Heart Disease Prediction

Soorya Harsha P[1], Sankar Guru S[2], Aravind V[3], Jona J B.[4]
[1, 2, 3, 4] Dept of Computer Applications
[1, 2, 3, 4] Coimbatore Institute of Technology, Coimbatore,India
(Affiliated to Anna University)

Abstract- *The proposed system leverages advanced machine learning techniques, including Logistic Regression, Decision Tree, and Random Forest algorithms, to enhance the accuracy of predicting patient outcomes based on the input data. By harnessing the power of these algorithms, the system aims to reduce the manual effort required for risk assessment and provides highly precise results, including future projections.*

These predictive models are valuable tools for healthcare professionals as they enable more effective patient treatment strategies. By offering insights into future health prospects, they empower doctors to intervene early and take proactive measures to prevent heart disease and other life-threatening conditions, thereby saving lives and improving overall patient care.

I. INTRODUCTION

Heart Disease has become one of the most leading causes of death on the planet and it has become the most life-threatening disease. The early prediction of heart disease will help in reducing death rate. In our day to day life, there are many factors that affect the human heart. The report by World Health Statistics puts emphasis on this fact: one out of three adults all over the world today, is suffering from hypertension, i.e., elevated levels of blood pressure – a condition which results in deaths from a stroke or leads to heart attacks and other heart related diseases. Heart diseases, also called by the name cardiovascular diseases, include a no. of factors that can affect the heart – which not only include heart attacks. Cardiovascular diseases can be considered a primary reason of casualty in numerous countries. Cardiovascular diseases result in the death of one person in an average of 34 seconds as stated by a study held in the USA. Hence, it is extremely beneficial to have a system for automating the medical diagnosis for predicting the possibility of heart disease before it happens based on certain physiological parameters.

A. PROBLEM STATEMENT

1. Delayed Diagnosis and Missed Opportunities:

- Individuals at risk may not be identified: Without early detection, individuals at high risk of developing heart disease may not be identified until they begin to experience symptoms, potentially leading to delayed diagnosis and treatment.
- Missed opportunities for prevention: Early identification of individuals at risk enables preventive measures such as lifestyle changes and medication, which can significantly reduce the risk of developing heart disease.
- Increased risk of complications: Delayed diagnosis and treatment can lead to more serious complications, potentially requiring more intensive and expensive interventions.

2. Suboptimal Treatment and Management:

- Less personalized treatment plans: Without a personalized prediction of risk and specific medical profile, treatment plans may not be as effective as they could be.
- Ineffective resource allocation: Without accurate risk assessment, resources for diagnosis, treatment, and monitoring may be allocated inefficiently.
- Potential for overtreatment: For individuals with a low risk of heart disease, unnecessary treatments and interventions could be prescribed, leading to potential side effects and increased costs.

3. Lack of Patient Empowerment and Engagement:

- Limited information and awareness: Individuals may not have access to information about heart disease risk factors, prevention strategies, and treatment options, hindering their ability to make informed decisions about their health.
- Challenges in adopting healthy behaviors: Without personalized recommendations and support, individuals may find it difficult to adopt and maintain healthy habits that reduce their risk of heart disease.
- Communication gap between patients and providers: Lack of access to a tool that facilitates communication and information sharing between patients and healthcare providers can hinder effective disease management.

B. OBJECTIVE

A heart disease predicting application project has several objectives, which can be broadly categorized into three main areas:

1. Early detection and prevention:
    - Identify individuals at risk of developing heart disease: The application can analyze patient data such as age, gender, medical history, lifestyle factors, and genetic information to predict the risk of developing heart disease. Early detection allows for timely intervention through preventive measures like lifestyle changes, medication, and closer monitoring.
    - Reduce the burden of heart disease: By identifying individuals at high risk, resources can be allocated more effectively for preventive measures and early treatment, potentially reducing the overall burden of heart disease on individuals, healthcare systems, and society.
2. Improved diagnosis and treatment:
    - Assist healthcare professionals in diagnosis: The application can be used as a decision support tool for doctors, providing additional insights into a patient's risk of heart disease and aiding in diagnosis.
    - Personalize treatment plans: Based on an individual's predicted risk and specific medical profile, the application can suggest personalized treatment plans, including medication, lifestyle modifications, and monitoring strategies.
    - Improve treatment effectiveness: By providing personalized and targeted treatment plans, the application can potentially improve the effectiveness of treatment and lead to better patient outcomes.
3. Enhanced patient engagement and self-management:
    - Empower patients to manage their health: The application can provide patients with information about heart disease, risk factors, and preventive measures. This empowers them to actively participate in managing their health and reducing their risk of developing heart disease.
4. Promote healthy lifestyle choices: The application can provide personalized recommendations for healthy lifestyle changes, including diet, exercise, and stress management. This can help patients adopt and maintain healthy habits to reduce their risk of heart disease.

