Heart Disease Prediction using Machine Learning

Sai Satwik Yarapothini

Date: 25-05-2023

Abstract:

Heart disease remains a significant global health concern being the biggest cause of death around the world, necessitating accurate and timely diagnosis for effective treatment and prevention. In recent years, machine learning techniques have shown promise in assisting healthcare professionals with early detection and prediction of heart disease. This technical report presents a comprehensive analysis of appropriate machine learning algorithms applied to heart disease prediction, aiming to identify the most effective model for accurate and reliable diagnosis.

The study utilizes a publicly available dataset containing clinical and demographic features of patients, encompassing a diverse range of attributes such as age, gender, blood pressure, cholesterol levels, and electrocardiogram readings. It then trains an appropriate model for accurate prediction of cardiac diseases.

1. Problem Statement:

Our goal is to develop a Robust and Accurate Machine learning model for predicting the risk of heart disease in individuals.The primary goal of this study is to leverage machine learning algorithms and advanced data analysis techniques to develop a predictive model that can accurately classify individuals as either having a high or low risk of heart disease based on a set of relevant features and medical indicators.

2. Market/Customer Need Assessment

2.1. Market Need Assessment

Market need assessment reveals a strong demand for accurate, scalable, and personalized heart disease prediction models using Machine learning.

1. Growing Prevalence of Heart Disease: Heart disease continues to be a leading cause of unexpected deaths worldwide. There is a significant need for accurate prediction models that can identify individuals at high risk of heart disease to enable early intervention and preventive measures. Machine learning algorithms can offer improved accuracy and efficiency compared to traditional risk assessment methods.

2.Potential for Population Health Management: Machine learning models for heart disease prediction can contribute to population health management initiatives. By analyzing large-

scale health data, these models can identify high-risk populations, detect emerging trends, and support policymakers in developing targeted public health interventions and preventive strategies.

3. Personalized Medicine and Precision Health: Machine learning-based heart disease prediction models can contribute to the field of personalized medicine and precision health. By considering a wide range of individual attributes and genetic factors, these models can offer tailored risk assessment and treatment strategies, leading to more effective and targeted healthcare interventions.

4. Potential for Early Detection and Prevention: Early detection of heart disease risk factors can help individuals make necessary lifestyle modifications and receive appropriate medical interventions. Machine learning models can analyze a wide range of factors, including medical history, lifestyle choices, biomarkers, and genetic data, to provide early detection and preventive recommendations, empowering individuals to take proactive steps to reduce their risk.

2.2. Customer Need Assessment

Customers require an accurate, personalized, user-friendly, and trustworthy predictive model for heart disease risk assessment that can detect the disease as early as possible. Meeting these customer needs will result in improved health outcomes, informed decision-making, and enhanced healthcare delivery.

1.Accurate Risk Assessment: Customers, including individuals and healthcare professionals, have a crucial need for accurate risk assessment of heart disease. They require a reliable predictive model that can accurately classify individuals as either having a high or low risk of heart disease based on their specific health data. The model should provide precise risk probabilities to guide decision-making and intervention strategies.

2. Early Detection and Prevention: Customers seek early detection and prevention of heart disease to minimize its impact on health and well-being. They need a predictive model that can identify potential risk factors and early signs of heart disease, allowing for timely intervention, lifestyle modifications, and targeted preventive measures. Early detection can help avoid complications and improve long-term health outcomes.

3. Personalized Risk Assessment: Customers desire a personalized risk assessment approach that takes into account their unique characteristics, including medical history, lifestyle choices, genetic factors, and biomarkers. They want a machine learning model that can analyze these individual attributes and provide tailored risk predictions and recommendations, considering their specific circumstances and potential risk factors.

4. Easy-to-Use and User-Friendly Interface: Customers, including healthcare professionals, value an intuitive and user-friendly interface for accessing and interacting with the predictive model. They need a platform or application that allows for easy input of relevant health

data, provides clear and understandable risk assessment results, and offers user-friendly visualization tools to aid in interpreting and communicating the predictions effectively.

3. Target Specifications and Characterization

By below target specifications and characterizations, my Product idea would get proper gateway to the final goal:

- Dataset Description: Provide a detailed description of the dataset used for training and evaluating the machine learning model. Specify the number of instances, variables, and the nature of the features, including demographic, clinical, lifestyle, and genetic attributes. Highlight any preprocessing steps applied to the dataset, such as handling missing values, data normalization, or outlier removal.

- Machine Learning Algorithms: Specify the machine learning algorithms utilized for heart disease prediction. Discuss the rationale behind the selection of these algorithms and their suitability for the task. Include algorithms like logistic regression, decision trees, random forests, support vector machines, or neural networks. Justify the choice of each algorithm based on its strengths and relevance to the problem at hand.

