

## **HEART DISEASE PREDICTION**

**Ch. Venkatarami Reddy**

**Date: 27-04-2022**

**GITHUB LINK TO PROJECT: [Heart\\_Disease\\_Prediction](#)**

## **Abstract:**

*Heart-related diseases or CardioVascular Diseases (CVD's) are the main reason for a huge number of deaths in the world over the last few decades and has emerged as the most life-threatening disease, not only in India but in the whole world. So, there is a need for a reliable, accurate and feasible system to diagnose such diseases in time for proper treatment. Machine Learning algorithms and techniques have been applied to various medical datasets to automate the analysis of large and complex data. Many researchers, in recent times, have been using several machine learning techniques to help the health care industry and the professionals in the diagnosis of heart-related diseases. Heart is the next major organ comparing to the brain which has more priority in the Human body. It pumps the blood and supplies it to all organs of the whole body. Prediction of occurrences of heart diseases in the medical field is significant work. Data analytics is useful for prediction from more information and it helps the medical center to predict various diseases. A huge amount of patient-related data is maintained on monthly basis. The stored data can be useful for the source of predicting the occurrence of future diseases. Some of the data mining and machine learning techniques are used to predict heart diseases, such as Artificial Neural Network (ANN), Random Forest, and Support Vector Machine (SVM). Prediction and diagnosing of heart disease become a challenging factor faced by doctors and hospitals both in India and abroad. To reduce the large scale of deaths from heart diseases, a quick and efficient detection technique is to be discovered. Data mining techniques and machine learning algorithms play a very important role in this area. The researchers accelerating their research works to develop software with the help of machine learning algorithms which can help doctors to decide both prediction and diagnosing of heart disease. The main objective of this research project is to predict the heart disease of a patient using machine learning algorithm*

**Keywords:** SVM (Supporting Vector Machine), Decision Tree, Random Forest, Logistic Regression, Adaboost, XG-Boost, Python-Programming, KNeighbors

## 1. Problem Statement

The major challenge in heart disease is its detection. There are instruments available which can predict disease either it is expensive or not efficient to calculate chance of heart disease in human. Early detection of cardiac disease 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 medical data.

## 2. Market/Customer Need Assessment

There has been a huge death in the medical sector. Our aim is to analyze the heart disease prediction algorithm for the system which will help doctors and hospitals to get a glance about heart disease predictions which will help them to plan the treatment accordingly.

## 3. Target Specification and characterization

**Finding disease based on the information that have been proposed by the MachineLearning algorithm via checking the dataset.**

If there are sufficient data of persons who are suffering from heart disease then our machine learning model will predict easily.

**Heart disease prediction will help doctors for treatment of person and can use the robust model to analyze if person is going to suffer from heart disease or not .**

## 4. External Search (information sources)

The dataset can be found on my repository which will be provided at the end of report. The dataset consists about the different groups of data which will be helpful to train our model. The sources of subsequent information is given below as references.

```
1 data = pd.read_csv('heart.csv')
2 data.head()
```

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
0	52	1	0	125	212	0	1	168	0	1.0	2	2	3	0
1	53	1	0	140	203	1	0	155	1	3.1	0	0	3	0
2	70	1	0	145	174	0	1	125	1	2.6	0	0	3	0
3	61	1	0	148	203	0	1	161	0	0.0	2	1	3	0
4	62	0	0	138	294	1	1	106	0	1.9	1	3	2	0

## 5. Benchmarking



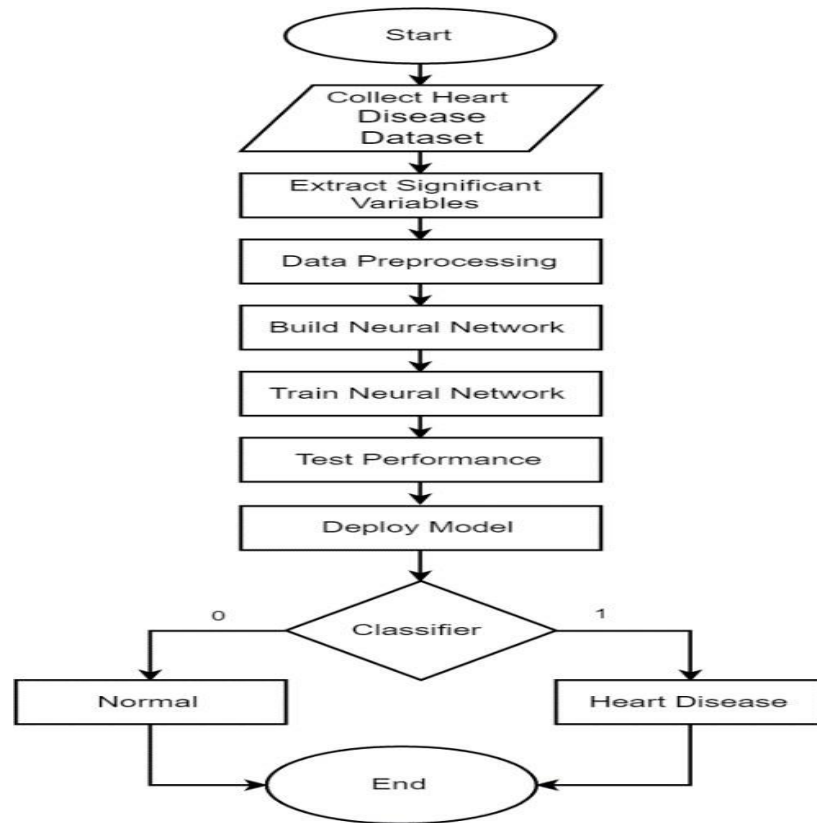
## 6. Applicable Constraints:

1. The use of cloud platforms to store the data gathered over the net.
2. Using the pandas service to clean and transform data.
3. For Evaluation of the model which is done with the help of Keras.
4. For modelling Classification and logistic regression is applied.

