Project Report

Team ID: PNT2022TMID10663

Team Leader: Lingeshwar S

Team Members: Logesh Prabu R M

Mathiazhagan M S

Mohamed Yusuf S

Prabakaran R

Project Title: Visualizing and Predicting Heart disease

1. INTRODUCTION:

1.1 PROJECT OVERVIEW:

According to the World Health Organization, every year 12 million deaths occur worldwide due to Heart Disease. Heart disease is one of the biggest causes of morbidity and mortality among the population of the world. Prediction of cardiovascular disease is regarded as one of the most important subjects in the section of data analysis. The load of cardiovascular disease is rapidly increasing all over the world from the past few years. Numerous studies have been undertaken in an effort to identify the most significant risk factors for heart disease and to precisely forecast the general risk Even the silent killer of heart disease, which causes the person's death without obvious symptoms. early heart disease detection plays a crucial role in determining whether high-risk individuals should make lifestyle modifications and in lessens the complications in turn.

Machine learning proves to be effective in assisting in making decisions and predictions from the large quantity of data produced by the health care industry. This project aims to predict future Heart Disease by analyzing data of patients which classifies whether they have heart disease or not using machine-learning algorithm. Machine Learning techniques can be a boon in this regard. There is a common set of basic risk factors that determine whether or not someone will ultimately be at risk for heart disease, despite the fact that heart disease can manifest itself in various ways. We may say that this technique can be very well fitted to accomplish the prediction of heart disease by gathering the data from many sources, classifying them under acceptable categories, and then analysing to obtain the needed data.

1.2 PURPOSE:

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

2.LITERATURE SURVEY:

2.1 EXISTING SYSTEM:

[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 opensource 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.

Advantage:

• The accuracy of the system is about 86.7%.

Disadvantage:

• The parameters and their values used are confidence.

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

Advantage:

• This algorithm gives an 87% accuracy.

Disadvantage:

• It can handle complicated, nonlinear, dependent data so it is found suitable for heart disease dataset as this dataset is also complicated, dependent and nonlinear in nature.

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

Advantage:

• Decision Tree has highest accuracy than Bayesian classifier.

Disadvantage:

• Some analysis has been led to think about the execution of prescient data mining strategy on the same dataset.

[4] 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 feed forwarded or feedback.

Advantage:

• This dataset is performed by using neural network algorithm multi-layer perceptron.

Disadvantage:

• One input layer and one output layer and one or more layers are hidden layers between these two input and output layers.

2.2 REFERENCES:

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

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

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

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

2.3 PROBLEM STATEMENT DEFINITION:

Problem Statement (PS)	I am (Customer)	I'm trying to	But	Because	Which makes me feel
PS-1	Heart diseases is said to be a big threat for the people above the age of 40.But now a days even the youngest people under the age of 40(between 30-40) might have a high chances of getting coronary artery diseases(CAD).This condition occurs when fatty substances called	Our solution is about to findout the persons who are all on the edge to caught by heart disease.	When they facing a problem of health illness they feel lonely,get depressed of them and their family,feel insecure etc.	The main reason of getting Cardio Vascular Diseases are diabetes,high cholesterol, blood pressure,smoking,mental depression,eating an unhealthy diet and family history of heart disease.	Cardiophobia is the main fear,Get tensed and collapsed when handling tough situations,And they want a good care from their loved ones.

	plaque builds up inside your coronary arteries.And other reasons are due to hyper tension rise in BP level(diabetes people below 80/120mmhg				
PS-2	Heart faliure due to shortness of breadth heart attack due to sudden cause of blockage in valve.In Medical feld it can be treated mostly using ECG. But when we come to technology feld for identifying and providing a solution in the feld of medicine,we must undergo several ideas to innovate things that make use of individuals who are all undergoes these problems	For this we taking a survey on people health conditions by age, gender and what type of foods they are intaking ,b y this we predict and visualize the people those who are all normal.	After knowing their illness can be treated,they have hope,confidenc e to tackle their problem and fight for they love.	The main reason of getting Cardio Vascular Diseases are diabetes,high cholesterol, blood pressure,smoking,ment a l depression,eating an unhealthy diet and family history of heart disease.	It is the most valuable thing for them and First of all they should have the self-belief that they'll be alright.

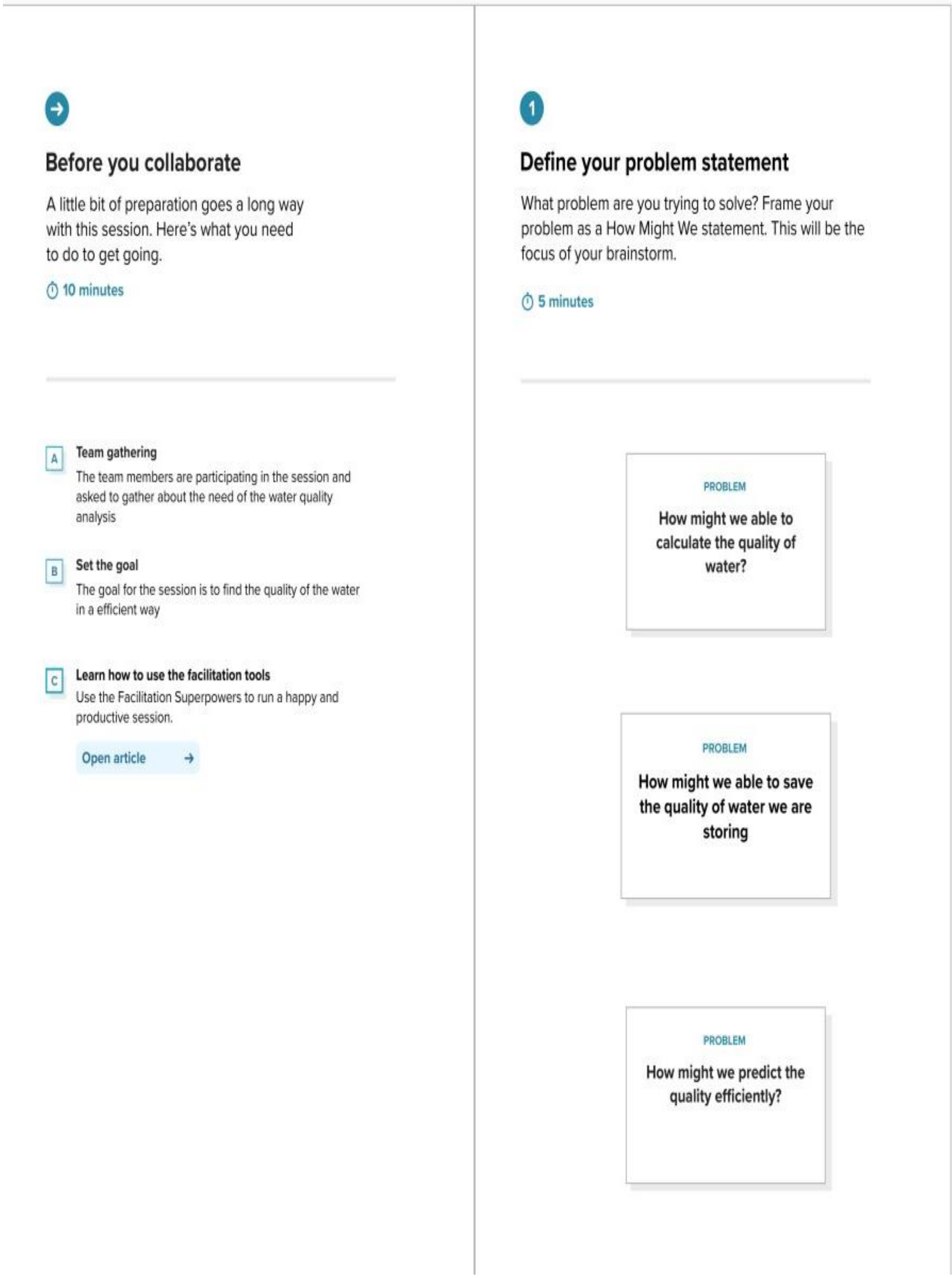
3. MILESTONE AND ACTIVITY LIST:

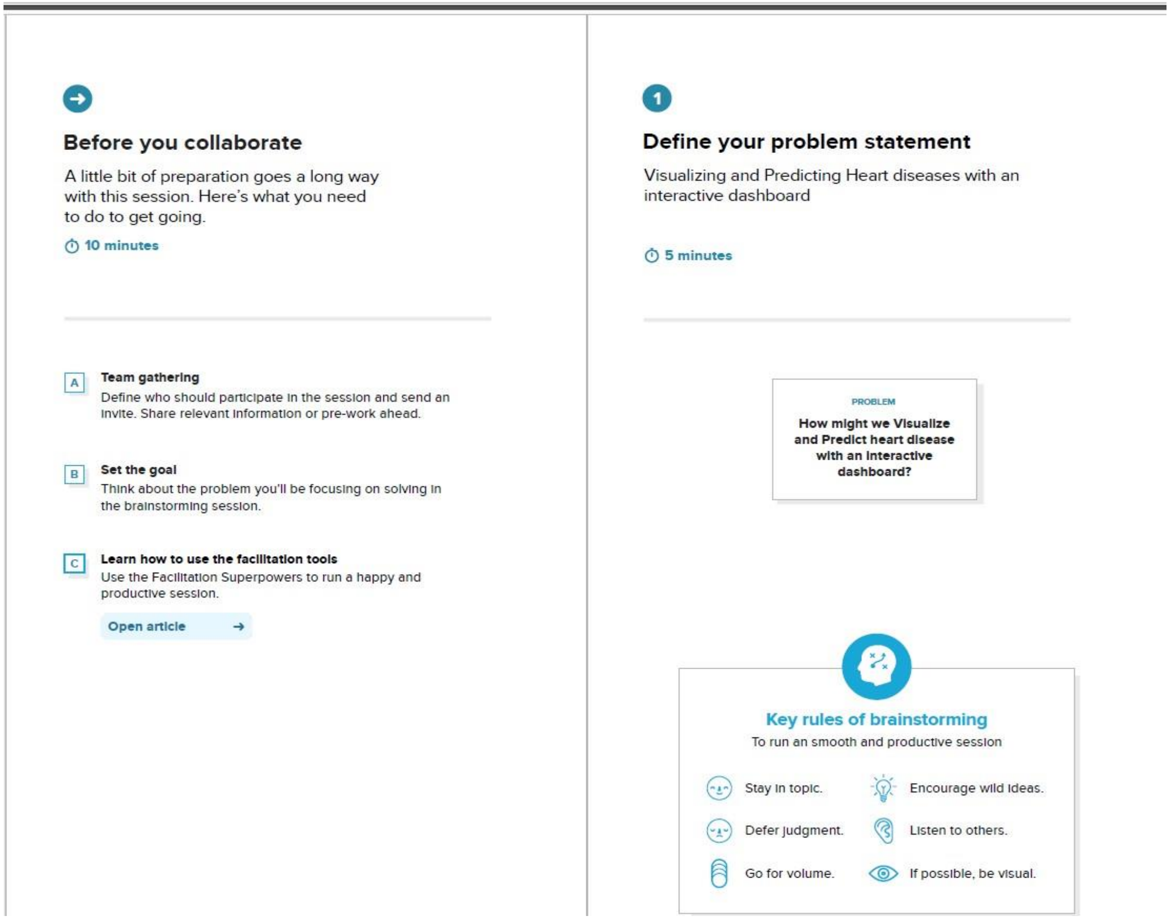
Activity Number	Activity Name	Detailed Activity Description	Status / Comments
1	**Preparation Phase**	Access the resources (courses) in project dashboard Access the guided project workspace Create GitHub account & collaborate with Project Repository in project workspace Set-up the Laptop / Computers based on the prerequisites for each technology track	It refers to do the listed activities in the preparation phase and done Prerequisites, Registration, Environment setup
2	**Ideation Phase**	Literature survey on the selected project & Information Gathering Preparation of Empathy Map Canvas to capture the user Pains & Gains, Prepare list of problem statements List the ideas by organizing the brainstorming session and prioritize the top 3 ideas based on the feasibility & importance	The activities in ideation phase refers to when gathering the idea for project information and picturize in Empathy map, referring the literature survey& brainstorming the ideas for this project.

