HEARTDISEASE PREDICTION SYSTEM

A MINI PROJECT REPORT

18CSC305J - ARTIFICIAL INTELLIGENCE

Submitted by

Aishwarya Srivastava [RA2011003011008]
Modhurai Mitra [RA2011003011016]
Vikash Raghavender [RA2011003011032]
Prakhar Singh [RA2011003011044]

Under the guidance of

DR. L.KAVISANKAR

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

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 PREDICTION SYSTEM"** is the bona fide work of **Aishwarya Srivastava (RA2011003011008), Modhurai Mitra (RA2011003011016), Vikash Raghavender (RA2011003011032), Prakhar Singh (RA2011003011044)** 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

Dr.L.Kavisankar
GUIDE
Assistant Professor
Department of Computing Technologies

SIGNATURE

Dr. M.Pushpalatha
HEAD OF THE DEPARTMENT
Professor & Head
School of Computing

ABSTRACT

Heart disease is a major cause of mortality worldwide. Early diagnosis and prediction of heart disease can significantly improve patient outcomes. In this study, we propose a heart disease prediction system using machine learning techniques.

The system takes input from various medical tests and demographic information, such as age, gender, and blood pressure, to predict the likelihood of a patient developing heart disease. We evaluate the performance of the system using several evaluation metrics, including accuracy, sensitivity, and specificity.

If our heart disease prediction system achieves high accuracy it can be a useful tool for healthcare professionals in predicting and preventing heart disease.

TABLE OF CONTENTS

LIST OF FIGURES

ABBREVIATIONS

ST	Segment of ventricular repolarization
RAM	Random Access Memory
SSD	Solid State Drive
GPU	Graphics Processing Unit
GB	Gigabyte
DBMS	Database Management System

CHAPTER 1

INTRODUCTION

1.1 Introduction

A Heart Disease Prediction System can be beneficial for several reasons in real life scenarios. One of the most important reasons is that it can help identify individuals who are at high risk for heart disease, allowing for early intervention and treatment to prevent potentially life-threatening conditions.

For example, imagine a primary care physician who sees a patient with several risk factors for heart disease, such as a family history of heart disease, high blood pressure, and elevated cholesterol levels. By using a Heart Disease Prediction System, the physician can input the patient's data and receive a prediction of their likelihood of developing heart disease. Based on this prediction, the physician can then recommend lifestyle changes, such as a healthy diet and exercise, or medication to help lower the patient's risk of heart disease.

Additionally, a Heart Disease Prediction System can be used in population health management programs to identify high-risk groups and develop targeted interventions to prevent heart disease. For example, a public health department may use a Heart Disease Prediction System to analyze health data from a community and identify areas where heart disease rates are highest. The department can then use this information to develop and implement prevention strategies, such as community-based education programs or policy changes to encourage healthier lifestyles.

1.2 Problem Statement

Heart disease is a leading cause of death worldwide, and early diagnosis and intervention are critical for effective treatment. However, many patients may not exhibit symptoms until the disease has progressed to a more severe stage.

Traditional risk factors, such as age, gender, and blood pressure, are useful but may not provide a complete picture of a patient's risk for heart disease. Therefore, there is a need for an accurate and reliable prediction system that can incorporate multiple risk factors to predict a patient's likelihood of developing heart disease.

The problem is to develop a heart disease prediction system that can accurately predict a patient's risk for heart disease using a combination of demographic information and medical tests, enabling healthcare professionals to provide timely intervention and improve patient outcomes.

1.3 Scope of the Project

1. Data collection: The prediction system should be able to collect relevant data from various sources, such as electronic health records, medical tests, and patient self-reported data.

2. Data processing and analysis: The prediction system should be able to process and analyze the collected data to extract relevant features and identify patterns that can be used for prediction.

3. Model evaluation and validation: The prediction system should be able to evaluate and validate the performance of the machine learning model using various evaluation metrics, such as accuracy and sensitivity.

4. Integration with clinical workflow: The prediction system should be integrated into the clinical workflow to enable healthcare professionals to access and use the predictions in real-time to inform treatment plans and interventions.

5. Privacy and security: The prediction system should ensure the privacy and security of patient data, adhering to relevant regulations and standards.