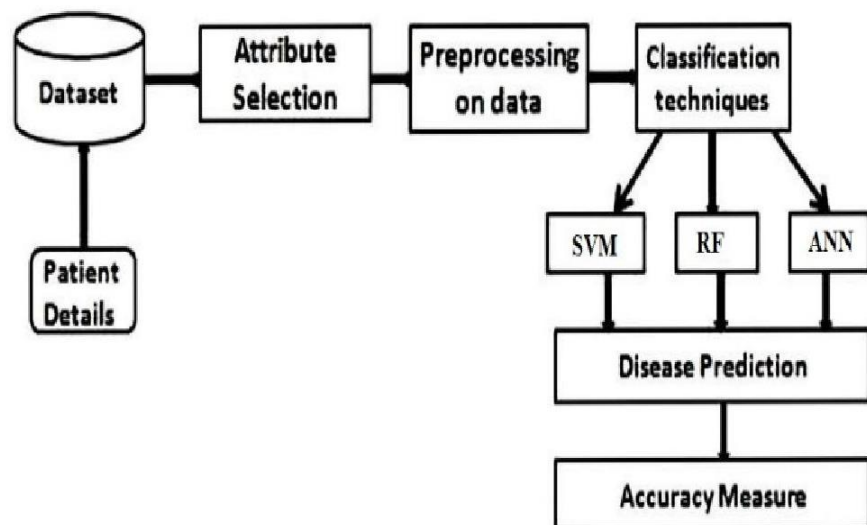
## 7. Concept Generation

This product requires the tool of machine learning models to be written from scratch in order to suit our needs. Tweaking these models for our use is less daunting than coding it up from scratch. A well-trained model can either be repurposed or built. But building a model with the resources and data we have is dilatory but possible. The customer might want to spend the least amount of time giving input data. This accuracy will take a little effort to nail, because it's imprudent to rely purely on Classic Machine Learning algorithm.

This will be the proposed flow chart that the system will look like:

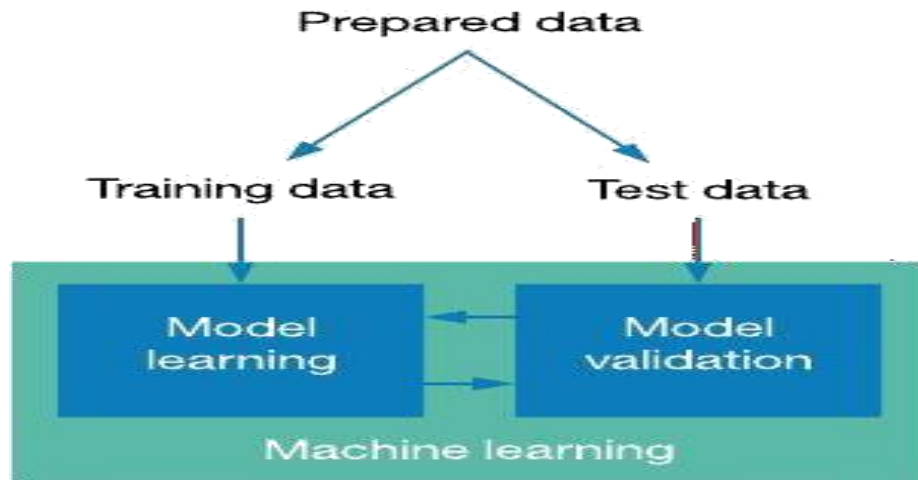


**Proposed System:**



## 7.1 Collection of datasets

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



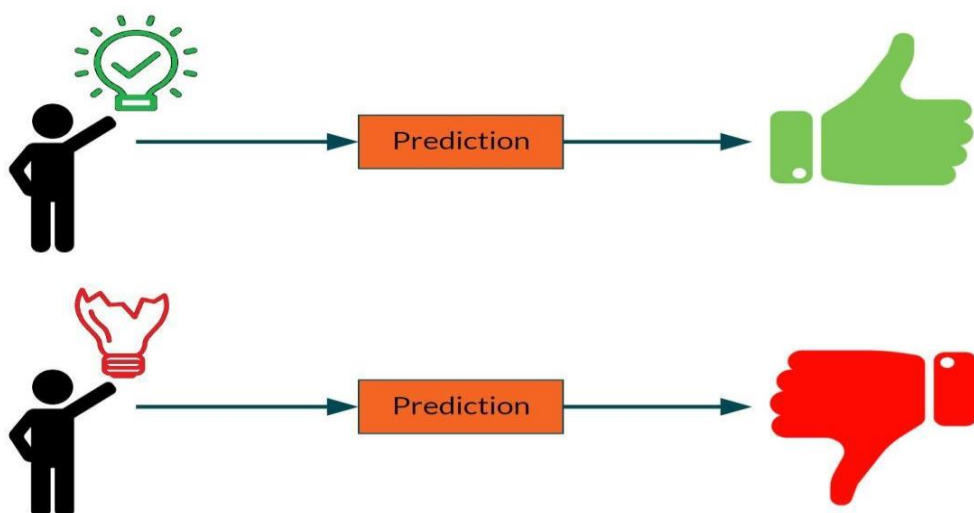
## 7.2 Pre-processing of Data

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.