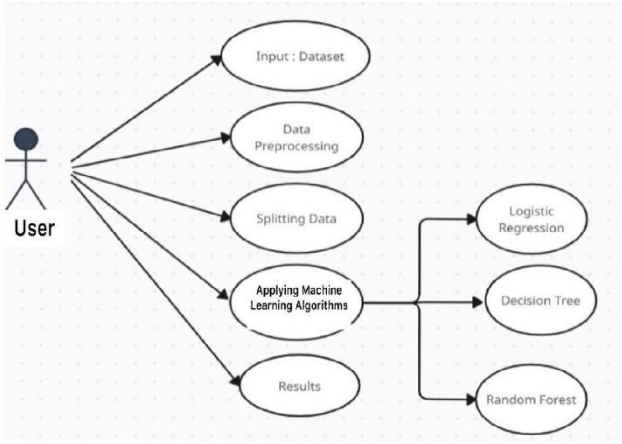
Overall, a heart disease predicting application project has the potential to significantly impact the prevention, diagnosis, and management of heart disease, leading to improved patient outcomes and reduced healthcare burdens.

II. SYSTEM DESIGN

The design activities are critical in this phase since they are where decisions are made that will eventually affect the software implementation's success and ease of maintenance. This decision has the most impact on a system's dependability and maintainability. The only method to correctly translate a customer's needs into final software or a system is through design.

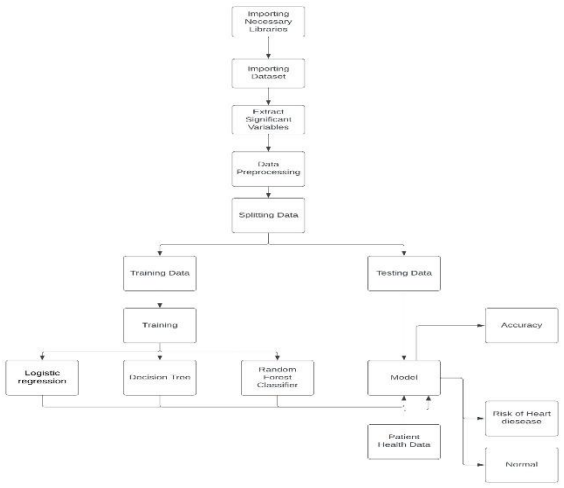
A. USE CASE DESIGN

A Use case diagram at its simplest is a representation of a user's interaction with the system that shows the relationship between the user and the different use cases in which the user is involved. A use case diagram can identify the different types of users of a system and the different use cases and will often be accompanied by other types of diagrams as well. The use cases are represented by either circles or ellipses. Fig below represents the Use case diagram for heart disease prediction.



B. SYSTEM FLOW

A system flow diagram is a visual representation of the flow of data, processes, and interactions within a system. It is a high-level view that provides an overview of how the components in a system work together. System flow diagrams are often used in system design, software development, and business analysis to convey the structure and functionality of a system Fig below represents the System Flow Diagram.



III. METHODOLOGY

1. Training Model

Importing the necessary libraries to train the model. Fig below represents the training model .

```
[ ] import numpy as np
    import pandas as pd
    from sklearn.model_selection import train_test_split
    from sklearn.linear_model import LogisticRegression
    from sklearn.tree import DecisionTreeClassifier
    from sklearn.ensemble import RandomForestClassifier
    from sklearn import tree
    from matplotlib import pyplot as plt
    from sklearn.metrics import accuracy_score

[ ] # uploading data to a Pandas DataFrame
    heart_data = pd.read_csv('/content/heart_disease_data.csv')

    # printing first 5 rows
    heart_data.head()
```