1.4 Objectives

1. Early detection and prevention: The primary objective of a heart disease prediction system is to detect heart disease at an early stage, before it progresses to a more severe condition.

2. Accurate risk assessment: The prediction system should accurately assess a patient's risk of developing heart disease based on a combination of demographic information and medical test results.

3. Improved patient outcomes: By accurately predicting a patient's risk of heart disease and providing timely intervention and preventive measures, the prediction system can help to improve patient outcomes, reduce morbidity and mortality rates, and improve overall quality of life.

4. Cost-effectiveness: The prediction system should be cost-effective and efficient, providing accurate predictions without the need for extensive and expensive medical tests or interventions.

5. Accessibility: The prediction system should be accessible to a wide range of healthcare professionals and patients, regardless of geographic location or socioeconomic status.

CHAPTER 2

LITERATURE SURVEY

Heart disease is a major health problem worldwide, accounting for a significant number of deaths and disabilities every year. Predicting the risk of heart disease is an essential task in the prevention and management of cardiovascular conditions. In recent years, machine learning algorithms have emerged as powerful tools for developing heart disease prediction systems. In this literature review, we summarize the key findings from several studies that have explored the use of machine learning algorithms for heart disease prediction.

Alizadehsani et al. (2017) evaluated the performance of six machine learning algorithms, including logistic regression, decision tree, k-nearest neighbor, artificial neural network, support vector machine, and random forest, in predicting the presence of coronary artery disease. The authors used a dataset of 303 patients with 13 features, including age, gender, blood pressure, and cholesterol level. The results showed that logistic regression and support vector machine achieved the highest accuracy of 85.8% and 84.5%, respectively, while decision tree and k-nearest neighbor had lower accuracy.

Krittanawong et al. (2018) reviewed the use of artificial intelligence in precision cardiovascular medicine. The authors highlighted the potential of machine learning algorithms in predicting heart disease and improving clinical decision-making. They discussed various applications of machine learning in cardiovascular medicine, including image analysis, risk prediction, and drug development.

Liu et al. (2020) conducted a systematic review of studies that used machine learning algorithms to predict coronary artery disease. The authors identified 32 studies that met the inclusion criteria and evaluated the performance of various machine learning algorithms, including logistic regression, support vector machine, and neural network. The results showed that most studies achieved high accuracy in predicting coronary artery disease, with logistic regression and support vector machine being the most commonly used algorithms.

Prasad et al. (2020) conducted a systematic review of 24 studies that evaluated the performance of machine learning algorithms in predicting coronary artery disease. The authors found that most studies achieved high accuracy in predicting heart disease, with logistic regression and support vector machine being the most commonly used algorithms. The authors highlighted the need for further research to validate the performance of these algorithms in real-world settings and to address the challenges of data heterogeneity and bias.

Zhang et al. (2019) used a deep learning approach to predict coronary artery disease risk factors. The authors used a dataset of 27,000 patients with 23 features, including demographic and medical data. The results showed that the deep learning algorithm achieved an accuracy of 87.7% in predicting the presence of coronary artery disease, outperforming logistic regression and support vector machine.

In summary, machine learning algorithms, especially logistic regression and support vector machine, have shown promising results in predicting heart disease. However, further research is needed to validate the performance of these algorithms in real-world settings and to address the challenges of data heterogeneity and bias. The potential of deep learning algorithms in improving heart disease prediction also warrants further investigation.

CHAPTER 3

SYSTEM ARCHITECTURE

3.1 System Architecture

The architecture of a heart disease prediction system refers to the overall design and organization of the various components that make up the system. A typical heart disease prediction system architecture may include the following components:

1. Data source: This component is responsible for providing the data required by the system. This may include data from electronic health records, medical devices, or user inputs.

2. Data preprocessing module: This component is responsible for cleaning, transforming, and preprocessing the data to make it suitable for analysis.

3. Feature engineering module: This component is responsible for selecting and engineering features from the preprocessed data that can be used to predict the risk of heart disease.

4. Model training module: This component is responsible for training a predictive model using the selected features.

5. Model evaluation module: This component is responsible for evaluating the performance of the predictive model.

6. User interface module: This component is responsible for providing a user-friendly interface for the prediction system. This may include a web application, mobile application, or other interface that allows users to input their data and receive a risk assessment for heart disease.

