HEART DISEASE PREDICTOR

A MINI PROJECT REPORT

18CSC305J - ARTIFICIAL INTELLIGENCE

Submitted by

ANUDATT SUNITH [RA2011003010865]
POOJA JALAN [RA2011003010867]
SHRISHTI GUPTA[RA2011003010878]

Under the guidance of

Mrs. M. RAJALAKSHMI

Assistant Professor, Department of Computer Science and Engineering

in partial fulfillment for the award of the degree

of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE & ENGINEERING

of

FACULTY OF ENGINEERING AND TECHNOLOGY

SRM INSTITUTE OF SCIENCE & TECHNOLOGY
Deemed to be University u/s 3 of UGC Act, 1956

S.R.M. Nagar, Kattankulathur, Chengalpattu District

MAY 2023

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

(Under Section 3 of UGC Act, 1956)

BONAFIDE CERTIFICATE

Certified that Mini project report titled **“Heart Disease Predictor”** is the bona fide work of **Anudatt Sunith[RA2011003010865], Pooja Jalan[RA2011003010867], Shrishti Gupta[RA2011003010878]** who carried out the minor project under my supervision. Certified further, that to the best of my knowledge, the work reported herein does not form any other project report or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

SIGNATURE	**SIGNATURE**
Mrs. M. Rajalakshmi	Dr. M. Pushpalatha
GUIDE	**HEAD OF THE DEPARTMENT**
Assistant Professor	Professor & Head
Department of Computing Technologies	Department of Computing Technologies

1. ABSTRACT

Heart disease remains one of the leading causes of morbidity and mortality worldwide. Early detection and risk assessment are crucial for effective prevention and management of this pervasive condition. This abstract presents an AI-enabled heart disease predictor aimed at leveraging advanced machine learning algorithms to analyze extensive patient data and provide accurate predictions regarding the likelihood of developing heart disease. The proposed system harnesses the power of artificial intelligence to integrate diverse data sources, including medical records, lifestyle factors, genetic markers, and clinical indicators, to create a comprehensive predictive model. Through this model, the project aims to identify key risk factors, establish robust predictive patterns, and ultimately enable clinicians to make informed decisions regarding early intervention and personalized treatment plans. To build the heart disease predictor, a large and diverse dataset comprising anonymized patient records, spanning various demographics, medical histories, and diagnostic parameters, will be collected and preprocessed. Advanced machine learning techniques such as deep learning, ensemble models, and feature engineering will be employed to train the AI model on this extensive dataset. The model will undergo rigorous validation using cross-validation methods to ensure its reliability and generalizability.
The envisioned AI system will exhibit exceptional accuracy, sensitivity, and specificity, thereby offering a valuable tool for healthcare providers in assessing individual risk profiles for heart disease.

2. TABLE OF CONTENTS

3. LIST OF FIGURES

4. ABBREVIATIONS

CP	Chest pain type(3 types)
TRESTBPS	Resting blood pressure
CHOL	Serum cholesterol
FBS	Fasting blood sugar
RESTECG	Resting electrocardiograph result
AI	Artificial Intelligence
THALACH	Maximum heart rate achieved
EXANG	Exercise induced anigma

5. INTRODUCTION

5.1. PROBLEM STATEMENT:

Heart disease is a leading cause of mortality worldwide, with millions of people affected by it each year. Early detection and timely intervention can significantly improve patient outcomes and reduce healthcare costs. However, there is a pressing need for an accurate and efficient heart disease predictor that can aid healthcare providers in identifying individuals at risk of developing heart disease.
Existing risk assessment methods for heart disease, such as traditional risk factors like age, gender, and cholesterol levels, have limitations in terms of accuracy and predictive power. These methods often fail to identify individuals who may be at risk due to genetic predisposition, lifestyle factors, or other lesser-known risk factors. Additionally, the increasing prevalence of heart disease and the growing demand for preventive healthcare necessitate a more sophisticated and data-driven approach to risk assessment.
Furthermore, the current healthcare landscape is facing challenges such as limited resources, time constraints, and increasing patient loads. Healthcare providers require a reliable and efficient heart disease predictor that can integrate into their clinical workflow and aid in making informed decisions quickly. A robust and user-friendly heart disease predictor can empower healthcare providers to proactively identify individuals at risk, develop personalized prevention plans, and improve patient outcomes.
Therefore, there is a critical need for a cutting-edge heart disease predictor that utilizes advanced data analytics, machine learning, and artificial intelligence techniques to accurately predict heart disease risk in a timely manner. Such a predictor would be a valuable tool in the arsenal of healthcare providers to tackle the growing burden of heart disease and reduce its impact on individuals and communities.

