**A study on**

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

**A Report Submitted**

**In partial fulfillment for the degree of**

## Bachelor of Technology

**In**

**Computer Science and Engineering Under**

**The Assam Royal Global University**



#### Submitted By-

###### Samrat Dey(192025023) Anwayajyoti Dey(192025004)

###### Thangjam Balkrishna Singh(192025030)

#### Under the guidance of Dr. ANUPAM DAS

#### Associate Professor

#### Department of Computer Science and Engineering Royal School of Engineering & Technology Guwahati-781035

#### Dec-2022



#### DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING ROYAL SCHOOL OF ENGINEERING & TECHNOLOGY

**CERTIFICATE OF APPROVAL**

It is certified that the work contained in the report entitled **"Detection of Deepfake"** by  **Anwayajyoti Dey** bearing Roll No 192025004 and **Samrat Dey** bearing Roll No 192025023 of **Bachelor of Technology, 8th Semester** under the **Computer Science Department**, **Royal School of Technology**, The Assam Royal Global University, Guwahati, Assam for the fulfillment of the degree of **Bachelor of Technology** has been carried out under my supervision and that work has not been submitted elsewhere for a degree.

###### Date: Project Guide:



**FORWARDING CERTIFICATE**

This is to certify that the project work entitled " HEART DISEASE PREDICTION USING MACHINE LEARNING ALGORITHMS" is hereby approved as a bonafide work of study as an engineering subject, carried out by the students – Samrat Dey(Roll No: 192025023), Anwayajyoti Dey(Roll No: 192025004) and Thangjam Balkrishna(Roll No: 192025030) of 7th Semester, B.Tech, Computer Science and Engineering Department under the guidance of Dr. Anupam Das, Assistant Professor, Computer Science and Engineering Department, The Assam Royal Global University. The work in the project is a genuine work carried out by the students as a prerequisite to the degree for which it has been submitted.

###### Date:

`

#### DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING ROYAL SCHOOL OF ENGINEERING & TECHNOLOGY

##### Declaration by the Candidate

We certify that this project entitled " HEART DISEASE PREDICTION USING MACHINE LEARNING ALGORITHMS", a perquisite towards partial fulfillment for the award of B.Tech degree in Computer Science and Engineering, Royal School of Engineering & Technology, Guwahati contains no materials previously published or written by another person, except where due reference has made in the text as in an accurate record of our work carried under the guidance and supervision of Dr. Anupam Das, Associate Professor, Department of Computer Science and Engineering.

Date-30-12-2022

|  |  |  |
| --- | --- | --- |
| Samrat Dey | Anwayajyoti Dey | Thangjam Balkrishna |
| (Roll No: 192025023) | (Roll No: 192025004) | (Roll No:192025030) |

**ACKNOWLEDGEMENT**

We would like to extend our gratitude and our sincere thanks to our honorable, esteemed guide, Dr. Anupam Das, Associate Professor, Computer Science and Engineering, Royal School of Engineering and Technology. He is not only a helpful teacher with deep vision but also most importantly a kind person. We sincerely thank for his exemplary guidance and encouragement. His trust and support inspired us in the most important moments of making rightdecisions and we are glad to work with him.

We would like to thank all our other faculty members for their guidance and ideas that helped us make this project successful. This project is by far the most significant accomplishment in our life and it would be impossible without these people who supported us and believed in us.

We would like to thank all our friends for all the thoughtful and mind stimulating discussions that we had, which made us think beyond the obvious. Last but not the least, we would like to thank our parents, who taught us the value of hard work.

**ABSTRACT**

Machine Learning is used across many ranges around the world. The health care industry is no exclusion. Machine Learning can play an essential role in predicting presence/absence of cardiovascular disorders, Heart diseases and more. Such information, if predicted well in advance, can provide important intuitions to doctors who can then adapt their diagnosis and dealing per patient basis. We work on predicting possible Heart Diseases in people using Machine Learning algorithms. In this project we perform SVM, Decision Tree, Random Forest, SGD Classifier, and Logistic Regression, among these Random Forest gives the better accuracy and predictive analysis.

**Keywords:** SVM; Decision Tree; Random Forest; SGD Classifier; Logistic Regression; Python Programming;

**Notations / Abbreviations**

1. ML - Machine Learning
2. SVM - Support Vector Machine
3. SGD - Stochastic Gradient Descent
4. SK Learn - Scikit Learn

III

|  |  |
| --- | --- |
| **Table of Contents** |  |
| Acknowledgement | I |
| Abstract | II |
| Notations/Abbreviations | III |
| **Chapter 1 INTRODUCTION** | **1** |
| 1.1 Motivation | 1 |
| 1.2 Aims and Objectives | 1 |
| 1.3 Problem Statement | 1 |
| 1.4 Project Description | 2 |
| 1.5 Dependencies Used | 2 |
| **Chapter 2 LITERATURE REVIEW** | **3-4** |
| **Chapter 3 METHODOLOGY** | **5** |
| 3.1 Existing System | 5 |
| 3.2 Proposed System | 5 |
| 3.2.1 Collecting of dataset | 5 |
| 3.2.2 Selection of attributes | 6 |
| 3.2.3 Pre processing of data | 7 |
| 3.2.4 Balancing of data | 7 |
| 3.2.5 Prediction of Disease | 8 |
| **Chapter 4 ALGORITHM** | **9** |
| 4.1 Logistic Regression Algorithm | 9 |
| 4.2 SVM | 10 |
| 4.3 Decision Tree Algorithm | 12 |
| 4.4 Random Forest Algorithm | 14 |
| 4.5 SGD Classifier | 16 |
| **Chapter 5 Experimental Analysis** | **17** |
| 5.1 System Configuration | 17 |
| 5.1.1 Hardware Requirements | 17 |
| 5.1.2 Software Requirements | 17 |
| 5.2 Sample Code | 17 |
| 5.3 Dataset Details | 20 |
| 5.4 Performance Analysis | 22 |