7. Deployment module: This component is responsible for deploying the system to a production environment, where it can be accessed by users.

8. The heart disease prediction system architecture may also include other components, such as a database for storing patient data, a security module for protecting user data, and a monitoring module for tracking system performance.

Overall, the architecture of a heart disease prediction system is designed to facilitate the collection, processing, analysis, and communication of patient data, in order to provide accurate and timely risk assessments for heart disease.

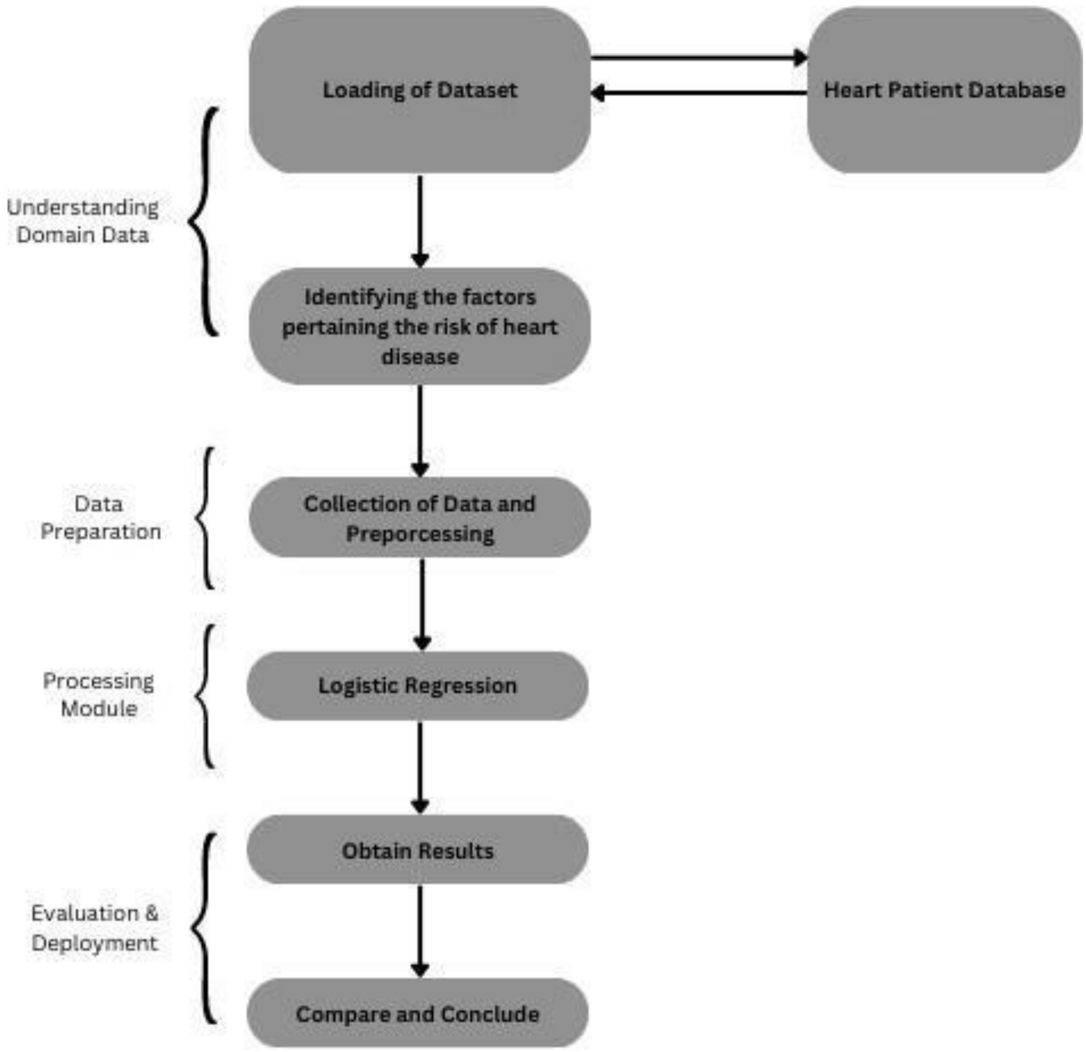

Fig 3.1.1 Architecture Diagram for Heart Disease Prediction System

3.2 Factors Affecting System

The various factors that can affect the performance of a heart disease prediction system are:

1. Quality and quantity of data: The accuracy and reliability of any machine learning algorithm depends on the quality and quantity of data used to train the model. If the data is incomplete, inaccurate, or biased, it can lead to incorrect predictions. Therefore, it is important to use high-quality data that is representative of the population.