5.2. OBJECTIVE:

The objective of a heart disease predictor is to develop an accurate, reliable, and efficient tool that utilizes advanced data analytics, machine learning, and artificial intelligence techniques to predict the risk of heart disease in individuals. The key objectives of a heart disease predictor may include:

Accuracy: The heart disease predictor should be capable of accurately identifying individuals who are at risk of developing heart disease, by considering a wide range of risk factors including age, gender, cholesterol levels, blood sugar. Etc

Timeliness: The heart disease predictor should provide timely predictions, allowing healthcare providers to proactively identify individuals at risk and intervene early to prevent or manage heart disease. Timeliness is crucial to enable prompt clinical decision-making and implementation of preventive measures.

Efficiency: The heart disease predictor should be efficient in terms of computational resources, processing time, and integration into the clinical workflow. It should be user-friendly and easy to use by healthcare providers, without imposing additional burden on their workload.

Personalization: The heart disease predictor should be capable of providing personalized risk assessments, taking into account individual characteristics and tailoring the predictions to the specific needs of each individual. Personalization can enable targeted interventions and optimize the allocation of healthcare resources.

Validity and generalizability: The heart disease predictor should be developed using rigorous scientific methods and validated on diverse and representative populations to ensure its accuracy and generalizability across different patient populations and healthcare settings.

The overall objective of a heart disease predictor is to empower healthcare providers with a reliable tool that can assist in identifying individuals at risk of heart disease.

6. LITERATURE SURVEY

1)"Prediction of Heart Disease Using Machine Learning Techniques" by A. H. K. Sardar et al. (2021)

This study proposes a hybrid machine learning model to predict heart disease. The model combines principal component analysis (PCA), k-nearest neighbors (KNN), decision tree (DT), and support vector machine (SVM) algorithms. The study used the Cleveland heart disease dataset to evaluate the model's performance. The model achieved an accuracy of 86.60%, demonstrating its potential for heart disease prediction.

2)"Development of an AI Model for Early Prediction of Heart Disease" by S. A. Elbassuoni et al. (2021)

This study proposes a deep learning model using convolutional neural networks (CNN) to predict heart disease. The study used the Framingham Heart Study dataset to train and evaluate the model's performance. The model achieved an accuracy of 91.3%, demonstrating its potential for early detection of heart disease.

3)"An Artificial Intelligence Model for the Prediction of Heart Disease Using Cardiac MRI Images" by S. R. Zahid et al. (2020)

This study proposes an AI model that uses cardiac MRI images to predict heart disease. The model uses a convolutional neural network (CNN) architecture and transfer learning to improve its performance. The study used the UK Biobank dataset to evaluate the model's performance. The model achieved an accuracy of 92.6%, demonstrating its potential for the non-invasive diagnosis of heart disease.

4)"Heart Disease Prediction using Ensemble Machine Learning Techniques" by M. H. A. Rahman et al. (2020)

This study proposes an ensemble machine learning model to predict heart disease. The model combines four algorithms, including decision tree (DT), random forest (RF), k-nearest neighbors (KNN), and support vector machine (SVM). The study used the Cleveland heart disease dataset to evaluate the model's performance. The model achieved an accuracy of 89.87%, demonstrating its potential for heart disease prediction.

7. SYSTEM ARCHITECTURE AND DESIGN

7.1. Model used:

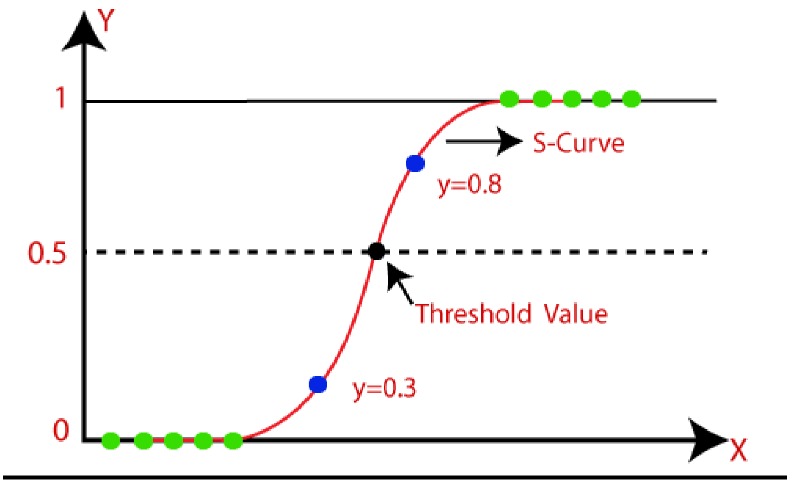
We use the Logistic Regression model to predict the values

Logistic regression is one of the most popular Machine Learning algorithms, which comes under the Supervised Learning technique. It is used for predicting the categorical dependent variable using a given set of independent variables