3	**Project Design Phase -I**		
3.1	Proposed Solution	Preparation of proposed solution document, which includes the novelty, feasibility of idea, business model, social impact, scalability of solution	The solution for the project is prepared as a standard Document structure from team members.
3.2	Problem Solution fit	Preparation of problem solution fit	Prepared problem is analyzed and make effective solutions for the problems.
3.3	Solution Architecture	Prepare an architecture for solution	Suitable block diagram template used prepare the solution architecture

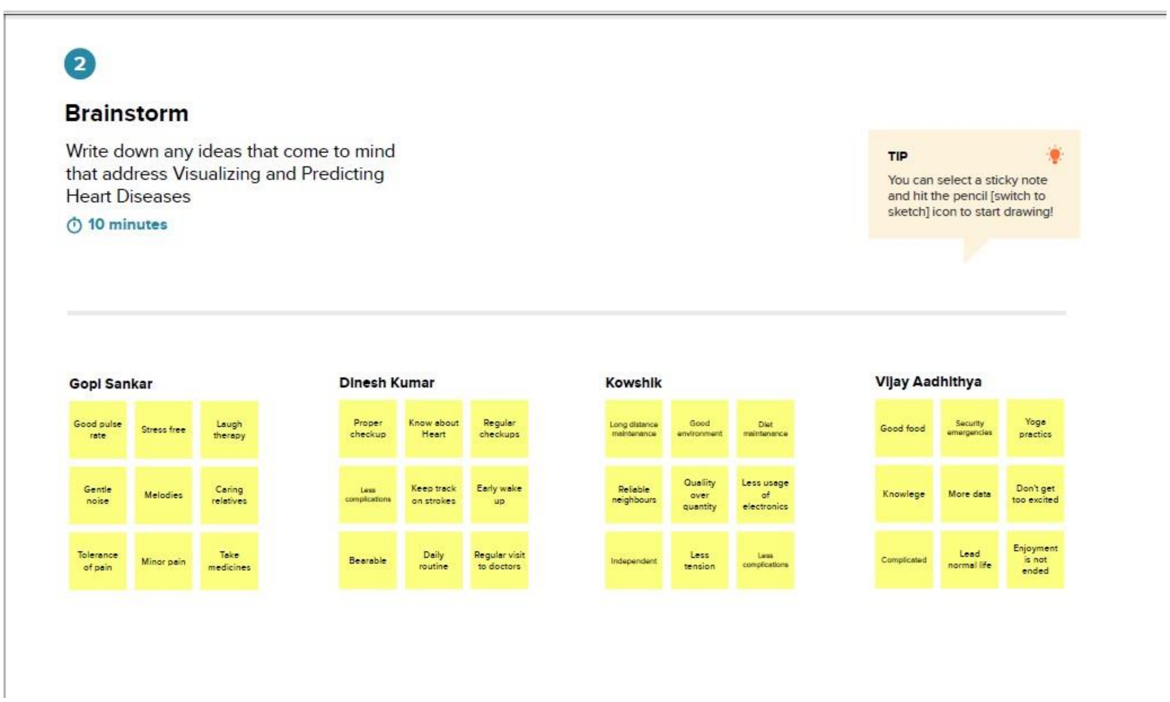
3.1 IDEATION & BRAINSTORMING:

STEP 1:




Before you collaborate
A little bit of preparation goes a long way with this session. Here's what you need to do to get going.
10 minutes
A
Team gathering
Define who should participate in the session and send an invite. Share relevant information or pre-work ahead.
B
Set the goal
Think about the problem you'll be focusing on solving in the brainstorming session.
C
Learn how to use the facilitation tools
Use the Facilitation Superpowers to run a happy and productive session.
Open article
1
Define your problem statement
Visualizing and Predicting Heart diseases with an interactive dashboard
5 minutes
PROBLEM
How might we Visualize and Predict heart disease with an interactive dashboard?
Key rules of brainstorming
To run an smooth and productive session
Stay in topic.
Encourage wild ideas.
Defer judgment.
Listen to others.
Go for volume.
If possible, be visual.

STEP 2:


2
Brainstorm
Write down any ideas that come to mind that address Visualizing and Predicting Heart Diseases
10 minutes
TIP
You can select a sticky note and hit the pencil [switch to sketch] icon to start drawing!
Gopi Sankar
Good pulse rate
Stress free
Laugh therapy
Gentle noise
Melodies
Caring relatives
Tolerance of pain
Minor pain
Take medicines
Dinesh Kumar
Proper checkup
Know about Heart
Regular checkups
Less complications
Keep track on strokes
Early wake up
Bearable
Daily routine
Regular visit to doctors
Kowshik
Long distance maintenance
Good environment
Diet maintenance
Reliable neighbours
Quality over quantity
Less usage of electronics
Independent
Less tension
Less complications
Vijay Aadhithya
Good food
Security emergencies
Yoga practics
Knowlege
More data
Don't get too excited
Complicated
Lead normal life
Enjoyment is not ended

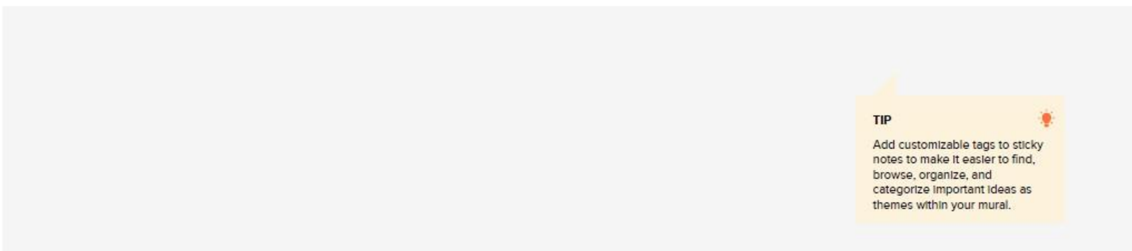
STEP 3:

3

Group ideas

Take turns sharing your ideas while clustering similar or related notes as you go. Once all sticky notes have been grouped, give each cluster a sentence-like label. If a cluster is bigger than six sticky notes, try and see if you and break it up into smaller sub-groups.

⏱ 20 minutes



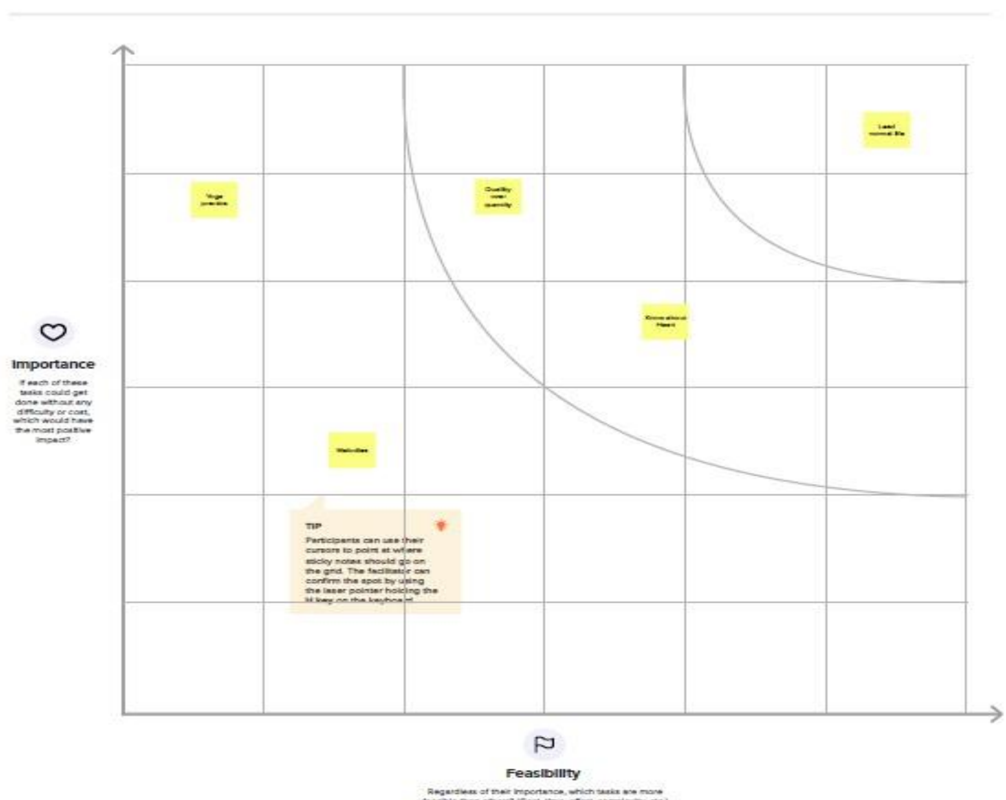
STEP 4:

4

Prioritize

Your team should all be on the same page about what's important moving forward. Place your ideas on this grid to determine which ideas are important and which are feasible.