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
0	63	1	3	145	233	1	0	150	0	2.3	0	0	1	1
1	37	1	2	130	250	0	1	187	0	3.5	0	0	2	1
2	41	0	1	130	204	0	0	172	0	1.4	2	0	2	1
3	56	1	1	120	236	0	1	178	0	0.8	2	0	2	1
4	57	0	0	120	354	0	1	163	1	0.6	2	0	2	1

Importing Necessary Libraries

2. Data Preprocessing

Analyzing the dataset for null values or Duplicate values. As, it may reduce the accuracy and may mislead the model. Fig 4.2 shows below the check for unique values.

```
[ ] # printing last 5 rows
    heart_data.tail()
```

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
298	57	0	0	140	241	0	1	123	1	0.2	1	0	3	0
299	45	1	3	110	264	0	1	132	0	1.2	1	0	3	0
300	68	1	0	144	193	1	1	141	0	3.4	1	2	3	0
301	57	1	0	130	131	0	1	115	1	1.2	1	1	3	0
302	57	0	1	130	236	0	0	174	0	0.0	1	1	2	0

```
    # to view the number of rows and columns in the dataset
    heart_data.shape

    (303, 14)

[ ] # to check for unique values
    heart_data['sex'].unique()

    array([1, 0])

[ ] #to check for duplicates
    heart_data.duplicated().sum()

    1
```

Check for Unique Values from the Dataset

3. Dropping the Null and Duplicate values :

After removing the Null and Duplicate values from the dataset. Using the isnull() function we are checking for null value count in the dataset . Fig below the Null count of the dataset after Data Preprocessing.

```
+ Code  + Text                                                    Connect

    # checking for missing values
    heart_data.isnull().sum()

    age       0
    sex       0
    cp        0
    trestbps  0
    chol      0
    fbs       0
    restecg   0
    thalach   0
    exang     0
    oldpeak   0
    slope     0
    ca        0
    thal      0
    target    0
    dtype: int64

[ ] # statistical measures about the data
    heart_data.describe()
```

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
count	302.000000	302.000000	302.000000	302.000000	302.000000	302.000000	302.000000	302.000000	302.000000	302.000000	302.000000	302.000000	302.000000	302.000000
mean	54.420503	0.682119	0.963576	131.602649	246.500000	0.149007	0.526490	149.569536	0.327815	1.043046	1.397351	0.718543	2.314570	0.543046
std	9.047797	0.466426	1.032044	17.563394	51.753489	0.356686	0.526027	22.903527	0.470196	1.161452	0.616274	1.006748	0.613026	0.498970
min	29.000000	0.000000	0.000000	94.000000	126.000000	0.000000	0.000000	71.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	48.000000	0.000000	0.000000	120.000000	211.000000	0.000000	0.000000	133.250000	0.000000	0.000000	1.000000	0.000000	2.000000	0.000000
50%	55.500000	1.000000	1.000000	130.000000	240.500000	0.000000	1.000000	152.500000	0.000000	0.800000	1.000000	0.000000	2.000000	1.000000

Check for Null Value Count

4. Dataset Splitting

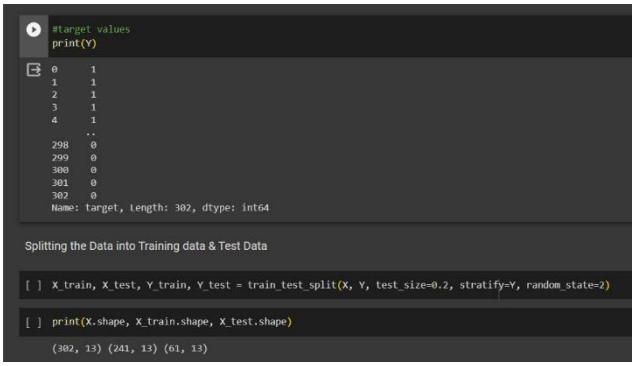
The dataset is divided into two parts: the first part contains all necessary features, and the second part only contains the target value (true or false). Fig below shows below the Dataset after splitting them into two halves.

```
+ Code  + Text

[ ] # checking the distribution of Target Variable
    heart_data['target'].value_counts()

    1    164
    0    138
    Name: target, dtype: int64

Splitting dataset

[ ] X = heart_data.drop(columns='target', axis=1)
    Y = heart_data['target']

    # Data for training
    print(X)

        age  sex  cp  trestbps  chol  fbs  restecg  thalach  exang  oldpeak  \
    0    63    1   3       145   233    1        0      150      0      2.3
    1    37    1   2       130   250    0        1      187      0      3.5
    2    41    0   1       130   204    0        0      172      0      1.4
    3    56    1   1       120   236    0        1      178      0      0.8
    4    57    0   0       120   354    0        1      163      1      0.6
    ..  ...  ...  ..       ...   ...  ...      ...      ...    ...      ...
    298  57    0   0       140   241    0        1      123      1      0.2
    299  45    1   3       110   264    0        1      132      0      1.2
    300  68    1   0       144   193    1        1      141      0      3.4
    301  57    1   0       130   131    0        1      115      1      1.2
    302  57    0   1       130   236    0        0      174      0      0.0

         slope  ca  thal
    0        0   0     1
    1        0   0     2
    2        2   0     2
    3        2   0     2
    4        2   0     2
```

