A Project Report

On

Heart Disease Prediction Using Random Forest

Submitted in partial fulfillment of the
requirement for the award of the degree of

BACHELOR OF COMPUTER APPLICATIONS

GALGOTIAS UNIVERSITY

(Established under Galgotias University Uttar Pradesh Act No. 14 of 2011)

Session 2023-24
in

Machine Learning

By
Anurag Kumar Jha
Roshni Giri
Tejasvini Arya

21SCSE1430008
21SCSE1430030
21SCSE1430032

Under the guidance of
Ms. Pragati Gupta (Assistant Professor)

SCHOOL OF COMPUTER APPLICATION AND TECHNOLOGY

GALGOTIAS UNIVERSITY, GREATER NOIDA

INDIA

April,2024

SCHOOL OF COMPUTER APPLICATION AND TECHNOLOGY
GALGOTIAS UNIVERSITY, GREATER NOIDA

CANDIDATE'S DECLARATION

We hereby certify that the work which is being presented in the project, entitled **"Heart Disease Prediction Using Random Forest"** in partial fulfillment of the requirements for the award of the BCA (Bachelor of Computer Applications) submitted in the School of Computer Application and Technology of Galgotias University, Greater Noida, is an original work carried out during the period of August, 2023 to April 2024, under the supervision of '**Ms. Pragati Gupta',** Department of Computer Science and Engineering/School of Computer Application and Technology , Galgotias University, Greater Noida.

The matter presented in the thesis/project/dissertation has not been submitted by me/us for the award of any other degree of this or any other places.

Anurag Kumar Jha (21SCSE1430008)

Roshni Giri (21SCSE1430030)

Tejasvini Arya (21SCSE1430032)

This is to certify that the above statement made by the candidates is correct to the best of my knowledge.

Ms. Pragati Gupta

Assistant Professor

CERTIFICATE

This is to certify that Project Report entitled **“Heart Disease Prediction Using Random Forest”** which is submitted by **“Anurag Kumar Jha, Roshni Giri, Tejasvini Arya”** in partial fulfillment of the requirement for the award of degree BCA in Department of School of Computer Application and Technology ,Galgotias University, Greater Noida, India is a record of the candidate own work carried out by them under my supervision. The matter embodied in this thesis is original and has not been submitted for the award of any other degree.

Signature of Examiner(s) **Signature of Supervisor(s)**

Date:

Place: Greater Noida

ABSTRACT

Heart disease remains a significant global health concern, and early prediction plays a pivotal role in effective prevention and management. This project leverages machine learning techniques to develop a heart disease prediction model using a comprehensive dataset encompassing critical patient attributes. The dataset includes features such as age, sex, chest pain type, resting blood pressure, cholesterol levels, fasting blood sugar, smoking history etc. The primary objective is to create a robust predictive model capable of identifying individuals at risk of heart disease based on these input variables. Random Forest will be explored to evaluate its effectiveness in predicting heart disease outcomes. The Random Forest algorithm was selected due to its capability to handle complex, high-dimensional data and provide robust predictive performance. Our study involves preprocessing the data, handling missing values to ensure the quality and relevance of attributes. The dataset is then divided into training and testing sets to evaluate the model's performance.

The outcomes of this project include the development of a predictive model that can estimate the probability of an individual suffering from heart disease. We assess the importance of each attribute in the prediction process, providing insights into which factors have the most significant impact on heart disease risk. This information can be valuable for healthcare professionals and policymakers in identifying at-risk individuals and implementing preventive measures.

By leveraging this algorithm, we aim to evaluate the contribution of it in predicting accuracy, providing valuable insights into the factors influencing heart disease. This project seeks to contribute to the field of healthcare by providing a data-driven approach to heart disease prediction, assisting healthcare professionals in early diagnosis and intervention.

TABLE OF CONTENTS

LIST OF FIGURES

ACRONYMS

ML	Machine Learning
RF	Random Forest
EDA	Exploratory Data Analysis
sklearn	Scikit learn
pd	Pandas
NumPy	Numerical Python
Matplotlib	Mathematical Plotting Library
GUI	Graphical User Interface
cp	Chest pain
trestbps	Resting blood pressure
chol	Cholesterol level
fbs	Fasting blood sugar level
thalach	Maximum heart rate achieved during exercise
restecg	Resting electrocardiographic results
exang	Exercise-induced angina
ca	Number of major vessels colored by fluoroscopy
thal	Thallium stress test result

CHAPTER-1

INTRODUCTION

1.1 Introduction

Heart disease remains a significant global health concern, contributing to a substantial number of preventable deaths each year. Timely and accurate prediction of heart disease risk is crucial for early intervention and improved patient outcomes. In this era of advanced data science and machine learning techniques, predictive models offer a promising avenue for identifying individuals at high risk of heart disease. This project aims to develop a Heart Disease Prediction System using the Random Forest algorithm, leveraging a comprehensive dataset that includes a wide array of features, such as age, sex, chest pain type, resting blood pressure, cholesterol levels, fasting blood sugar, smoking history, and more. Additionally, the system is designed with a user-friendly graphical user interface (GUI) to ensure accessibility and ease of use for healthcare professionals and individuals interested in assessing their heart disease risk.

Cardiovascular diseases are multifaceted, often influenced by a complex interplay of factors. Traditional risk assessment methods, while valuable, have limitations in terms of accuracy and efficiency. Machine learning, particularly the Random Forest algorithm, has shown promise in providing more accurate predictions by learning from extensive datasets, considering interactions between various risk factors, and adapting to changing patterns over time. By analyzing a diverse set of features, our model will be able to provide more precise and individualized risk assessments.

1.2 Formulation of Problem

This project's core challenge is to develop a precise heart disease risk prediction model using the Random Forest algorithm. This model utilizes a variety of health-related features, such as age, sex, chest pain type, blood pressure, cholesterol levels, fasting blood sugar, and smoking history, to estimate an individual's likelihood of developing heart disease.