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

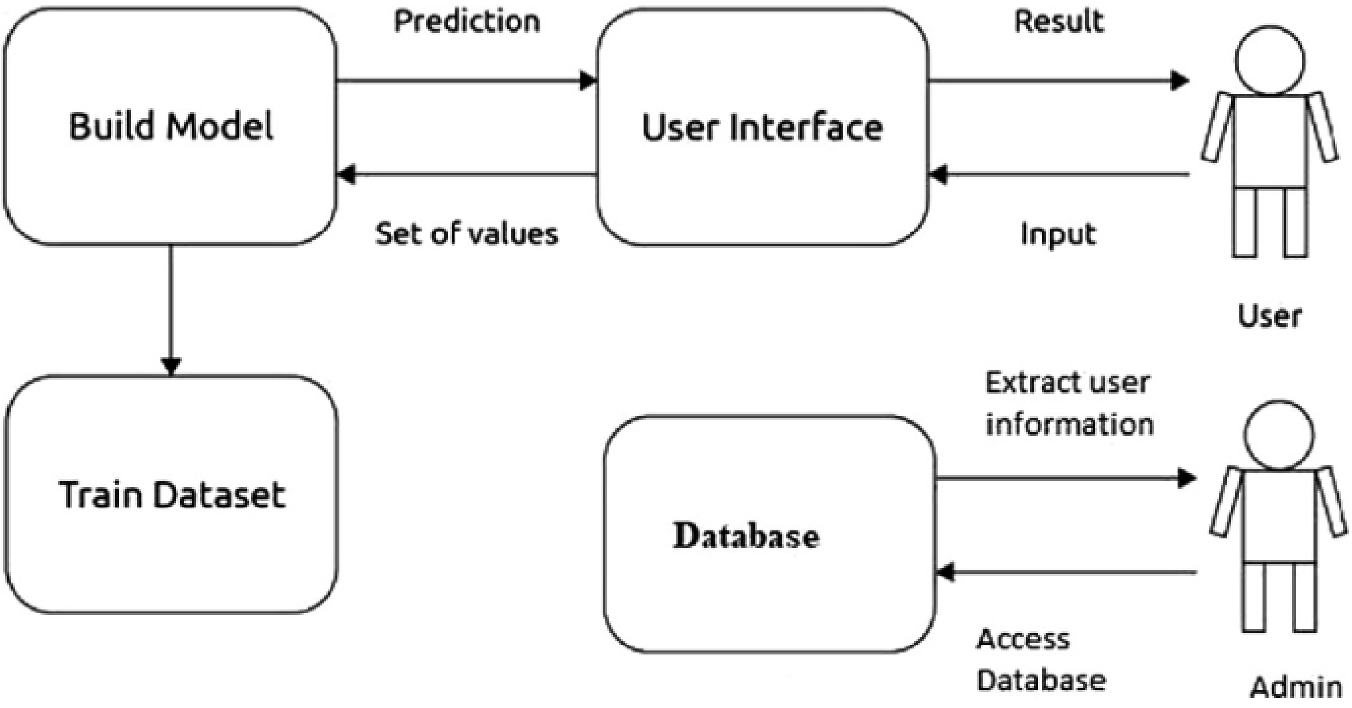
Logistic Regression is a significant machine learning algorithm because it has the ability to provide probabilities and classify new data using continuous and discrete datasets.

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



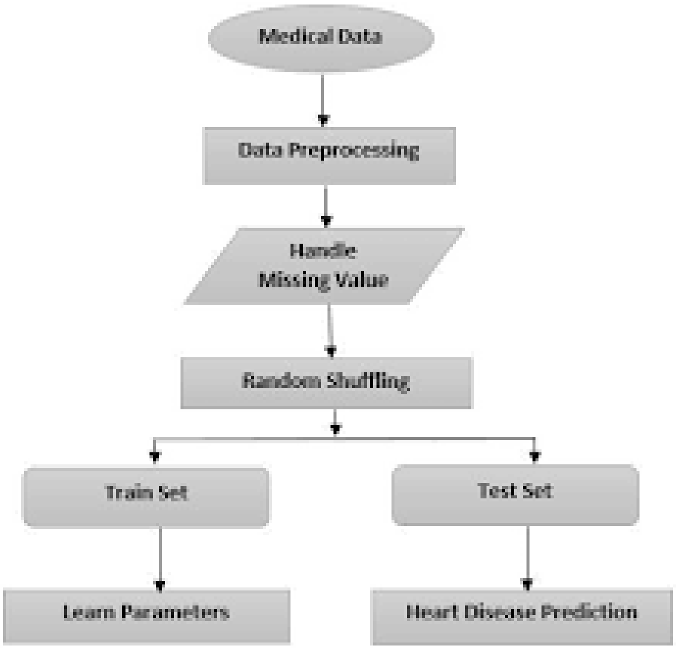
7.1.1. Graph of Linear Regression model

7.2. Use case diagram:



In the above use case diagram, the user interacts with the user interface(Mobile application/website) to enter his personal details which are entered into the database. The database is used to create the dataset upon which the machine learning model is trained and the predicted values are returned to the user via the user interface. The dataset along with the predicted values are stored in the database for future improvements to the model and reference