Splitting the Dataset into two halves

5.Splitting Dataset Into Training And Testing Data:

Printing the target values of the dataset. After that, we are splitting the dataset into 80:20 ratio for Training and Testing. Fig shows below the target values of the dataset and Splitting of Dataset .



Splitting dataset into Training and testing dataset.

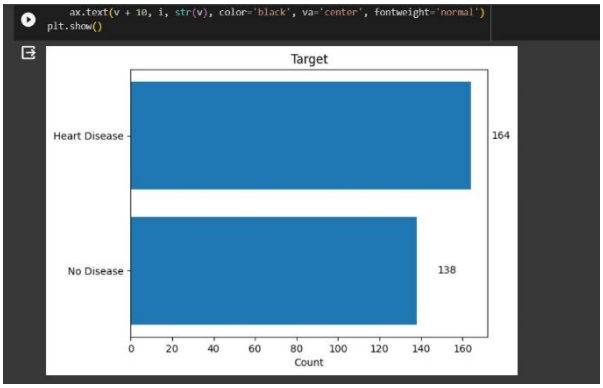
6.Feature Selection

To extract the features from the dataset we use Pearson Correlation in the dataset.

Check variables with high correlation using the correlation function and a threshold value of > 0.8. Since there are no correlation variables, the result set is empty. Fig 4.8 show below the result set of Correlation variables.

7. Graphs

Plotting a graph to determine the number of Affected and Unaffected individuals in the dataset. Fig 4.9 shows below the graph count of Affected and Unaffected Persons in the dataset.



Count of Affected and Unaffected Persons in the Dataset

Plotting a graph to determine the number of Males and Females, who were affected and Not Affected.

Plotting a graph to determine the total number of Affected and Not Affected cases in the dataset.

8. Model Training

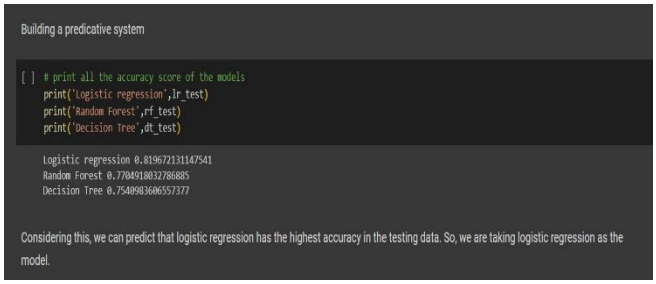
Logistic Regression : Initializing the Logistic Regression algorithm to variable lr and train the model using fit function with training dataset as parameter.

Decision tree Classifier:Initializing the Decision tree Classifier to variable decision_tree and training the model using predict function with training dataset as parameter. Fig 4.14 shows the accuracy of the trained model.

9. Overall Accuracy Of Models

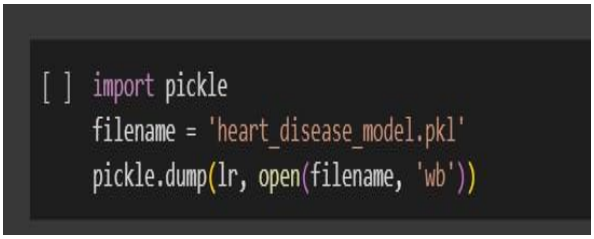
After training the models, displaying the accuracy of all the models with the test dataset.

Fig shows the overall accuracy of all the models.