The objective of this project is to leverage the Random Forest machine learning algorithm to predict the risk of heart disease in individuals based on a set of medical and demographic features. Early detection of heart disease is crucial for timely intervention and prevention. By creating a robust predictive model, the project aims to assist healthcare professionals and individuals in assessing their risk factors for heart disease.

The project's ultimate objective is to develop a user-friendly application or interface that enables users to input their personal information and receive a reliable assessment of their heart disease risk. Such a tool could have a significant impact on public health by aiding in the early identification of individuals at risk, allowing them to make informed decisions about their lifestyle and healthcare, and potentially reducing the burden of heart disease on society.

1.2.1 Tools and Programming Language Used

In this project, the primary technology utilized is the Random Forest algorithm for heart disease prediction. Random Forest is a powerful machine learning algorithm known for its ability to handle classification tasks effectively, making it well-suited for predicting heart disease outcomes based on a set of attributes. Random Forest leverages the strength of multiple decision trees to create a robust and accurate predictive model.

Random Forest is a popular machine learning algorithm that belongs to the supervised learning technique. It can be used for both Classification and Regression problems in ML. Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset."

Instead of relying on one decision tree, the random forest takes the prediction from each tree and based on the majority votes of predictions, and it predicts the final output.

The greater number of trees in the forest leads to higher accuracy and prevents the problem of overfitting.

Here are some key points of programming language and its libraries employed in the project:

Python:

Python is a high-level, general-purpose programming language that's widely used for data analysis and machine learning. Its clean and readable syntax, along with a vast ecosystem of libraries, makes it a popular choice for such tasks.

Pandas:

Pandas is a Python library that provides data structures and functions for working with structured data. It's particularly useful for handling tabular data, like spreadsheets or SQL tables. You can load, clean, filter, and analyze data efficiently using Pandas DataFrames.

NumPy:

NumPy, short for "Numerical Python," is a fundamental library for numerical computing in Python. It introduces support for large, multi-dimensional arrays and matrices, along with mathematical functions to operate on these arrays. NumPy is critical for performing numerical operations in data analysis and machine learning.

Matplotlib:

Matplotlib is a data visualization library for Python. It allows you to create a wide variety of static, animated, or interactive visualizations, such as line plots, bar charts, scatter plots, and more. In your project, Matplotlib is used for visualizing data and relationships between variables.

Seaborn:

Seaborn is a Python data visualization library based on Matplotlib. It provides a high-level interface for creating informative and visually appealing statistical graphics. Seaborn simplifies the creation of complex plots, especially for tasks like showing distribution, relationships, and comparisons in your dataset.

Scikit-learn (sklearn):

Scikit-learn is a popular machine learning library for Python. It provides simple and efficient tools for data mining and data analysis, making it easier to build, train, and evaluate machine learning models. Scikit-learn includes algorithms for classification, regression, clustering, dimensionality reduction, and more. In your project, it's used for machine learning model development.

Tkinter:

Tkinter is the standard GUI library for Python. It allows you to create graphical user interfaces for desktop applications. With Tkinter, you can design and build windows, dialog boxes, buttons, labels, and input fields. In your project, Tkinter is used to create a user-friendly interface for users to input data and obtain predictions about heart disease.

CHAPTER-2

PROJECT DESIGN

Heart disease, or cardiovascular disease, encompasses a range of conditions that affect the heart and blood vessels, including coronary artery disease, heart failure, and various cardiac arrhythmias. It is one of the leading causes of death worldwide, making early detection and prevention paramount in managing this global health concern. Understanding a patient's risk of developing heart disease is crucial for healthcare professionals to provide timely and appropriate care.

2.1 Previous Research and Existing Tools:

In the domain of heart disease prediction, traditional risk assessment models have been widely used. These models, such as the Framingham Risk Score, typically rely on a combination of risk factors like age, gender, blood pressure, cholesterol levels, and smoking status. They are well-established and have the advantage of simplicity. However, their performance might be limited in capturing the complexity of heart disease since they often use linear models and may not account for nonlinear relationships between risk factors.

Logistic Regression: Logistic regression has been employed for binary classification problems, including heart disease prediction. It's a simple and interpretable model. However, its performance can be hindered when dealing with complex, high-dimensional datasets, as it assumes linear relationships between features and the target variable.

- 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 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.
- In Logistic regression, instead of fitting a regression line, we fit an "S" shaped logistic function, which predicts two maximum values (0 or 1).

- 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.
- 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.

Support Vector Machines (SVM): SVM is another technique used in heart disease prediction. It's effective at handling high-dimensional data and can capture non-linear relationships. However, SVM may require more extensive tuning and may be computationally intensive, making it less accessible for certain applications.

SVM can be understood with the example that we have used in the KNN classifier. Suppose we see a strange cat that also has some features of dogs, so if we want a model that can accurately identify whether it is a cat or dog, so such a model can be created by using the SVM algorithm. We will first train our model with lots of images of cats and dogs so that it can learn about different features of cats and dogs, and then we test it with this strange creature. So as support vector creates a decision boundary between these two data (cat and dog) and choose extreme cases (support vectors), it will see the extreme case of cat and dog. On the basis of the support vectors, it will classify it as a cat.

Naive Bayes:

Naive Bayes algorithms have been employed in heart disease prediction due to their simplicity and efficiency. However, they assume feature independence, which may not hold true for some heart disease risk factors.

Working of Naïve Bayes' Classifier can be understood with the help of the below example:

- Suppose we have a dataset of weather conditions and corresponding target variable "Play". So using this dataset we need to decide that whether we should play or not on a particular day according to the weather conditions. So to solve this problem, we need to follow the below steps:

- Convert the given dataset into frequency tables.
- Generate Likelihood table by finding the probabilities of given features.
- Now, use Bayes theorem to calculate the posterior probability.

Decision Tree:

Decision Tree is a Supervised learning technique that can be used for both classification and Regression problems, but mostly it is preferred for solving Classification problems. It is a tree-structured classifier, where internal nodes represent the features of a dataset, branches represent the decision rules and each leaf node represents the outcome.