- Evaluation Metrics: Define the evaluation metrics used to assess the performance of the predictive model. Common metrics for classification tasks include accuracy, precision, recall, F1 score, and area under the receiver operating characteristic curve (AUC-ROC). Discuss the interpretation and significance of each metric and explain how they reflect the model's ability to predict heart disease accurately.

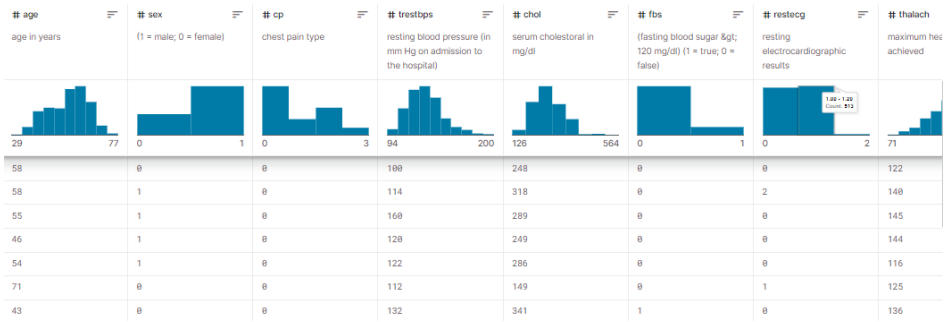
- Model Training and Validation: Describe the methodology employed for model training and validation. Discuss the use of cross-validation techniques, such as k-fold cross-validation, to ensure robustness and generalizability of the model. Explain the training/validation split ratio and any hyperparameter tuning performed to optimize the model's performance.

- Deployment Considerations: Address the deployment considerations and practical implications of implementing the predictive model in real-world healthcare settings. Discuss aspects such as scalability, computational requirements, integration with existing healthcare systems, and regulatory compliance. Evaluate the feasibility of integrating the model into telemedicine platforms or wearable devices for remote monitoring and personalized risk assessment.

- Ethical and Privacy Considerations: Discuss ethical considerations related to the use of personal health data for heart disease prediction. Address concerns related to data privacy, informed consent, and potential biases in the dataset or model predictions.

4. External Search [Online information references/links]

- The dataset for training the model is Heart Dataset which is gathered from four databases: Cleveland, Hungary, Switzerland, and Long Beach V and can be found from Kaggle or any other online sources . The current dataset which is used in this report is obtained from Kaggle in csv format.
  Link: https://www.kaggle.com/datasets/johnsmith88/heart-disease-dataset?resource=download



# age age in years	# sex (1 = male; 0 = female)	# cp chest pain type	# trestbps resting blood pressure (in mm Hg on admission to the hospital)	# chol serum cholestoral in mg/dl	# fbs (fasting blood sugar > 120 mg/dl) (1 = true; 0 = false)	# restecg resting electrocardiographic results	# thalach maximum hea achieved
29 77	0 1	0 3	94 200	126 564	0 1	0 2	71
58	0	0	100	248	0	0	122
58	1	0	114	318	0	2	140
55	1	0	160	289	0	0	145
46	1	0	120	249	0	0	144
54	1	0	122	286	0	0	116
71	0	0	112	149	0	1	125
43	0	0	132	341	1	0	136

- Improvements in making the product even better and accurate than the pre-planned product are made using the following references with links attached below:
  1. https://towardsdatascience.com/predicting-presence-of-heart-diseases-using-machine-learning-36f00f3edb2c?gi=ec21c6b32b0b
  2. https://www.analyticsvidhya.com/blog/2022/02/heart-disease-prediction-using-machine-learning/
  3. https://iopscience.iop.org/article/10.1088/1757-899X/1022/1/012072/meta
- The Machine learning model is made more accurate using the following reference:
  https://youtu.be/lrXhh7Ffmbo

5. Bench marking alternate products

In addition to developing the own Machine learning model for heart disease prediction, it is essential to benchmark your solution against existing alternate products or approaches in order to assess its performance and identify the unique advantages and disadvantages of the new Machine Learning model. Some of the existing Machine learning software that are used to predict heart diseases are:

✓ The Framingham Heart Study Risk Score: While not a machine learning model per se, the Framingham Risk Score is a widely used statistical model for predicting the 10-year risk of developing cardiovascular disease. It incorporates factors such as age, sex, blood pressure, cholesterol levels, smoking status, and diabetes status. The score is available as an online calculator and is commonly used in clinical practice.