### 7.3 Prediction of Disease

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



### Classifiers Used for Experiments.

#	Attributes	Description	Values
1	Age	Patient's age in years	Continuous Value
2	Sex	Sex of Patient	1 = Male 0 = Female
3	Cp	Chest pain	Value 1: typical angina Value 2: atypical angina Value 3: non-angina pain Value 4: asymptomatic
4	Trestbps	Resting blood pressure	Continuous value in mm/Hg
5	Chol	Serum cholesterol in mg/dl	Continuous value in mg/dl
6	Fbs	Fasting blood sugar	1 $\geq$ 120 mg/dl 0 $\leq$ 120 mg/dl
7	Restcg	Resting electrocardiographic results	0 = normal 1 = having_ST_T wave abnormal 2 = left ventricular hypertrophy
8	Thalach	Maximum heart rate achieved	Continuous value
9	Exang	Exercise induced angina	1: yes 0: no
10	Oldpeak	ST depression induced by exercise relative to rest	Continuous value
11	Slope	the slope of the peak exercise ST segment	1: upsloping 2: flat 3: down sloping
12	Ca	number of major vessels colored by fluoroscopy	0-3 value
13	Thal	defect type	3 = normal 6 = fixed defect 7 = reversible defect
14	num	diagnosis of heart disease	no_heart_disease have_heart_disease

## 7.4 Machine Learning

In Machine Learning, classifications refer to a predictive modeling problem, where a class label is predicted for a given example of input data.

### 7.4.1 Supervised Learning

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

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

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

### 7.4.2 Unsupervised learning

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

- Unsupervised learning is helpful for finding useful insights from the data.
- Unsupervised learning is much similar to how a human learns to think by their own experiences, which makes it closer to the real AI.
- Unsupervised learning works on unlabeled and uncategorized data which make unsupervised learning more important.
- In real-world, we do not always have input data with the corresponding output so to solve such cases, we need unsupervised learning.

### 7.4.3 Reinforcement learning

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



## 7.5 ALGORITHMS

### 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.

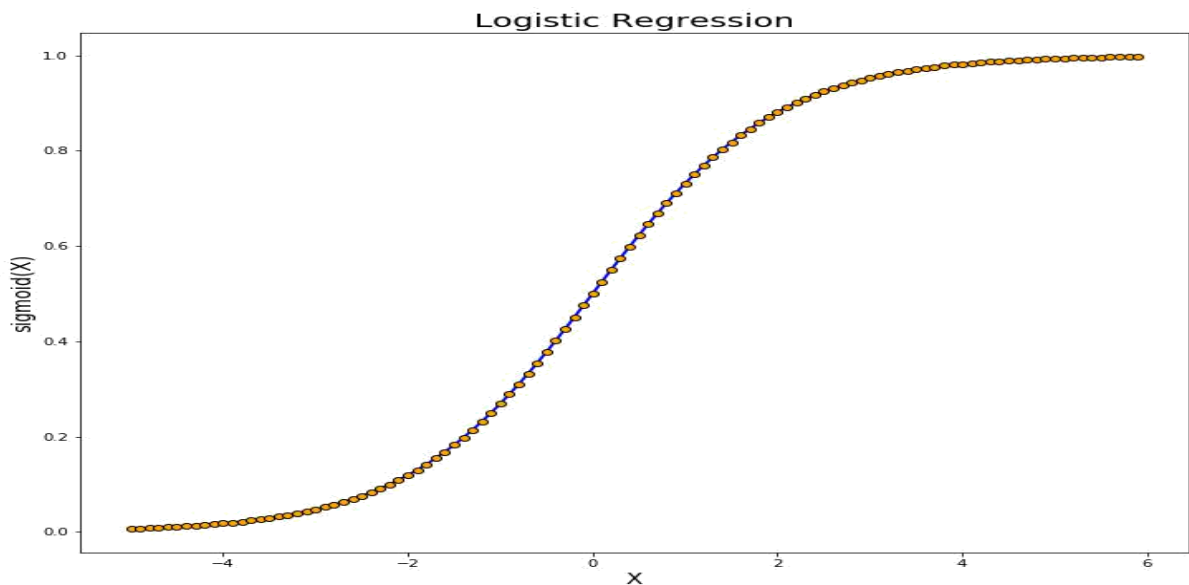
Logistic regression predicts the output of a categorical dependent variable. Therefore, the outcome must be a categorical or discrete value. It can be either Yes or No, 0 or 1, true or False, etc. but instead of giving the exact value as 0 and 1, it gives the probabilistic values which lie between 0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