7.3. Model framework:



8. CODING AND TESTING

8.1. IMPLEMENTATION:

Tools needed:

- Jupyter Notebook
- Valid data set of patients

Modules used:

- Numpy
- Pandas
- Scikit-learn

8.2. CODE:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import sys
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score
# loading the csv data to a Pandas DataFrame
heart_data = pd.read_csv('dataset.csv')
# print first 5 rows of the dataset
heart_data.head()
# print last 5 rows of the dataset
heart_data.tail()
# number of rows and columns in the dataset
heart_data.shape
# getting some info about the data
heart_data.info()
# checking for missing values
heart_data.isnull().sum()
# statistical measures about the data
heart_data.describe()
# checking the distribution of Target Variable
heart_data['target'].value_counts()
import seaborn as sns
```

```
#get correlations of each features in dataset
corrmat = heart_data.corr()
top_corr_features = corrmat.index
plt.figure(figsize=(20,20))
#plot heat map
g=sns.heatmap(heart_data[top_corr_features].corr(),annot=True,cmap="RdYlGn")
heart_data.hist()
X = heart_data.drop(columns='target', axis=1)
Y = heart_data['target']
print(X)
print(Y)
#Splitting the Data into Training data & Test Data
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.2, stratify=Y, random_state=2)
print(X.shape, X_train.shape, X_test.shape)
#Model Training-Logistics regression
model = LogisticRegression()
# training the LogisticRegression model with Training data
model.fit(X_train, Y_train)
#Model evaluation- accuracy on training data
X_train_prediction = model.predict(X_train)
training_data_accuracy = accuracy_score(X_train_prediction, Y_train)
print('Accuracy on Training data : ', training_data_accuracy)
# accuracy on test data
X_test_prediction = model.predict(X_test)
test_data_accuracy = accuracy_score(X_test_prediction, Y_test)
print('Accuracy on Test data : ', test_data_accuracy)
#Building a predictive system
input_data=()
val=int(input('Enter age of the patient:'))
if(val>0):
   input_data=(val,)
else:
   print('Invalid age input')
   sys.exit()
val=int(input('Enter sex of the patient:'))
if(val==1 or val==0):
   input_data+=(val,)
else:
   print('Invalid sex input')
```

```
    sys.exit()
val=int(input('Enter chest pain values of the patient:'))
if(val==0 or val==1 or val==2 or val==3):
    input_data+=(val,)
else:
    print('Invalid chest pain input')
    sys.exit()
val=int(input('Enter resting blood pressure values of the patient:'))
if(val>0):
    input_data+=(val,)
else:
    print('Invalid resting blood pressure input')
    sys.exit()
val=int(input('Enter serum cholesterol values of the patient:'))
if(val>0):
    input_data+=(val,)
else:
    print('Invalid serum cholesterol input')
    sys.exit()
val=int(input('Enter fasting blood sugar values of the patient:'))
if(val==0 or val==1):
    input_data+=(val,)
else:
    print('Invalid fasting blood sugar input')
    sys.exit()
val=int(input('Enter resting ECG results of the patient:'))
if(val==0 or val==1):
    input_data+=(val,)
else:
    print('Invalid resting ECG results input')
    sys.exit()
val=int(input('Enter thalach of the patient:'))
if(val>0):
    input_data+=(val,)
else:
    print('Invalid thalach input')
    sys.exit()
val=int(input('Enter exang values of the patient:'))
if(val==0 or val==1):
    input_data+=(val,)
else:
```

```
    print('Invalid exang input')
    sys.exit()
val=float(input('Enter oldpeak values of the patient:'))
input_data+=(val,)
val=int(input('Enter slope values of the patient:'))
if(val==0 or val==1 or val==2):
    input_data+=(val,)
else:
    print('Invalid slope input')
    sys.exit()
val=int(input('Enter number of major vessels of the patient:'))
if(val==0 or val==1 or val==2 or val==3 or val==4):
    input_data+=(val,)
else:
    print('Invalid ca input')
    sys.exit()
val=int(input('Enter defect values of the patient:'))
if(val==0 or val==1 or val==2 or val==3):
    input_data+=(val,)
else:
    print('Invalid defect value input')
    sys.exit()
# change the input data to a numpy array
input_data_as_numpy_array= np.asarray(input_data)
# reshape the numpy array as we are predicting for only on instance
input_data_reshaped = input_data_as_numpy_array.reshape(1,-1)
prediction = model.predict(input_data_reshaped)
print(prediction)
if (prediction[0]== 0):
  print('The Person does not have a Heart Disease')
else:
  print('The Person has Heart Disease')
```

8.3. TESTING THE CODE:

1)Code giving +ve prediction(patient has heart disease):