⏱ 20 minutes



STEP 5:

After you collaborate

You can export the mural as an image or pdf to share with members of your company who might find it helpful.

Quick add-ons

A **Share the mural**
Share a view link to the mural with stakeholders to keep them in the loop about the outcomes of the session.

B **Export the mural**
Export a copy of the mural as a PNG or PDF to attach to emails, include in slides, or save in your drive.

Keep moving forward

Strategy blueprint
Define the components of a new idea or strategy.
Open the template →

Customer experience journey map
Understand customer needs, motivations, and obstacles for an experience.
Open the template →

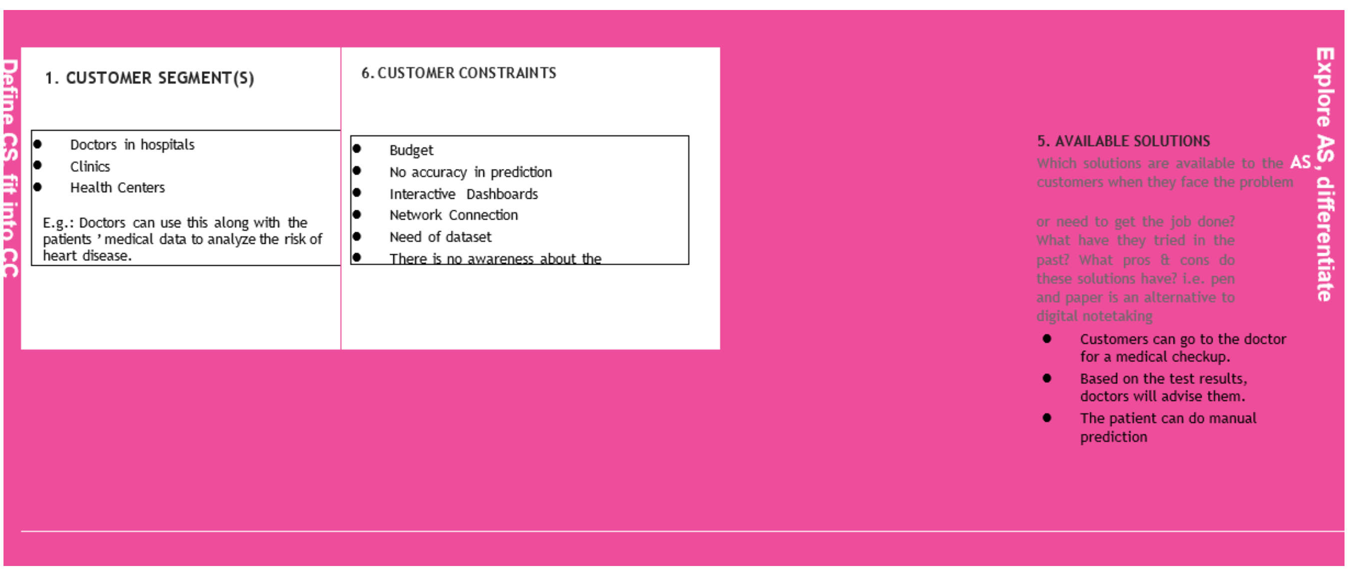
Strengths, weaknesses, opportunities & threats
Identify strengths, weaknesses, opportunities, and threats (SWOT) to develop a plan.
Open the template →

3.2 PROPOSED SOLUTION:

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	To analyse which patients are most likely to suffer from heart disease based on given parameters.It can provide visualization dashboards and uses this information to easily visualize and predict the patient details.
2.	Idea / Solution description	Parameters in data set helps hospitals to identify the patient heart condition and their health condition.A dashboard using cognitive analysis can be created to present the data and utilize it for future use
3.	Novelty / Uniqueness	Many tests are taken by doctors to detect presence of heart disease. The parameters used are often understood only by medical professional.Time can be saved.To provide a significant contribution in computing strength scores with significant predictors in Heart disease prediction.
4.	Social Impact / Customer Satisfaction	● Reduces the patient’s risk level ● Reduces the medical costs ● Save human lives ● Handy Interactive dashboard ● It will make the hospital to work efficiently ● It help the hospitals to know the health records of the heart patient

5.	Business Model (Revenue Model)	● Awareness can be created among the patients through ads ● Updates will be updated according to the necessity for the patients ● No complexity ● Data security ● This project can be converted to an software kit, webpage or even an application which users can interact with.
6.	Scalability of the Solution	☐ Machine learning ☐ Scalable dataset ☐ Adding new characteristics ☐ Easy prediction of the patient details with heart ☐ Disease ☐ Maintains best user experiences

3.3 PROBLEM SOLUTION FIT:



you address for your customers? There could be more than one; explore different sides.

- Visualizations give doctors very good insights on the potential chances for a patient to get heart disease.
- It is also very useful to explain to patients so that they can easily understand the risk factor and take care of themselves to reduce the likelihood of getting heart disease.
- Standard of Data: The outcome is fully depends on the accurate and relative dataset
- Visualizing and predicting heart disease

Tap into BE, understand RC

do this job?

i.e. customers have to do it because of the change in regulations.

- Not storing and analyzing data properly to help doctors make informed decisions
- Increasing in heart disease will not be identified firstly is major reason.
- There is a possibility of considering every heart disease as same
- There is no idea about relation between similar heart disease

i.e.directlyrelated:findtherightsolarpanel installer,calculateusageandbenefits;indirectly associated: customers spend free time on volunteering work (i.e.Greenpeace)

Ensure data is stored in organizedandsequential order an excel sheet for example r from the start so that is ready be used foranalysis.

The customer need accurate results For the various datasets

Tap into BE, understand RC

3. TRIGGERS TR

What triggers customers to act? i.e., seeing their neighbor installing solar panels, reading about a more efficient solution in the news.

Patients who have a history with heart disease orthosepatients whoare currently experiencing similar symptoms to those who have heart disease.
Similarity of heart disease is not identified

-
-

4. EMOTIONS: BEFORE/ AFTER EM

How do customers feel when they face a problem or a job and afterwards?

i.e.lost,insecure >confident,incontrol- use itinyour communication strategy & design.

Feeling afraid and depressed.
Develop a feeling of awareness which mean people
There is huge uncertainity in knowing the accurate and correct Reason for a disease and predicting it.

10. YOUR SOLUTION SL

Ifyou areworkingon anexistingbusiness,writedownyour currentsolutionfirst,fillinthecanvas,andcheckhowmuchit fits reality.

Ifyouareworkingonanewbusinessproposition,thenkeepit blankuntilyoufillinthecanvasandcomeupwithasolution that fits within customer limitations, solves a problem and matches customer behavior.

To clean data and provide visualizations to help doctors in their diagnosis of patient as well as make customers more aware ofthis issue.

8. CHANNELS of BEHAVIOUR CH

8.1 ONLINE

What kind of actions do customers take online? Extract online channels from #7

8.2 OFFLINE

Whatkindofactionsdocustomerstakeoffline?Extractoffline channelsfrom #7anduse them forcustomerdevelopment.

ONLINE:
Users look at the data and compare it with their testresults Upload data.Prepare data,Exploration of data.
OFFLINE: Doctors use it as a tool to diagnose patients and make accurate predictions.

4. REQUIREMENT ANALYSIS:

4.1 FUNCTIONAL REQUIREMENTS:

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through Facebook Registration through Gmail Registration through google
FR-2	Account creation	User fill Gmail and password for account creation
FR-3	User Confirmation	Confirmation via Email Confirmation via OTP
FR-4	Personal details for account	Apart from the basic details, user need to enter details such as name, age, sex, height, weight, previous medical records, etc
FR-5	Regular medical condition updation in app	Entry present medical records, symptoms, etc
FR-6	Doctor consultation	Expert doctor consultation through app

4.2 NON-FUNCTIONAL REQUIREMENTS:

Following are the non-functional requirements of the proposed solution.

FR No.	Non-Functional Requirement	Description
NFR-1	Usability	As usability is a prerequisite for success of health and wellness mobile apps, our proposed solution aims to provide insights and suggestions for improving usability experience of the mobile health app by exploring the degree of alignment between app insiders and users.
NFR-2	Security	Our proposed solution can empower patients, streamline communication, and provide real-time monitoring and self-management of medical conditions by building a secure app that puts security, privacy and compliance by considering authentication, privilege management, secure data storage and communication, compliance and testing and installation.