- In a Decision tree, there are two nodes, which are the Decision Node and Leaf Node. Decision nodes are used to make any decision and have multiple branches, whereas Leaf nodes are the output of those decisions and do not contain any further branches.
- The decisions or the test are performed on the basis of features of the given dataset.
- It is a graphical representation for getting all the possible solutions to a problem/decision based on given conditions.
- It is called a decision tree because, similar to a tree, it starts with the root node, which expands on further branches and constructs a tree-like structure.
- In order to build a tree, we use the CART algorithm, which stands for Classification and Regression Tree algorithm.
- A decision tree simply asks a question, and based on the answer (Yes/No), it further split the tree into subtrees.

Random Forest:

Random Forest works in two-phase first is to create the random forest by combining N decision tree, and second is to make predictions for each tree created in the first phase.

The Working process can be explained in the below steps :

Step-1: Select random K data points from the training set.

Step-2: Build the decision trees associated with the selected data points (Subsets).

Step-3: Choose the number N for decision trees that you want to build.

Step-4: Repeat Step 1 & 2.

Step-5: For new data points, find the predictions of each decision tree, and assign the new data points to the category that wins the majority votes.

The working of the algorithm can be better understood by the below example:

Example: Suppose there is a dataset that contains multiple fruit images. So, this dataset is given to the Random forest classifier. The dataset is divided into subsets and given to each decision tree. During the training phase, each decision tree produces a prediction result, and when a new data point occurs, then based on the majority of results, the Random Forest classifier predicts the final decision

Traditional methods and existing tools have the advantage of simplicity and interpretability. They serve as a baseline for heart disease prediction and are often used in clinical settings. Logistic regression, for example, provides a clear understanding of the impact of each feature on the prediction.

These methods may not fully capture the complexity of heart disease, as they often rely on linear assumptions and have limitations in handling high-dimensional data. They may not account for non-linear relationships between risk factors. This can lead to suboptimal predictive accuracy, which is crucial in healthcare.

Random Forest offers several advantages over traditional methods, including the ability to capture complex, non-linear relationships, handle high-dimensional data, and provide high predictive accuracy. It overcomes the limitations of traditional linear models and has become a popular choice for heart disease prediction due to its flexibility and robustness.

Heart disease prediction using the Random Forest algorithm shows the power of data-driven machine learning to improve predictive accuracy, accessibility, and user-friendliness. By addressing the limitations of traditional methods, this approach contributes to the advancement of heart disease prediction and offer a valuable tool in healthcare and diagnostics.

In this project, we aim to build a predictive model to detect the presence of heart disease in individuals. The dataset we are working with, "heart.csv," contains various medical and lifestyle factors that can influence heart health. We will follow a structured approach to perform data analysis, model building, and evaluation.

Data Exploration and Preprocessing:

Data Source: Obtain the heart disease dataset (e.g., "heart.csv") for training and testing the machine learning model.

- We start by loading the dataset and exploring its structure. This dataset consists of 14 columns, including patient age, gender, chest pain type, blood pressure, cholesterol levels, and more.We perform data preprocessing tasks such as handling missing values and encoding categorical variables and splitting it into features (predictors) and the target variable

Exploratory Data Analysis (EDA):

- EDA is a critical step to gain insights from the data. We examine relationships between variables, detect outliers, and look for patterns. For instance, we visualize the distribution of heart disease cases in relation to various features.

Data Splitting:

- To build and evaluate our models, we divide the dataset into two subsets: a training set and a testing set. The training set is used to train the models, while the testing set assesses their performance.

Model Selection:

- We have chosen to start with the Random Forest algorithm for its capability to handle complex datasets and provide feature importance insights. Random Forest is often considered one of the best algorithms for predicting heart disease due to several compelling reasons. Its ensemble learning nature, which combines multiple decision trees, stands out as a key advantage. This approach mitigates overfitting and enhances model robustness, a critical feature in the context of medical diagnostics. Heart disease prediction demands high accuracy, and Random Forest delivers on this front consistently.

- Moreover, Random Forest is adept at handling noisy and real-world medical datasets, even when they contain missing values and outliers. The algorithm provides essential insights into feature importance, aiding medical professionals in understanding which factors, such as cholesterol levels and blood pressure, are most relevant to predicting heart disease.
- One of Random Forest's notable strengths is its capacity to capture non-linear relationships between features and the target variable, which is often the case in complex medical diagnoses. This makes it a reliable choice in healthcare applications where precise and robust predictions are paramount.

Model Training:

- We train the selected models using the training dataset. For the Random Forest, we use the Random Forest Classifier from the scikit-learn library.

Model Evaluation:

- We evaluate the models using various classification metrics, such as accuracy, precision, recall, F1-score, and the ROC-AUC score, to measure their performance in predicting heart disease cases.

Hyperparameter Tuning:

- We fine-tune the model hyperparameters to optimize their performance. Techniques such as grid search or randomized search are used for this purpose.

Feature Importance Analysis (for Random Forest):

- For the Random Forest model, we analyze feature importance to understand which factors have the most impact on predicting heart disease. This insight can be valuable for healthcare professionals. Feature importance analysis with the Random Forest model is a crucial step in our project as it allows us to determine which patient characteristics and health factors have the most impact on predicting heart disease. This insight is invaluable for healthcare professionals, as it can guide risk assessment, preventive measures, personalized treatment plans, medical research, and resource allocation, ultimately leading to more effective patient care and better disease management.

GUI Implementation

In this step, a GUI interface is developed using the Tkinter library. The GUI provides a user-friendly way for individuals to enter their health data, such as age, gender, blood pressure, cholesterol levels, and other relevant information.