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

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

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

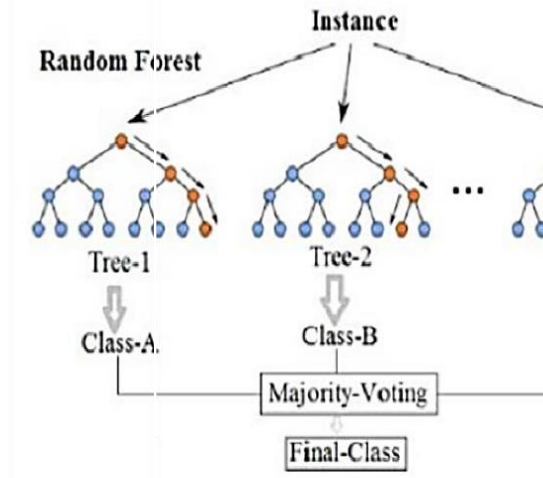
Logistic Regression is a significant machine learning algorithm because it has the ability to provide probabilities and classify new data using continuous and discrete datasets.



### 2. RANDOM FOREST

Random Forest is a supervised machine learning algorithm. This technique can be used for both regression and classification tasks but generally performs better in classification tasks. As the name suggests, Random Forest technique considers multiple decision trees before giving an output. So, it is basically an ensemble of decision trees. This technique is based on the belief that more number of trees would converge to the right decision. For classification, it uses a voting system and then decides the class whereas in regression it takes the mean of all the outputs of each of the decision trees. It works well with large datasets with high dimensionality.

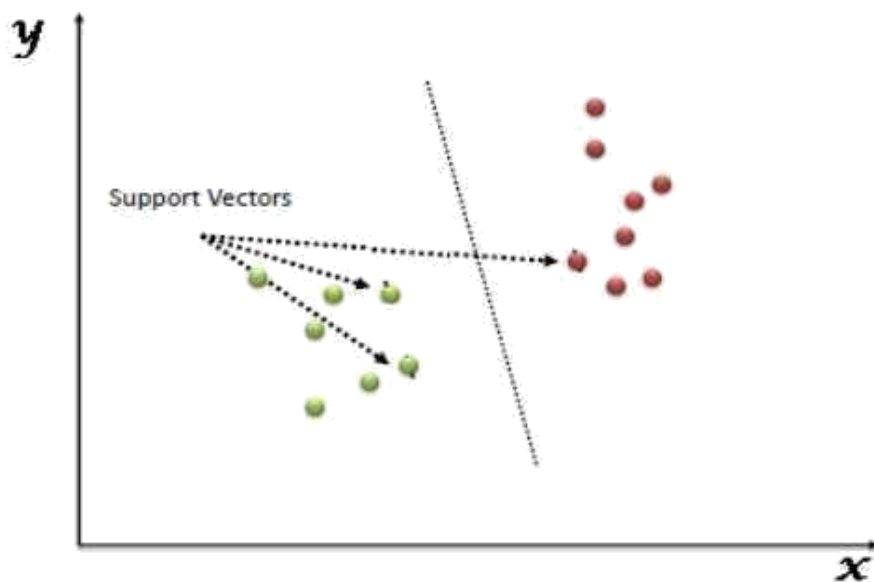
## Random Forest Simplified



### 3. SVM (Support Vector Machine)

Support vector machines exist in different forms, linear and non-linear. A support vector machine is a supervised classifier. What is usual in this context, two different datasets are involved with SVM, training and a test set. In the ideal situation the classes are linearly separable. In such situation a line can be found, which splits the two classes perfectly. However not only one line splits the dataset perfectly, but a whole bunch of lines do. From these lines the best is selected as the "separating line".

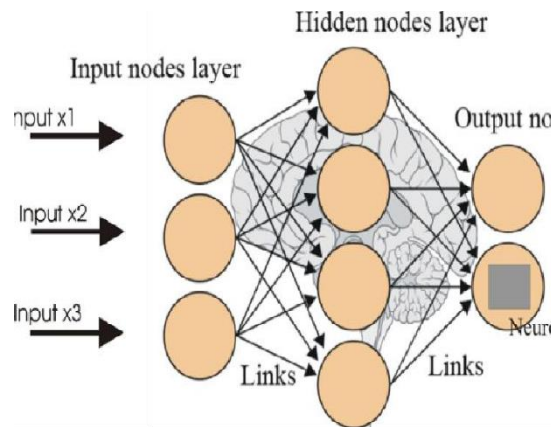
A SVM can make some errors to avoid over-fitting. It tries to minimize the number of errors that will be made. Support vector machines classifiers are applied in many applications. They are very popular in recent research. This popularity is due to the good overall empirical performance. Comparing the naive Bayes and the SVM classifier, the SVM has been applied the most.



#### 4. Artificial Neural Network

These are used to model/simulate the distribution, functions or mappings among variables as modules of a dynamic system associated with a learning rule or a learning algorithm. The modules here simulate neurons in nervous system and hence ANN collectively refers to the neuron simulators and their synapsis simulating interconnections between these modules in different layers.