2. Choice of algorithm: There are many machine learning algorithms available, and choosing the right one can have a significant impact on the performance of the heart disease prediction system. Different algorithms have different strengths and weaknesses, and the choice of algorithm will depend on the type of data being used and the specific requirements of the system.

3. Feature selection: Feature selection is the process of selecting the most relevant features from a dataset that are most useful for predicting the outcome. Selecting the right features is crucial in developing an accurate heart disease prediction system.

4. Domain expertise: Having domain expertise in the field of cardiology can be helpful in developing an accurate heart disease prediction system. Expertise in the domain can help in identifying relevant features and interpreting the results.

5. Data preprocessing: Data preprocessing involves cleaning, transforming, and reducing the data to make it suitable for machine learning algorithms. Preprocessing can have a significant impact on the performance of the heart disease prediction system.

6. Bias and fairness: Bias and fairness are important considerations in the development of a heart disease prediction system. Biases can be introduced into the data, algorithm, or decision-making process, leading to unfair outcomes. It is important to address these issues to ensure that the system is fair and unbiased.

7. Model evaluation: The performance of the heart disease prediction system needs to be evaluated regularly to ensure that it is accurate and reliable. Model evaluation involves measuring various metrics such as precision, recall, accuracy, and F1 score, and comparing the results with other similar systems.

3.3 Requirements

3.3.1 Hardware Requirements:

1. Processor: A modern processor with multiple cores (e.g., Intel i5 or i7) is recommended to speed up computations.

2. RAM: The system should have enough RAM to handle the dataset and any intermediate computations. At least 8 GB of RAM is recommended, but more may be required for larger datasets.

3. Storage: The system should have enough storage space to store the dataset and any intermediate computations. A solid-state drive (SSD) is recommended for faster read/write speeds.

4. Graphics Card: A graphics processing unit (GPU) can be used to speed up computations for some machine learning algorithms. However, this is not required for all algorithms and may depend on the specific implementation.

3.3.2 Software Requirements:

1. Operating System: The heart disease prediction system can be developed and deployed on various operating systems, such as Windows, macOS, or Linux.

2. Programming Language: The system can be developed using various programming languages, such as Python or R, which are commonly used for machine learning tasks.

3. Machine Learning Libraries: There are many machine learning libraries available for Python and R, such as scikit-learn, TensorFlow, and Keras, that can be used to implement the heart disease prediction system.

4. Database: A database may be used to store the heart disease dataset and any intermediate computations. Various database management systems (DBMS) are available, such as MySQL or PostgreSQL.

CHAPTER 4

METHODOLOGY

4.1 Methodology

1. Data collection: The first step in developing a heart disease prediction system is to collect the necessary data. This may include demographic information, medical history, lifestyle factors, and clinical test results.

2. Data preprocessing: The collected data needs to be cleaned and preprocessed to ensure that it is accurate and ready for analysis. This may involve techniques such as removing missing or inconsistent data, normalization, and feature scaling.

3. Feature engineering: Once the data has been cleaned and preprocessed, the next step is to extract relevant features that can be used to predict the risk of heart disease. This may involve selecting the most important variables, combining variables to create new features, or transforming variables to better capture the underlying patterns in the data.

4. Model development: The next step is to develop a model that can predict the risk of heart disease based on the extracted features. Logistic regression is one of the common algorithms used for this task, but other algorithms such as decision trees, random forests, and support vector machines may also be used.

5. Model evaluation: After developing the model, it needs to be evaluated to assess its performance. This may involve splitting the data into training and testing sets and using metrics such as accuracy, sensitivity, specificity, or area under the receiver operating characteristic curve (AUC-ROC) to evaluate the model's performance.

6. Deployment: Finally, the developed and evaluated model can be deployed as a heart disease prediction system. This may involve integrating the model into a user-friendly interface, such as a web application or mobile app, that allows users to input their data and receive a personalized risk assessment for heart disease.