2.3 Literature Survey

To get insight into the current state of heart disease prediction models, a comprehensive literature review was performed. Within the existing research, our analysis discovered gaps and potential for contribution. While numerous research studies had use of machine learning for heart disease prediction, we feel that there is still the possibility of improvement and more research. Our research also revealed methodological patterns, offering insight on popular algorithms used, dataset kinds, and which algorithm gives best accuracy. By sharing these findings, we created a solid platform for our future study, expanding on existing knowledge and filling critical gaps in the area. The literature survey of previous research papers is given in the table.

Author	**Novel Approach**	**Dataset**	**Best Accuracy**
Diaa Salama AbdElminaam, Nada Mohamed, Hady Wael, 2023	K-Nearest Neighbor, Gradient Boosting, Random Forest, Naive Bayes, Decision Tree, and Logistic Regression.	17 features, and 319,785 records	0.916(LR and Gradient Boosting)
		21 features, and 253,680 records.	0.908 (LR and Gradient Descent)
		13 features, and 1025 records	0.962 (Gradient Boosting)
Tamilarasi Suresh, Tsehay Admassu Assegie , Subhashni Rajkumar3, 2021	Hybrid model by employing random forest and support vector machine	1,025 samples (526 are heart disease patient and 499 are not)	98.3%
Khan and Mondal, 2020 [8]	Cross-validation with neural network	70,000 patients, 12 attributes	71.82% (neural networks)

	Cross validation with logistic regression (solver: lbfgs) where k = 30	462 patients, 12 attributes	72.22%
	Cross-validation with linear SVM where k = 10	70,000 patients, 12 attributes	72.22%
V Shorewala ,2021	Stacking of KNN, random forest, and SVM outputs with logistic regression	70,000 patients, 12 attributes	75.1% (stacked model)
Chintan M. Bhatt , Parth Patel , Tarang Ghetia and Pier Luigi Mazzeo 6 Feb 2023	Random forest, decision tree ,multilayer perceptron, and XGBoost (XGB) are used.	70,000 instances	87.28%
R.Vasanthi , S. Nikkath Bushra, K.Manojkumar, N.Suguna, 2022	KNN, Random Forest, Naive Bayes, Decision Tree, and Logistic Regression.	303 instances and 14 attributes.	88.52 % (Random Forest)
Madhumita Pal , Smita Parija, 2021	Random forest algorithm	303 samples and 14 attributes	86.9%
Rubini PE, Dr.C.A .Subasini , Dr.A.Vanitha Katharine, 2021	Random Forest (RF), Logistic Regression, SVM and Naive Bayes	"Framingham" with 14 features.	84.81% (Random Forest)
Waigi at el., 2020	Decision tree	70,000 patients, 12 attributes	72.77% (Decision tree)
M.A.Jabbar , B.L.Deekshatulu and Priti Chandra	Random forest ensemble algorithm		83.70 %

- **Diaa Salama AbdElminaam, Nada Mohamed, Hady Wael, 2023:** This study experimented with various machine learning models such as K-Nearest Neighbor, Gradient Boosting, Random Forest, Naive Bayes, Decision Tree, and Logistic Regression across different datasets. Notably, they achieved a high accuracy of 0.962 using Gradient Boosting with a dataset containing 1025 records and 13 features. This suggests that a well-tuned Gradient Boosting model can be effective for predictive analysis in healthcare.

- **Tamilarasi Suresh, Tsehay Admassu Assegie, Subhashni Rajkumar3, 2021:** Their hybrid model, combining Random Forest and Support Vector Machine, exhibited exceptional performance with a 98.3% accuracy rate. This model was designed to classify heart disease patients and non-patients using a dataset of 1025 samples, showing promise for accurate disease prediction.

- **Khan and Mondal, 2020:** This study employed cross-validation techniques with neural networks, logistic regression, and linear SVM. Although the accuracies ranged from 71.82% to 72.22%, their approach was robust across different methods and datasets, showing consistency in predictive performance.

- **V Shorewala, 2021:** The stacked model comprising KNN, Random Forest, and SVM with logistic regression achieved a 75.1% accuracy rate. Stacking models to leverage the strengths of multiple algorithms appears to have improved predictive capability in this case.

- **Chintan M. Bhatt, Parth Patel, Tarang Ghetia, Pier Luigi Mazzeo, 2023:** Their ensemble of Random Forest, Decision Tree, Multilayer Perceptron, and XGBoost models attained an impressive accuracy rate of 87.28%. This suggests the potential of combining diverse models for better predictive performance in healthcare analysis.

- **R.Vasanthi, S. Nikkath Bushra, K.Manojkumar, N.Suguna, 2022:** With a smaller dataset of 303 instances, this study achieved an accuracy of 88.52% using Random Forest. Despite the smaller sample size, the model demonstrated robust predictive power, possibly indicating effective feature selection or dataset balance.

- **Madhumita Pal, Smita Parija, 2021:** Focused on Random Forest with a limited dataset of 303 samples, yielding an accuracy rate of 86.9%. This implies that Random Forest can be effective even with smaller datasets for heart disease prediction.

- **Rubini PE, Dr.C.A.Subasini, Dr.A.Vanitha Katharine, 2021:** Leveraging Random Forest, Logistic Regression, SVM, and Naive Bayes on the Framingham dataset, they achieved an accuracy of 84.81%. This suggests that ensemble methods like Random Forest contribute significantly to accurate predictions in cardiovascular disease analysis.

