnesses are found as the excellent wellspring of death on the planet lately, they are likewise the ones that can be controlled and overseen successfully. The entire exactness in administration of a sickness lies on the legitimate season of location of that illness. The proposed work makes an endeavor to distinguish these heart sicknesses at beginning phase to evade appalling outcomes. Records of huge arrangement of clinical information made by clinical specialists are accessible for dissecting and separating important information from it. Information mining procedures are the methods for extricating important and concealed data from the enormous measure of information accessible. Generally the clinical data set comprises of discrete data. Subsequently, dynamic utilizing discrete information becomes mind boggling and intense errand. AI (ML) which is subfield of information mining handles enormous scope all around organized dataset productively. In the clinical field, AI can be utilized for determination, location and expectation of different illnesses. The primary objective of this paper is to give a device to specialists to recognize coronary illness as beginning phase [5]. This thus will assist with giving compelling treatment to patients and stay away from serious outcomes. ML assumes a vital part to recognize the concealed discrete examples and subsequently examine the given information. After investigation of information ML strategies help in coronary illness expectation and early determination. This paper presents execution examination of different ML methods, for example, Naive Bayes, Decision Tree, Logistic Regression and Random Forest for anticipating coronary illness at a beginning phase [3].

I RELATED WORK

Senthil Kumar Mohan et al[1] proposed Effective Heart Disease Prediction Using Hybrid Machine Learning Techniques in which system that goal is to finding basic incorporates by applying Machine Learning achieving improving the precision in the assumption for cardiovascular ailment. The assumption model is made with different mixes of features and a couple of known course of action systems. We produce an improved display level with an accuracy level of 88.7%

through the forecast model for coronary illness with crossover random woodland with a direct model (HRFLM)

they similarly instructed about Diverse information mining approaches and assumption methods, Such as, KNN, LR, SVM, NN, and Vote have been genuinely acclaimed of late to recognize and anticipate coronary illness.

Sonam Nikhar et al [2] has developed the paper named as Prediction of Heart Disease Using Machine Learning Algorithms by This investigation intends to give a point-by-point depiction of NaïveBayes and decision tree classifier that are applied in our assessment particularly in the forecast of Heart

Disease. Some examination has been directed to consider the execution of judicious information mining procedure on the comparable dataset, and the outcome uncovers that Decision Tree beats over Bayesian grouping framework.

Aditi Gavhane et al [3] Prediction of Heart Disease Using Machine Learning", In this paper proposed framework they utilized the neural organization calculation multi-layer perceptron (MLP) to prepare and test the dataset. In this calculation there will be numerous layers like one for input, second for yield and at least one layers are shrouded layers between these two information and yield layers. Every hub in info layer is associated with yield hubs through these concealed layers. This association is allocated for certain loads. There is another character input called inclination which is with weight b , which added to hub to adjust the perceptron. The association between the hubs can be feedforwarded or input dependent on the necessity.

Abhay Kishore et al [4] created Heart Attack Prediction Using Deep Learning in which This paper proposes a cardiovascular failure forecast framework utilizing Deep learning techniques, unequivocally Recurrent Neural System to foresee the likely possibilities of heart related contaminations of the patient. Repetitive Neural Network is a very earth shattering portrayal figuring that uses Deep Learning approach in Artificial Neural Network. The paper discusses in detail the critical modules of the structure close by the connected theory. The proposed model profound learning and information mining to give the exact results least bumbles. This paper gives an orientation and perspective for the headway of another sort of coronary episode expectation stage. Expectation stage.

Lakshmana Rao et al [5] Machine Learning Techniques for Heart Disease Prediction in which the contributing components for coronary illness are more (circulatory strain, diabetes, flow smoker, elevated cholesterol, etc.). Thus, it is hard to recognize coronary illness. Various frameworks in information mining and neural frameworks have been used to find the reality of coronary illness among individuals. The possibility of CHD illness is stupefying, moreover, thusly, the infection should be managed carefully. Not doing early ID, may affect the heart or cause unexpected passing. The point of view of restorative science besides, information tunneling is

utilized

for finding different kinds of metabolic AI a system that makes the structure acquire from past data tests, models without being explicitly tweaked. AI makes reasoning reliant on chronicled data.

Santhana Krishnan et al [6] Prediction of coronary illness utilizing AI calculation This Paper predicts coronary illness for Male Patient utilizing Classification Techniques. The itemized data about Coronary Heart sicknesses like its Facts, Common Types, and Risk Factors has been clarified in this paper. The Data Mining device utilized is WEKA (Waikato Environment for Knowledge Analysis), a decent Data Mining Tool for Bioinformatics Fields. The every one of the three accessible Interface in WEKA is utilized here; Naive Bayes, Artificial Neural Networks and Decision Tree are Main Data Mining Techniques and through this strategies coronary illness is anticipated in this System. The primary Methodology utilized for forecast is Decision Trees like CART, C4.5, CHAID, J48, ID3 Algorithms, and Naive Bayes Techniques.

Avinash Golande et al[7] proposed Heart Disease Prediction Using Effective Machine Learning Techniques in which Specialists use a couple of information mining methodologies that are accessible to help the specialists or specialists recognize the coronary illness. Normally used procedure used are decision tree, k-nearest and Naïve Bayes. Other special portrayal-based procedures used are pressing count, Part thickness, successive irrelevant smoothing out and neural frameworks, straight Kernel selfarranging aide and SVM (Bolster Vector Machine). The accompanying territory clearly gives nuances of frameworks that were used in the assessment.