* 1. [Performance Measures 22](#_TOC_250002)
  2. [Input and Output 23](#_TOC_250001)
  3. [Result 25](#_TOC_250000)

Chapter 6 CONCLUSION 26

# LIST OF FIGURES.

|  |  |  |  |
| --- | --- | --- | --- |
| **S. No.** | **Figure No.** | **Description** | **Page No.** |
| 1. | 3.2.1 | Collection of data set | 6 |
| 2. | 3.2.3 | Data Pre-Processing | 7 |
| 3. | 4.1 | Linear Regression | 9 |
| 4. | 4.2 | Support Vector Machine | 11 |
| 5. | 5.3 | Dataset Details | 20 |
| 6. | 5.5 | Performance Measures | 22 |
| 7. | 5.6 | Input and Output | 23 |

**Chapter 1**

**INTRODUCTION**

* 1. **Motivation**

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

* 1. **Aims and Objectives**

The main objective of this project is to build a system that will help in detecting whether the person is having heart disease or not.

* 1. **PROBLEM STATEMENT**

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

* 1. **Project Description**

**Machine** Learning helps in predicting the Heart diseases, and the predictions made are quite accurate. A data set is formed by taking into consideration some of the information of 1000 individuals. The problem is based on the given information about each individual we have to calculate that whether that individual will suffer from heart disease.

* 1. **Dependencies Used**

The following dependencies have been chosen for the development of this project:

1. Numpy
2. Pandas
3. Matplotlib
4. Seaborn
5. Scikit Learn

**Chapter 2**

**LITERATURE REVIEW**

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

[1] Purushottam ,et ,al proposed a paper “Efficient Heart Disease Prediction System”using hill climbing and decision tree algorithms .They used Cleveland dataset and preprocessing of data is performed before

using classification algorithms. The Knowledge Extraction is done based on Evolutionary Learning (KEEL), an open source data mining tool that fills the missing values in the data set.A decision tree follows top- down order. For each actual node selected by hill-climbing algorithm a node is selected by a test at each level. The parameters and their values used are confidence. Its minimum confidence value is 0.25. The accuracy of the system is about 86.7

Lakshmana Rao et al,[2] Machine Learning Techniques for Heart Disease Prediction in which the contributing elements for heart disease are more (circulatory strain, diabetes, current smoker, high cholesterol, etc..).

Sonam Nikhar et al [3] has built up the paper titled as Prediction of Heart Disease Using Machine Learning Algorithms by This exploration plans to give a point by point portrayal of logistic regression classifier that are applied in our examination especially in the prediction of Heart Disease.

1. Aditi Gavhane et al proposed a paper “Prediction of Heart Disease Using Machine Learning”, in which training and testing of dataset is performed by using neural network algorithm multi-layer perceptron. In this algorithm there will be one input layer and one output layer and one or more layers are hidden layers between these two input and output layers. Through hidden layers each input node is connected to output layer. This connection is assigned with some random weights. The other input is called bias which is assigned with weight based on requirement the connection between the nodes can be feedforwarded or feedback.
2. Lakshmana Rao et al,proposed “Machine Learning Techniques for Heart Disease Prediction” in which the contributing elements for heart disease are more. So, it is difficult to distinguish heart disease.To find the seriousness of the heart disease among people different neural systems and data mining techniques are used.
3. Abhay Kishore et alproposed “Heart Attack Prediction Using Deep Learning” in which heart attack prediction system by using Deep learning techniques and to predict the probable aspects of heart related infections of the patient Recurrent Neural System is used. This model uses deep learning and data mining to give the best precise model and least blunders. This paper acts as strong reference model for another type of heart attack prediction models.

**Chapter 3**

**METHODOLOGY**

* 1. **EXISTING SYSTEM**

Heart disease is even being highlighted as a silent killer which leads to the death of a person without obvious symptoms. The nature of the disease is the cause of growing anxiety about the disease & its consequences. Hence continued efforts are being done to predict the possibility of this deadly disease in prior. So that various tools & techniques are regularly being experimented with to suit the present-day health needs. Machine Learning techniques can be a boon in this regard. Even though heart disease can occur in different forms, there is a common set of core risk factors that influence whether someone will ultimately be at risk for heart disease or not. By collecting the data from various sources, classifying them under suitable headings & finally analyzing to extract the desired data we can conclude. This technique can be very well adapted to the do the prediction of heart disease. As the well-known quote says “Prevention is better than cure”, early prediction & its control can be helpful to prevent & decrease the death rates due to heart disease.

* 1. **PROPOSED SYSTEM**

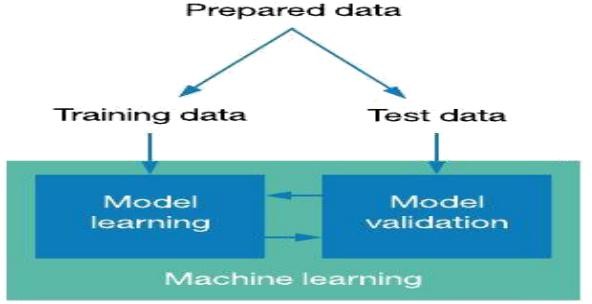
The working of the system starts with the collection of data and selecting the important attributes. Then the required data is pre-processed into the required format. The data is then divided into two parts training and testing data. The algorithms are applied and the model is trained using the training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