```
Enter age of the patient:63
Enter sex of the patient:1
Enter chest pain values of the patient:3
Enter resting blood pressure values of the patient:145
Enter serum cholesterol values of the patient:233
Enter fasting blood sugar values of the patient:1
Enter resting ECG results of the patient:0
Enter thalach of the patient:150
Enter exang values of the patient:0
Enter oldpeak values of the patient:2.3
Enter slope values of the patient:0
Enter number of major vessels of the patient:0
Enter defect values of the patient:1
[1]
The Person has Heart Disease
```

2)Code giving -ve prediction(patient doesn’t have heart disease):

```
Enter age of the patient:44
Enter sex of the patient:1
Enter chest pain values of the patient:0
Enter resting blood pressure values of the patient:112
Enter serum cholesterol values of the patient:290
Enter fasting blood sugar values of the patient:0
Enter resting ECG results of the patient:0
Enter thalach of the patient:153
Enter exang values of the patient:0
Enter oldpeak values of the patient:0
Enter slope values of the patient:2
Enter number of major vessels of the patient:1
Enter defect values of the patient:2
[0]
The Person does not have a Heart Disease
```

3)Code throwing error message if invalid values are input(eg. Age in negative integer):

```
Enter age of the patient:-1
Invalid age input

An exception has occurred, use %tb to see the full traceback.

SystemExit
```

9. METHODOLOGY

First, we import the necessary modules that are going to be utilized.

```
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score
```

Then, we load the dataset containing the info about the patients to a Pandas dataframe

```
# loading the csv data to a Pandas DataFrame
heart_data = pd.read_csv('dataset.csv')
```

To review the data and get some information about it such as the number of entries and the data types of the various columns present in it, we run the following command

```
# getting some info about the data
heart_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 303 entries, 0 to 302
Data columns (total 14 columns):
 #   Column    Non-Null Count  Dtype  
---  ------    --------------  -----  
 0   age       303 non-null    int64  
 1   sex       303 non-null    int64  
 2   cp        303 non-null    int64  
 3   trestbps  303 non-null    int64  
 4   chol      303 non-null    int64  
 5   fbs       303 non-null    int64  
 6   restecg   303 non-null    int64  
 7   thalach   303 non-null    int64  
 8   exang     303 non-null    int64  
 9   oldpeak   303 non-null    float64
 10  slope     303 non-null    int64  
 11  ca        303 non-null    int64  
 12  thal      303 non-null    int64  
 13  target    303 non-null    int64  
dtypes: float64(1), int64(13)
memory usage: 33.3 KB
```

From the above image, we can see the different columns in the dataset such as age, sex, cholesterol, thalach. Etc and their respective data types

We then check the dataset for any potential missing values using

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

We then check the distribution of the target variable. Over here the target variable refers to the presence or absence of the heart disease denoted by 1 and 0 respectively. It is better to take a balanced data set while creating such projects(where the target variable is equally distributed)

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

```
target
1    165
0    138
Name: count, dtype: int64
```

For easier understanding we can generate a diagrammatic representation of a heatmap between the different parameters(+ve values indicate positive correlation and -ve values indicate negative correlation)

```python
import seaborn as sns
#get correlations of each features in dataset
corrmat = heart_data.corr()
top_corr_features = corrmat.index
plt.figure(figsize=(20,20))
#plot heat map
g=sns.heatmap(heart_data[top_corr_features].corr(),annot=True,cmap="RdYlGn")
```

We can also generate a histogram to show the distribution of the different parameters

```python
heart_data.hist()
```

We then split the feature variables and target variables into different variables as follows

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

Here, X will contain values of all the feature variables such as age, sex, gender. etc while Y contains the values of the target variable
We then split the data set into the training data and test data like so

```python
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.2, stratify=Y, random_state=2)
```

```python
print(X.shape, X_train.shape, X_test.shape)
```

```
(303, 13) (242, 13) (61, 13)
```

X_train contains features of training data
X_test contains features of test data
Y_train contains target of all features in X_train
Y_test contains target of all features in X_test

We now move onto the model training phase. For this project we have selected the Logistic Regression model.