- ✓ Zebra Medical Vision: It utilizes machine learning algorithms for medical imaging analysis. Their deep learning models can analyze medical images, such as chest X-rays, to detect abnormalities related to heart diseases, including coronary artery disease and heart failure. Their software aims to assist radiologists and healthcare professionals in interpreting images and identifying potential cardiovascular conditions.

- ✓ Aidoc: It applies machine learning algorithms to medical imaging data for automated detection and prioritization of critical findings. Their software can analyze CT scans and identify signs of conditions related to heart diseases, including aortic dissection, pulmonary embolism, or cardiac masses. The software aims to assist radiologists in the timely detection and diagnosis of cardiovascular abnormalities.

- ✓ AliveCor: It has developed a mobile ECG device called KardiaMobile, which captures electrocardiogram (ECG) readings using a smartphone. Their machine learning algorithms analyze the ECG data to detect abnormalities such as arrhythmias and provide indications of potential heart conditions. The software helps individuals track their heart health and share the results with healthcare professionals for diagnosis and treatment planning.

Our aim is to build a Heart disease prediction product which comprises of useful features from the above benchmarked products and give accurate results to the customers accordingly.

6. Applicable Patents

The Patentability of software and algorithms are subject to specific requirements and guidelines set by the Indian Patent Office. It is important to assess whether your software meets the criteria for patentability, prepare a detailed patent application describing your software invention then file the patent application with the Indian Patent Office.

Patents that are applicable in this product are:

1. Anaconda Navigator Patent for developing the Software on Jupyter Notebook

2. Patents of any pre-existing libraries & algorithms if any

7. Applicable Regulations

The development of machine learning software in India is subject to various regulations and guidelines set by the Indian government. Important regulations include:

- ✓ Intellectual Property Rights (IPR): Protecting intellectual property of a unique software is crucial when developing machine learning software. In India, software and algorithms can be protected through copyright laws. It is advisable to ensure that you have proper documentation, including agreements with developers and employees, to establish ownership and protection of your software's intellectual property.

- ✓ Data Protection and Privacy: Machine learning software often relies on large amounts of data for training and prediction. It is essential to comply with data protection and privacy laws in India, such as the Information Technology (Reasonable Security Practices and Procedures and Sensitive Personal Data or Information) Rules, 2011. It is important to obtain appropriate consent for data collection, storage, and usage and implement necessary security measures to protect user data.

- ✓ Regulatory Compliance: Depending on the application and use case of the machine learning software, it needs to comply with specific regulatory frameworks. Current software comes under healthcare sector, it may need to adhere to regulations set by the Central Drugs Standard Control Organization (CDSCO) or the Medical Council of India (MCI).

- ✓ Ethical and Bias Considerations: Machine learning algorithms and models should be developed with ethical considerations in mind, ensuring fairness, transparency, and accountability. It is essential to avoid biases in the data used for training and testing, as well as in the design and implementation of the algorithms. Following ethical guidelines and frameworks, such as those provided by NITI Aayog's Responsible AI for All, can help ensure responsible development and deployment of machine learning software.

- ✓ Import and Export Regulations: If your machine learning software involves the import or export of specific technologies, data, or components, you should comply with the relevant import/export regulations set by the Directorate General of Foreign Trade (DGFT) and other concerned authorities.

8. Business Model[Monetization Idea]

The Monetization idea for this Heart Disease Prediction System is to partner with healthcare providers or insurance companies and offer the system as a value-added service. Implementation of the monetization idea is done as follows:

- Partnership with Healthcare Providers: Product Developers collaborate with hospitals, clinics, or healthcare networks to integrate the heart disease prediction system which are offered as premium service that healthcare providers can provide to their patients for heart disease risk assessment and monitoring into their existing patient management infrastructure.

  - ✓ We monetize this by charging healthcare providers a licensing fee or a percentage of the revenue generated from the system's usage.

- Collaboration with Insurance Companies: Product Developers partner with insurance providers to incorporate the heart disease prediction system as part of their risk assessment process where Insurance companies can offer policyholders the option to

undergo heart disease risk assessment using the system, allowing for more accurate premium calculations.

    - We monetize this by negotiating revenue-sharing agreements with insurance companies based on the number of policyholders utilizing the system or the reduction in claims due to early detection.

- Data Analytics and Research: Product Developers will Aggregate and anonymize the data generated from the heart disease prediction system to provide valuable insights and research findings to pharmaceutical companies, researchers, or public health organizations.

    - We monetize this by collaborating with these entities by offering data analytics services, customized research reports, or access to specific datasets Charge fees based on the scope and complexity of the analytics or research services provided.