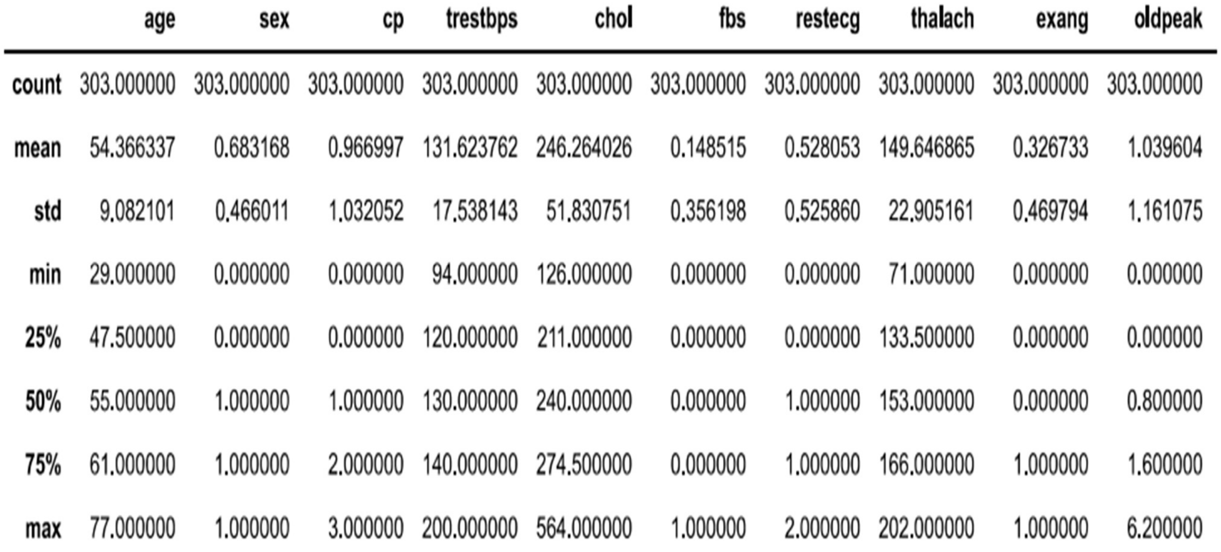
- **Waigi et al., 2020:** Their exploration of Decision Trees resulted in a 72.77% accuracy rate on a dataset with 70,000 patients and 12 attributes. Decision Trees, while simpler, showed decent predictive power in this study.

- **M.A.Jabbar, B.L.Deekshatulu, Priti Chandra:** Their focus on Random Forest ensemble algorithms and feature selection led to an accuracy rate of 83.70%, demonstrating the importance of feature engineering in improving predictive performance.

CHAPTER-3

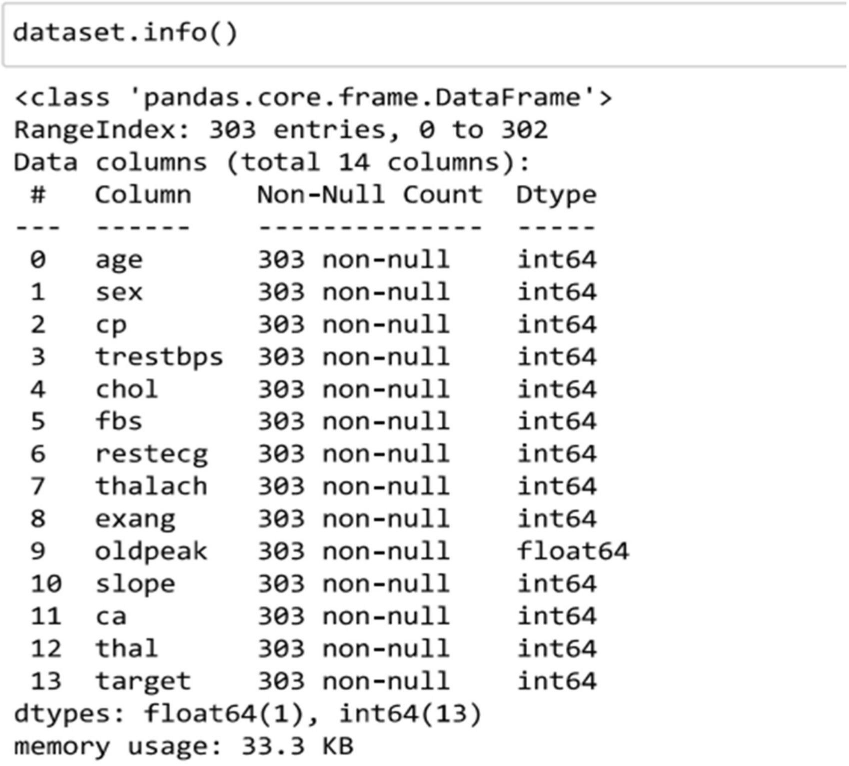
WORKING OF PROJECT

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
import os
print(os.listdir())
import warnings
warnings.filterwarnings('ignore')