Neural Network is built by stacking together multiple neurons in layers to produce a final output. First layer is the input layer and the last is the output layer. All the layers in between is called hidden layers. Each neuron has an activation function. Some of the popular Activation functions are Sigmoid, ReLU, tanh etc. The parameters of the network are the weights and biases of each layer. The goal of the neural network is to learn the network parameters such that the predicted outcome is the same as the ground truth. Back-propagation along loss-function is used to learn the network parameters



#### 5. Decision Tree Algorithm

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

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

## 8. Concept Development

The concept can be developed by using The appropriate python frameworks and libraries for the same and for its deployment, The cloud services has to be chosen accordingly to the need.

### 1. Importing the Libraries

```
[ ] 1 import pandas as pd
    2 import numpy as np
    3 from sklearn.preprocessing import StandardScaler
    4 from sklearn.model_selection import train_test_split
    5 from sklearn.linear_model import LogisticRegression
    6 from sklearn.metrics import accuracy_score
    7 from sklearn import svm
    8 from sklearn.neighbors import KNeighborsClassifier
    9 from sklearn.tree import DecisionTreeClassifier
   10 from sklearn.ensemble import RandomForestClassifier
   11 from sklearn.ensemble import GradientBoostingClassifier
   12 import seaborn as sns
   13 import joblib
   14 from tkinter import *
   15 import matplotlib.pyplot as plt
```

### 2. Importing Dataset

```
[ ] 1 data = pd.read_csv('heart.csv')
```

### 3. Taking care of Missing Values

```
▶ 1 data.isnull().sum()
```

### 4. Taking care of Duplicate Values

```
[ ] 1 data_dup = data.duplicated().any()
```

```
[ ] 1 data_dup
```

True

```
[ ] 1 data = data.drop_duplicates()
```

```
[ ] 1 data_dup = data.duplicated().any()
```

```
[ ] 1 data_dup
```

False

## 5. Data Processing

```
[ ] 1 cate_val = [] # Columns which contains categorical values.  
    2 cont_val = [] # Columns which contains numerical values.  
    3  
    4 for column in data.columns :  
    5     if data[column].nunique() <= 10 :  
    6         cate_val.append(column)  
    7     else :  
    8         cont_val.append(column)
```

```
[ ] 1 cate_val  
  
    ['sex', 'cp', 'fbs', 'restecg', 'exang', 'slope', 'ca', 'thal', 'target']
```

```
[ ] 1 cont_val  
  
    ['age', 'trestbps', 'chol', 'thalach', 'oldpeak']
```

## 6. Encoding Categorical Data

```
[ ] 1 cate_val  
  
    ['sex', 'cp', 'fbs', 'restecg', 'exang', 'slope', 'ca', 'thal', 'target']
```


```
[ ] 1 data['cp'].unique()  
  
    array([0, 1, 2, 3])
```

```
[ ] 1 cate_val.remove("target")  
    2 cate_val.remove("sex")  
    3  
    4 # These cols are already contains 0's & 1's so no need of Encoding  
    5  
    6 data = pd.get_dummies(data, columns = cate_val, drop_first = True)
```

```
[ ] 1 data.head()
```

## 7. Feature Scaling

```
[ ] 1 st = StandardScaler()  
    2 data[cont_val] = st.fit_transform(data[cont_val])
```

```
 1 data.head()
```

## 8. Splitting The Dataset Into Training Set and Test Set

```
[ ] 1 X = data.drop("target", axis = 1)

[ ] 1 Y = data["target"]

[ ] 1 X_train , X_test, Y_train , Y_test = train_test_split(X,Y,test_size = 0.2, random_state=40)

[ ] 1 X_train, Y_train

[ ] 1 X_test, Y_test
```

## 9. Logistic Regression

```
▶ 1 data.head()

[ ] 1 log = LogisticRegression()
    2 log.fit(X_train, Y_train)

[ ] 1 y_pred1 = log.predict(X_test)

[ ] 1 accuracy_score(Y_test, y_pred1)

0.9508196721311475
```

## 10. SVC (Support Vector Classifier)

```
[ ] 1 svm = svm.SVC()

[ ] 1 svm.fit(X_train, Y_train)

▶ 1 y_pred2 = svm.predict(X_test)
    2 accuracy_score(Y_test, y_pred2)

0.8688524590163934
```

## 11. KNeighbors Classifier

```
[ ] 1 knn = KNeighborsClassifier()

▶ 1 knn.fit(X_train, Y_train)

[ ] 1 y_pred3 = knn.predict(X_test)

[ ] 1 accuracy_score(Y_test, y_pred3)

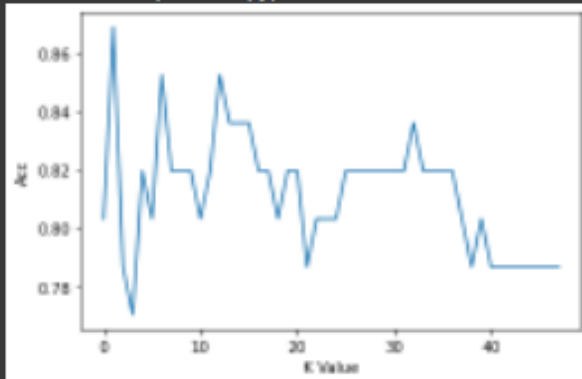
0.819672131147541
```

```
[ ] 1 score = []
    2
    3 for k in range(1,49) :
    4     knn = KNeighborsClassifier(n_neighbors = k)
    5     knn.fit(X_train, Y_train)
    6     y_pred = knn.predict(X_test)
    7     score.append(accuracy_score(Y_test, y_pred))
```

```
[ ] 1 score
```

```
▶ 1 plt.plot(score)
   2 plt.xlabel("K Value")
   3 plt.ylabel("Acc")
   4 plt.show
```

```
↳ <function matplotlib.pyplot.show>
```



```
[ ] 1 knn = KNeighborsClassifier(n_neighbors = 2)
    2 knn.fit(X_train, Y_train)
    3 y_pred = knn.predict(X_test)
    4 accuracy_score(Y_test, y_pred)
```

```
0.8688524590163934
```