Overall, the methodology of a heart disease prediction system involves collecting and preprocessing data, extracting relevant features, developing and evaluating a predictive model, and deploying the model as a user-friendly system for predicting the risk of heart disease.

4.2 Method of Approach

The next step is to select a suitable machine learning algorithm for the task at hand. There are several types of machine learning algorithms that can be used for developing a Heart Disease Prediction System, including:

- Logistic Regression: This is a linear algorithm that is commonly used for binary classification tasks, such as predicting whether a patient is at high risk of heart disease.

- Decision Trees: This algorithm uses a tree-like structure to classify patients based on a series of decisions or rules. Decision trees are useful for tasks that require both classification and feature selection.

- Random Forest: This is an ensemble algorithm that combines multiple decision trees to make more accurate predictions. Random forest algorithms are particularly useful for tasks that require high accuracy and robustness.

Once the algorithm has been selected, it is trained using the preprocessed health data. The training process involves feeding the algorithm a set of labeled data (i.e., data with known outcomes) and allowing it to learn the patterns and relationships in the data. Once the algorithm has been trained, it can be used to make predictions about the likelihood of a patient developing heart disease based on their health data.

4.3 Algorithm Used

Logistic regression is basically a supervised classification algorithm. In a classification problem, the target variable (or output), y, can take only discrete values for a given set of features (or inputs), X. Contrary to popular belief, logistic regression is a regression model. The model builds a regression model to predict the probability that a given data entry belongs to the category numbered as “1”. Just like Linear regression assumes that the data follows a linear function, Logistic regression models the data using the sigmoid function.

Logistic regression becomes a classification technique only when a decision threshold is brought into the picture. The setting of the threshold value is a very important aspect of Logistic regression and is dependent on the classification problem itself.

In logistic regression, we generally compute the probability which lies between the interval 0 and 1 (inclusive of both). Then probability can be used to classify the data. For example, if the computed probability comes out to be greater than 0.5, then the data belonged to class A and otherwise, for less than 0.5, the data belonged to class B.

4.4 Comparing With Other Algorithms

Logistic regression, decision trees, and random forests are all popular algorithms used in machine learning. Each algorithm has its own strengths and weaknesses, and the choice of algorithm depends on the specific problem at hand. Here are some reasons why logistic regression may be preferred over decision trees and random forests for heart disease prediction:

1. Interpretability: Logistic regression produces a linear equation that can be easily understood and interpreted by humans. In contrast, decision trees and random forests can be more difficult to interpret, especially for large and complex datasets.

2. Regularization: Logistic regression can be regularized to prevent overfitting, which can occur when the model is too complex and fits the noise in the data instead of the underlying pattern. Decision trees and random forests can also be regularized, but the choice of hyperparameters can be more difficult and time-consuming.

3. Performance: Logistic regression is generally faster and requires less memory than decision trees and random forests, especially for large datasets. Logistic regression can also handle a large number of features, whereas decision trees and random forests may struggle with high-dimensional datasets.

4. Non-linear Relationships: Logistic regression can handle non-linear relationships between input variables and the output variable using polynomial or interaction terms. However, decision trees and random forests may be better suited for capturing non-linear relationships between variables.

Overall, logistic regression is a better choice for heart disease prediction if interpretability, regularization, and computational efficiency are important considerations.

4.5 Logistic Regression

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, etc. but instead of giving the exact value, it gives the probabilistic values which lie between 0 and 1.It is much similar to the Linear Regression except that how they are used. Linear Regression is used for solving Regression problems, whereas Logistic regression is used for solving the classification problems.

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

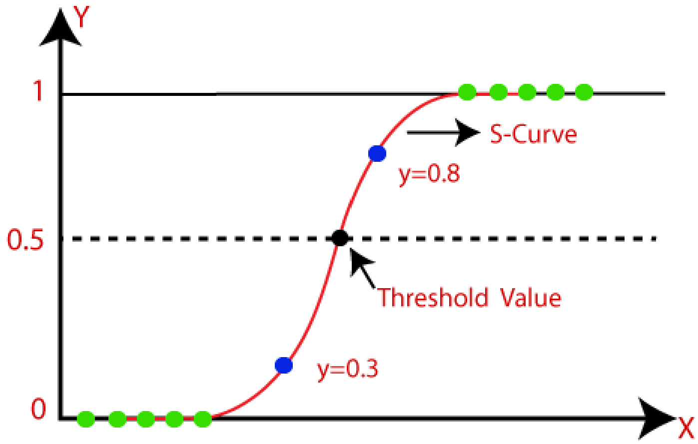

Fig 4.5.1 Logistic Function (Sigmoid Function)

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.