II EXISTING SYSTEM

Relatively few structures use the available clinical data for assumption purposes and whether or not they do, they are restricted by the gigantic number of affiliations. In existing framework, the information subtleties are gotten from the patient. At that point from the client inputs, utilizing ML procedures coronary illness is dissected. Presently, the got results are contrasted and the aftereffects of illness patients gathered from the UCI research facility is utilized to find designs with NN, DT, SupportVector machines SVM, and Naive Bayes. The outcomes are contrasted for execution and exactness and these calculations. The proposed mixture strategy returns aftereffects of 80% for F- measure, contending with the other existing techniques.

DISADVANTAGES OF THE EXISTING SYSTEM

- Detection is past the domain of creative mind at a past stage.
- In the current structure, practical use of various assembled data is monotonous
- Prediction of cardiovascular sickness results isn't exact.
- Data mining strategies doesn't assist with giving powerful dynamic.
- Cannot handle colossal datasets for tolerant records.

III PROPOSED SYSTEM

In the proposed work explained different AI calculations and ran after finding the best calculation by examining their highlights and work predicts coronary illness by investigating the previously mentioned four order calculations and does execution examination. Each calculation has given distinctive outcome in various circumstances. The goal of this examination is to successfully foresee if the patient experiences coronary illness. Further it is dissected just minimal precision is accomplished for prescient model of coronary illness and henceforth more mind boggling models are expected to build the exactness of anticipating the early coronary illness. The wellbeing proficient enters the information esteems from the patient's wellbeing report.

The information is taken care of into model which predicts the likelihood of having coronary illness. Fig. 1 shows the whole cycle included. The finding of coronary illness is generally founded on signs, indications and actual assessment of the patient. In this undertaking we will utilize a portion of the AI calculations like Logistic Regression, K-closest neighbors, Support Vector Machine, Decision Tree Classifier, Random woods. In proposed procedure came for early expectation of coronary illness with high exactness and least expense and intricacy.

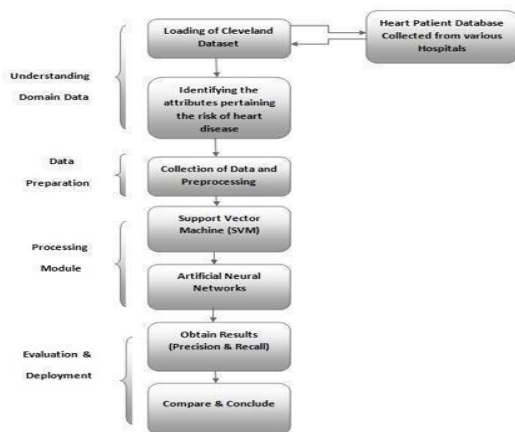


FIG 1 OVERVIEW OF THE PROPOSED SYSTEM ADVANTAGES OF THE PROPOSED SYSTEM

- High execution and precision rate.
- Increased precision for compelling coronary illness analysis.
- Handles roughest(enormous) measure of information utilizing random woodland calculation and highlight determination.
- Reduce the time intricacy of specialists.
- Cost viable for patients.

IV MACHINE LEARNING ALGORITHMS AND TECHNIQUES USED

NAÏVE BAYES

The Bayes Theorem. It accepts freedom among indicators, i.e., the credits or highlights ought to be not associated to each other or ought not, in at any rate, be identified with one another. Regardless of whether there is reliance, still every one of these highlights or qualities freely add to the likelihood and that is the reason it is called Naïve

SUPPORT VECTOR MACHINE

Support Vector Machine is a very mainstream administered AI technique (having a pre-characterized target variable) which can be utilized as a classifier just as an indicator. For arrangement, it finds a hyper-plane in the element space that separates between the classes. A SVM model addresses the preparation information focuses as focuses in the element space, planned so that focuses having a place with discrete classes are isolated by an edge as wide as could be expected. The test information focuses are then planned into that equivalent space and are ordered dependent on which side of the edge they fall.

K – NEAREST NEIGHBOR

A nonparametric procedure for design order which is famously known the K-Nearest Neighbor rule[13]. K-Nearest Neighbor procedure is quite possibly the most rudimentary yet exceptionally successful arrangement methods. It makes no suspicions about the information and is for the most part be utilized for grouping assignments when there is less or no earlier information about the information appropriation. This calculation includes finding the k closest information focuses in the preparation set to the information point for which an objective worth is inaccessible and doling out the normal estimation of the discovered information focuses to it

DECISION TREE

Decision tree is an of administered learning calculation. This strategy is for the most part utilized in characterization issues. It performs easily with consistent and absolute credits. This calculation partitions the populace into at least two comparable sets dependent on the main indicators. Decision Tree calculation, first figures the entropy of every single property. At that point the dataset is part with the assistance of the factors or indicators with greatest data gain or least entropy. These two stages are performed recursively with the leftover ascribes.

RANDOM FOREST

Random Forest is additionally a prevalently directed AI calculation. This strategy can be utilized for both relapse and grouping undertakings however by and large performs better in arrangement assignments. As the name recommends, Random Forest procedure considers numerous decision trees prior to giving a yield. Thus, it is essentially a troupe of decision trees. This procedure depends on the conviction that more number of trees would unite to the correct

decision. For characterization, it utilizes a democratic framework and afterward chooses the class though in relapse

V

V

V

V

V



between all models, while the exhibition of the Naïve Bayes has shown the least precision in this investigation.