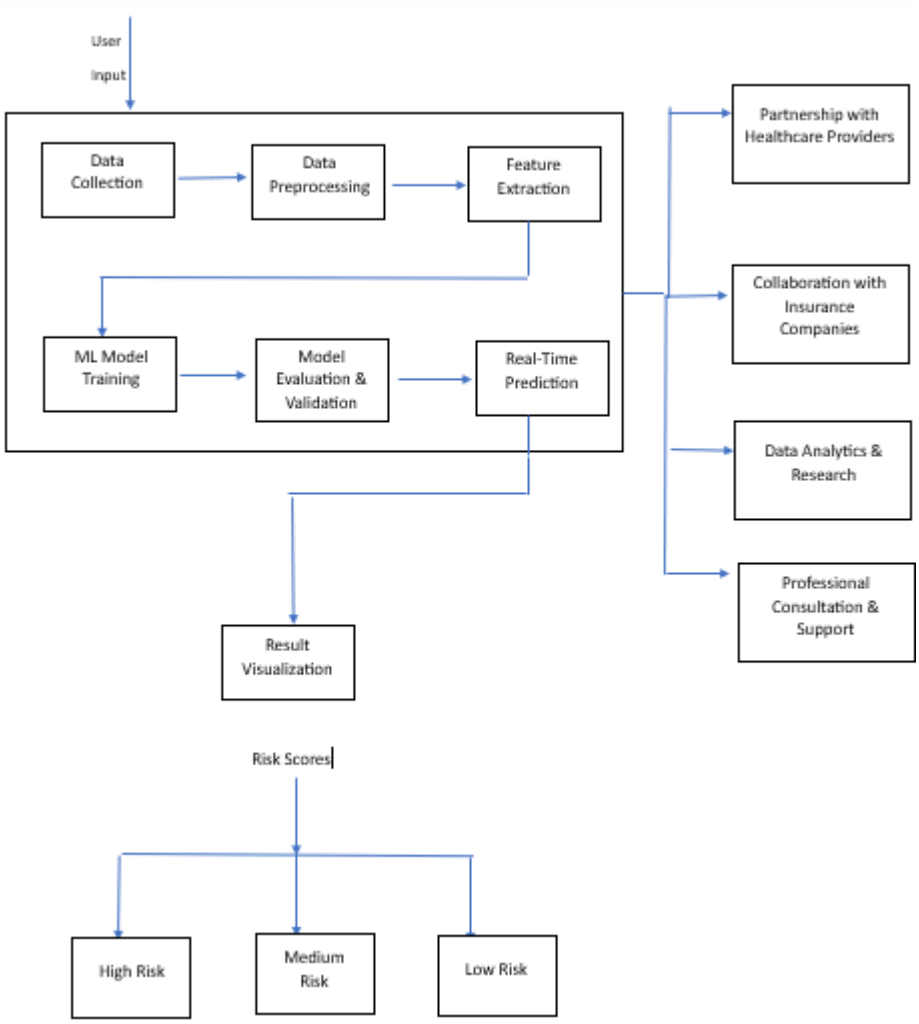
- Professional Consultation and Support: Product Developers offer additional services such as expert consultation, personalized risk assessment reports, or telemedicine consultations based on the heart disease prediction system's results.

    - We monetize this by charging a fee for these value-added services, providing users with access to healthcare professionals who can provide guidance, answer questions, and offer recommendations based on the system's predictions.

The team will conduct constant market research and assess the specific needs and preferences of your target customers to tailor the monetization strategy accordingly.

9. Final Product Prototype with Schematic Diagram

The Schematic diagram representing the final product prototype only provides a high-level overview of the product implementation and the actual implementation may involve additional components, algorithms, and considerations specific to the heart disease prediction system which is being developed.



The explanation of components of the Heart disease prediction system in this schematic diagram is as follows:

- Data Collection: System collects relevant data from various sources, such as electronic health records (EHR), patient demographics, medical imaging, laboratory test results, and lifestyle data.
    - This data is stored in a secure database for further processing and analysis.
- Data Preprocessing: Collected data goes through preprocessing steps to clean and transform it into a suitable format for machine learning.
    - Preprocessing may involve data normalization, feature selection, handling missing values, and addressing data quality issues.
- Feature Extraction: Relevant features are extracted from the preprocessed data to represent the patient's health status.
    - Feature extraction techniques can include statistical measures, signal processing algorithms, or domain-specific algorithms.
- Machine Learning Model Training: The extracted features are used to train a machine learning model, such as a classification algorithm (e.g., logistic regression, random forest, or support vector machine).