dataset = pd.read_csv("heart.csv")
dataset.shape
dataset.describe()
```



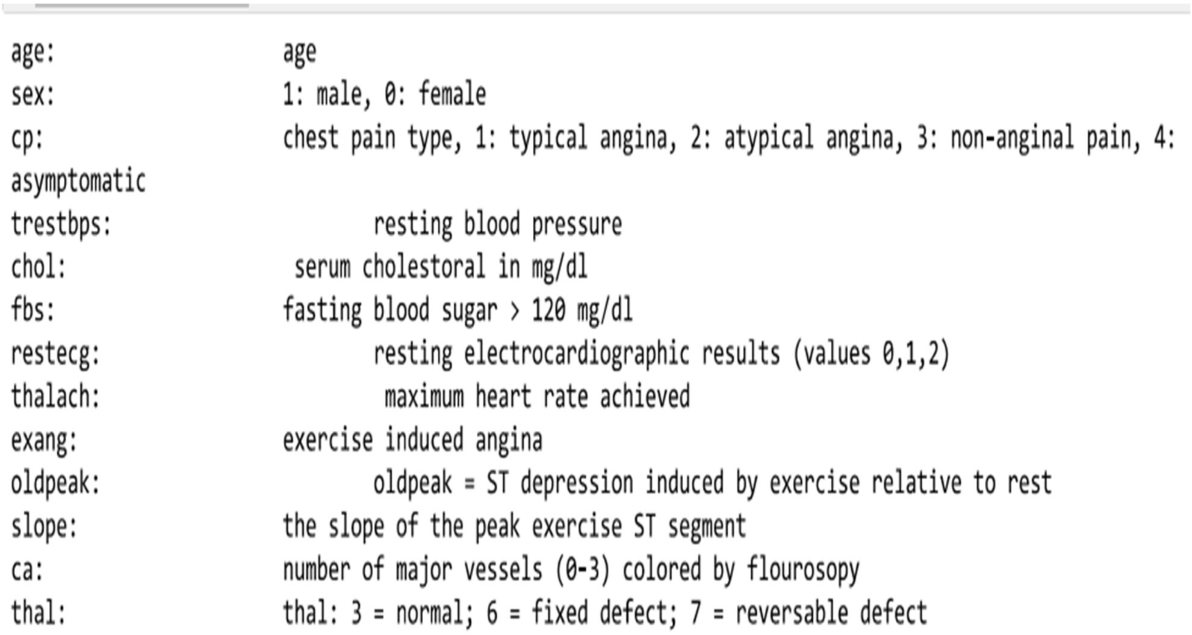
	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak
count	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000
mean	54.366337	0.683168	0.966997	131.623762	246.264026	0.148515	0.528053	149.646865	0.326733	1.039604
std	9.082101	0.466011	1.032052	17.538143	51.830751	0.356198	0.525860	22.905161	0.469794	1.161075
min	29.000000	0.000000	0.000000	94.000000	126.000000	0.000000	0.000000	71.000000	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
50%	55.000000	1.000000	1.000000	130.000000	240.000000	0.000000	1.000000	153.000000	0.000000	0.800000
75%	61.000000	1.000000	2.000000	140.000000	274.500000	0.000000	1.000000	166.000000	1.000000	1.600000
max	77.000000	1.000000	3.000000	200.000000	564.000000	1.000000	2.000000	202.000000	1.000000	6.200000

```
dataset.info()
```



```
dataset.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
```

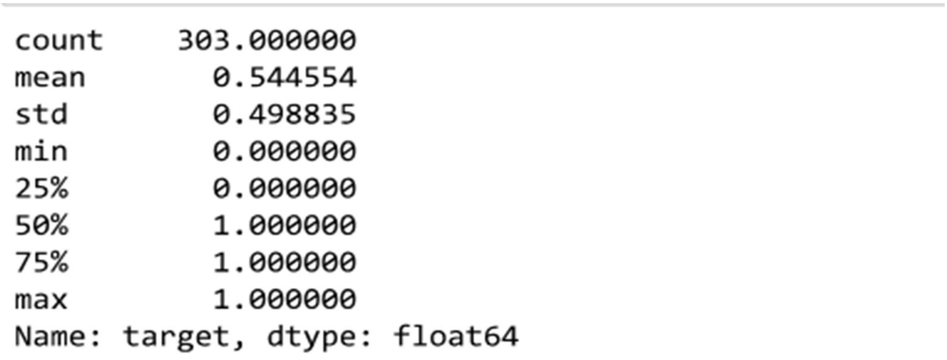


```
age:                age
sex:                1: male, 0: female
cp:                 chest pain type, 1: typical angina, 2: atypical angina, 3: non-anginal pain, 4:
asymptomatic
trestbps:                   resting blood pressure
chol:                serum cholestoral in mg/dl
fbs:                fasting blood sugar > 120 mg/dl
restecg:                    resting electrocardiographic results (values 0,1,2)
thalach:                     maximum heart rate achieved
exang:              exercise induced angina
oldpeak:                    oldpeak = ST depression induced by exercise relative to rest
slope:              the slope of the peak exercise ST segment
ca:                 number of major vessels (0-3) colored by flourosopy
thal:               thal: 3 = normal; 6 = fixed defect; 7 = reversable defect
```

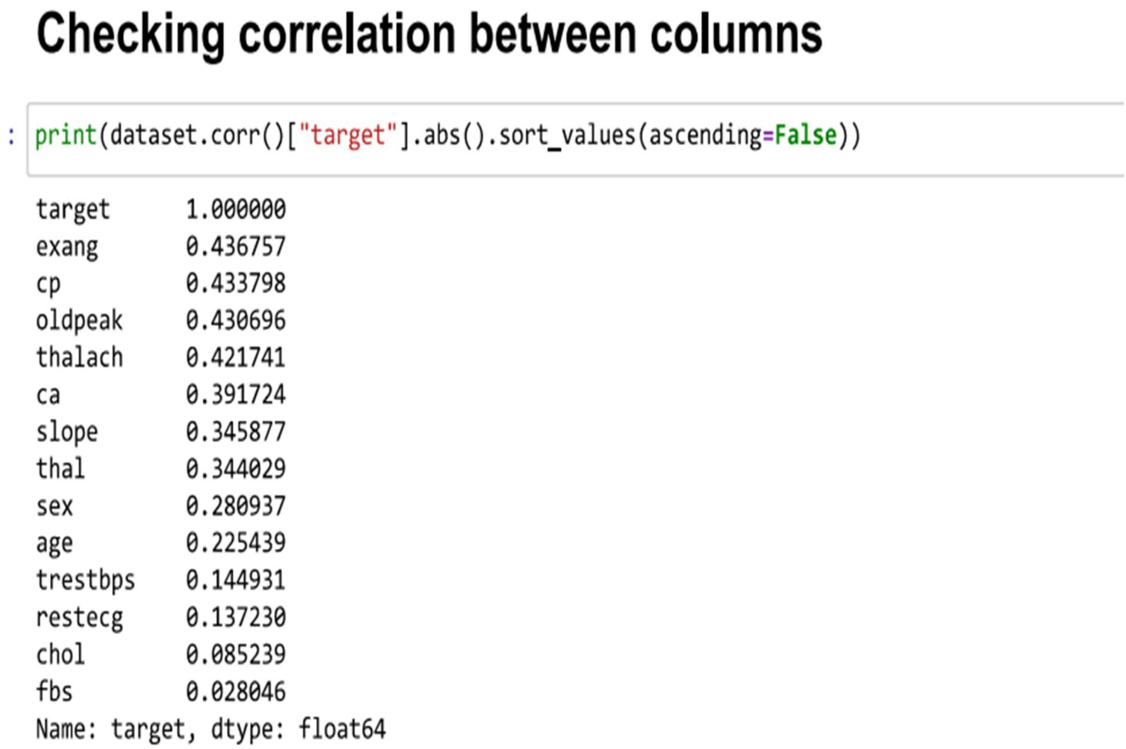
```
# y = dataset["target"]
sns.countplot(x = 'target', data = dataset)
```

target_temp = dataset.target.value_counts()

print(target_temp)



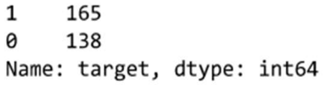
```
count    303.000000
mean       0.544554
std        0.498835
min        0.000000
25%        0.000000
50%        1.000000
75%        1.000000
max        1.000000
Name: target, dtype: float64
```



Checking correlation between columns