1.) Collection of Dataset 2.) Selection of attributes 3.) Data Pre-Processing 4.) Balancing of Data

5.) Disease Prediction

* + 1. **Collection of data set**

Initially, we collect a data set for our heart disease prediction system. After the collection of the dataset, we split the dataset into training data and testing data. The training dataset is used for prediction model learning and testing data is used for evaluating the prediction model. For this project, 80% of training data is used and 20% of data is used for testing

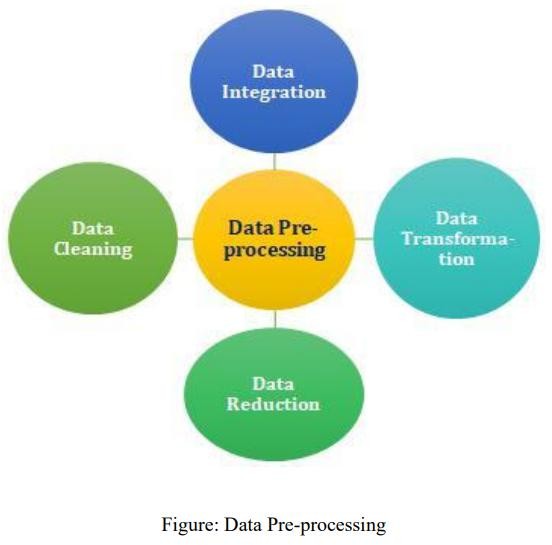


* + 1. **Selection of attributes**

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

* + 1. **Pre-Processing of Data**

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



* + 1. **Balancing of Data**

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

1. Under Sampling: In Under Sampling, dataset balance is done by the reduction of the size of the ample class. This process is considered when the amount of data is adequate.
2. Over Sampling: In Over Sampling, dataset balance is done by increasing the size of the scarce samples. This process is considered when the amount of data is inadequate.
   * 1. **Prediction of Disease**

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

**Chapter 4**

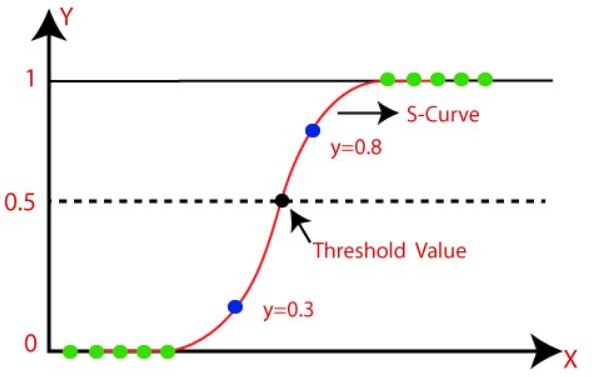
**ALGORITHM**

* 1. **Logistic Regression Algorithm**

Logistic regression is one of the most popular Machine Learning algorithms, which comes under the Supervised Learning technique. It is used for predicting the categorical dependent variable using a given set of independent variables. Logistic regression predicts the output of a categorical dependent variable. Therefore the outcome must be a categorical or discrete value. It can be either Yes or No, 0 or 1, true or False, etc. but instead of giving the exact value as 0 and 1, **it gives the probabilistic values which lie between 0 and 1**.

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

Logistic Regression can be used to classify the observations using different types of data and can easily determine the most effective variables used for the classification. The below image is showing the logistic function:



* 1. **SVM (Support Vector Machine)**

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

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

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

The followings are important concepts in SVM –

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

Hyperplane - As we can see in the above diagram, it is a decision plane or space which is divided between a set of objects having different classes. Margin - It may be defined as the gap between two lines on the closest data points of different classes.

It can be calculated as the perpendicular distance from the line to the 12 support vectors. Large margin is considered as a good margin and small margin is considered as a bad margin.

**Types of SVM:**

SVM can be of two types:

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

is called as Non-linear SVM classifier.

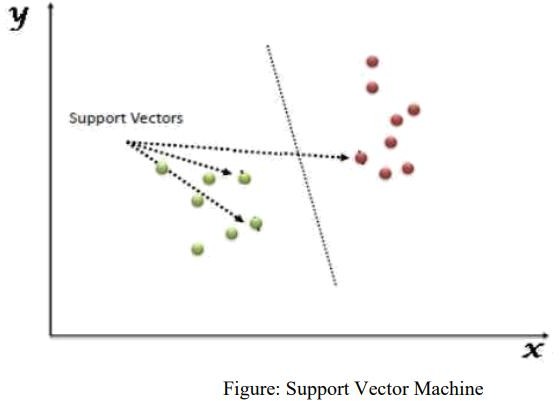
The objective of the support vector machine algorithm is to find a hyper-plane in an N dimensional space (N - the number of features) that distinctly classifies the data points.

###### The advantages of support vector machines are:

* Effective in high dimensional spaces.
* Still effective in cases where the number of dimensions is greater than the number of samples.
* Uses a subset of training points in the decision function (called support vectors), so it is also memory efficient.
* Versatile: different kernel functions can be specified for the decision function. Common kernels are provided, but it is also possible to specify custom kernels.

###### The disadvantages of support vector machines include:

* If the number of features is much greater than the number of samples, avoid over-fitting in choosing Kernel functions and regularization term is crucial. SVMs do not directly provide probability estimates, these are calculated using an expensive five-fold cross-validation.



* 1. **DECISION TREE ALGORITHM**

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

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

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

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

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

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

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

* + 1. **Information Gain:**

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

Entropy is the measure of uncertainty of a random variable, it characterizes the impurity of an

arbitrary collection of examples. The higher the entropy the more the information content.