    - The model learns patterns and relationships from the training data to predict the likelihood of heart disease.
- Model Evaluation and Validation: The trained model is evaluated using validation datasets to assess its performance and accuracy.
    - Evaluation metrics, such as accuracy, precision, recall, and F1 score, are used to measure the model's predictive capabilities.
- Real-time Prediction: Once the model is trained and validated, it can be deployed to make real-time predictions. New patient data or input features are fed into the trained model, and the model generates a heart disease prediction or risk score.
- Result Visualization and Reporting: System presents the prediction results to healthcare professionals or individuals in a user-friendly format, such as a graphical dashboard or report.

    ✓ The visualization may include risk scores, probability estimates, and other relevant information for informed decision-making.

10. Code Implementation on Small Scale

10.1. Exploratory Data Analysis(EDA)

- The dataset used in training this model can be obtained from Kaggle website and consists of the following attributes:

1. age
2. sex
3. chest pain type (4 values)
4. resting blood pressure
5. serum cholestoral in mg/dl
6. fasting blood sugar > 120 mg/dl
7. resting electrocardiographic results (values 0,1,2)
8. maximum heart rate achieved
9. exercise induced angina
10. oldpeak = ST depression induced by exercise relative to rest
11. the slope of the peak exercise ST segment
12. number of major vessels (0-3) colored by flourosopy
13. thal: 0 = normal; 1 = fixed defect; 2 = reversable defect

The names and social security numbers of the patients were recently removed from the database, replaced with dummy values.

- We then perform the steps of EDA in order to train the Machine learning model with suitable data in format by the use of pre-existing Python libraries.

Importing Libraries

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

Import the Dataset

```
df=pd.read_csv("https://raw.githubusercontent.com/PRIYANG-BHATT/Datasets/main/DS/heart.csv")
```

-Getting the overview of the dataset using suitable methods

Getting Insights of Dataset

```
print(df.head())
print(df.shape)
print(df.info())
```

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

   ca  thal  target
0   2     3       0
1   0     3       0
2   0     3       0
3   1     3       0
4   3     2       0
(1025, 14)
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1025 entries, 0 to 1024
Data columns (total 14 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   age       1025 non-null   int64
 1   sex       1025 non-null   int64
 2   cp        1025 non-null   int64
 3   trestbps  1025 non-null   int64
 4   chol      1025 non-null   int64
 5   fbs       1025 non-null   int64
 6   restecg   1025 non-null   int64
 7   thalach   1025 non-null   int64
 8   exang     1025 non-null   int64
 9   oldpeak   1025 non-null   float64
 10  slope     1025 non-null   int64
 11  ca        1025 non-null   int64
 12  thal      1025 non-null   int64
 13  target    1025 non-null   int64
dtypes: float64(1), int64(13)
memory usage: 112.2 KB
None
```

Data Filtering

Checking Missing Values

```
print(df.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
```

Checking Duplicate Values

Checking Duplicate Values

```
df_dup=df.duplicated().any()
print(df_dup)
```

```
True
```

```
#Removing Duplicates in the Data
df=df.drop_duplicates()
```

```
df_dup=df.duplicated().any()
print(df_dup)
```

```
False
```

Data Preprocessing

```
cate_val=[]
nume_val=[]
for column in df.columns:
    if df[column].nunique() <=10:
        cate_val.append(column)
    else:
        nume_val.append(column)
```

```
cate_val
```

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

```
nume_val
```

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

```
cate_val.remove('sex')
cate_val.remove('target')
```

```
df=pd.get_dummies(df,columns=cate_val,drop_first=True)
```

```
df.head()
```

	age	sex	trestbps	chol	thalach	oldpeak	target	cp_1	cp_2	cp_3	...	exang_1	slope_1	slope_2	ca_1	ca_2	ca_3	ca_4	thal_1	thal_2	thal_3
0	52	1	125	212	168	1.0	0	0	0	0	...	0	0	1	0	1	0	0	0	0	1
1	53	1	140	203	155	3.1	0	0	0	0	...	1	0	0	0	0	0	0	0	0	1
2	70	1	145	174	125	2.6	0	0	0	0	...	1	0	0	0	0	0	0	0	0	1
3	61	1	148	203	161	0.0	0	0	0	0	...	0	0	1	1	0	0	0	0	0	1
4	62	0	138	294	106	1.9	0	0	0	0	...	0	1	0	0	0	1	0	0	1	0

5 rows × 23 columns

Feature Scaling