NFR-3	Reliability	Measuring reliability can improve the quality and value of health care apps. Our proposed solution will provide accurate prediction of disease with a lower risk of errors that cause harm to user and reduces the death rate. Our solution provides Safety to user's data with lot of benefits simply in home which is Efficient without wasting equipment, supplies, ideas, and energy.
NFR-4	Performance	The performance of this project is to reduce heart disease death rate by earlier accurate disease prediction. Our solution offers services such as disease prevention, diagnosis and treatment, and rehabilitation.
NFR-5	Availability	Availability is important because, while there are often shortages in human resources, deployed providers are frequently inappropriately absent or, when present, are not actively delivering health care because they are engaged in other duties. Our proposed solution provides immediate access to care anytime anywhere
NFR-6	Scalability	It can be integrated with smart watch and apps for further advancements which is very helpful for earlier prediction. And further, we can provide live doctor consultancy, keep up the old data records for increasing accurate prediction and advices to prevent heart disease. Notifies alerts to nearby hospital when person is at risk

5. PROJECT DESIGN:

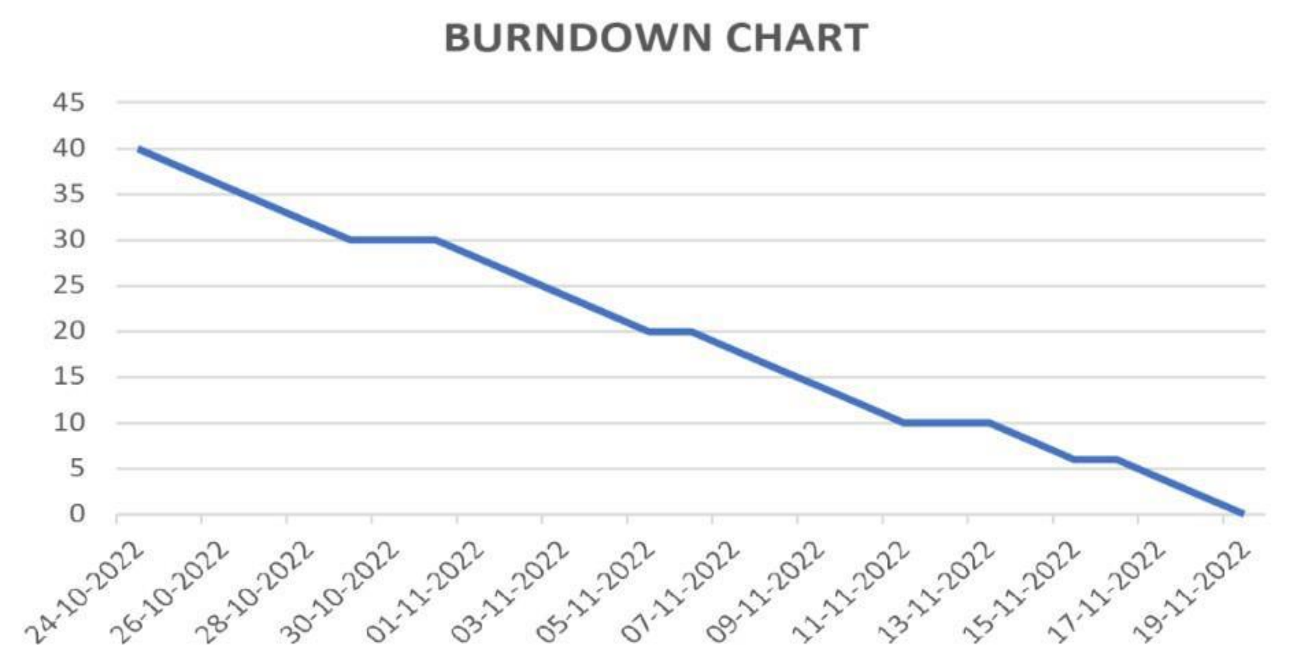
5.1 TECHNICAL ARCHITECTURE:

6. PROJECT PLANNING AND SCHEDULING:

6.1. SPRINT PLANNING &ESTIMATION:

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	2	High	1
Sprint-1		USN-2	As a user, I will receive confirmation email once I have registered for the application	1	High	2
Sprint-2		USN-3	As a user, I can register for the application through Facebook	2	Low	4
Sprint-1		USN-4	As a user, I can register for the application through Gmail	2	Medium	3
Sprint-1	Login	USN-5	As a user, I can log into the application by entering email & password	1	High	2
Sprint-2	Dashboard	USN-6	Able to view only his medical records	2	High	4
Sprint-2		USN-7	View the possibilities of occurrence of heart disease	1	High	2
Sprint-3	Helpdesk	USN-8	Admin be able to view queries	2	High	4
Sprint-3		USN-9	Admin be able to answer queries	2	High	4
Sprint-4	User Profile	USN-10	Able to update users medical records	1	Medium	4
Sprint-4		USN-11	Able to add/ Delete users	2	High	2
Sprint-4		USN-12	Able to view/ organize the user details	1	High	2

6.2 REPORTS FROM JIRA:



7.CODING & SOLUTIONING:

7.1 FEATURE 1:

The proposed system is the machine learning model where we could able to predict the quality of the water from giving the necessary details regarding the water body. This part deals with creating a model from the random forest algorithm. With the dataset we will be finding out the water quality index and using that we split the data into the training and testing set. Then the model will be created using the splitted data. After the model is created the accuracy of the model will be determined and model is deployed in the pickle. There is also another method to deploy a model using the IBM cloud.

The below code is the model created from the random forest algorithm,

```
import numpy as np
import pickle
import sklearn
from flask import Flask, render_template, request, redirect, url_for, flash
import sqlite3
model = pickle.load(open('models.pkl', 'rb'))
app = Flask(__name__)
@app.route('/')
def index():
    return render_template('Heart_Disease_Classifier.html', title='Home')
@app.route('/Heart_Disease_Classifier')
def Heart_Disease_Classifier():
       return render_template('Heart_Disease_Classifier.html')
@app.route('/predict', methods =['POST'])
def predict():
  features = [float(i) for i in request.form.values()]
  #Convert features to array
```

```
  array_features = [np.array(features)]
  #Predict features
  prediction = model.predict(array_features)
  output = prediction
  if output == 1:
    return render_template('Heart_Disease_Classifier.html', result = 'The patient is
not likely to have heart disease!')
  else:
    return render_template('Heart_Disease_Classifier.html', result = 'The patient is
likely to have heart disease!')
if __name__ == '__main__':
   app.run(debug=True,port=5100)
```

7.2 FEATURE 2:

The model is deployed in the IBM cloud with the following code.

```
from ibm_watson_machine_learning import APIClient
```

```
<html>
<head>
<!-- Bootstrap CSS -->
<link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css"
integrity="sha384-
JcKb8q3iqJ61gNV9KGb8thSsNjpSL0n8PARn9HuZOnIxN0hoP+VmmDGMN5t9
UJ0Z" crossorigin="anonymous">
<script src="https://code.jquery.com/jquery-3.5.1.slim.min.js" integrity="sha384-
DfXdz2htPH0lsSSs5nCTpuj/zy4C+OGpamoFVy38MVBnE+IbbVYUew+OrCXa
Rkfj" crossorigin="anonymous"></script>
<script
src="https://cdn.jsdelivr.net/npm/popper.js@1.16.1/dist/umd/popper.min.js"
integrity="sha384-
9/reFTGAW83EW2RDu2S0VKaIzap3H66lZH81PoYlFhbGU+6BZp6G7niu735S
k7lN" crossorigin="anonymous"></script>
<script
src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"
```

```
integrity="sha384-
B4gt1jrGC7Jh4AgTPSdUtOBvfO8shuf57BaghqFfPlYxofvL8/KUEfYiJOMMV+r
V" crossorigin="anonymous"></script>
<title>Heart Disease Test</title>
</head>
<body>
<!-- Java Script -->
<script src="https://code.jquery.com/jquery-3.5.1.slim.min.js" integrity="sha384-
DfXdz2htPH0lsSSs5nCTpuj/zy4C+OGpamoFVy38MVBnE+IbbVYUew+OrCXa
Rkfj" crossorigin="anonymous"></script>
<script
src="https://cdn.jsdelivr.net/npm/popper.js@1.16.1/dist/umd/popper.min.js"
integrity="sha384-
9/reFTGAW83EW2RDu2S0VKaIzap3H66lZH81PoYlFhbGU+6BZp6G7niu735S
k7lN" crossorigin="anonymous"></script>
<script
src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"
integrity="sha384-
B4gt1jrGC7Jh4AgTPSdUtOBvfO8shuf57BaghqFfPlYxofvL8/KUEfYiJOMMV+r
V" crossorigin="anonymous"></script>


<!-- Navbar-->
<nav class="navbar navbar-dark" style="background-color: rgb(13, 102, 87);">
<span class="navbar-brand mb-0 h1">Heart Disease Test</span>
</nav>
<div class="container">
<br>
<!--Form-->
<form action = "{{url_for('predict')}}" method ="POST" >
<fieldset>
<legend style="color: rgb(41, 15, 134);"><b>Heart Disease Test
Form</b></legend><br>
<div class="card card-body" style="background-color: rgb(194 245 236 / 56%);">
<div class="form-group  row">
```

```
<div class="col-sm-3">
<label for="age">Age</label>
<input type="number" class="form-control" id="age" name="age" required>
</div>
<div class="col-sm-3">
<label for="sex">Sex</label>
<select class="form-control" id="sex"  name="sex" required>
<option disabled selected value> -- Select an Option -- </option>
<option value = "0">Female</option>
<option value = "1">Male</option>
</select>
</div>
</div>
<br>
<div class="form-group  row">
<div class="col-sm">
<label for="cp">Chest Pain Type</label>
<select class="form-control" id="cp" name = "cp" required>
<option disabled selected value> -- Select an Option -- </option>
<option value = "1">Typical Angina</option>
<option value = "2">Atypical Angina</option>
<option value = "3">Non-anginal Pain</option>
<option value = "4">Asymptomatic</option>
</select>
</div>
<div class="col-sm">
<label for="trestbps">Resting Blood Pressure in mm Hg</label>
<input type="number" class="form-control" id="trestbps" name="trestbps"
required>
</div>
<div class="col-sm">
<label for="chol">Serum Cholestoral in mg/dl</label>
<input type="number" class="form-control" id="chol" name="chol" required>
</div>
<div class="col-sm">
```

```html
<label for="fbs">Fasting Blood Sugar > 120 mg/dl</label>
<select class="form-control" id="fbs" name="fbs" required>
<option disabled selected value> -- Select an Option -- </option>
<option value = "0">False</option>
<option value = "1">True</option>
</select>
</div>
</div>

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