* + 1. **Gini Index:**

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

The most notable types of Decision Tree algorithms are:-

* + - 1. **IDichotomiser 3 (ID3):** This algorithm uses Information Gain to decide which attribute is to be used to classify the current subset of the data. For each level of the tree, information gain is calculated for the remaining data recursively.
      2. **C4.5:** This algorithm is the successor of the ID3 algorithm. This algorithm uses either Information gain or Gain ratio to decide upon the classifying attribute. It is a direct improvement from the ID3 algorithm as it can handle both continuous and missing attribute values.

**Working:**

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

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

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

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

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

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

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

to improve the performance of the model.

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

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

**Assumptions:**

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

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

**Algorithm Steps:**

It works in four steps:

* Select random samples from a given dataset.
* Construct a Decision Tree for each sample and get a prediction result from each Decision Tree.
* Perform a vote for each predicted result.
* Select the prediction result with the most votes as the final prediction.

**Advantages:**

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

**Disadvantages:**

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

* 1. **SGD Classifier (Stochastic Gradient Descent)**

The word 'stochastic’ means a system or process linked with a random probability. Hence, in Stochastic Gradient Descent, a few samples are selected randomly instead of the whole data set for each iteration. In Gradient Descent, there is a term called "batch" which denotes the total number of samples from a dataset that is used for calculating the gradient for each iteration. In typical Gradient Descent optimization, like Batch Gradient Descent, the batch is taken to be the whole dataset. Although using the whole dataset la really useful for getting to the minima in a less noisy and less random manner, the problem arises when our dataset gets big. Suppose, you have a million samples in your dataset, so if you use a typical Gradient Descent optimization technique, you will have to use all of the one million samples for completing one Iteration while performing the Gradient Descent and it has to be done for every iteration until the minima are reached. Hence, it becomes computationally very expensive to perform. This problem is solved by Stochastic Gradient Descent. In SGD, it uses only a single sample, i.e., a batch size of one, to perform each iteration. The sample randomly shuffled and selected for performing the iteration.

So, in SGD, we find out the gradient of the cost function of a single example at each iteration instead of the sum of the gradient of the cost function of all the examples.

In SGD, since only one sample from the dataset is chosen at random for each iteration, the path taken by the algorithm to reach the minima is usually noisier than your typical Gradient Descent algorithm. But that doesn’t matter all that much because the path taken by the algorithm does not matter, as long as we reach the minima and with a significantly shorter training time.

**Chapter 5**

**EXPERIMENTAL ANALYSIS**

* 1. **SYSTEM CONFIGURATION**
     1. **Hardware requirements:**

Processor : Any Update Processor Ram : Min 4GB

Hard Disk : Min 100GB

* + 1. **Software requirements:**

Operating System : Windows family Technology : Python3.7

IDE : Jupiter notebook

* 1. **SAMPLE CODE**

### #Importing the Dependencies

import numpy as np import pandas as pd

import matplotlib.pyplot as plt import seaborn as sns

from sklearn.model\_selection import train\_test\_split from sklearn.linear\_model import LogisticRegression from sklearn import svm

from sklearn.linear\_model import SGDClassifier from sklearn import tree

from sklearn.ensemble import RandomForestClassifier from sklearn.metrics import accuracy\_score

### #Data Collection and Processing

heart\_data=pd.read\_csv("/content/Updated datasheet.csv")

### #print first 5 rows of the dataset

heart\_data.head()

### #Number of rows and coloumn in the dataset

heart\_data.shape

### #Getting some information about the data

heart\_data.info()

### # Checking for missing values

heart\_data.isnull().sum()

### # Statistical measures about the data

heart\_data.describe()

ax = sns.countplot(heart\_data["target"]) target\_temp = heart\_data.target.value\_counts() print(target\_temp)

### #Splitting the Features and Target

X = heart\_data.drop(columns='target',axis=1) Y = heart\_data['target']

### #Splitting into Training data and Test data

X\_train, X\_test, Y\_train, Y\_test = train\_test\_split(X, Y, test\_size=0.2, stratify=Y, random\_state=2)

### #Model Training

LR\_classifier = LogisticRegression(random\_state=0) clf = svm.SVC()

sgd=SGDClassifier()

forest=RandomForestClassifier(n\_estimators=20, random\_state=12,max\_depth=6)

treee = tree.DecisionTreeClassifier(criterion = 'entropy',random\_state=0,max\_depth = 6)

#Training the model with Training data LR\_classifier.fit(X\_train, Y\_train) clf.fit(X\_train, Y\_train)

sgd.fit(X\_train, Y\_train) treee.fit(X\_train, Y\_train)

forest.fit(X\_train, Y\_train)

### #Accuracy Score

# Accuracy on training data Y\_pred=LR\_classifier.predict(X\_train) Y\_predsvm=clf.predict(X\_train) Y\_predsgd=sgd.predict(X\_train) Y\_predtree=treee.predict(X\_train) Y\_predforest=forest.predict(X\_train)

# Accuracy on train data print(accuracy\_score(Y\_train, Y\_pred)) print(accuracy\_score(Y\_train, Y\_predsvm)) print(accuracy\_score(Y\_train, Y\_predsgd)) print(accuracy\_score(Y\_train, Y\_predtree)) print(accuracy\_score(Y\_train, Y\_predforest))

Y\_pred=LR\_classifier.predict(X\_test) Y\_predsvm=clf.predict(X\_test) Y\_predsgd=sgd.predict(X\_test) Y\_predtree=treee.predict(X\_test) Y\_predforest=forest.predict(X\_test)