10. Dumping Model

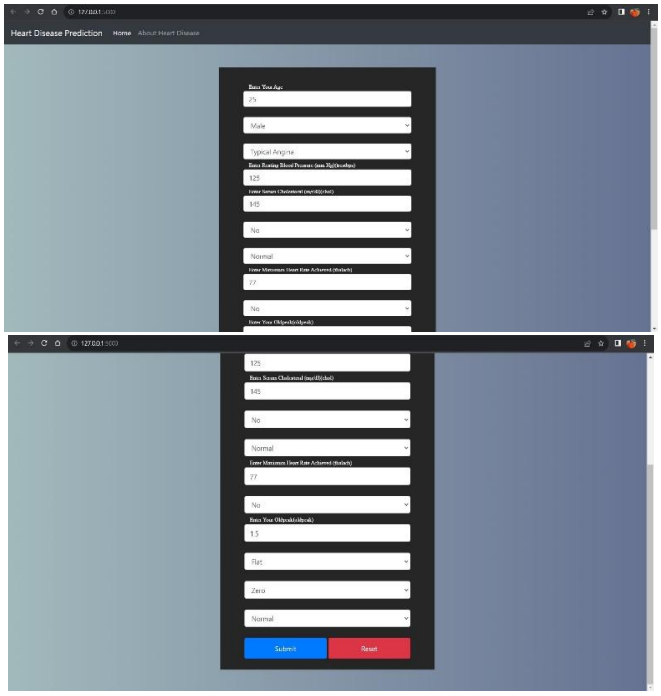
Dumping the model Logistic Regression which has the highest accuracy of 81%. Fig shows the dumping of the model.



Plotting a graph to visualize the parameters due to which the patient has hif=gher chances of heart disease. Fig 4.18 shows the parameters due to which the patient has higher chances of heart disease.

11. Home Page

Home page of the GUI where the users should enter their medical data into the form and click “submit”. shows the home page.

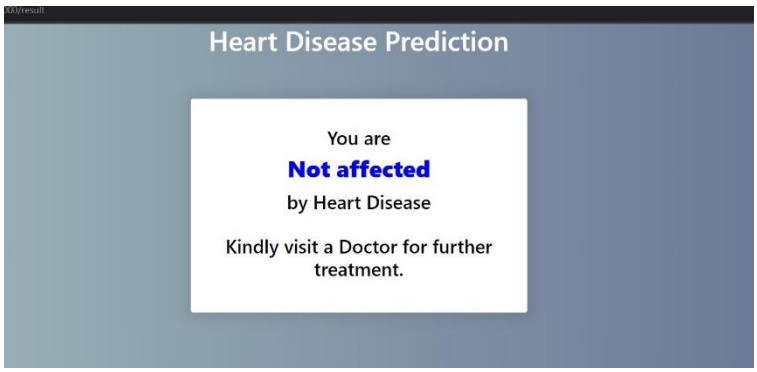


12.About Page

This page contains the symptoms and causes of heart disease.

13.Result Page

This page displays the result whether the patient is affected by heart disease or not.And, also it shows the parameters by which the patient is affected. Fig shows the Result page of Affected person. Fig shows the result page of Not Affected person.



REFERENCES

[1] Ordonez C **"**Association Rule Discovery With The Train and Test Approach for Heart Disease Prediction**",** IEEE Transactions on Information Technology in Biomedicine, Volume 10 Issue 2, pg 334-43, 2006.

[2] Shinde R, Arjun S, Patil P and Waghmare J. **"**An Intelligent Heart Disease Prediction

[3] System, International Journal of Computer Science and Information Technologies**",** Volume Issue 1, pg 637-9, 2015.

[4] Victor Chang, Vallabhanent Rupa Bhavani, Ariel Qianwen Xu, Ma Hossain. "An Artificial Intelligence Model for Heart Disease Detection Using Machine Learning", Healthcare Analytics, Volume 2, pg 116, 2022.

[5] Mohan, Senthilkumar, "Chandrasegar Thirumalai, and Gautam Srivastava, Effective heart disease prediction using hybrid machine learning techniques", IEEE Volume 7, pg 81542-81554, 2016.

[6] https://machinemantra.in/heart-disease-prediction-in-python/

[7] https://www.tensorflow.org/resources/learn-ml

[8] https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8898839/

[9] https://www.hindawi.com/journals/jhe/