```
df.head()
```

	age	sex	trestbps	chol	thalach	oldpeak	target	cp_1	cp_2	cp_3	...	exang_1	slope_1	slope_2	ca_1	ca_2	ca_3	ca_4	thal_1	thal_2	thal_3
0	52	1	125	212	168	1.0	0	0	0	0	...	0	0	1	0	1	0	0	0	0	1
1	53	1	140	203	155	3.1	0	0	0	0	...	1	0	0	0	0	0	0	0	0	1
2	70	1	145	174	125	2.6	0	0	0	0	...	1	0	0	0	0	0	0	0	0	1
3	61	1	148	203	161	0.0	0	0	0	0	...	0	0	1	1	0	0	0	0	0	1
4	62	0	138	294	106	1.9	0	0	0	0	...	0	1	0	0	0	1	0	0	1	0

5 rows × 23 columns

```
#Feature Scaling for Distance based algorithms
from sklearn.preprocessing import StandardScaler
st=StandardScaler()
df[nume_val]=st.fit_transform(df[nume_val])
df.head()
```

	age	sex	trestbps	chol	thalach	oldpeak	target	cp_1	cp_2	cp_3	...	exang_1	slope_1	slope_2	ca_1	ca_2	ca_3	ca_4	thal_1	thal_2
0	-0.267966	1	-0.376556	-0.667728	0.806035	-0.037124	0	0	0	0	...	0	0	1	0	1	0	0	0	0
1	-0.157260	1	0.478910	-0.841918	0.237495	1.773958	0	0	0	0	...	1	0	0	0	0	0	0	0	0
2	1.724733	1	0.764066	-1.403197	-1.074521	1.342748	0	0	0	0	...	1	0	0	0	0	0	0	0	0
3	0.728383	1	0.935159	-0.841918	0.499898	-0.899544	0	0	0	0	...	0	0	1	1	0	0	0	0	0
4	0.839089	0	0.364848	0.919336	-1.905464	0.739054	0	0	0	0	...	0	1	0	0	0	1	0	0	1

5 rows × 23 columns

Splitting the Dataset into Train & Test Set

```
X=df.drop('target',axis=1)
y=df['target']
```

```
X
```

	age	sex	trestbps	chol	thalach	oldpeak	cp_1	cp_2	cp_3	fbs_1	...	exang_1	slope_1	slope_2	ca_1	ca_2	ca_3	ca_4	thal_1	thal_2
0	-0.267966	1	-0.376556	-0.667728	0.806035	-0.037124	0	0	0	0	...	0	0	1	0	1	0	0	0	0
1	-0.157260	1	0.478910	-0.841918	0.237495	1.773958	0	0	0	1	...	1	0	0	0	0	0	0	0	0
2	1.724733	1	0.764066	-1.403197	-1.074521	1.342748	0	0	0	0	...	1	0	0	0	0	0	0	0	0
3	0.728383	1	0.935159	-0.841918	0.499898	-0.899544	0	0	0	0	...	0	0	1	1	0	0	0	0	0
4	0.839089	0	0.364848	0.919336	-1.905464	0.739054	0	0	0	1	...	0	1	0	0	0	1	0	0	1
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
723	1.503322	0	-0.661712	-0.687083	-1.511859	0.394086	0	1	0	0	...	0	1	0	0	0	0	0	0	1
733	-1.153610	0	-1.346085	-2.041893	1.112172	-0.382092	0	1	0	0	...	0	1	0	0	0	0	0	0	1
739	-0.267966	1	-0.205463	0.164513	0.499898	-0.899544	0	0	0	0	...	1	0	1	1	0	0	0	0	0
843	0.506972	1	1.619532	0.512893	-1.074521	-0.899544	0	0	1	0	...	0	0	1	0	0	0	0	0	1
878	-0.046555	1	-0.661712	-1.132235	-1.599327	0.307844	0	0	0	0	...	0	1	0	1	0	0	0	0	0

302 rows × 22 columns

```
y
```

```
0      0
1      0
2      0
3      0
4      0
      ..
723    1
733    1
739    0
843    0
878    0
Name: target, Length: 302, dtype: int64
```

```
from sklearn.model_selection import train_test_split
X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.2,random_state=42)
```

10.2. Machine Learning Modelling

We now train different classifiers using different classification algorithms in order to choose which one is the most appropriate for the given data:

Implementing Different Classification Algorithms

1. Logistic Regression

```
from sklearn.linear_model import LogisticRegression
log = LogisticRegression()
log.fit(X_train,y_train)
```

```
LogisticRegression()
```

```
y_pred1=log.predict(X_test)
```

```
from sklearn.metrics import accuracy_score
print(accuracy_score(y_test,y_pred1) *100)
```

```
78.68852459016394
```

2. Support Vector Classifier(SVC)