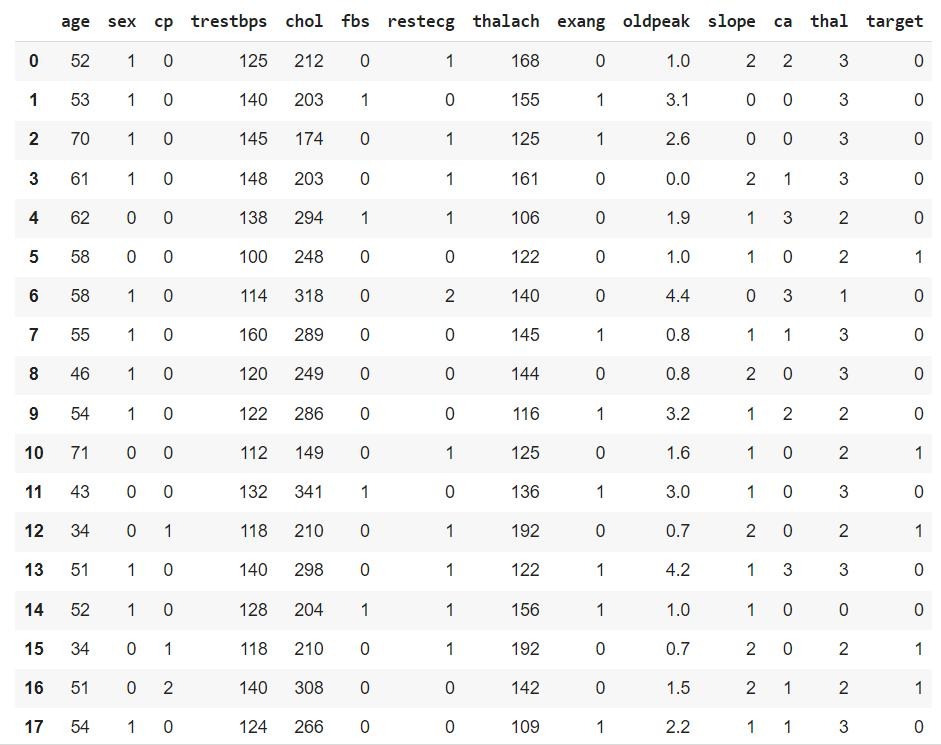
# Accuracy on test data print(accuracy\_score(Y\_test, Y\_pred)) print(accuracy\_score(Y\_test, Y\_predsvm)) print(accuracy\_score(Y\_test, Y\_predsgd)) print(accuracy\_score(Y\_test, Y\_predtree)) print(accuracy\_score(Y\_test, Y\_predforest))

import pickle

pickle.dump(forest, open('Random\_forest\_model.pkl', 'wb'))

filename = 'heart\_disease\_prediction.sav' pickle.dump(forest, open(filename, 'wb'))

* 1. **DATASET DETAILS**
* 14 attributes are considered for the prediction of the output.



**Input dataset attributes**

age: The person's age in years

sex: The person's sex (1 = male, 0 = female)

cp: The chest pain experienced (Value 1: typical angina, Value 2: atypical angina, Value

3: non-anginal pain, Value 4: asymptomatic) trestbps: The person's resting blood pressure (mm Hg on admission to the hospital) chol: The person's cholesterol measurement in mg/dl

fbs: The person's fasting blood sugar (> 120 mg/dl, 1 = true; 0 = false)

restecg: Resting electrocardiographic measurement (0 = normal, 1 = having ST-T wave abnormality,

2 = showing probable or definite left ventricular hypertrophy by Estes' criteria) thalach: The person's maximum heart rate achieved

exang: Exercise induced angina (1 = yes; 0 = no)

oldpeak: ST depression induced by exercise relative to rest ('ST' relates to positions on the ECG plot.

See more here)

slope: the slope of the peak exercise ST segment (Value 1: upsloping, Value 2: flat, Value 3:

downsloping)

ca: The number of major vessels (0-3)

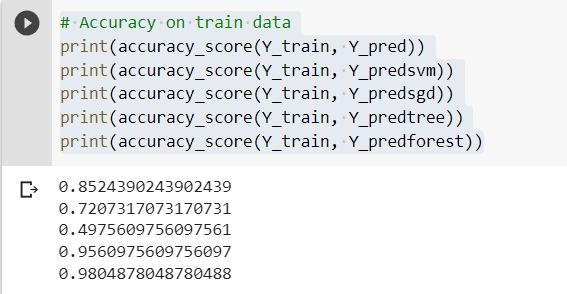
thal: A blood disorder called thalassemia (0 = normal; 1 = fixed defect; 2 = reversable defect) target: Heart disease (0 = no, 1 = yes)

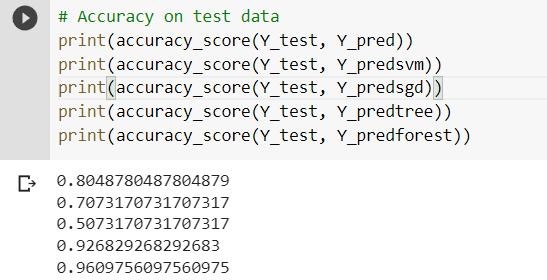
* 1. **PERFORMANCE ANALYSIS**

In this project, various machine learning algorithms like SVM, Decision Tree, Random Forest, Logistic Regression, SGD Classifier are used to predict heart disease. In Heart Disease dataset, 14 attributes are considered for the prediction of heart disease. Various attributes of the patient like gender, chest pain type, fasting blood pressure, serum cholesterol, exang, etc are considered for this project. The accuracy for individual algorithms has to measure and whichever algorithm is giving the best accuracy,that is considered for the heart disease prediction. Accuracy- Accuracy is the ratio of the number of correct predictions to the total number of inputs in the dataset.