FIG 2 HEART DISEASE PREDICTION APPLICATION

The best two model in our investigation are known as decision tree and Random Forest. Each model altogether improved the exhibitions in past work and shown the agreeable upgrade, which is more prominent than 85%. In correlation with the past work delineated in the third segment of Table 1, the exploration applied the trial utilizing five calculations [7]. That exploration utilized the equivalent dataset (UCI) with highlight determination approach. The precision of our model has improved the presentation of the classifiers. For instance, Naïve Bayes precision expanded 3%, Logistic Regression and SVM upgraded 5%, Decision Tree improved the exactness proportion 11%, and finally the Random Forest AI classifier expanded 8%.

Nonetheless, in past work, the investigation likewise utilized a similar stage, which is Rapid Miner. However, precision execution enlarged in our work may be a result of utilizing 10-overlap cross approval, while the past work utilized 5-crease cross approval approach. More emphasis during the learning stage can assist with creating more precise outcomes. Another conceivable explanation for the positive advancement in the precision is the size of the dataset, which has been enhanced in this examination as talked about in information outline area. It features that the huge size of the dataset can make positive effect on classifier precision as it upgrades the learning cycle.

V CONCLUSION

With the expanding number of passing's because of heart sicknesses, it has gotten compulsory to build up a framework to foresee heart infections adequately and precisely. The inspiration for the investigation was to locate the most productive ML calculation for identification of heart sicknesses. This examination thinks about the precision score of Decision Tree, Logistic Regression, Random Forest and Naive Bayes calculations for anticipating coronary illness utilizing UCI AI store dataset.

calculation with precision score of 88.52% for forecast of coronary illness. In future the work can be upgraded by building up a web application dependent on the Random Forest calculation just as utilizing a bigger dataset when contrasted with the one utilized in this examination which will assist with giving better outcomes and help wellbeing experts in foreseeing the coronary illness adequately and effectively.

REFERENCES:

- [1] Senthilkumar Mohan, Chandrasegar Thirumalai, Gautam Srivastava —Effective Heart Disease Prediction Using Hybrid Machine Learning Techniques|, Digital Object Identifier 10.1109/ACCESS.2019.2923707, IEEE Access, VOLUME 7, 2019 S.P. Bingulac, —On the Compatibility of Adaptive Controllers,| Proc. Fourth Ann. Allerton Conf. Circuits and Systems Theory, pp. 8-16, 1994. (Conference proceedings)
- [2] Sonam Nikhar, A.M. Karandikar” Prediction of Heart Disease Using Machine Learning Algorithms” International Journal of Advanced Engineering, Management and Science (IJAEMS) Infogain Publication,[Vol-2, Issue-6, June- 2016].I.S. Jacobs and C.P. Bean, “Fine particles, thin films and exchange anisotropy,” in Magnetism, vol. III, G.T. Rado and H. Suhl, Eds. New York: Academic, 1963, pp. 271-350.
- [3] Aditi Gavhane, Gouthami Kokkula, Isha Pandya, Prof. Kailas Devadkar (PhD),” Prediction of Heart Disease Using Machine Learning”, Proceedings of the 2nd International conference on Electronics, Communication and Aerospace Technology (ICECA 2018).IEEE Conference Record # 42487; IEEE Xplore ISBN:978-1- 5386-0965-1
- [4] Abhay Kishore¹, Ajay Kumar², Karan Singh³, Maninder Punia⁴, Yogita Hambir⁵,” Heart Attack Prediction Using Deep Learning”, International Research Journal of Engineering and Technology (IRJET), Volume: 05 Issue: 04 | Apr-2018.
- [5] A.Lakshmanarao, Y.Swathi, P.Sri Sai Sundareswar,” Machine Learning Techniques For Heart Disease Prediction”, International Journal Of Scientific & Technology Research Volume 8, Issue 11, November 2019.
- [6] Mr.Santhana Krishnan.J, Dr.Geetha.S,” Prediction of Heart Disease Using Machine Learning Algorithms”,2019 1st International Conference on Innovations in Information and Communication Technology(ICIICT),doi:10.1109/ICIICT1.2019.8741465.
- [7] Avinash Golande, Pavan Kumar T,” Heart Disease Prediction Using Effective Machine Learning Techniques”, International Journal of Recent Technology and Engineering (IJRTE) ISSN: 2277-3878, Volume-8, Issue-1S4, June 2019.
- [8] V.V. Ramalingam, Ayantan Dandapath, M Karthik Raja,” Heart disease prediction using machine learning techniques: a survey”, International Journal of Engineering & Technology, 7 (2.8) (2018) 684-687.
- [9] V. Manikantan and S. Latha, “Predicting the analysis of heart disease symptoms using

medicinal data mining methods”, International Journal of Advanced Computer Theory and Engineering, vol. 2, pp.46-51, 2013.

[10] M. S. Amin, Y. K. Chiam, K. D. Varathan, “Identification of significant features and data mining techniques in predicting heart disease,” Telematics Inform., vol. 36, pp. 82–93, Mar.2019.

[11] S. M. S. Shah, S. Batool, I. Khan, M. U. Ashraf, S. H. Abbas, and S. A. Hussain, “Feature extraction through parallel probabilistic principal component analysis for heart disease diagnosis,” Phys. A, Stat. Mech. Appl., vol. 482, pp. 796–807, 2017. doi:10.1016/j.physa.2017.04.113.