```
from sklearn import svm
svm=svm.SVC()
svm.fit(X_train,y_train)
```

```
SVC()
```

```
y_pred2=svm.predict(X_test)
```

```
print(accuracy_score(y_test,y_pred2) *100)
```

```
80.32786885245902
```

3. KNeighbors Classifier

```
from sklearn.neighbors import KNeighborsClassifier
knn=KNeighborsClassifier()
knn.fit(X_train,y_train)
```

```
KNeighborsClassifier()
```

```
y_pred3=knn.predict(X_test)
```

```
print(accuracy_score(y_test,y_pred3) *100)
```

```
73.77049180327869
```

```
score=[]
for k in range(1,40):
    knn=KNeighborsClassifier(n_neighbors=k)
    knn.fit(X_train,y_train)
    y_pred=knn.predict(X_test)
    score.append(accuracy_score(y_test,y_pred))
```

```
print("The Maximum accuracy available in k-nearest neighbors is :" ,max(score)*100)
```

```
The Maximum accuracy available in k-nearest neighbors is : 80.32786885245902
```

4. Non-Linear ML Algorithms

```
df=pd.read_csv("https://raw.githubusercontent.com/PRIYANG-BHATT/Datasets/main/DS/heart.csv")
print(df.head())
print(df.shape)
```

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

   ca  thal  target
0   2     3       0
1   0     3       0
2   0     3       0
3   1     3       0
4   3     2       0
(1025, 14)
```

```
df=df.drop_duplicates()
print(df.shape)
```

```
(302, 14)
```

```
#Seggregating Dataset into Independent & Dependent Variable
X=df.drop('target',axis=1)
y=df['target']
from sklearn.model_selection import train_test_split
X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.2,random_state=42)
```

a. Decision Tree Classifier

```
from sklearn.tree import DecisionTreeClassifier
dt= DecisionTreeClassifier()
dt.fit(X_train,y_train)
```

```
DecisionTreeClassifier()
```

```
y_pred4=dt.predict(X_test)
```

```
print(accuracy_score(y_test,y_pred4) *100)
```

```
70.49180327868852
```

b. Random Forest Classifier

```
from sklearn.ensemble import RandomForestClassifier
rf= RandomForestClassifier()
rf.fit(X_train,y_train)
```

```
RandomForestClassifier()
```

```
y_pred5=rf.predict(X_test)
```

```
print(accuracy_score(y_test,y_pred5) *100)
```

```
85.24590163934425
```

c. Gradient Boosting Classifier

```
from sklearn.ensemble import GradientBoostingClassifier
gbc = GradientBoostingClassifier()
gbc.fit(X_train,y_train)
```

```
GradientBoostingClassifier()
```

```
y_pred6=gbc.predict(X_test)
```

```
print(accuracy_score(y_test,y_pred6) *100)
```

```
80.32786885245902
```

10.3. Data Visualization

After training the models with different classifiers we choose the best classification algorithm by the use of Data Visualization.

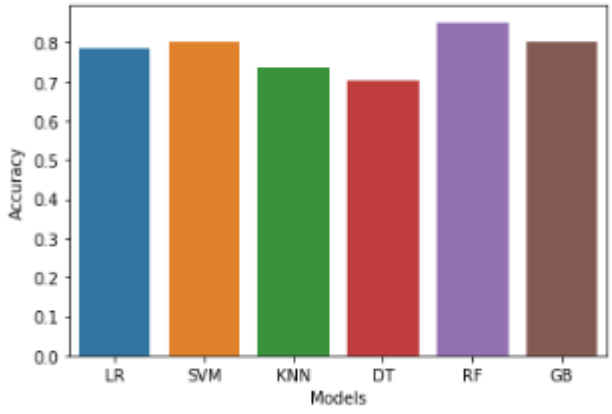
Choosing Best Classifier

```
final_df = pd.DataFrame({'Models':['LR','SVM','KNN','DT','RF','GB'],'Accuracy':[accuracy_score(y_test,y_pred1),
                                                                                 accuracy_score(y_test,y_pred2),
                                                                                 accuracy_score(y_test,y_pred3),
                                                                                 accuracy_score(y_test,y_pred4),
                                                                                 accuracy_score(y_test,y_pred5),
                                                                                 accuracy_score(y_test,y_pred6)]})
print(final_df)
```

```
  Models  Accuracy
0     LR  0.786885
1    SVM  0.803279
2    KNN  0.737705
3     DT  0.704918
4     RF  0.852459
5     GB  0.803279
```

```
#Visulaizing the Comparison between Different Classifiers & Choosing the Best one
sns.barplot(final_df['Models'],final_df['Accuracy'])
```

```
<AxesSubplot:xlabel='Models', ylabel='Accuracy'>
```



We can now choose the Random Forest Classification in order to train the model with whole available data and predict Heart disease for new data.