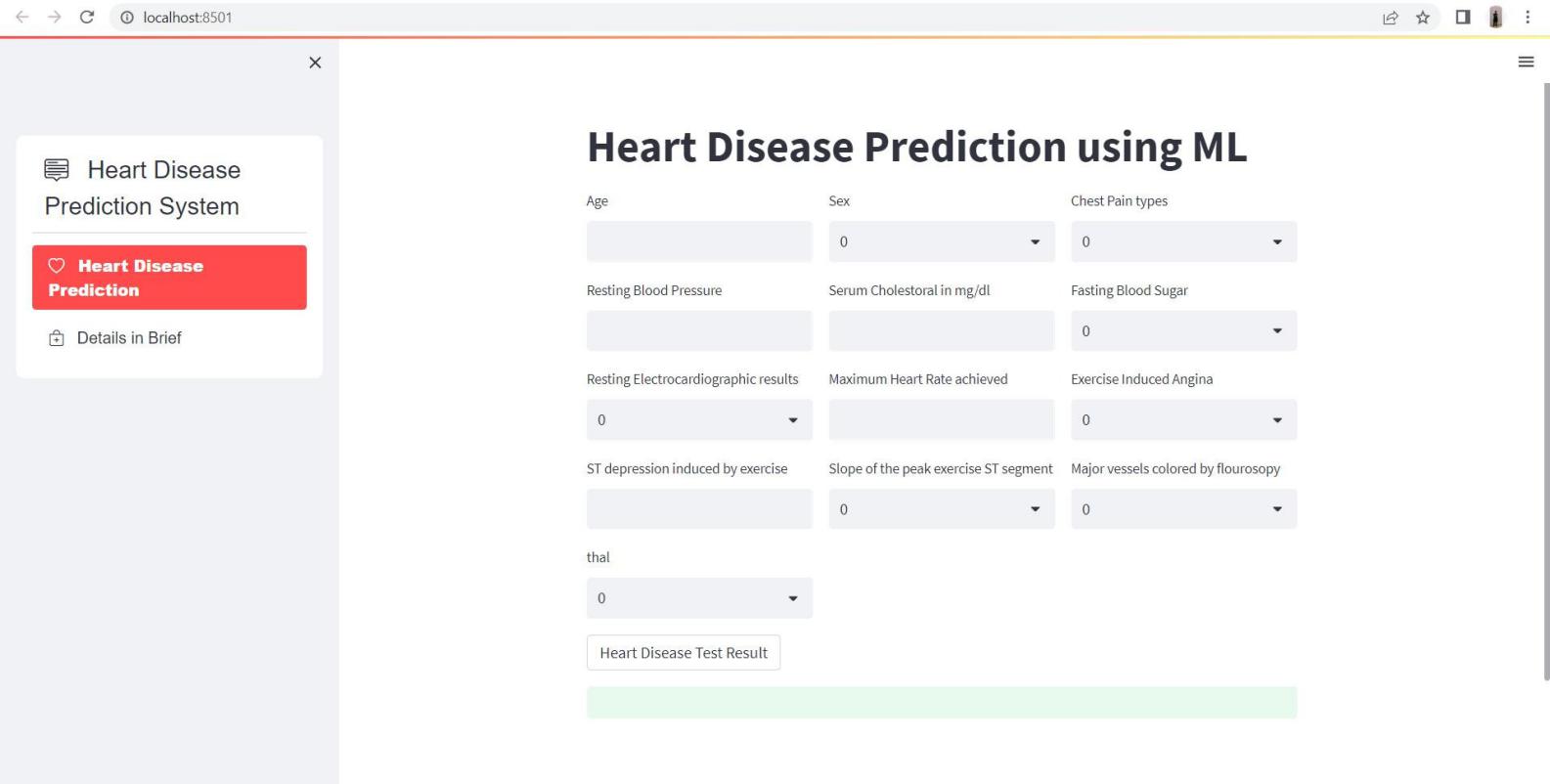
# PERFORMANCE MEASURES

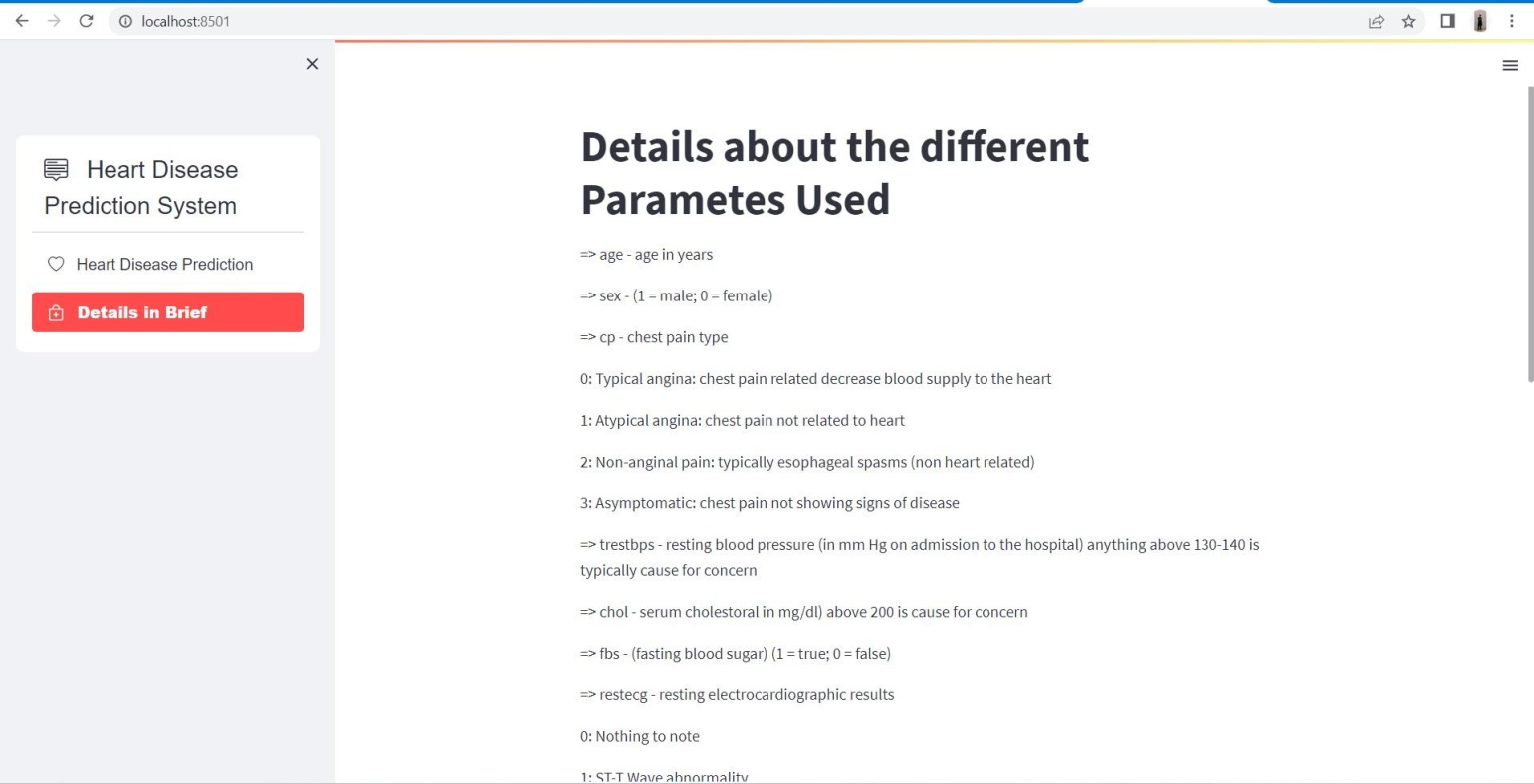
### The highest accuracy is given by Random Forest.

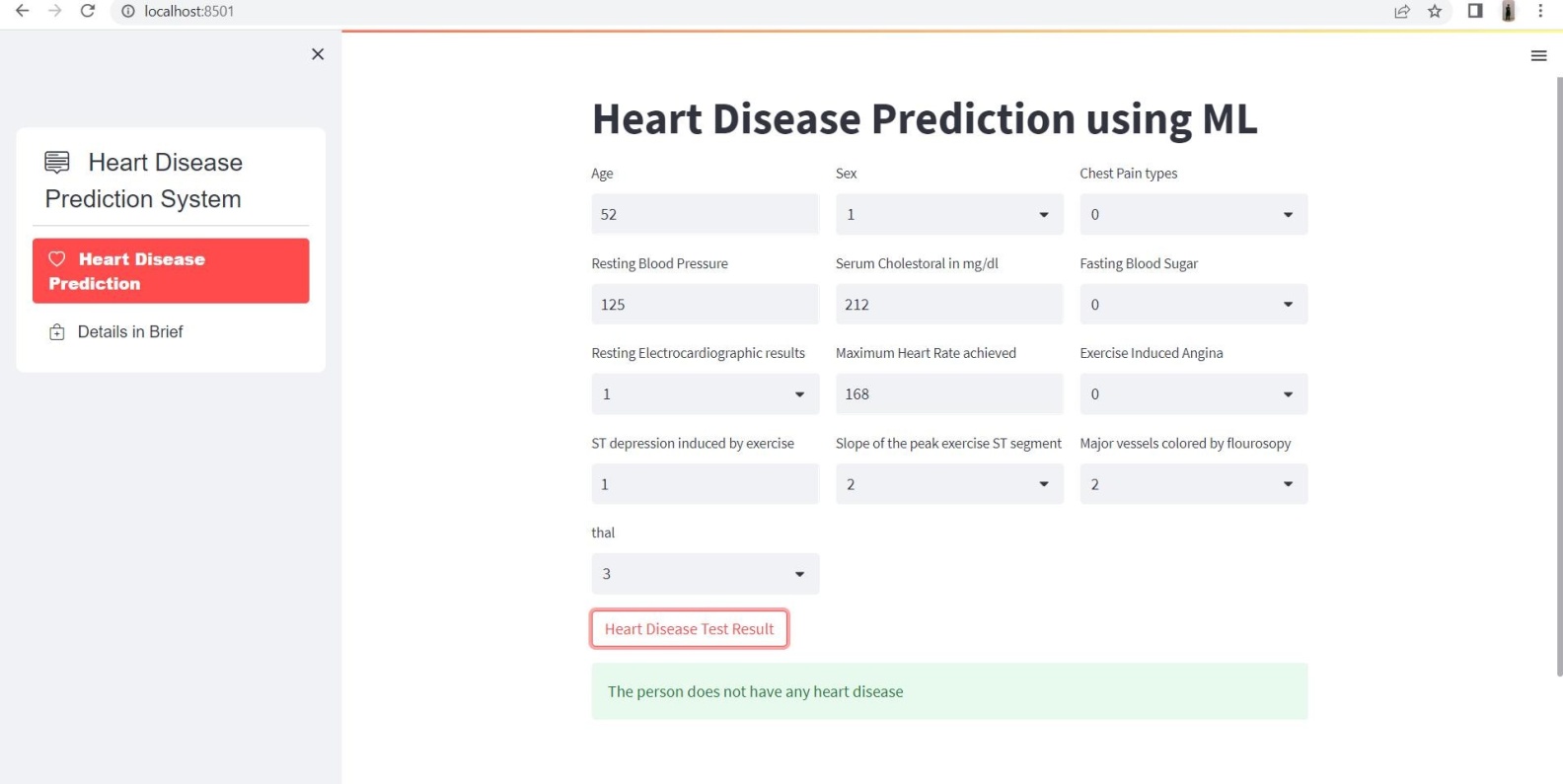




# INPUT AND OUTPUT







# RESULT

After performing the machine learning approach for training and testing we find that accuracy of the Random Forest is better compared to other algorithms.