[12] Stephen F. Weng, Jenna Reips, Joe Kai1, Jonathan M. Garibaldi, Nadeem Qureshi, —Can machine-learning improve cardiovascular risk prediction using routine clinical data?¶, PLOS ONE | <https://doi.org/10.1371/journal.pone.0174944> April 4, 2017.

[13] N. Al-milli, __Backpropagation neural network for prediction of heart disease, “J. Theor. Appl. Inf. Technol., vol. 56, no. 1, pp.131–135, 2013.

[14] A. S. Abdullah and R. R. Rajalaxmi, “A data mining model for predicting the coronary heart

C. Source

Code HREAT DISEASE

PREDICTION #!/usr/bin/env

python

coding: utf-8

HEART DISEASE ANALYSIS USING MEACHINE LEARNING ALGORITHM'S

INTRODUCTION

We have a data which classified if patients have heart disease or not according to features in it. We will try to use this data to create a model which tries predict if a patient has this disease or not. We will use logistic regression (classification) algorithm.

Create data dictionary

1.age - age in years

2.sex - (1 = male; 0 = female)

3.cp - chest pain type

\$\\;\\;\\;\\;\$3.1: Typical angina: chest pain related decrease blood supply to the heart

\$\\;\\;\\;\\;\$3.2: Atypical angina: chest pain not related to heart

\$\\;\\;\\;\\;\$3.3: Non-anginal pain: typically esophageal spasms (non heart related)

\$\\;\\;\\;\\;\$3.4: Asymptomatic: chest pain not showing signs of disease

4.trestbps - resting blood pressure (in mm Hg on admission to the hospital) anything above 130-140 is typically cause for concern

5.chol - serum cholesterol in mg/dl

\$\\;\\;\\;\\;\$5.1:serum = LDL + HDL + .2 *

triglycerides
 # \$\\;\\;\\;\\;\$5.2:above 200 is cause for

concern

6.fbs - (fasting blood sugar > 120 mg/dl) (1 = true; 0 = false)

\$\\;\\;\\;\\;\$6.1'>126' mg/dL signals diabetes

7.restecg - resting electrocardiographic

results
 # \$\\;\\;\\;\\;\$7.1: Nothing to note

\$\\;\\;\\;\\;\$7.2: ST-T Wave abnormality

\$\\;\\;\\;\\;\$can range from mild symptoms to severe

problems
 # \$\\;\\;\\;\\;\$signals non-normal heart beat

\$\\;\\;\\;\\;\$7.3: Possible or definite left ventricular

hypertrophy
 # \$\\;\\;\\;\\;\$Enlarged heart's main pumping

chamber

8.thalach - maximum heart rate achieved

9.exang - exercise induced angina (1 = yes; 0 = no)

10.oldpeak - ST depression induced by exercise relative to rest looks at stress
of heart during exercise unhealthy heart

\$\\;\\;\\;\\;\$will stress more

11.slope - the slope of the peak exercise ST segment

\$\\;\\;\\;\\;\$11.1: Upsloping: better heart rate with exercise

(uncommon)
 # \$\\;\\;\\;\\;\$11.2: Flatsloping: minimal change (typical

healthy heart)

\$\\;\\;\\;\\;\$11.3: Downsloping: signs of unhealthy heart

12.ca - number of major vessels (0-3) colored by flourosopy

\$\\;\\;\\;\\;\$colored vessel means the doctor can see the blood passing

through
 # \$\\;\\;\\;\\;\$the more blood movement the better (no clots)

13.thal - thalium stress result

\$\\;\\;\\;\\;\$1,3: normal

\$\\;\\;\\;\\;\\;\$: fixed defect: used to be defect but ok now

\$\\;\\;\\;\\;\\;\$7: reversable defect: no proper blood movement when

exercising
 # 14.target - have disease or not (1=yes, 0=no) (= the predicted attribute)

Importing Libraries

In[1]:

import numpy as np

import pandas as pd

import matplotlib.pyplot as

plt import seaborn as sns

from sklearn.linear_model import

LogisticRegression from sklearn.model_selection

import train_test_split

Input data files are available in the "../input/" directory.

Any results you write to the current directory are saved as output.