```
<input type="number" step="any" class="form-control" id="oldpeak"
name="oldpeak" required>
</div>
</div>
<br>
<div class="form-group  row">
<div class="col-sm">
<label for="slope">Slope of the Peak Exercise ST Segment </label>
<select class="form-control" id="slope" name="slope" required>
<option disabled selected value> -- Select an Option -- </option>
<option value = "1">Upsloping</option>
<option value = "2">Flat</option>
<option value = "3">Downsloping</option>
</select>
</div>
<div class="col-sm">
<label for="ca">Number of Vessels Colored by Flourosopy</label>
<select class="form-control" id="ca" name = "ca" required>
<option disabled selected value> -- Select an Option -- </option>
<option value = "0">0</option>
<option value = "1">1</option>
<option value = "2">2</option>
<option value = "3">3</option>
</select>
</div>
<div class="col-sm">
<label for="thal">Thalassemia</label>
<select class="form-control" id="thal" name = "thal" required>
<option disabled selected value> -- Select an Option -- </option>
<option value = "3">Normal</option>
<option value = "6">Fixed defect</option>
<option value = "7">Reversable defect</option>
</select>
</div>
</div>
```

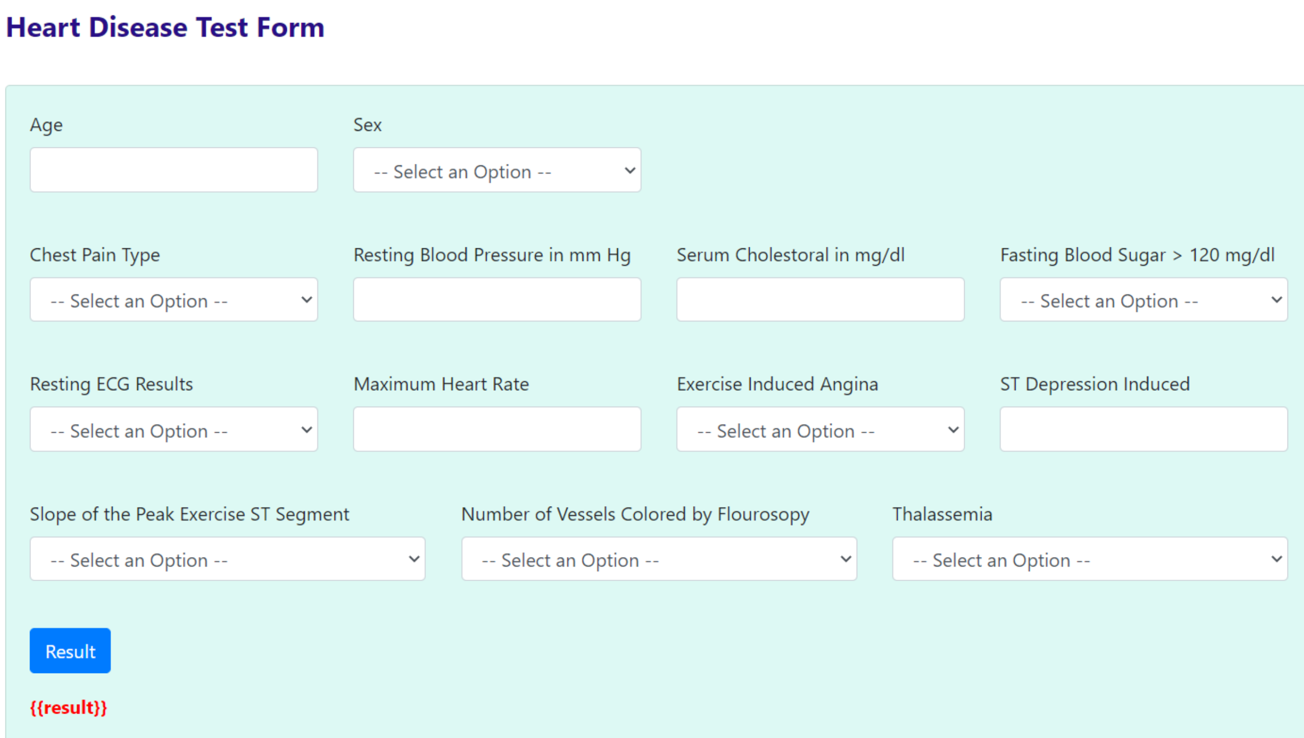
```
<br>
<div class="form-group">
<input class="btn btn-primary" type="submit" value="Result">
</div>