CHAPTER 5

CODING AND TESTING

5.1 Code

Importing the Dependencies

```
[ ] 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
```

Data Collection and Processing

```
[ ] # loading the csv data to a Pandas DataFrame
    heart_data = pd.read_csv('/content/data.csv')
```

```
[ ] # print first 5 rows of the dataset
    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

```
# print last 5 rows of the dataset
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

```
[ ] # number of rows and columns in the dataset
    heart_data.shape
```

```
(303, 14)
```

```python
# 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
```

```python
# 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
```

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

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
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	303.000000	303.000000
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	2.313531	0.544554
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	0.612277	0.498835
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%	47.500000	0.000000	0.000000	120.000000	211.000000	0.000000	0.000000	133.500000	0.000000	0.000000	1.000000	0.000000	2.000000	0.000000
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	2.000000	1.000000
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	3.000000	1.000000
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	3.000000	1.000000

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

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

1 --> Defective Heart

0 --> Healthy Heart

Splitting the Features and Target

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

```
print(X)
```

```
     age  sex  cp  trestbps  chol  ...  exang  oldpeak  slope  ca  thal
0     63    1   3       145   233  ...      0      2.3      0   0     1
1     37    1   2       130   250  ...      0      3.5      0   0     2
2     41    0   1       130   204  ...      0      1.4      2   0     2
3     56    1   1       120   236  ...      0      0.8      2   0     2
4     57    0   0       120   354  ...      1      0.6      2   0     2
..   ...  ...  ..       ...   ...  ...    ...      ...    ...  ..   ...
298   57    0   0       140   241  ...      1      0.2      1   0     3
299   45    1   3       110   264  ...      0      1.2      1   0     3
300   68    1   0       144   193  ...      0      3.4      1   2     3
301   57    1   0       130   131  ...      1      1.2      1   1     3
302   57    0   1       130   236  ...      0      0.0      1   1     2

[303 rows x 13 columns]
```

```
print(Y)
```

```
0      1
1      1
2      1
3      1
4      1
      ..
298    0
299    0
300    0
301    0
302    0
Name: target, Length: 303, dtype: int64
```

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)
```

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

Model Training

Logistic Regression

```
model = LogisticRegression()
```

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

5.2 Testing

Model Evaluation

Accuracy Score

```
[ ] # 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
```

Fig 5.2.1 Accuracy Testing

CHAPTER 6

SCREENSHOTS AND RESULTS

6.1 Dataset

	A	B	C	D	E	F	G	H	I	J	K	L
1	Age	Sex	ChestPainType	RestingBP	Cholesterol	FastingBS	RestingECG	MaxHR	ExerciseAngina	Oldpeak	ST_Slope	HeartDisease
2	40	M	ATA	140	289	0	Normal	172	N	0	Up	0
3	49	F	NAP	160	180	0	Normal	156	N	1	Flat	1
4	37	M	ATA	130	283	0	ST	98	N	0	Up	0
5	48	F	ASY	138	214	0	Normal	108	Y	1.5	Flat	1
6	54	M	NAP	150	195	0	Normal	122	N	0	Up	0
7	39	M	NAP	120	339	0	Normal	170	N	0	Up	0
8	45	F	ATA	130	237	0	Normal	170	N	0	Up	0
9	54	M	ATA	110	208	0	Normal	142	N	0	Up	0
10	37	M	ASY	140	207	0	Normal	130	Y	1.5	Flat	1
11	48	F	ATA	120	284	0	Normal	120	N	0	Up	0
12	37	F	NAP	130	211	0	Normal	142	N	0	Up	0
13	58	M	ATA	136	164	0	ST	99	Y	2	Flat	1
14	39	M	ATA	120	204	0	Normal	145	N	0	Up	0
15	49	M	ASY	140	234	0	Normal	140	Y	1	Flat	1
16	42	F	NAP	115	211	0	ST	137	N	0	Up	0
17	54	F	ATA	120	273	0	Normal	150	N	1.5	Flat	0
18	38	M	ASY	110	196	0	Normal	166	N	0	Flat	1
19	43	F	ATA	120	201	0	Normal	165	N	0	Up	0
20	60	M	ASY	100	248	0	Normal	125	N	1	Flat	1
21	36	M	ATA	120	267	0	Normal	160	N	3	Flat	1
22	43	F	TA	100	223	0	Normal	142	N	0	Up	0
23	44	M	ATA	120	184	0	Normal	142	N	1	Flat	0
24	49	F	ATA	124	201	0	Normal	164	N	0	Up	0
25	44	M	ATA	150	288	0	Normal	150	Y	3	Flat	1
26	40	M	NAP	130	215	0	Normal	138	N	0	Up	0
27	36	M	NAP	130	209	0	Normal	178	N	0	Up	0
28	53	M	ASY	124	260	0	ST	112	Y	3	Flat	0
29	52	M	ATA	120	284	0	Normal	118	N	0	Up	0
30	53	F	ATA	113	468	0	Normal	127	N	0	Up	0
31	51	M	ATA	125	188	0	Normal	145	N	0	Up	0

6.1.2 Factors Taken In Dataset

1. Age
2. Sex
3. Chest Pain Type
4. RestingBP – Resting Blood Pressure
5. Cholesterol
6. FastingBS - Fasting Blood Sugar
7. RestingECG - Resting Electrocardiogram
8. MaxHR - Max Heart Rate
9. Exercise Angina
10. Old Peak – ST depression induced by exercise relative to rest
11. ST Slope

6.2 Result

Building a Predictive System