Read Data

ln[2]:

```
# We are reading our data  
df = pd.read_csv("heart.csv")
```

```
# In[3]:
```

```
# First 5 rows of our data  
df.head()
```

```
# In[4]:
```

```
print(df.info())
```

```
# ## Data Exploration
```

```
# In[5]:
```

```
df.target.value_counts()
```

```
# In[6]:
```

```
pd.set_option("display.float",  
"{:.2f}".format) df.describe()
```

```
# In[7]:
```

```
df.target.value_counts().plot(kind="bar", color=["salmon", "lightblue"])
```

```
# In[8]:
```

```
countNoDisease = len(df[df.target == 0])
```

```
countHaveDisease = len(df[df.target == 1])
```

```
print("Percentage of Patients Haven't Heart Disease:  
{:.2f}%".format((countNoDisease / (len(df.target))*100)))
```

```
print("Percentage of Patients Have Heart Disease:  
{:.2f}%".format((countHaveDisease / (len(df.target))*100)))
```

```
# In[9]:
```

```
sns.countplot(x='sex', data=df,  
palette="mako_r") plt.xlabel("Sex (0 = female, 1=  
male)")  
plt.show()
```

```
# In[10]:
```

```
countFemale = len(df[df.sex == 0])  
countMale = len(df[df.sex == 1])  
print("Percentage of Female Patients: {:.2f}%".format((countFemale  
/ (len(df.sex))*100)))  
print("Percentage of Male Patients: {:.2f}%".format((countMale / (len(df.sex))*100)))
```

```
# In[11]:
```

```
df.groupby('target').mean()
```

```
# In[12]:
```

```
pd.crosstab(df.age,df.target).plot(kind="bar",figsize=(20,6))
```

```
plt.title('Heart Disease Frequency for  
Ages') plt.xlabel('Age')  
plt.ylabel('Frequency')  
plt.savefig('heartDiseaseAndAges.png')  
plt.show()
```

```
# In[13]:
```

```
categorical_val = []
```

```
continous_val = []
```

```
for column in df.columns:
```

```
print('=====')  
print('=====')
```

```
    print(f"{column} :
```

```
    {df[column].unique()}") if
```

```
    len(df[column].unique()) <= 10:
```

```
        categorical_val.append(column)
```

```
    else:
```

```
        continous_val.append(column)
```

```
# In[14]:
```

```
plt.figure(figsize=(15, 15))
```

```
for i, column in enumerate(categorical_val, 1):
```

```
    plt.subplot(3, 3, i)
```

```
    df[df["target"] == 0][column].hist(bins=35, color='blue', label='Have Heart Disease  
= NO', alpha=0.8)
```

```
    df[df["target"] == 1][column].hist(bins=35, color='red', label='Have Heart Disease  
= YES', alpha=0.8)
```

```
    plt.legend()
```

```
    plt.xlabel(column
```

```
)
```

```
# In[15]:
```

```
plt.figure(figsize=(15, 15))
```

```
for i, column in enumerate(continous_val, 1):
```

```
    plt.subplot(3,3 , i)
```

```
    df[df["target"] == 0][column].hist(bins=35, color='blue', label='Have Heart Disease  
= NO', alpha=0.8)
```

```
    df[df["target"] == 1][column].hist(bins=35, color='red', label='Have Heart Disease  
= YES', alpha=0.8)
```

```
    plt.legend()
```

```
    plt.xlabel(column
```

)

```
# In[16]:
```

```
pd.crosstab(df.sex,df.target).plot(kind="bar",figsize=(15,6),color=['#1CA53B','#AA  
1 111' ])
```

```
plt.title('Heart Disease Frequency for
```

```
Sex') plt.xlabel('Sex (0 = Female, 1 =
```

```
Male') plt.xticks(rotation=0)
```

```
plt.legend(["Haven't Disease", "Have
```

```
Disease"]) plt.ylabel('Frequency')
```

```
plt.show()
```

```
# In[17]:
```

```
plt.scatter(x=df.age[df.target==1], y=df.thalach[(df.target==1)],
```

```
c="red") plt.scatter(x=df.age[df.target==0],
```

```
y=df.thalach[(df.target==0)]) plt.legend(["Disease", "Not Disease"])
```

```
plt.xlabel("Age")
```

```
plt.ylabel("Maximum Heart
```

```
Rate") plt.show()
```

ln[18]:

```
pd.crosstab(df.slope,df.target).plot(kind="bar",figsize=(15,6),color=['#DAF7A6','#F  
F 5733' ])
```

```
plt.title('Heart Disease Frequency for Slope')
```

```
plt.xlabel('The Slope of The Peak Exercise ST Segment ')
```

```
plt.xticks(rotation = 0)
```

```
plt.ylabel('Frequency')
```

```
plt.show()
```

```
# In[19]:
```

```
pd.crosstab(df.fbs,df.target).plot(kind="bar",figsize=(15,6),color=['#FFC300','#581  
8 45' ])
```

```
plt.title('Heart Disease Frequency According To FBS')
```

```
plt.xlabel('FBS - (Fasting Blood Sugar > 120 mg/dl) (1 = true; 0 = false)')
```

```
plt.xticks(rotation = 0)
```

```
plt.legend(["Haven't Disease", "Have
```

```
Disease"]) plt.ylabel('Frequency of Disease or
```

```
Not') plt.show()
```

```
# In[20]:
```

```

pd.crosstab(df.cp,df.target).plot(kind="bar",figsize=(15,6),color=['#11A5AA','#AA1
1 90' ])

plt.title('Heart Disease Frequency According To Chest Pain Type')

plt.xlabel('Chest Pain Type')

plt.xticks(rotation = 0)

plt.ylabel('Frequency of Disease or
Not') plt.show()

```

Creating Dummy Variables

Since 'cp', 'thal' and 'slope' are categorical variables we'll turn them into dummy variables.