## Non-Linear ML Algorithms

```
▶ 1 data = pd.read_csv('heart.csv')
   2 data.head()
```

```
[ ] 1 data = data.drop_duplicates()
    2 data.shape
```

```
(302, 14)
```

```
[ ] 1 X = data.drop('target', axis = 1)
    2 Y = data['target']
```

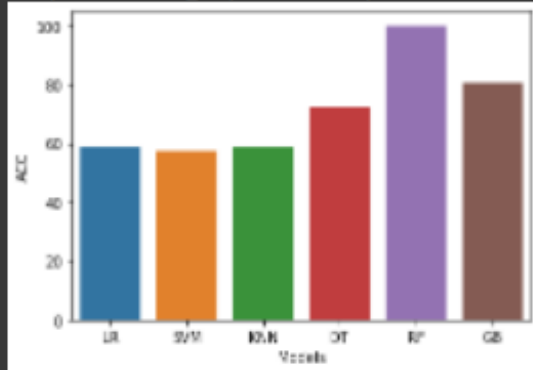
```
[ ] 1 X_train , X_test, Y_train , Y_test = train_test_split(X,Y,test_size = 0.2, random_state=42)
```





```
1 sns.barplot(final_data['Models'], final_data['ACC'])
```

```
[ ] /usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarning: Pass the following variables as keyword a
FutureWarning
<matplotlib.axes._subplots.AxesSubplot at 0x7fa3b6f04a10>
```



So Random Forest model gives a good accuracy so it is best model

```
[ ] 1 X = data.drop("target", axis=1)
2 Y = data['target']
```

```
[ ] 1 X.shape
```

```
(382, 13)
```

```
[ ] 1 rf = RandomForestClassifier()
2 rf.fit(X,Y)
```

```
RandomForestClassifier()
```

## 15. Prediction on New Data

```
[ ] 1 new_data = pd.DataFrame({
2     'age': 52,
3     'sex' : 1,
4     'cp' : 0,
5     'trestbps' : 125,
6     'chol' : 212,
7     'fbs' : 0,
8     'restecg' : 1,
9     'thalach' : 168,
10    'exang' : 0,
11    'oldpeak' : 1.0,
12    'slope' : 2,
13    'ca' : 2,
14    'thal' : 3,
15 }, index=[0])
16
17 new_data
```

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal
0	52	1	0	125	212	0	1	168	0	1.0	2	2	3

```
[ ] 1 p = rf.predict(new_data)
    2 if p[0] == 0:
    3     print("No Disease")
    4 else :
    5     print("Suffering from Disease")
```

No Disease

## 16. Save Model Using Joblib

```
[ ] 1 joblib.dump(rf, 'model_joblib_heart')

    ['model_joblib_heart']

[ ] 1 model = joblib.load('model_joblib_heart')

▶ 1 model.predict(new_data)

array([0])
```

## 9. Final Report Prototype

The product takes the following functions to perfect and provide a good result.

### Back-end

Model Development: This must be done before testing the model. A lot of manual supervised machine learning must be performed to optimize the automated tasks.