```
[ ]  input_data = (62,0,0,140,268,0,0,160,0,3.6,0,2,2)

     # 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')
```

```
[0]
The Person does not have a Heart Disease
```

CHAPTER 7

CONCLUSION AND FUTURE ENHANCEMENTS

7.1 Conclusion

Heart disease prediction systems can play an important role in improving the diagnosis and treatment of cardiovascular diseases. With the help of machine learning algorithms such as logistic regression, healthcare professionals can identify patients at risk of heart disease and provide timely interventions to prevent or manage the condition.

However, developing an accurate heart disease prediction system requires careful consideration of various factors such as data quality, feature selection, algorithm selection, and model evaluation. It is also important to ensure the system's reliability, interpretability, and scalability, while maintaining patient privacy and data security.

Despite these challenges, heart disease prediction systems have the potential to significantly improve the health outcomes of patients and reduce healthcare costs. By leveraging the power of machine learning and data analytics, we can work towards a future where heart disease is effectively prevented and managed, leading to better quality of life for individuals and communities.

7.2 Future Enhancement

Incorporating more data sources: The prediction system can be enhanced by incorporating additional data sources, such as wearable devices, to provide a more comprehensive view of patient health.

Introducing more advanced machine learning techniques: The prediction system can be enhanced by incorporating more advanced machine learning techniques, such as deep learning, to improve the accuracy of the predictive model.

Integrating with electronic health records: The prediction system can be enhanced by integrating with electronic health records, to enable more seamless and efficient sharing of patient data.

REFERENCES

1. Alizadehsani, R., Habibi, J., Hosseini, M. J., Mashayekhi, H., & Hosseini, S. M. (2017). Prediction of coronary artery disease using machine learning algorithms. CVD Prevention and Control, 12(3), 77-83.

2. Krittanawong, C., Zhang, H., Wang, Z., Aydar, M., & Kitai, T. (2018). Artificial intelligence in precision cardiovascular medicine. Journal of the American College of Cardiology, 71(23), 2668-2679.

3. Prasad, K. S., Choudhary, S. K., Jaiswal, M., Singh, A. R., & Bharti, S. K. (2020). Machine learning in predicting coronary artery disease: a systematic review. Journal of Medical Systems, 44(9), 1-10.

4. Zhang, Y., Jiang, C., Wang, X., Cai, D., & Zhu, H. (2019). Predicting coronary artery disease risk factors using deep learning. IEEE Access, 7, 154982-154989.

5. Liu, M., Yin, Y., & Wang, Y. (2020). Application of machine learning in predicting coronary artery disease. Journal of Healthcare Engineering, 2020, 1-9.