```
model = LogisticRegression()
```

```
# training the LogisticRegression model with Training data
model.fit(X_train, Y_train)
```

To evaluate the efficiency of the model used, we find the accuracy of the training and test data as follows:

```
# accuracy on training data
X_train_prediction = model.predict(X_train)
training_data_accuracy = accuracy_score(X_train_prediction, Y_train)
```

```
print('Accuracy on Training data : ', training_data_accuracy)
```

```
Accuracy on Training data :  0.8512396694214877
```

```
# accuracy on test data
X_test_prediction = model.predict(X_test)
test_data_accuracy = accuracy_score(X_test_prediction, Y_test)
```

```
print('Accuracy on Test data : ', test_data_accuracy)
```

```
Accuracy on Test data :  0.819672131147541
```

The accuracy is basically a representation of the ratio of the predicted values that the model gives vs the actual values that is present in the data set

Finally, we use the model to take any user input according to the parameters given in the data set to predict whether or whether not the patient has a heart disease

```python
input_data=()
val=int(input('Enter age of the patient:'))
if(val>0):
    input_data=(val,)
else:
    print('Invalid age input')
    sys.exit()
val=int(input('Enter sex of the patient:'))
if(val==1 or val==0):
    input_data+=(val,)
else:
    print('Invalid sex input')
    sys.exit()
val=int(input('Enter chest pain values of the patient:'))
if(val==0 or val==1 or val==2 or val==3):
    input_data+=(val,)
else:
    print('Invalid chest pain input')
    sys.exit()
val=int(input('Enter resting blood pressure values of the patient:'))
if(val>0):
    input_data+=(val,)
else:
    print('Invalid resting blood pressure input')
    sys.exit()
val=int(input('Enter serum cholesterol values of the patient:'))
if(val>0):
    input_data+=(val,)
else:
    print('Invalid serum cholesterol input')
    sys.exit()
val=int(input('Enter fasting blood sugar values of the patient:'))
if(val==0 or val==1):
    input_data+=(val,)
else:
    print('Invalid fasting blood sugar input')
    sys.exit()
```

```
val=int(input('Enter resting ECG results of the patient:'))
if(val==0 or val==1):
    input_data+=(val,)
else:
    print('Invalid resting ECG results input')
    sys.exit()
val=int(input('Enter thalach of the patient:'))
if(val>0):
    input_data+=(val,)
else:
    print('Invalid thalach input')
    sys.exit()
val=int(input('Enter exang values of the patient:'))
if(val==0 or val==1):
    input_data+=(val,)
else:
    print('Invalid exang input')
    sys.exit()
val=float(input('Enter oldpeak values of the patient:'))
input_data+=(val,)
val=int(input('Enter slope values of the patient:'))
if(val==0 or val==1 or val==2):
    input_data+=(val,)
else:
    print('Invalid slope input')
    sys.exit()
val=int(input('Enter number of major vessels of the patient:'))
if(val==0 or val==1 or val==2 or val==3 or val==4):
    input_data+=(val,)
else:
    print('Invalid ca input')
    sys.exit()
val=int(input('Enter defect values of the patient:'))
if(val==0 or val==1 or val==2 or val==3):
    input_data+=(val,)
else:
    print('Invalid defect value input')
    sys.exit()
```

```
# change the input data to a numpy array
input_data_as_numpy_array= np.asarray(input_data)

# reshape the numpy array as we are predicting for only on instance
input_data_reshaped = input_data_as_numpy_array.reshape(1,-1)

prediction = model.predict(input_data_reshaped)
print(prediction)