In[21]:

```

a = pd.get_dummies(df['cp'], prefix = "cp")
b = pd.get_dummies(df['thal'], prefix = "thal")
c = pd.get_dummies(df['slope'], prefix = "slope")

```

In[22]:

```
frames = [df, a, b, c]
```

```
df = pd.concat(frames, axis = 1)
```

```
df.head()
```

```
# In[23]:
```

```
df = df.drop(columns = ['cp', 'thal', 'slope'])
```

```
df.head()
```

```
# ## Creating Model for Logistic
```

```
Regression # <br>
```

We can use sklearn library or we can write functions ourselves. Let's them both.
Firstly we will write our functions after that we'll use sklearn library to calculate score.

```
# In[24]:
```

```
y = df.target.values
```

```
x_data = df.drop(['target'], axis = 1)
```

Normalize Data


```
#
```

```
# In[25]:
```

```
# Normalize
```

```
x = (x_data - np.min(x_data)) / (np.max(x_data) - np.min(x_data)).values
```

```
# We will split our data. 80% of our data will be train data and 20% of it will be  
test data.
```

```
# In[26]:
```

```
x_train, x_test, y_train, y_test = train_test_split(x,y,test_size = 0.2,random_state=0)
```

```
# In[27]:
```

```
#transpose
```

matrices x_{train} =

$x_{\text{train}}.T y_{\text{train}}$ =

$y_{\text{train}}.T$

```
x_test =
```

```
x_test.T y_test
```

```
= y_test.T
```

```
# Let's say weight = 0.01 and bias = 0.0
```

```
# In[28]:
```

```
#initialize
```

```
def initialize(dimension):
```

```
    weight = np.full((dimension,1),0.01)
```

```
    bias = 0.0
```

```
    return weight,bias
```

```
# ### Sigmoid
```

```
Function # <br>
```

```
# 
```

```
# In[29]:
```

```
def sigmoid(z):
```

```
y_head = 1/(1+ np.exp(-z))
```

```
return y_head
```

```
# By the way in formulas;
```

```
<br> # *  $h_0(x^i) = y_{head}$ 
```

```
# *  $y^i =$ 
```

```
y_train # *  $x^i$ 
```

```
= x_train
```

```
# In[30]:
```

```
def
```

```
forwardBackward(weight,bias,x_train,y_train):
```

```
# Forward
```

```
y_head = sigmoid(np.dot(weight.T,x_train) + bias)
```

```
loss = -(y_train*np.log(y_head) + (1-y_train)*np.log(1-y_head))
```

```
cost = np.sum(loss) / x_train.shape[1]
```

```
# Backward
```

```
derivative_weight = np.dot(x_train,((y_head-y_train).T))/x_train.shape[1]
```

```
derivative_bias = np.sum(y_head-y_train)/x_train.shape[1]

gradients = {"Derivative Weight" : derivative_weight, "Derivative Bias" :
derivative_bias}
```

```
return cost,gradients
```

```
# In[31]:
```

```
def update(weight,bias,x_train,y_train,learningRate,iteration) :
```

```
    costList = []
```

```
    index = []
```

```
    #for each iteration, update weight and bias
```

```
    values for i in range(iteration):
```

```
        cost,gradients = forwardBackward(weight,bias,x_train,y_train)
```

```
        weight = weight - learningRate * gradients["Derivative
```

```
Weight"] bias = bias - learningRate * gradients["Derivative
```

```
Bias"]
```

```
        costList.append(cost
```

```
        ) index.append(i)
```

```
    parameters = {"weight": weight,"bias": bias}
```

```
    print("iteration:",iteration)
```

```
print("cost:",cost)
```

```
plt.plot(index,costList)
```

```
plt.xlabel("Number of  
Iteration") plt.ylabel("Cost")  
plt.show()
```

```
return parameters, gradients
```

```
# In[32]:
```

```
def predict(weight,bias,x_test):  
    z = np.dot(weight.T,x_test) +  
    bias y_head = sigmoid(z)  
  
    y_prediction = np.zeros((1,x_test.shape[1]))  
  
    for i in range(y_head.shape[1]):  
        if y_head[0,i] <= 0.5:  
            y_prediction[0,i] =  
0 else:  
            y_prediction[0,i] =  
1 return y_prediction
```

In[33]:

```
def logistic_regression(x_train,y_train,x_test,y_test,learningRate,iteration):

    dimension = x_train.shape[0]

    weight,bias = initialize(dimension)


    parameters,                gradients
    = update(weight,bias,x_train,y_train,learningRate,iteration)


    y_prediction = predict(parameters["weight"],parameters["bias"],x_test)


    print("Manuel Test Accuracy: {:.2f}%".format((100 - np.mean(np.abs(y_prediction
- y_test))*100)))
```