<!--Prediction Result-->
<div id ="result">
<strong style="color:red">{{result}}</strong>
</div>
</div>
</fieldset>
</form>
</div>
</body>
</html>
```

8. RESULTS:

8.1. PERFORMANCE METRICS:

The accuracy score achieved using Random Forest Classifier is: 85.19 %
The accuracy score achieved using K-Nearest Neighbors Classifier is: 79.63 %
The accuracy score achieved using Navie Bayes Classifier is: 66.67 %
The accuracy score achieved using Decision Tree Classifier is: 70.37 %



9. CONCLUSION AND FUTURE SCOPE:

Heart diseases are a major killer in India and throughout the world, application of promising technology like machine learning to the initial prediction of heart diseases will have a profound impact on society. The early prognosis of heart disease can aid in making decisions on lifestyle changes in high-risk patients and in turn reduce the complications, which can be a great milestone in the field of medicine. The number of people facing heart diseases is on a raise each year. This prompts for its early diagnosis and treatment. The utilization of suitable technology support in this regard can prove to be highly beneficial to the medical fraternity and patients. In this paper, the seven different machine learning algorithms used to measure the performance are SVM, Decision Tree, Random Forest, Naïve Bayes, Logistic Regression, Adaptive Boosting, and Extreme Gradient Boosting applied on the dataset. The expected attributes leading to heart disease in patients are available in the dataset which contains 76 features and 14 important features that are useful to evaluate the system are selected among them. If all the features taken into the consideration then the efficiency of the system the author gets is less. To increase efficiency, attribute selection is done. In this n features have to be selected for evaluating the model which gives more accuracy. The correlation of some features in the dataset is almost equal and so they are removed. If all the attributes present in the dataset are taken into account then the efficiency decreases considerably. All the seven machine learning methods accuracies are compared based on which one prediction model is generated. Hence, the aim is to use various evaluation metrics like confusion matrix, accuracy, precision, recall, and f1-score which predicts the disease efficiently. Comparing all seven the extreme gradient boosting classifier gives the highest accuracy of 81%.

10. APPENDIX:

SOURCE CODE:

HEART DISEASE PREDICTION.ipynb:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from matplotlib import rcParams
from matplotlib.cm import rainbow
import seaborn as sns
%matplotlib inline
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.preprocessing import LabelEncoder
from sklearn import tree
from warnings import filterwarnings
filterwarnings("ignore")
#model validation
from sklearn.metrics import
log_loss,roc_auc_score,precision_score,f1_score,recall_score,roc_curve,auc,plot_r
oc_curve
from sklearn.metrics import classification_report,
confusion_matrix,accuracy_score,fbeta_score,matthews_corrcoef
from sklearn import metrics
#extra
from sklearn.pipeline import make_pipeline, make_union
from sklearn.preprocessing import PolynomialFeatures
from sklearn.feature_selection import SelectFwe, f_regression
from sklearn.ensemble import RandomForestClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.naive_bayes import GaussianNB
dataset =
pd.read_csv('DataSet\Heart_Disease_Prediction.csv',sep=',',encoding="utf-8")
type(dataset)
pandas.core.frame.DataFrame
dataset.shape
```

```
(270, 14)
dataset.info()
dataset.columns
Index(['age', 'sex', 'cp', 'trestbps', 'chol', 'fbs', 'restecg', 'thalach',
       'exang', 'oldpeak', 'slope', 'ca', 'thal', 'target'],
      dtype='object')
dataset.describe()
print('cp ',dataset['cp'].unique())
print('fbs ',dataset['fbs'].unique())
print('restecg ',dataset['restecg'].unique())
print('exang ',dataset['exang'].unique())
print('slope ',dataset['slope'].unique())
print('ca ',dataset['ca'].unique())
print('thal ',dataset['thal'].unique())
fig, (ax1) = plt.subplots(nrows=1, ncols=1, sharey=False, figsize=(14,6))

ax1 = dataset['target'].value_counts().plot.pie( x="Heart disease" ,y ='no.of
patients',
                   autopct = "%1.0f%%",labels=["Heart Disease","Normal"], startangle =
60,ax=ax1);
ax1.set(title = 'Percentage of Heart disease patients in Dataset')
plt.show()
y = dataset["target"]
rcParams['figure.figsize'] = 8,6
plt.bar(dataset['target'].unique(), dataset['target'].value_counts(), color = ['blue',
'green'])
plt.xticks([1, 2])
plt.xlabel('Target Classes (1 =no disease; 2 = disease)')
plt.ylabel('Samples')
plt.title('Count of each Target Class')
target_temp = dataset.target.value_counts()
print(target_temp)
dataset.hist(edgecolor='black',layout = (7, 2),
            figsize = (10, 30),
            color=['purple'])
dataset["sex"].unique()
array([1, 0], dtype=int64)
# Number of males and females
```

```
F = dataset[dataset["sex"] == 0].count()["target"]
M = dataset[dataset["sex"] == 1].count()["target"]

# Create a plot
figure, ax = plt.subplots(figsize = (6, 4))
ax.bar(x = ['Female', 'Male'], height = [F, M])
plt.xlabel('Gender')
plt.title('Number of Males and Females in the dataset')
plt.show()
pd.crosstab(dataset.sex,dataset.target).plot(kind="bar",figsize=(20,10),color=['blue'
,'#AA1111' ])
plt.title('Heart Disease Frequency for Sex')
plt.xlabel('Sex (0 = Female, 1 = Male)')
plt.xticks(rotation=0)
plt.legend(["Don't have Disease", "Have Disease"])
plt.ylabel('Frequency')
plt.show()
countFemale = len(dataset[dataset.sex == 0])
countMale = len(dataset[dataset.sex == 1])
print("Percentage of Female
Patients:{:.2f}%".format((countFemale)/(len(dataset.sex))*100))
print("Percentage of Male
Patients:{:.2f}%".format((countMale)/(len(dataset.sex))*100))
# Display age distribution based on heart disease
sns.distplot(dataset[dataset['target'] == 1]['age'], label='Do not have heart disease')
sns.distplot(dataset[dataset['target'] == 2]['age'], label = 'Have heart disease')
plt.xlabel('Frequency')
plt.ylabel('Age')
plt.title('Age Distribution based on Heart Disease')
plt.legend()
plt.show()
print('Min age of people who do not have heart disease: ',
min(dataset[dataset['target'] == 1]['age']))
print('Max age of people who do not have heart disease: ',
max(dataset[dataset['target'] == 1]['age']))
print('Average age of people who do not have heart disease: ',
dataset[dataset['target'] == 1]['age'].mean())
print('Min age of people who have heart disease: ', min(dataset[dataset['target'] ==
2]['age']))
```

```
print('Max age of people who have heart disease: ', max(dataset[dataset['target'] ==
2]['age']))
print('Average age of people who have heart disease: ', dataset[dataset['target'] ==
2]['age'].mean())
pd.crosstab(dataset.age,dataset.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()
plt.figure(figsize=(12, 10))
dataset.age.hist(bins=80)
print(f"The most of the patients have a mean age of : {dataset.age.mean()}")
categorial = [('sex', ['female', 'male']),
              ('cp', ['typical angina', 'atypical angina', 'non-anginal pain',
'asymptomatic']),
              ('fbs', ['fbs > 120mg', 'fbs < 120mg']),
              ('restecg', ['normal', 'ST-T wave', 'left ventricular']),
              ('exang', ['yes', 'no']),
              ('slope', ['upsloping', 'flat', 'downsloping']),
              ('thal', ['normal', 'fixed defect', 'reversible defect'])]
def plotGrid(isCategorial):
    if isCategorial:
        [plotCategorial(x[0], x[1], i) for i, x in enumerate(categorial)]
    else:
        [plotContinuous(x[0], x[1], i) for i, x in enumerate(continuous)]
def plotCategorial(attribute, labels, ax_index):
    sns.countplot(x=attribute, data=dataset, ax=axes[ax_index][0])
    sns.countplot(x='target', hue=attribute, data=dataset, ax=axes[ax_index][1])
    avg = dataset[[attribute, 'target']].groupby([attribute], as_index=False).mean()
    sns.barplot(x=attribute, y='target', hue=attribute, data=avg,
ax=axes[ax_index][2])

    for t, l in zip(axes[ax_index][1].get_legend().texts, labels):
        t.set_text(l)
    for t, l in zip(axes[ax_index][2].get_legend().texts, labels):
        t.set_text(l)
fig_categorial, axes = plt.subplots(nrows=len(categorial), ncols=3, figsize=(15,
30))
```

```
plotGrid(True)
continuous = [('trestbps', 'blood pressure in mm Hg'),
              ('chol', 'serum cholestoral in mg/d'),
              ('thalach', 'maximum heart rate achieved'),
              ('oldpeak', 'ST depression by exercise relative to rest'),
              ('ca', '# major vessels: (0-3) colored by flourosopy')]
def plotContinuous(attribute, xlabel, ax_index):
    sns.distplot(dataset[[attribute]], ax=axes[ax_index][0])
    axes[ax_index][0].set(xlabel=xlabel, ylabel='density')
    sns.violinplot(x='target', y=attribute, data=dataset, ax=axes[ax_index][1])
fig_continuous, axes = plt.subplots(nrows=len(continuous), ncols=2, figsize=(15,
22))

plotGrid(isCategorial=False)
fig, ax = plt.subplots(4,2, figsize = (14,14))
((ax1, ax2), (ax3, ax4), (ax5, ax6), (ax7, ax8)) = ax

labels = ["Male", "Female"]
values = dataset['sex'].value_counts().tolist()[:2]
ax1.pie(x=values, labels=labels,
autopct="%1.1f%%",colors=['#AAb3ff','#CC80FF'],shadow=True,
startangle=45,explode=[0.1, 0.1])
ax1.set_title("Sex", fontdict={'fontsize': 12},fontweight ='bold')

labels = ["Typical angina", "Atypical angina","non-anginal pain","asymptomatic"]
values = dataset['cp'].value_counts().tolist()
ax2.pie(x=values, labels=labels,
autopct="%1.1f%%",colors=['#AAb3ff','#CC80FF','#DD00AA','#FF0099'],shadow
=True,startangle=45,explode=[0.1, 0.1, 0.1, 0.2])
ax2.set_title("Chest Pain", fontdict={'fontsize': 12},fontweight ='bold')

labels = dataset['fbs'].value_counts().index.tolist()[:2]
values = dataset['fbs'].value_counts().tolist()
ax3.pie(x=values, labels=labels,
autopct="%1.1f%%",colors=['#AAb3ff','#CC80FF'],shadow=True,
startangle=45,explode=[0.1, 0.15])
ax3.set_title("Fasting Blood Sugar", fontdict={'fontsize': 12},fontweight ='bold')
```

```
labels = dataset['restecg'].value_counts().index.tolist()[:3]
values = dataset['restecg'].value_counts().tolist()
ax4.pie(x=values, labels=labels, autopct="%1.1f%%",
colors=['#AAb3ff','#CC80FF','#DD00AA'],shadow=True,startangle=45,explode=[
0.05, 0.05, 0.05])
ax4.set_title("Resting Blood Pressure", fontdict={'fontsize': 12},fontweight
='bold')

labels = dataset['exang'].value_counts().index.tolist()[:2]
values = dataset['exang'].value_counts().tolist()
ax5.pie(x=values, labels=labels, autopct="%1.1f%%",
colors=['#AAb3ff','#CC80FF'],shadow=True, startangle=45,explode=[0.1, 0.1])
ax5.set_title("Exercise induced Angina", fontdict={'fontsize': 12},fontweight
='bold')

labels = dataset['slope'].value_counts().index.tolist()[:3]
values = dataset['slope'].value_counts().tolist()
ax6.pie(x=values, labels=labels, autopct="%1.1f%%",
colors=['#AAb3ff','#CC80FF','#DD00AA'],shadow=True,startangle=45,explode=[
0.1, 0.1, 0.1])
ax6.set_title("Peak exercise ST_segment Slope", fontdict={'fontsize':
12},fontweight ='bold')

labels = dataset['ca'].value_counts().index.tolist()[:4]
values = dataset['ca'].value_counts().tolist()
ax7.pie(x=values, labels=labels, autopct="%1.1f%%", shadow=True,
startangle=45,explode=[0.05, 0.07, 0.1,
0.1],colors=['#AAb3ff','#CC80FF','#DD00AA','#FF0099'])
ax7.set_title("Major vessels", fontdict={'fontsize': 12},fontweight ='bold')

labels = dataset['thal'].value_counts().index.tolist()[:3]
values = dataset['thal'].value_counts().tolist()
ax8.pie(x=values, labels=labels, autopct="%1.1f%%", shadow=True,
startangle=45,explode=[0.1, 0.1, 0.1],colors=['#AAb3ff','#CC80FF','#DD00AA'])
ax8.set_title("Types of Thalassemia", fontdict={'fontsize': 12},fontweight ='bold')

plt.tight_layout()
plt.show()

plt.savefig("PiePlots.png")
```

```
plt.figure(figsize=(10,10))
sns.heatmap(pd.DataFrame(dataset.corr()['target']).sort_values(by='target').transpo
se().drop('target',axis=1).transpose(),annot=True,cmap='twilight')
plt.savefig("TargetCorrelations.png")
X = dataset.drop('target',axis=1)
Y = dataset['target']
from sklearn.feature_selection import SelectKBest, chi2
fs = SelectKBest(score_func=chi2, k='all')
fs.fit(X, Y)
per = []
for i in fs.scores_:
    per.append(round(((i/sum(fs.scores_))*100),3))

features_data = pd.DataFrame({'Feature':X.columns,'Scores':fs.scores_,'Importance
(%)':per}).sort_values(by=['Scores'],ascending=False)

plt.figure(figsize=(9,4))
sns.barplot( 'Importance
(%)','Feature',orient='h',data=features_data,palette='twilight_shifted_r')
insignificant = features_data.loc[features_data['Importance
(%)']<0.005]['Feature'].unique()
features_data = features_data.set_index('Feature')
features_data
plt.savefig("FeatureImportance.png")
# Display fasting blood sugar in bar chart
dataset.groupby(dataset['fbs']).count()['target'].plot(kind = 'bar', title = 'Fasting
Blood Sugar', figsize = (8, 6))
plt.xticks(np.arange(2), ('fbs < 120 mg/dl', 'fbs > 120 mg/dl'), rotation = 0)
plt.show()
pd.crosstab(dataset.fbs,dataset.target).plot(kind = "bar", figsize = (8, 6))
plt.title('Heart Disease Frequency According to Fasting Blood Sugar')
plt.xlabel('Fasting Blood Sugar')
plt.xticks(np.arange(2), ('fbs < 120 mg/dl', 'fbs > 120 mg/dl'), rotation = 0)
plt.ylabel('Frequency')
plt.show()
dataset["cp"].unique()
array([4, 3, 2, 1], dtype=int64)
plt.figure(figsize=(26, 10))
sns.barplot(dataset["cp"],y)
```

```
pd.crosstab(dataset.cp,dataset.target).plot(kind = "bar", figsize = (8, 6))
plt.title('Heart Disease Frequency According to Chest Pain Type')
plt.xlabel('Chest Pain Type')
plt.xticks(np.arange(4), ('typical angina', 'atypical angina', 'non-anginal pain',
'asymptomatic'), rotation = 0)
plt.ylabel('Frequency')
plt.show()
dataset["trestbps"].unique()
plt.figure(figsize=(26, 10))
sns.barplot(dataset["trestbps"],y)
fig, (axis1, axis2) = plt.subplots(1, 2,figsize=(25, 5))
ax = sns.distplot(dataset[dataset['target'] == 1]['trestbps'], label='Do not have heart
disease', ax = axis1)
ax.set(xlabel='People Do Not Have Heart Disease')
ax = sns.distplot(dataset[dataset['target'] == 2]['trestbps'], label = 'Have heart
disease', ax = axis2)
ax.set(xlabel='People Have Heart Disease')
plt.show()
# Get min, max and average of the  blood pressure of the people do not have heart
diseas
print('Min blood pressure of people who do not have heart disease: ',
min(dataset[dataset['target'] == 1]['trestbps']))
print('Max blood pressure of people who do not have heart disease: ',
max(dataset[dataset['target'] == 1]['trestbps']))
print('Average blood pressure of people who do not have heart disease: ',
dataset[dataset['target'] == 1]['trestbps'].mean())
```

```
Min blood pressure of people who do not have heart disease:  94
Max blood pressure of people who do not have heart disease:  180
Average blood pressure of people who do not have heart disease:  128.8666666666
6667
```

```
# Get min, max and average of the blood pressure of the people have heart diseas
print('Min blood pressure of people who have heart disease: ',
min(dataset[dataset['target'] == 2]['trestbps']))
print('Max blood pressure of people who have heart disease: ',
max(dataset[dataset['target'] == 2]['trestbps']))
print('Average blood pressure of people who have heart disease: ',
dataset[dataset['target'] == 2]['trestbps'].mean())
```

```
Min blood pressure of people who have heart disease:  100
Max blood pressure of people who have heart disease:  200
```