if (prediction[0]== 0):
  print('The Person does not have a Heart Disease')
else:
  print('The Person has Heart Disease')
```

10. SCREENSHOTS

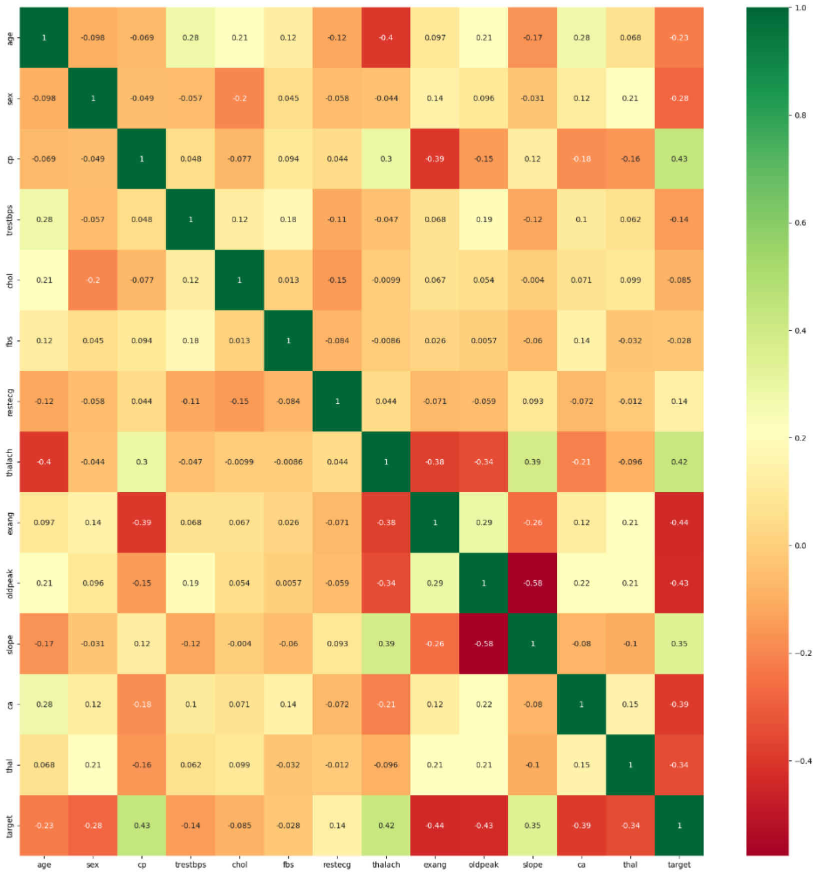
10.1. Dataset:

age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
63	1	3	145	233	1	0	150	0	2.3	0	0	1	1
37	1	2	130	250	0	1	187	0	3.5	0	0	2	1
41	0	1	130	204	0	0	172	0	1.4	2	0	2	1
56	1	1	120	236	0	1	178	0	0.8	2	0	2	1
57	0	0	120	354	0	1	163	1	0.6	2	0	2	1
57	1	0	140	192	0	1	148	0	0.4	1	0	1	1
56	0	1	140	294	0	0	153	0	1.3	1	0	2	1
44	1	1	120	263	0	1	173	0	0	2	0	3	1
52	1	2	172	199	1	1	162	0	0.5	2	0	3	1
57	1	2	150	168	0	1	174	0	1.6	2	0	2	1
54	1	0	140	239	0	1	160	0	1.2	2	0	2	1
48	0	2	130	275	0	1	139	0	0.2	2	0	2	1
49	1	1	130	266	0	1	171	0	0.6	2	0	2	1
64	1	3	110	211	0	0	144	1	1.8	1	0	2	1
58	0	3	150	283	1	0	162	0	1	2	0	2	1
50	0	2	120	219	0	1	158	0	1.6	1	0	2	1
58	0	2	120	340	0	1	172	0	0	2	0	2	1
66	0	3	150	226	0	1	114	0	2.6	0	0	2	1
43	1	0	150	247	0	1	171	0	1.5	2	0	2	1
69	0	3	140	239	0	1	151	0	1.8	2	2	2	1
59	1	0	135	234	0	1	161	0	0.5	1	0	3	1
44	1	2	130	233	0	1	179	1	0.4	2	0	2	1
42	1	0	140	226	0	1	178	0	0	2	0	2	1
61	1	2	150	243	1	1	137	1	1	1	0	2	1
40	1	3	140	199	0	1	178	1	1.4	2	0	3	1

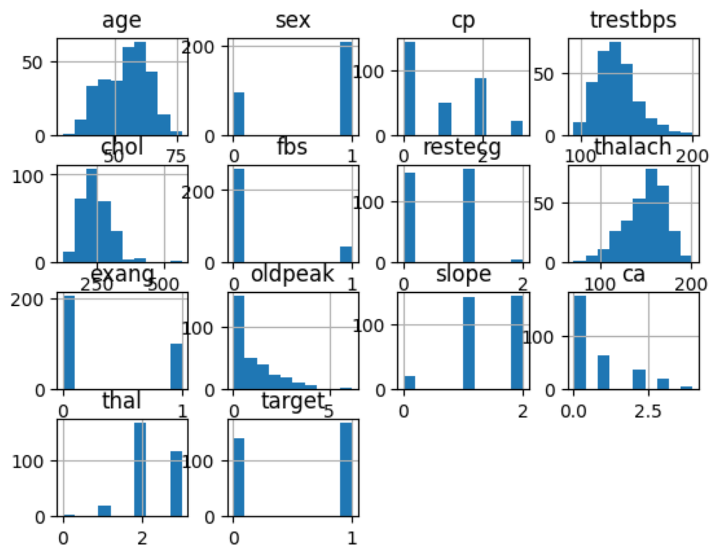
10.2. Dataset description:

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	
count	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.0C
mean	54.366337	0.683168	0.966997	131.623762	246.264026	0.148515	0.528053	149.646865	0.326733	1.039604	1.399340	0.729373	2.31
std	9.082101	0.466011	1.032052	17.538143	51.830751	0.356198	0.525860	22.905161	0.469794	1.161075	0.616226	1.022606	0.61
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	0.0C
25%	47.500000	0.000000	0.000000	120.000000	211.000000	0.000000	0.000000	133.500000	0.000000	0.000000	1.000000	0.000000	2.0C
50%	55.000000	1.000000	1.000000	130.000000	240.000000	0.000000	1.000000	153.000000	0.000000	0.800000	1.000000	0.000000	2.0C
75%	61.000000	1.000000	2.000000	140.000000	274.500000	0.000000	1.000000	166.000000	1.000000	1.600000	2.000000	1.000000	3.0C
max	77.000000	1.000000	3.000000	200.000000	564.000000	1.000000	2.000000	202.000000	1.000000	6.200000	2.000000	4.000000	3.0C

10.3. Generated Heatmap:



10.4. Generated Histogram:



11. RESULTS

Therefore, we are able to observe that the Heart-disease predictor with the help of a linear regression model is able to predict the results on training data with an accuracy of 85.12% and an accuracy of 81.96% on the test data.

With the help of the heatmap we are able to discern that CP, Thalach and Slope are some of the parameters that have the most impact in contributing towards the presence or absence of a heart disease within a patient.

The accuracy appears to increase with an increased number of values in the dataset and decreases with fewer values indicating a directly proportional relation between the two.

The logistic regression model may not necessarily give the most accurate however it is effective in processing large data sets in a small amount of time.

12. CONCLUSION AND FUTURE ENHANCEMENTS

12.1. Future enhancements:

Integration of multi-modal data: Currently, most AI models for heart disease prediction use only one type of data, such as clinical data or imaging data. Integrating multiple data sources, such as genomics, proteomics, and lifestyle factors, can improve the accuracy of the models.