It is concluded that extreme Random Forest is best with 98% accuracy and the comparison is shown below. TABLE: Accuracy comparison of algorithms Algorithm Accuracy

|  |  |
| --- | --- |
| **Algorithm** | **Accuracy** |
| Logistic Regression | 80.4% |
| SVM | 72.7% |
| SGD Classifier | 49.7% |
| Decision Tree | 95.6% |
| Random Forest Classifier | 98.0% |

# Chapter 6

**CONCLUSION**

### Heart diseases are a major killer in India and throughout the world, application of promising technology like machine learning to the initial prediction of heart

### diseases will have a profound impact on society. The early prognosis of heart disease can aid in making decisions on lifestyle changes in high-risk patients and in turn reduce the complications, which can be a great milestone in the field of medicine. The number of people facing heart diseases is on a raise each year. This prompts for its early diagnosis and treatment. The utilization of suitable technology support in this regard can prove to be highly beneficial to the medical fraternity and patients. In this report, the five different machine learning algorithms used to measure the performance are SVM, Decision Tree, Random Forest, Logistic Regression, and SGD Classifier are applied on the dataset.

### The expected attributes leading to heart disease in patients are available in the

### dataset which contains 14 important features that are useful to evaluate the system are selected among them. All the five machine learning methods accuracies are compared based on which one prediction model is generated. Comparing all five the Random Forest classifier gives the highest accuracy of 98%

**APPENDIX**

**Python**

Python is an interpreted, high-level, general purpose programming language created by Guido Van Rossum and first released in 1991, Python's design philosophy emphasizes code Readability with its notable use of significant White space. Its language constructs and object oriented approach aim to help programmers write clear, logical code for small and large-scale projects. Python is dynamically typed and garbage collected. It supports multiple programming paradigms, including procedural, object-oriented, and functional programming.

**Sklearn**

Scikit-learn (Sklearn) is the most useful and robust library for machine learning in Python. It provides a selection of efficient tools for machine learning and statistical modeling including classification, regression, clustering and dimensionality reduction via a consistent interface in Python. This library, which is largely written in Python, is built upon NumPy, SciPy and Matplotlib.

**Numpy**

NumPy is a library for the python programming language, adding support for large, multi- dimensional arrays and matrices, along with a large collection of high level mathematical functions to operate on these arrays. The ancestor of NumPy, Numeric, was originally created by Jim with contributions from several other developers. In 2005, Travis created NumPy by incorporating features of the competing Numarray into Numeric, with extensive modifications. NumPy is open source software and has many contributors.

**Matplotlib**

Matplotlib is a plotting library for the Python programming language and its numerical mathematics extension NumPy. It provides an object-oriented API for embedding plots into applications using general-purpose GUI toolkits like Tkinter, wxPython, Qt, or GTK. There is also a procedural "pylab" interface based on a statemachine (like OpenGL), designed to closely resemble that of MATLAB, though its use is discouraged.

**Seaborn**

Seaborn is a Python data visualization library based on matplotlib. It provides a high level interface for drawing attractive and informative statistical graphics. Seaborn is a

library in Python predominantly used for making statistical graphics. Seaborn is a data visualization library built on top of matplotlib and closely integrated with pandas data

structures in Python. Visualization is the central part of Seaborn which helps in exploration and understanding of data.

**Pandas**

Pandas is a Python library used for working with data sets. It has functions for analyzing, cleaning, exploring, and manipulating data.Pandas allows us to analyze big data and make conclusions based on statistical theories. Pandas can clean messy data sets, and make them readable and relevant. Relevant data is very important in data science.

**Pickle**

The [pickle](https://docs.python.org/3/library/pickle.html) module implements binary protocols for serializing and de-serializing a Python object structure. “Pickling” is the process whereby a Python object hierarchy is converted into a byte stream,

and “unpickling” is the inverse operation, whereby a byte stream (from a [binary file](https://docs.python.org/3/glossary.html) or [bytes-like object](https://docs.python.org/3/glossary.html)) is converted back into an object hierarchy.

**REFERENCES**

1. Soni J, Ansari U, Sharma D & Soni S (2011). Predictive data mining for medical diagnosis: an overview of heart disease prediction. International Journal of Computer Applications, 17(8), 43-8
2. 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.
3. 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].
4. Shinde R, Arjun S, Patil P & Waghmare J (2015). An intelligent heart disease prediction system using k-means clustering and Naïve Bayes algorithm. International Journal of Computer Science and Information Technologies, 6(1), 637-9.
5. Bashir S, Qamar U & Javed M Y (2014, November). An ensemble-based decision support framework for intelligent heart disease diagnosis. In International Conference on Information Society (i-Society 2014) (pp. 259-64). IEEE. ICCRDA 2020 IOP Conf. Series: Materials Science and Engineering 1022 (2021) 012072 IOP Publishing doi:10.1088/1757-899X/1022/1/012072 9
6. Jee S H, Jang Y, Oh D J, Oh B H, Lee S H, Park S W & Yun Y D (2014). A coronary heart disease prediction model: the Korean Heart Study. BMJ open, 4(5)