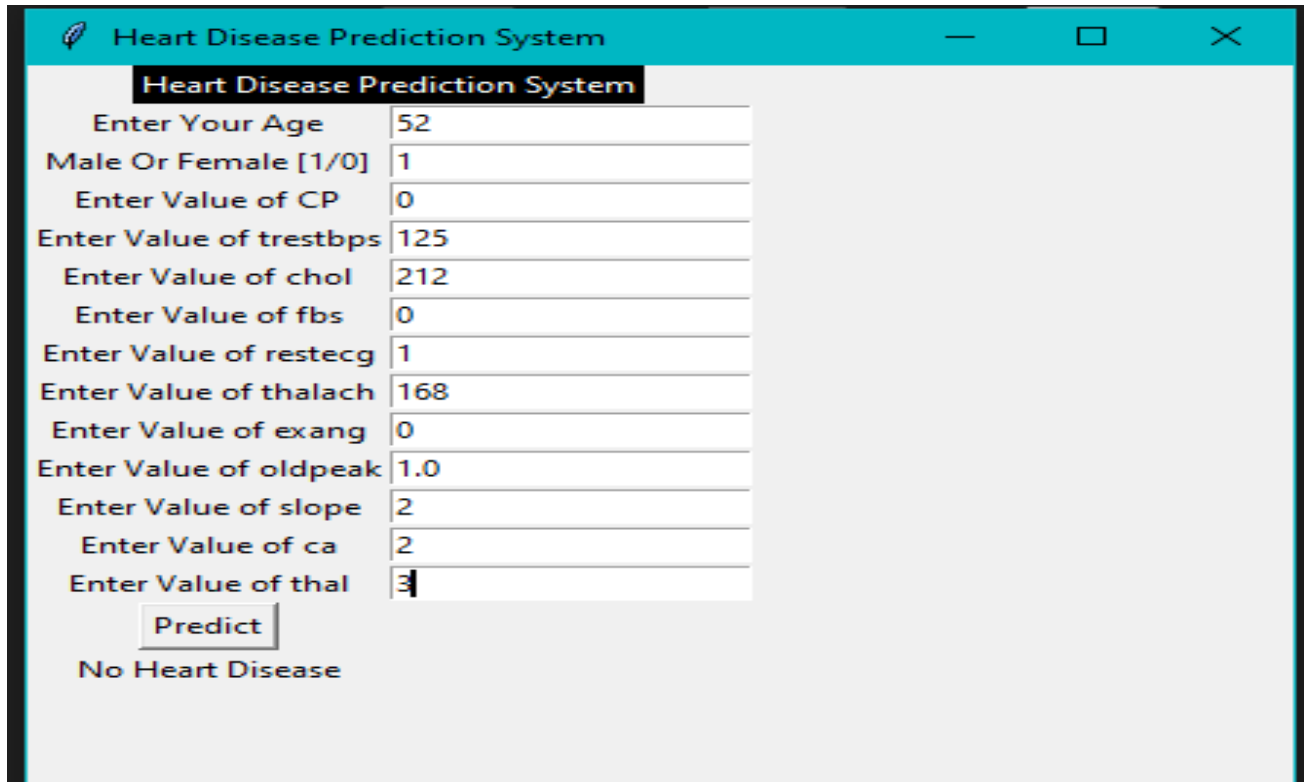
1. Performing EDA to realize the dependent and independent features.
2. Algorithm training and optimization must be done to minimize overfitting of the model and hyperparameter tuning.

### Front End

1. Different user interface: The user must be given many options to choose form in terms of parameters. This can only be optimized after a lot of testing and analysis all the edge cases. We used Tkinter as front end which is a python framework

## 10. Product details - How does it work?

An interactive user system will take inputs regarding different health condition values from the user and the user will get to know about whether he is suffering from a heart disease or not.