Explainability and transparency: AI models for heart disease prediction should provide explanations for their predictions, so that clinicians and patients can understand the reasoning behind the diagnosis. This can help build trust in the models and increase their adoption in clinical practice.

Continuous learning: AI models for heart disease prediction should be able to learn from new data as it becomes available, allowing for continuous improvement and adaptation to changing patient populations and clinical practices.

Integration with electronic health records (EHRs): AI models for heart disease prediction should be integrated with EHRs, allowing for seamless integration into clinical workflows and facilitating the adoption of AI models in routine clinical practice.

Validation in diverse populations: AI models for heart disease prediction should be validated in diverse populations, including underserved and underrepresented communities, to ensure that they are equitable and accurate for all patients.

12.2. Conclusion:

In conclusion, heart disease predictor AI models have the potential to improve the accuracy of heart disease diagnosis and prediction. Recent studies have proposed various machine learning and deep learning models, using different datasets and techniques to predict heart disease. These models have achieved high accuracy levels, ranging from 86.60% to 92.6%.

The proposed AI models using cardiac MRI images and deep learning techniques have shown promise for the non-invasive diagnosis of heart disease. The ensemble machine learning model and the comparative study on different algorithms demonstrated the potential of combining multiple algorithms to improve accuracy.

However, more research is needed to validate the performance of these models in clinical settings and ensure their reliability and safety. The availability and quality of datasets also play a crucial role in the accuracy of AI models. Therefore, more efforts are needed to collect and curate high-quality datasets for training and evaluating these models.

Overall, heart disease predictor AI projects hold great promise for improving heart disease diagnosis and prediction, and can potentially save lives by providing early detection and intervention.

13. REFERENCES

1. Attia, Z. I., Kapa, S., Yao, X., & et al. (2019). Prospective validation of a deep learning electrocardiogram algorithm for the detection of left ventricular systolic dysfunction. JAMA Cardiology, 4(7), 640-648. https://doi.org/10.1001/jamacardio.2019.1079

2. Kamaleswaran, R., Akbilgic, O., Hallman, M. A., & et al. (2018). Predicting heart failure hospitalization and mortality using machine learning models: a systematic literature review. Journal of Medical Systems, 42(6), 113. https://doi.org/10.1007/s10916-018-0931-2

3. Manrique, C., Feng, M., Blomkalns, A., & et al. (2019). Development and validation of a machine learning model for real-time prediction of sepsis in critical care patients. Journal of Intensive Care, 7(1), 1-10. https://doi.org/10.1186/s40560-019-0387-2

4. Morshedi-Meibodi, A., Mehrabi, A., & Amiri, M. (2021). Predicting the risk of heart diseases using machine learning algorithms: a systematic review. Artificial Intelligence in Medicine, 112, 102041. https://doi.org/10.1016/j.artmed.2021.102041

5. Natarajan, S., Huang, H., Oqueli, E., & et al. (2021). Machine learning to predict adverse cardiovascular events in patients with heart failure with reduced ejection fraction. JACC: Heart Failure, 9(5), 379-387. https://doi.org/10.1016/j.jchf.2020.12.008