Average blood pressure of people who have heart disease: 134.44166666666666

```
dataset["restecg"].unique()
```

array([2, 0, 1], dtype=int64)

```
# Display electrocardiographic results in bar chart
dataset.groupby(dataset['restecg']).count()['target'].plot(kind = 'bar', title = 'Resting Electrocardiographic Results', figsize = (8, 6))
plt.xticks(np.arange(3), ('normal', 'ST-T wave abnormality', 'probable or left ventricular hypertrophy'))
plt.show()
```

```
# Display resting electrocardiographic results based on the target
pd.crosstab(dataset.restecg,dataset.target).plot(kind = "bar", figsize = (8, 6))
plt.title('Heart Disease Frequency According to Resting Electrocardiographic Results')
plt.xticks(np.arange(3), ('normal', 'ST-T wave abnormality', 'probable or left ventricular hypertrophy'))
plt.xlabel('Resting Electrocardiographic Results')
plt.ylabel('Frequency')
plt.show()
dataset["exang"].unique()
# Display exercise induced angina in bar chart
dataset.groupby(dataset['exang']).count()['target'].plot(kind = 'bar', title = 'Exercise Induced Angina',  figsize = (8, 6))
plt.xticks(np.arange(2), ('No', 'Yes'), rotation = 0)
plt.show()
pd.crosstab(dataset.exang,dataset.target).plot(kind = "bar", figsize = (8, 6))
plt.title('Heart Disease Frequency According to Exercise Induced Angina')
plt.xlabel('Exercise Induced Angina')
plt.xticks(np.arange(2), ('No', 'Yes'), rotation = 0)
plt.ylabel('Frequency')
plt.show()
dataset["slope"].unique()
# Display slope of the peak exercise ST segment in bar chart
dataset.groupby(dataset['slope']).count()['target'].plot(kind = 'bar', title = 'Slope of the Peak Exercise ST Segment', figsize = (8, 6))
plt.xticks(np.arange(3), ('upsloping', 'flat', 'downsloping'), rotation = 0)
plt.show()
pd.crosstab(dataset.slope,dataset.target).plot(kind = "bar", figsize = (8, 6))
plt.title('Heart Disease Frequency According to Slope of the Peak Exercise ST Segment')
```

```
plt.xlabel('Slope')
plt.xticks(np.arange(3), ('upsloping', 'flat', 'downsloping'), rotation = 0)
plt.ylabel('Frequency')
plt.show()
dataset["ca"].unique()
dataset.groupby(dataset['ca']).count()['target'].plot(kind = 'bar', title = 'Number of
Major Vessels Colored by Flourosopy',
                                  figsize = (8, 6))
plt.show()
pd.crosstab(dataset.ca,dataset.target).plot(kind = "bar", figsize = (8, 6))
plt.title('Heart Disease Frequency According to Number of Major Vessels Colored
by Flourosopy')
plt.xlabel('number of vessels')
plt.xticks(rotation = 0)
plt.ylabel('Frequency')
plt.show()
dataset["thal"].unique()
sns.distplot(dataset["thal"])
dataset.groupby(dataset['thal']).count()['target'].plot(kind = 'bar', title =
'Thalassemia')
plt.xticks(np.arange(3), ('normal', 'fixed defect', 'reversible defect'), rotation = 0)
plt.show()
pd.crosstab(dataset.thal,dataset.target).plot(kind = "bar", figsize = (8, 6))
plt.title('Heart Disease Frequency According to Thalassemia')
plt.xlabel('Thalassemia')
plt.xticks(np.arange(3), ('normal', 'fixed defect', 'reversible defect'), rotation = 0)
plt.ylabel('Frequency')
plt.show()
plt.figure(figsize=(20,10))
sns.scatterplot(x='chol',y='thal',data=dataset,hue='target')
plt.show()
plt.figure(figsize=(20,10))
sns.scatterplot(x='thal',y='trestbps',data=dataset,hue='target')
plt.show()
plt.figure(figsize=(20, 10))
plt.scatter(x=dataset.age[dataset.target==2], y=dataset.thal[(dataset.target==2)],
c="green")
plt.scatter(x=dataset.age[dataset.target==1], y=dataset.thal[(dataset.target==1)])
plt.legend(["Disease", "Not Disease"])
plt.xlabel("Age")
```

```
plt.ylabel("Maximum Heart Rate")
plt.show()
plt.figure(figsize=(15,5))
plt.subplot(1,2,1)
sns.histplot(data=dataset,hue='target',x='thalach',bins=20,element='poly')
plt.subplot(1,2,2)
sns.histplot(data=dataset,hue='target',x='oldpeak',bins=20,element='poly')
plt.savefig("Thalach&oldpeak_Histplot.png")
plt.figure(figsize=(15,5))
plt.subplot(1,2,1)
sns.histplot(data=dataset,hue='target',x='trestbps',bins=20,element='poly')
plt.subplot(1,2,2)
sns.histplot(data=dataset,hue='target',x='chol',bins=20,element='poly')
plt.savefig("Resting_blood_pressure&chol_Histplot.png")
sns.pairplot(data=dataset)
corr_matrix = dataset.corr()
top_corr_feature = corr_matrix.index
plt.figure(figsize=(20, 20))
sns.heatmap(dataset[top_corr_feature].corr(), annot=True, cmap="RdYlGn",
annot_kws={"size":15})
dataset = pd.get_dummies(dataset, columns = ['sex', 'cp', 'fbs', 'restecg', 'exang',
'slope', 'ca', 'thal'])
standardScaler = StandardScaler()
columns_to_scale = ['age', 'trestbps', 'chol', 'thalach', 'oldpeak']
dataset[columns_to_scale] =
standardScaler.fit_transform(dataset[columns_to_scale])
dataset.head()
dataset.describe()
Y = dataset['target'].values
X = dataset.drop('target',axis=1).values
X.shape
(270, 28)
Y.shape
(270,)

# Split the dataset into training and testing.
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.2,
random_state=0)
```

```
print("Training features have {0} records and Testing features have {1} records.".\
    format(X_train.shape[0], X_test.shape[0]))
print('--------------Training Set----------------')
print(X_train.shape)
print(Y_train.shape)
print('----------------Test Set-----------------')
print(X_test.shape)
print(Y_test.shape)
max_accuracy = 0

for x in range(500):
    rf_classifier = RandomForestClassifier(random_state=x)
    rf_classifier.fit(X_train,Y_train)
    Y_pred_rf = rf_classifier.predict(X_test)
    current_accuracy = round(accuracy_score(Y_pred_rf,Y_test)*100,2)
    if(current_accuracy>max_accuracy):
        max_accuracy = current_accuracy
        best_x = x

print(max_accuracy)
print(best_x)
rf_classifier = RandomForestClassifier(random_state=best_x)
rf_classifier.fit(X_train,Y_train)
Y_pred_rf = rf_classifier.predict(X_test)
Y_pred_rf.shape
score_rf = round(accuracy_score(Y_pred_rf,Y_test)*100,2)
score_rf
y_pred_rfe = rf_classifier.predict(X_test)

plt.figure(figsize=(10, 8))
CM=confusion_matrix(Y_test,y_pred_rfe)
sns.heatmap(CM, annot=True)

TN = CM[0][0]
FN = CM[1][0]
TP = CM[1][1]
FP = CM[0][1]
specificity = TN/(TN+FP)
loss_log = log_loss(Y_test, y_pred_rfe)
```

```
acc= accuracy_score(Y_test, y_pred_rfe)
roc=roc_auc_score(Y_test, y_pred_rfe)
prec = precision_score(Y_test, y_pred_rfe)
rec = recall_score(Y_test, y_pred_rfe)
f1 = f1_score(Y_test, y_pred_rfe)

mathew = matthews_corrcoef(Y_test, y_pred_rfe)
model_results =pd.DataFrame([['Random Forest',acc, prec,rec,specificity, f1,roc,
loss_log,mathew]],
            columns = ['Model', 'Accuracy','Precision', 'Sensitivity','Specificity', 'F1
Score','ROC','Log_Loss','mathew_corrcoef'])

model_results
Y_pred_rf = np.around(Y_pred_rf)
print(metrics.classification_report(Y_test,Y_pred_rf))
Y_pred_rf = np.around(Y_pred_rf)
print(metrics.classification_report(Y_test,Y_pred_rf))
knn_classifier= KNeighborsClassifier(n_neighbors=31,leaf_size=30)
knn_classifier.fit(X_train,Y_train)
Y_pred_knn = knn_classifier.predict(X_test)
score_knn = round(accuracy_score(Y_pred_knn,Y_test)*100,2)
score_knn
knn_classifier= KNeighborsClassifier(n_neighbors=31,leaf_size=30)
knn_classifier.fit(X_train,Y_train)
Y_pred_knn = knn_classifier.predict(X_test)
score_knn = round(accuracy_score(Y_pred_knn,Y_test)*100,2)
score_knn
Y_pred_knn = np.around(Y_pred_knn)
print(metrics.classification_report(Y_test,Y_pred_knn))
Y_pred_knn = np.around(Y_pred_knn)
print(metrics.classification_report(Y_test,Y_pred_knn))
dt_classifier = DecisionTreeClassifier(
   max_depth=20,
   min_samples_split=2,
   min_samples_leaf=1,
   min_weight_fraction_leaf=0.00001,
   max_features='auto',
   random_state=46)
dt_classifier.fit(X_train, Y_train)
Y_pred_dt=dt_classifier.predict(X_test)
```

```
score_dt = round(accuracy_score(Y_pred_dt,Y_test)*100,2)
score_dt
y_pred_dte = dt_classifier.predict(X_test)

plt.figure(figsize=(10, 8))
CM=confusion_matrix(Y_test,y_pred_dte)
sns.heatmap(CM, annot=True)

TN = CM[0][0]
FN = CM[1][0]
TP = CM[1][1]
FP = CM[0][1]
specificity = TN/(TN+FP)
loss_log = log_loss(Y_test, y_pred_dte)
acc= accuracy_score(Y_test, y_pred_dte)
roc=roc_auc_score(Y_test, y_pred_dte)
prec = precision_score(Y_test, y_pred_dte)
rec = recall_score(Y_test, y_pred_dte)
f1 = f1_score(Y_test, y_pred_dte)

mathew = matthews_corrcoef(Y_test, y_pred_dte)
model_results =pd.DataFrame([['Decision Tree',acc, prec,rec,specificity, f1,roc,
loss_log,mathew]],
            columns = ['Model', 'Accuracy','Precision', 'Sensitivity','Specificity', 'F1
Score','ROC','Log_Loss','mathew_corrcoef'])

model_results
nb_classifier = GaussianNB( var_smoothing=1e-50)
nb_classifier.fit(X_train,Y_train)
nb_classifier.predict(X_test)
Y_pred_nb = nb_classifier.predict(X_test)
score_nb = round(accuracy_score(Y_pred_nb,Y_test)*100,2)
score_nb
y_pred_nbe = nb_classifier.predict(X_test)

plt.figure(figsize=(10, 8))
CM=confusion_matrix(Y_test,y_pred_nbe)
sns.heatmap(CM, annot=True)

TN = CM[0][0]
```

```
FN = CM[1][0]
TP = CM[1][1]
FP = CM[0][1]
specificity = TN/(TN+FP)
loss_log = log_loss(Y_test, y_pred_nbe)
acc= accuracy_score(Y_test, y_pred_nbe)
roc=roc_auc_score(Y_test, y_pred_nbe)
prec = precision_score(Y_test, y_pred_nbe)
rec = recall_score(Y_test, y_pred_nbe)
f1 = f1_score(Y_test, y_pred_nbe)
mathew = matthews_corrcoef(Y_test, y_pred_nbe)
model_results =pd.DataFrame([['Naive Bayes ',acc, prec,rec,specificity, f1,roc,
loss_log,mathew]],
            columns = ['Model', 'Accuracy','Precision', 'Sensitivity','Specificity', 'F1
Score','ROC','Log_Loss','mathew_corrcoef'])
model_results
Y_pred_nb = np.around(Y_pred_nb)
print(metrics.classification_report(Y_test,Y_pred_nb))
plot_roc_curve(nb_classifier,X_test,Y_test)
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('Receiver Operating Characteristic Curve');
plt.savefig("GNB.png")
scores = [score_rf,score_knn,score_nb,score_dt]
Models = ["Random Forest Classifier"," K-Nearest Neighbors Classifier","Navie
Bayes Classifier","Decision Tree Classifier"]

for i in range(len(Models)):
  print("The accuracy score achieved using "+Models[i]+" is: "+str(scores[i])+"
%")
Sns.set(style="darkgrid",rc={'figure.figsize':(20,10)})
plt.xlabel("Models")
plt.ylabel("Accuracy score")

sns.barplot(Models,scores)
plt.savefig("AccuracyScores.png")
import pickle
with open('models.pkl', 'wb') as file:
 pickle.dump(rf_classifier, file)
```