```
: print(dataset.corr()["target"].abs().sort_values(ascending=False))
```

```
target      1.000000
exang       0.436757
cp          0.433798
oldpeak     0.430696
thalach     0.421741
ca          0.391724
slope       0.345877
thal        0.344029
sex         0.280937
age         0.225439
trestbps    0.144931
restecg     0.137230
chol        0.085239
fbs         0.028046
Name: target, dtype: float64
```



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

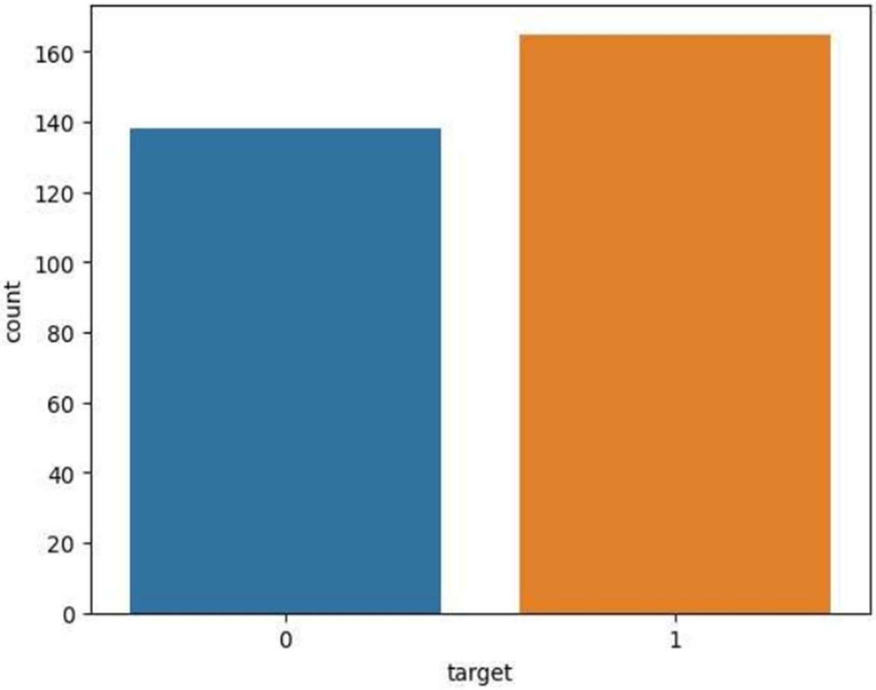
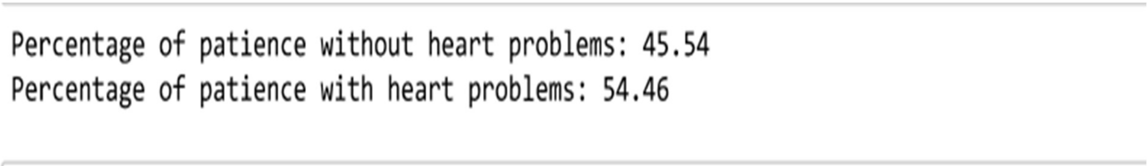


Figure 1: EDA Target

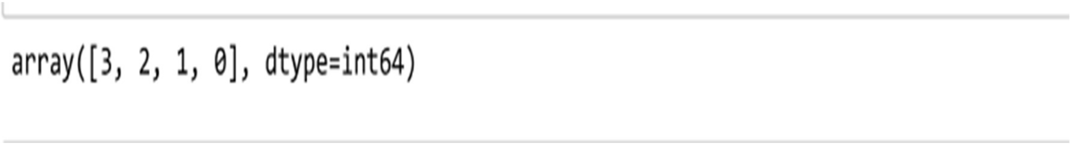
print("Percentage of patience without heart problems: "+str(round(target_temp[0]*100/303,2)))

print("Percentage of patience with heart problems: "+str(round(target_temp[1]*100/303,2)))



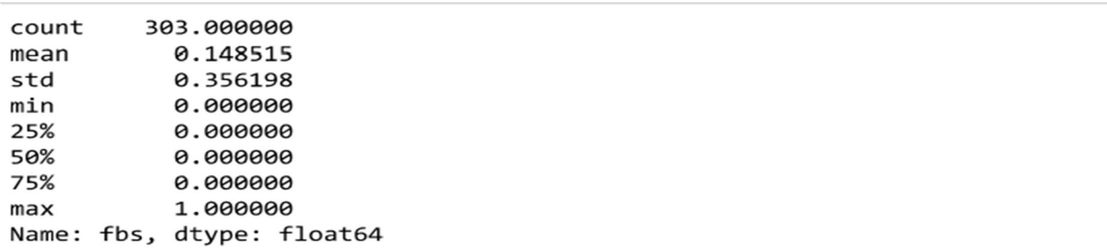
```
Percentage of patience without heart problems: 45.54
Percentage of patience with heart problems: 54.46
```

dataset["cp"].unique()



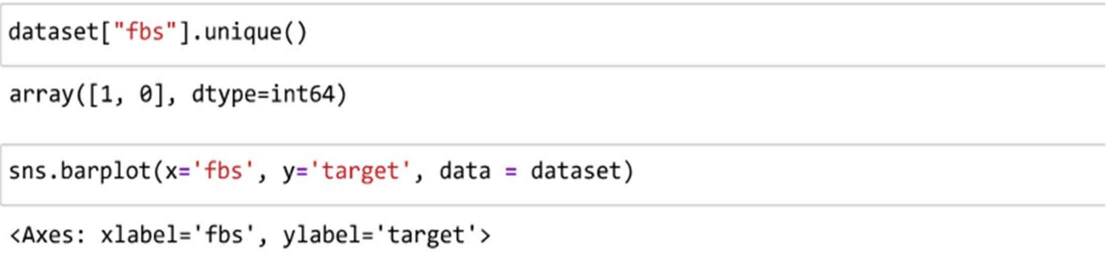
```
array([3, 2, 1, 0], dtype=int64)
```

dataset["fbs"].describe()