In[34]:

```
logistic_regression(x_train,y_train,x_test,y_test,1,100)
```

```
# ## Manuel Test Accuracy for Logistic Regression is
<font color="red">86.89%</font></font><br>
```

```
# Let's find out sklearn's score.
```

```
# ### Sklearn Logistic Regression
```

```
# In[35]:
```

```
accuracies = {}
```

```
lr = LogisticRegression()
```

```
lr.fit(x_train.T,y_train.T)
```

```
acc = lr.score(x_test.T,y_test.T)*100
```

```
accuracies['Logistic Regression'] = acc
```

```
print("Test Accuracy {:.2f}%".format(acc))
```

```
# ## Test Accuracy for Logistic Regression <font color="red">**86.89%**</font>  
accuracy.</font>
```

```
# ## K-Nearest Neighbour (KNN)
```

```
Classification # <br>
```

```
# Let's see what will be score if we use KNN algorithm.
```

```
# In[36]:
```

```
# KNN Model
```

```
from sklearn.neighbors import KNeighborsClassifier
```

```
knn = KNeighborsClassifier(n_neighbors = 2) # n_neighbors means k
knn.fit(x_train.T, y_train.T)
prediction = knn.predict(x_test.T)

print("{} NN Score: {:.2f}%".format(2, knn.score(x_test.T, y_test.T)*100))
```

```
# In[47]:
```

```
# try to find best k value
```

```
scoreList = []
```

```
for i in range(1,20):
```

```
    knn2 = KNeighborsClassifier(n_neighbors = i) # n_neighbors means k
```

```
    knn2.fit(x_train.T, y_train.T)
```

```
    scoreList.append(knn2.score(x_test.T, y_test.T))
```

```
plt.plot(range(1,20), scoreList)
```

```
plt.xticks(np.arange(1,20,1))
```

```
plt.xlabel("K value")
```

```
plt.ylabel("Score")
```

```
plt.show()
```

```
acc =  
max(scoreList)*100  
accuracies['KNN'] = acc
```

```
print("Maximum KNN ACCURACY is {:.2f}%".format(acc))
```

```
# As you can see above if we define k as 3-7-8 we will reach maximum score.
```

```
<br> # ## KNN Model's Accuracy is <font color="red">88.52%</font></font>
```

```
# ## Support Vector Machine (SVM) Algorithm
```

```
# Now we will use SVM algorithm.
```

```
# In[38]:
```

```
from sklearn.svm import SVC
```

```
# In[39]:
```

```
svm = SVC(random_state = 1)
```

```
svm.fit(x_train.T, y_train.T)
```

```
acc = svm.score(x_test.T, y_test.T)*100
```

```
accuracies['SVM'] = acc
```

```
print("Test Accuracy of SVM Algorithm: {:.2f}%".format(acc))
```

```
# ## Test Accuracy of SVM Algorithm is <font color="red"> 86.89%</font></font>
```

```
# ## Naive Bayes Algorithm
```

```
# **Naive Bayes Algorithm**
```

```
# <br>
```

```
# 
```

```
# In[40]:
```

```
from sklearn.naive_bayes import
```

```
GaussianNB nb = GaussianNB()
```

```
nb.fit(x_train.T, y_train.T)
```

```
acc = nb.score(x_test.T, y_test.T)*100
```

```
accuracies['Naive Bayes'] = acc
```

```
print("Accuracy of Naive Bayes: {:.2f}%".format(acc))
```

```
# ## Accuracy of Naive Bayes: <font color="red">86.89%</font></font>
```

Decision Tree Algorithm

```
# In[41]:
```

```
from sklearn.tree import DecisionTreeClassifier
```

```
dtc = DecisionTreeClassifier()
```

```
dtc.fit(x_train.T, y_train.T)
```

```
acc = dtc.score(x_test.T, y_test.T)*100
```

```
accuracies['Decision Tree'] = acc
```

```
print("Decision Tree Test Accuracy {:.2f}%".format(acc))
```

```
#    ##    Test    Accuracy    of    Decision    Tree    Algorithm:  
<font color="red">78.69%</font></font>
```

```
# ## Random Forest Classification
```

```
# In[42]:
```

```
# Random Forest Classification
```

```
from sklearn.ensemble import RandomForestClassifier
```

```
rf = RandomForestClassifier(n_estimators = 1000, random_state = 1)
```

```
rf.fit(x_train.T, y_train.T)
```

```
acc = rf.score(x_test.T,y_test.T)*100
```

```
accuracies['Random Forest'] = acc
```

```
print("Random Forest Algorithm Accuracy Score : {:.2f}%".format(acc))
```

```
# ## Test Accuracy of Random Forest: <font color="red">88.52%</font></font>
```

```
# ## Comparing Models
```

```
# In[43]:
```

```
colors = ["purple", "green", "orange", "magenta", "#CFC60E", "#0FBBAE"]
```

```
sns.set_style("whitegrid")
```

```
plt.figure(figsize=(16,5))
```

```
plt.yticks(np.arange(0,100,10))
```

```
plt.ylabel("Accuracy %")
```

```
plt.xlabel("Algorithms")
```

```
sns.barplot(x=list(accuracies.keys()), y=list(accuracies.values()), palette=colors)
```

```
plt.show()
```

```
# Our models work fine but best of them are KNN and Random Forest with  
88.52% of accuracy. Let's look their confusion matrixes.
```