Training Random Forest Classifier on Entire Dataset

```
X=df.drop('target',axis=1)
y=df['target']
from sklearn.ensemble import RandomForestClassifier
rf= RandomForestClassifier()
rf.fit(X,y)
```

```
RandomForestClassifier()
```

Prediction on New Data using RFC

```
new_data=pd.DataFrame({
    'age': 52,
    'sex ': 1,
    'cp': 0,
    'trestbps': 125,
    'chol': 212,
    'fbs': 0,
    'restecg':1,
    'thalach':168,
    'exang': 0,
    'oldpeak': 1.0,
    'slope': 2,
    'ca': 2,
    'thal':3,
},index=[0])
print(new_data)
```

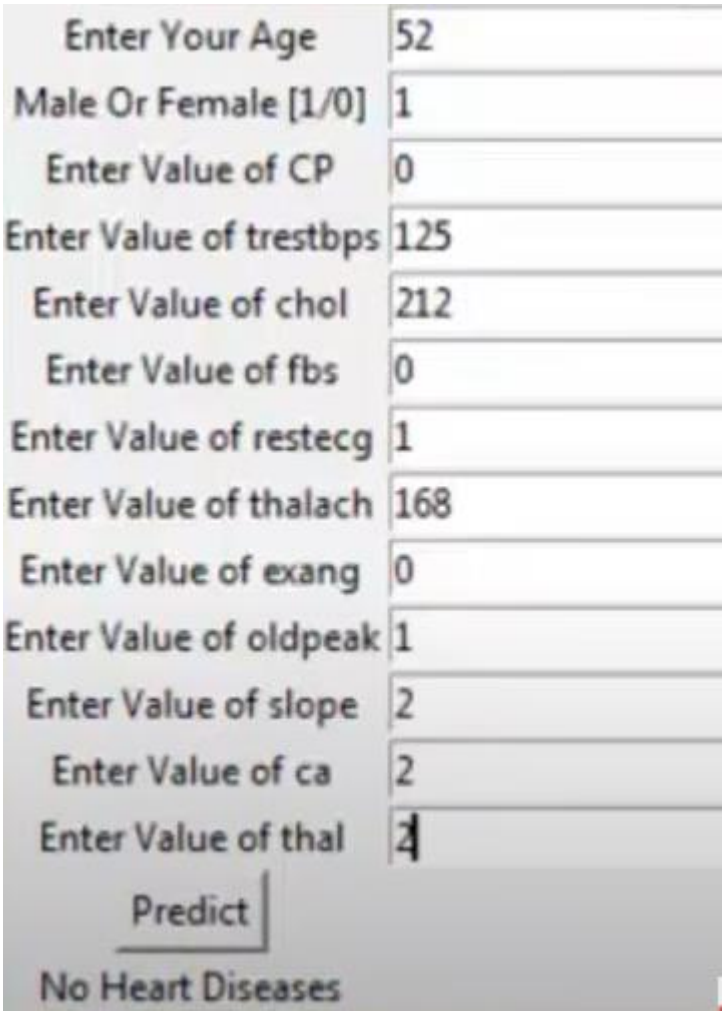
```
   age  sex   cp  trestbps  chol  fbs  restecg  thalach  exang  oldpeak  \
0   52     1   0       125   212    0        1      168      0      1.0

   slope  ca  thal
0      2   2     3
```

```
p=rf.predict(new_data)
if p[0]==0:
    print("No Disease")
else:
    print("Disease")
```

```
No Disease
```

The final output we get in order to predict the Heart Disease is generated from the GUI in Anaconda Navigator which is as follows:



✓ The GitHub link to the code implementation is attached below for further references:

https://github.com/satwikwrites7/My-Machine-Learning-Projects-.git

11. Conclusion

In conclusion, heart disease prediction using machine learning holds great promise for transforming cardiovascular healthcare. By harnessing the power of data and advanced algorithms, we can make significant strides in preventing, diagnosing, and managing heart diseases, ultimately leading to a healthier population and improved quality of life. Throughout the report, we discussed the importance of heart disease prediction, the challenges associated with traditional approaches, and the potential benefits of adopting machine learning techniques. We examined the various stages of developing a heart disease prediction system, including data collection, preprocessing, feature extraction, model training, and real-time prediction. We need to acknowledge the potential impact of heart disease prediction using machine learning and explore it even more for secure mankind.