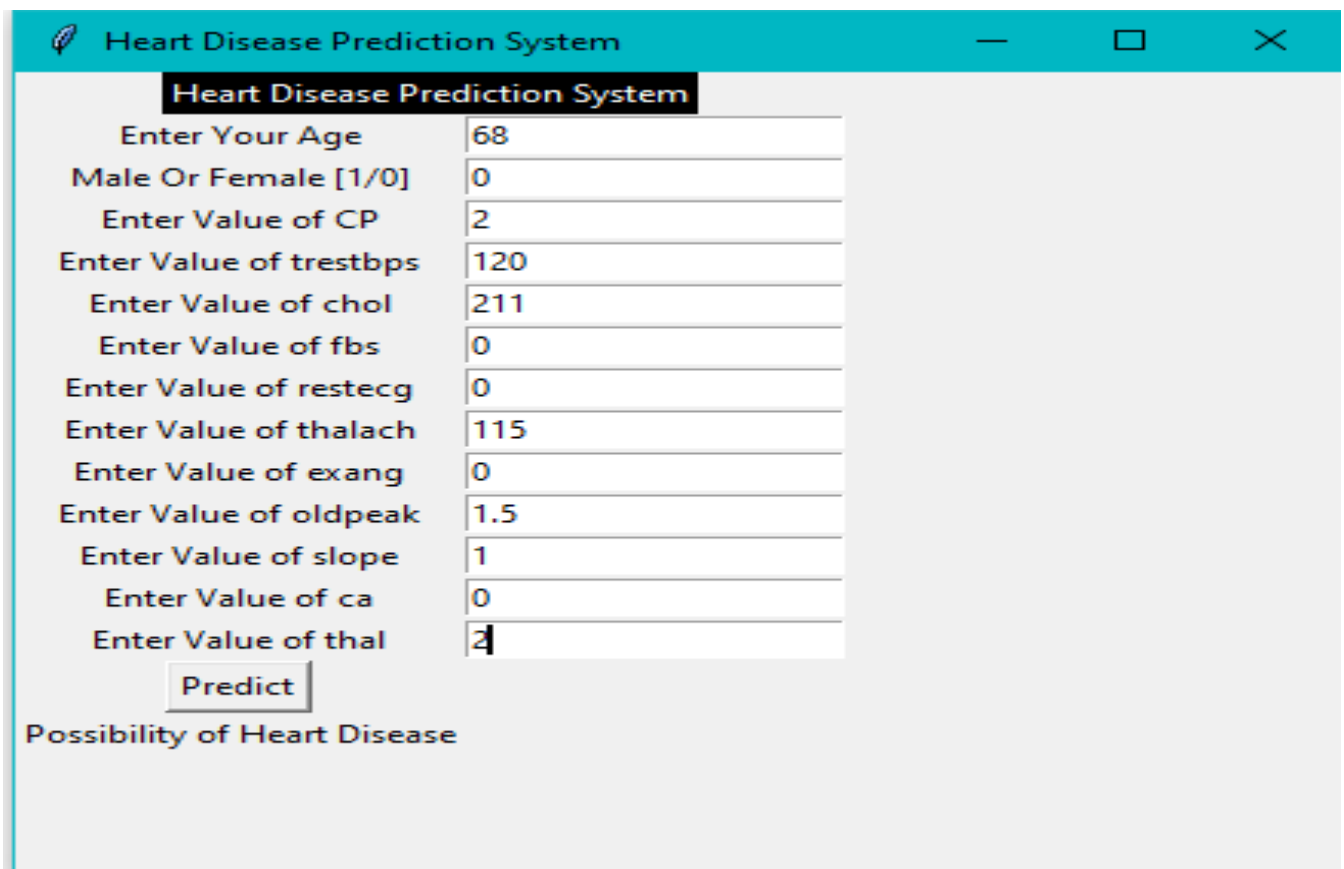


**Heart Disease Prediction System**

Enter Your Age	52
Male Or Female [1/0]	1
Enter Value of CP	0
Enter Value of trestbps	125
Enter Value of chol	212
Enter Value of fbs	0
Enter Value of restecg	1
Enter Value of thalach	168
Enter Value of exang	0
Enter Value of oldpeak	1.0
Enter Value of slope	2
Enter Value of ca	2
Enter Value of thal	3

**Predict**

No Heart Disease



**Heart Disease Prediction System**

Enter Your Age	68
Male Or Female [1/0]	0
Enter Value of CP	2
Enter Value of trestbps	120
Enter Value of chol	211
Enter Value of fbs	0
Enter Value of restecg	0
Enter Value of thalach	115
Enter Value of exang	0
Enter Value of oldpeak	1.5
Enter Value of slope	1
Enter Value of ca	0
Enter Value of thal	2

**Predict**

Possibility of Heart Disease

## 11.References/Source of Information

- [1] Mr. ChalaBeyene, Prof. Pooja Kamat, “Survey on Prediction and Analysis the Occurrence of Heart Disease Using Data Mining Technique”, International Journal of Pure and Applied Mathematics, 2018.
- [2] Mohan, Senthilkumar, Chandrasegar Thirumalai, and Gautam Srivastava, “Effective heart disease prediction using hybrid machine learning techniques” IEEE Access 7 (2019): 81542-81554
- [3] Ali, Liaqat, et al, “An optimized stacked support vector machines based expert system for the effective prediction of heart failure” IEEE Access 7 (2019): 54007-54014.
- [4] Singh Yeshvendra K., Nikhil Sinha, and Sanjay K. Singh, “Heart Disease Prediction System Using Random Forest”, International Conference on Advances in Computing and Data Sciences. Springer, Singapore, 2016.
- [5] Prerana T H M1, Shivaprakash N C2 , Swetha N3 “Prediction of Heart Disease Using Machine Learning, Algorithms- Naïve Bayes, Introduction to PAC Algorithm, Comparison of Algorithms and HDPS”
- [6] B.L DeekshatuluaPriti Chandra “Classification of Heart Disease Using K- Nearest Neighbor and Genetic Algorithm” International Conference on Computational Intelligence: Modeling Techniques and Applications (CIMTA) 2013.
- [7] S. Shilaskar and A.Ghatol, “Feature selection for medical diagnosis :Evaluation for cardiovascular diseases,” Expert Syst. Appl., vol. 40, no. 10, pp. 4146–4153, Aug. 2013.
- [8] C.-L. Chang and C.-H. Chen, “Applying decision tree and neural network to increase quality of dermatologic diagnosis,” Expert Syst. Appl., vol. 36, no. 2, Part 2, pp. 4035–4041, Mar. 2009.
- [9] T. Azar and S. M. El-Metwally, “Decision tree classifiers for automated medical diagnosis,” Neural Comput. Appl., vol. 23, no. 7–8, pp. 2387–2403, Dec. 2013. [10] Y. C. T. Bo Jin, “Support vector machines with
- [10] genetic fuzzy feature transformation for biomedical data classification.,” Inf Sci, vol. 177, no. 2, pp. 476–489, 2007.
- [11] N. Esfandiari, M. R. Babavalian, A.-M. E. Moghadam, and V. K. Tabar, “Knowledge discovery in medicine: Current issue and future trend,” Expert Syst. Appl., vol. 41, no. 9, pp. 4434–4463, Jul. 2014.
- [12] E. Hassanien and T. Kim, “Breast cancer MRI diagnosis approach using support vector machine and pulse coupled neural networks,” J. Appl. Log., vol. 10, no. 4, pp. 277–284, Dec. 2012.
- [13] Sanjay Kumar Sen 1, Dr. Sujata Dash 21Asst. Professor, Orissa Engineering College, Bhubaneswar, Odisha – India.
- [14] B.L Deekshatulua Priti Chandra “Reader, PG Dept. Of Computer Application North Orissa University, Baripada, Odisha – India. Empirical Evaluation of Classifiers Performance Using Data Mining Algorithm”

GITHUB LINK TO PROJECT: [Heart Disease Prediction](#)