Confusion Matrix

```
# In[44]:
```

```
# Predicted values
```

```
y_head_lr = lr.predict(x_test.T)
```

```
knn3 = KNeighborsClassifier(n_neighbors = 3)
```

```
knn3.fit(x_train.T, y_train.T)
```

```
y_head_knn = knn3.predict(x_test.T)
```

```
y_head_svm = svm.predict(x_test.T)
```

```
y_head_nb = nb.predict(x_test.T)
```

```
y_head_dtc = dtc.predict(x_test.T)
```

```
y_head_rf = rf.predict(x_test.T)
```

```
# In[45]:
```

```
from sklearn.metrics import confusion_matrix
```

```
cm_lr = confusion_matrix(y_test,y_head_lr)
```

```
cm_knn = confusion_matrix(y_test,y_head_knn)
```

```
cm_svm = confusion_matrix(y_test,y_head_svm)
```

```
cm_nb = confusion_matrix(y_test,y_head_nb)
```

```
cm_dtc = confusion_matrix(y_test,y_head_dtc)
```

```
cm_rf = confusion_matrix(y_test,y_head_rf)
```

```
# In[46]:
```

```
plt.figure(figsize=(24,12))
```

```
plt.suptitle("Confusion Matrixes",fontsize=24)
```

```
plt.subplots_adjust(wspace = 0.4, hspace= 0.4)
```

```
plt.subplot(2,3,1)
```

```
plt.title("Logistic Regression Confusion Matrix")
```

```
sns.heatmap(cm_lr,annot=True,cmap="Blues",fmt="d",cbar=False,  
annot_kws={"size": 24})
```

```
plt.subplot(2,3,2)
```

```
plt.title("K Nearest Neighbors Confusion Matrix")
```

```
sns.heatmap(cm_knn,annot=True,cmap="Blues",fmt="d",cbar=False,  
annot_kws={"size": 24})
```

```
plt.subplot(2,3,3)
```

```
plt.title("Support Vector Machine Confusion Matrix")
```

```
sns.heatmap(cm_svm,annot=True,cmap="Blues",fmt="d",cbar=False,  
annot_kws={"size": 24})
```

```
plt.subplot(2,3,4)
```

```
plt.title("Naive Bayes Confusion Matrix")
```

```
sns.heatmap(cm_nb,annot=True,cmap="Blues",fmt="d",cbar=False,  
annot_kws={"size": 24})
```

```
plt.subplot(2,3,5)
```

```
plt.title("Decision Tree Classifier Confusion Matrix")
```

```
sns.heatmap(cm_dtc,annot=True,cmap="Blues",fmt="d",cbar=False,  
annot_kws={"size": 24})
```

```
plt.subplot(2,3,6)
```

```
plt.title("Random Forest Confusion Matrix")
```

```
sns.heatmap(cm_rf,annot=True,cmap="Blues",fmt="d",cbar=False,  
annot_kws={"size": 24})
```

```
plt.show()
```

Heart disease Application

```
import streamlit as
```

```
st import pandas as
```

```
pd
```

```
from sklearn.ensemble import
```

```
RandomForestClassifier st.write(""" # Heart Disease
```

```
Prediction App
```

```
This app predicts that *the person has heart disease or not*
```

```
""")
```

```
def user_input_features():
```

```
    st.write("""**1. Select Age :**""")
```

```
    age = st.slider('', 0, 100, 25)
```

```
    st.write("""**You selected this option **""",age)
```

```
    st.write("""**2. Select Gender :**""")
```

```
    sex = st.selectbox("(1=Male,
```

```
0=Female)",["1","0"]) st.write("""**You selected
```

```
this option **""",sex)
```

```
    st.write("""**3. Select Chest Pain Type :**""")
```

```
    cp = st.selectbox("(1 = Typical Angina, 2 = Atypical Angina, 3 = Non—anginal  
Pain, 4 = Asymptotic) : ",["1","2","3","4"])
```

```
    st.write("""**You selected this option **""",cp)
```

```
st.write("""**4. Select Resting Blood Pressure  
:  
**""") trestbps = st.slider('In mm/Hg unit', 0, 200,  
110) st.write("""**You selected this option  
**""",trestbps)
```

```
st.write("""5. Select Serum Cholesterol

:****") chol = st.slider('In mg/dl unit', 0, 600,

115) st.write("""You selected this option

****",chol)
```

```
st.write("""6. Maximum Heart Rate Achieved (THALACH) :****")

thalach = st.slider('', 0, 220, 115)

st.write("""You selected this option ****",thalach)
```

```
st.write("""7. Exercise Induced Angina (Pain in chest while exersice) :****")

exang = st.selectbox("(1=Yes, 0=No)",["1","0"])

st.write("""You selected this option ****",exang)
```

```
st.write("""8. Oldpeak (ST depression induced by exercise relative to rest) :****")

oldpeak = float(st.slider('', 0.0, 10.0, 2.0))

st.write("""You selected this option ****",oldpeak)
```

```
st.write("""9. Slope (The slope of the peak exercise ST segment) :****")

slope = st.selectbox("(Select 0, 1 or 2)",["0","1","2"])

st.write("""You selected this option ****",slope)
```

```
st.write("""10. CA (Number of major vessels (0-3) colored by flourosopy)

:****") ca = st.selectbox("(Select 0, 1, 2 or 3)",["0","1","2","3"])
```

```
st.write("""**You selected this option **""",ca)
```

```

st.write("""**11. Thal :**""")

thal = float(st.slider('3 = normal; 6 = fixed defect; 7 = reversable defect', 0.0,
10.0, 3.0))

st.write("""**You selected this option **""",thal)


data = {'age': age, 'sex': sex, 'cp': cp, 'trestbps': trestbps, 'chol': chol, 'thalach':
thalach, 'exang': exang, 'oldpeak': oldpeak, 'slope': slope, 'ca': ca, 'thal': thal,}

features = pd.DataFrame(data, index=[0])

return features

df = user_input_features()

st.subheader('Given Inputs :

') st.write(df)


heart =

pd.read_csv("heart.csv") X =

heart.iloc[:,0:11].values

Y = heart.iloc[:,[11]].values


model = RandomForestClassifier()

model.fit(X, Y)


prediction = model.predict(df)

st.subheader('Prediction :')

```

```
df1=pd.DataFrame(prediction,columns=['0'])
```

```
df1.loc[df1['0'] == 0, 'Chances of Heart Disease'] = 'No'
```

```
df1.loc[df1['0'] == 1, 'Chances of Heart Disease'] = 'Yes'
```

```
st.write(df1)
```

```
prediction_proba =
```

```
model.predict_proba(df)
```

```
st.subheader('Prediction Probability in %
```

```
:') st.write(prediction_proba * 100)
```