APP.py:

```
import numpy as np
import pickle
import sklearn
from flask import Flask, render_template, request, redirect, url_for, flash
import sqlite3
model = pickle.load(open('models.pkl', 'rb'))
app = Flask(__name__)
@app.route('/')
def index():
    return render_template('Heart_Disease_Classifier.html', title='Home')
@app.route('/Heart_Disease_Classifier')
def Heart_Disease_Classifier():
        return render_template('Heart_Disease_Classifier.html')
@app.route('/predict', methods =['POST'])
def predict():
  features = [float(i) for i in request.form.values()]
  #Convert features to array
  array_features = [np.array(features)]
  #Predict features
  prediction = model.predict(array_features)
  output = prediction
  if output == 1:
    return render_template('Heart_Disease_Classifier.html', result = 'The patient is not likely to have heart disease!')
  else:
    return render_template('Heart_Disease_Classifier.html', result = 'The patient is likely to have heart disease!')
if __name__ == '__main__':
   app.run(debug=True,port=5100)
```

HEART DISEASE CLASSIFIER.HTML:

```
<html>
<head>
<!-- Bootstrap CSS -->
```

```
<link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css"
integrity="sha384-
JcKb8q3iqJ61gNV9KGb8thSsNjpSL0n8PARn9HuZOnIxN0hoP+VmmDGMN5t9
UJ0Z" crossorigin="anonymous">
<script src="https://code.jquery.com/jquery-3.5.1.slim.min.js" integrity="sha384-
DfXdz2htPH0lsSSs5nCTpuj/zy4C+OGpamoFVy38MVBnE+IbbVYUew+OrCXa
Rkfj" crossorigin="anonymous"></script>
<script
src="https://cdn.jsdelivr.net/npm/popper.js@1.16.1/dist/umd/popper.min.js"
integrity="sha384-
9/reFTGAW83EW2RDu2S0VKaIzap3H66lZH81PoYlFhbGU+6BZp6G7niu735S
k7lN" crossorigin="anonymous"></script>
<script
src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"
integrity="sha384-
B4gt1jrGC7Jh4AgTPSdUtOBvfO8shuf57BaghqFfPlYxofvL8/KUEfYiJOMMV+r
V" crossorigin="anonymous"></script>
<title>Heart Disease Test</title>
</head>
<body>
<!-- Java Script -->
<script src="https://code.jquery.com/jquery-3.5.1.slim.min.js" integrity="sha384-
DfXdz2htPH0lsSSs5nCTpuj/zy4C+OGpamoFVy38MVBnE+IbbVYUew+OrCXa
Rkfj" crossorigin="anonymous"></script>
<script
src="https://cdn.jsdelivr.net/npm/popper.js@1.16.1/dist/umd/popper.min.js"
integrity="sha384-
9/reFTGAW83EW2RDu2S0VKaIzap3H66lZH81PoYlFhbGU+6BZp6G7niu735S
k7lN" crossorigin="anonymous"></script>
<script
src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"
integrity="sha384-
B4gt1jrGC7Jh4AgTPSdUtOBvfO8shuf57BaghqFfPlYxofvL8/KUEfYiJOMMV+r
V" crossorigin="anonymous"></script>
```

```
<!-- Navbar-->
<nav class="navbar navbar-dark" style="background-color: rgb(13, 102, 87);">
<span class="navbar-brand mb-0 h1">Heart Disease Test</span>
</nav>
<div class="container">
<br>
<!--Form-->
<form action = "{{url_for('predict')}}" method ="POST" >
<fieldset>
<legend style="color: rgb(41, 15, 134);"><b>Heart Disease Test
Form</b></legend><br>
<div class="card card-body" style="background-color: rgb(194 245 236 / 56%);">
<div class="form-group  row">
<div class="col-sm-3">
<label for="age">Age</label>
<input type="number" class="form-control" id="age" name="age" required>
</div>
<div class="col-sm-3">
<label for="sex">Sex</label>
<select class="form-control" id="sex"  name="sex" required>
<option disabled selected value> -- Select an Option -- </option>
<option value = "0">Female</option>
<option value = "1">Male</option>
</select>
</div>
</div>
<br>
<div class="form-group  row">
<div class="col-sm">
<label for="cp">Chest Pain Type</label>
<select class="form-control" id="cp" name = "cp" required>
<option disabled selected value> -- Select an Option -- </option>
<option value = "1">Typical Angina</option>
<option value = "2">Atypical Angina</option>
<option value = "3">Non-anginal Pain</option>
```

```html
<option value = "4">Asymptomatic</option>
</select>
</div>
<div class="col-sm">
<label for="trestbps">Resting Blood Pressure in mm Hg</label>
<input type="number" class="form-control" id="trestbps" name="trestbps"
required>
</div>
<div class="col-sm">
<label for="chol">Serum Cholestoral in mg/dl</label>
<input type="number" class="form-control" id="chol" name="chol" required>
</div>
<div class="col-sm">
<label for="fbs">Fasting Blood Sugar > 120 mg/dl</label>
<select class="form-control" id="fbs" name="fbs" required>
<option disabled selected value> -- Select an Option -- </option>
<option value = "0">False</option>
<option value = "1">True</option>
</select>
</div>
</div>
<br>
<div class="form-group row">
<div class="col-sm">
<label for="restecg">Resting ECG Results </label>
<select class="form-control" id="restecg" name="restecg" required>
<option disabled selected value> -- Select an Option -- </option>
<option value = "0">Normal  </option>
<option value = "1">Having ST-T wave abnormality  </option>
<option value = "2">Probable or definite left ventricular hypertrophy</option>
</select>
</div>
<div class="col-sm">
<label for="thalach">Maximum Heart Rate</label>
```

```
<input type="number" class="form-control" id="thalach" name="thalach"
required>
</div>
<div class="col-sm">
<label for="exang">Exercise Induced Angina </label>
<select class="form-control" id="exang" name="exang" required>
<option disabled selected value> -- Select an Option -- </option>
<option value = "0">No</option>
<option value = "1">Yes</option>
</select>
</div>
<div class="col-sm">
<label for="oldpeak">ST Depression Induced</label>
<input type="number" step="any" class="form-control" id="oldpeak"
name="oldpeak" required>
</div>
</div>
<br>
<div class="form-group  row">
<div class="col-sm">
<label for="slope">Slope of the Peak Exercise ST Segment </label>
<select class="form-control" id="slope" name="slope" required>
<option disabled selected value> -- Select an Option -- </option>
<option value = "1">Upsloping</option>
<option value = "2">Flat</option>
<option value = "3">Downsloping</option>
</select>
</div>
<div class="col-sm">
<label for="ca">Number of Vessels Colored by Flourosopy</label>
<select class="form-control" id="ca" name = "ca" required>
<option disabled selected value> -- Select an Option -- </option>
<option value = "0">0</option>
<option value = "1">1</option>
<option value = "2">2</option>
```

```
<option value = "3">3</option>
</select>
</div>
<div class="col-sm">
<label for="thal">Thalassemia</label>
<select class="form-control" id="thal" name = "thal" required>
<option disabled selected value> -- Select an Option -- </option>
<option value = "3">Normal</option>
<option value = "6">Fixed defect</option>
<option value = "7">Reversable defect</option>
</select>
</div>
</div>
<br>
<div class="form-group">
<input class="btn btn-primary" type="submit" value="Result">
</div>
<!--Prediction Result-->
<div id ="result">
<strong style="color:red">{{result}}</strong>
</div>
</div>
</fieldset>
</form>
</div>
</body>
</html>
```

GITHUB LINK:

https://github.com/IBM-EPBL/IBM-Project-23311-1659877753.git

PROJECT DEMOSTRATION

https://drive.google.com/file/d/1FzakzLZ-sGFVIU14MbZs_VhZqILT-OiP/view?usp=sharing