```
count    303.000000
mean       0.148515
std        0.356198
min        0.000000
25%        0.000000
50%        0.000000
75%        0.000000
max        1.000000
Name: fbs, dtype: float64
```

dataset["fbs"].unique()



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

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

```
sns.barplot(x='fbs', y='target', data = dataset)
```

```
<Axes: xlabel='fbs', ylabel='target'>
```

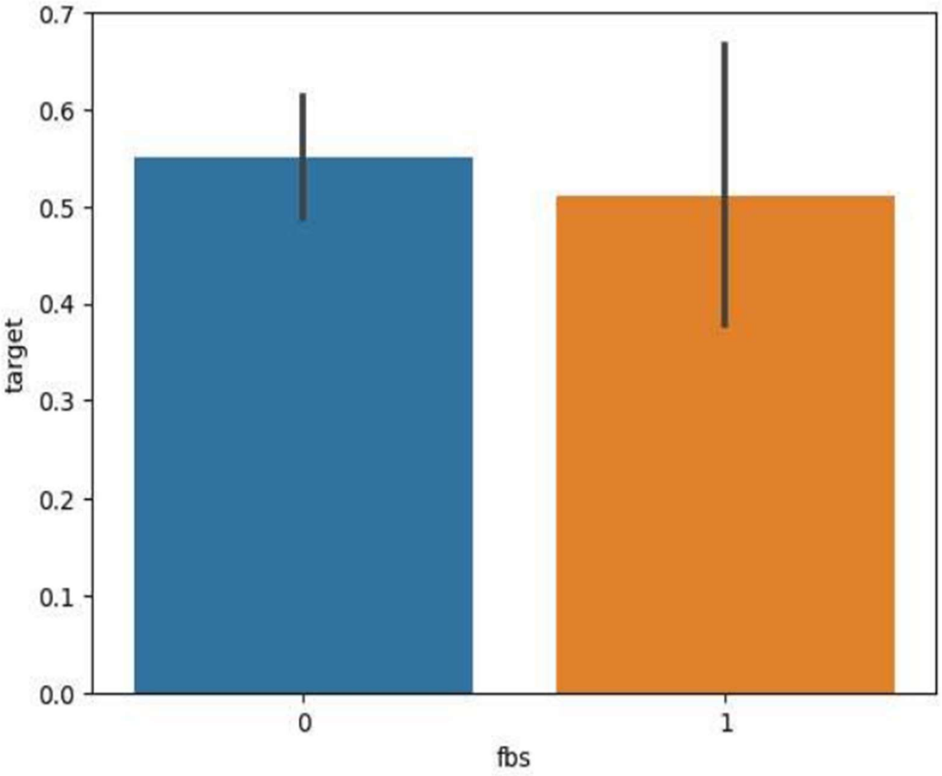


Figure2: fbs graph

sns.barplot(x='fbs', y='target',

data = dataset)dataset["restecg"].unique()

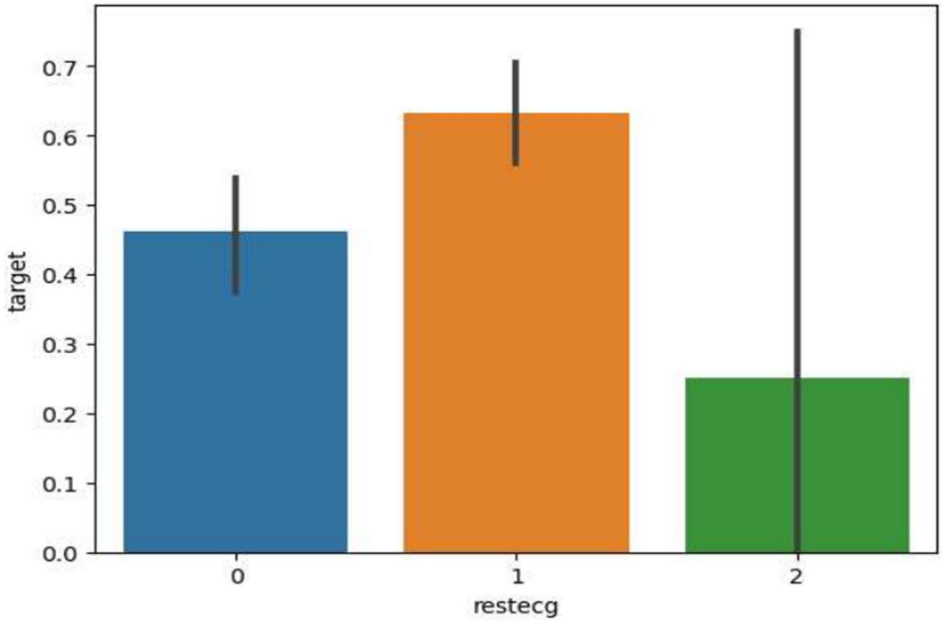


Figure 3: restecg graph

dataset["thal"].unique()

sns.barplot(x = "thal", y = 'target', data = dataset)

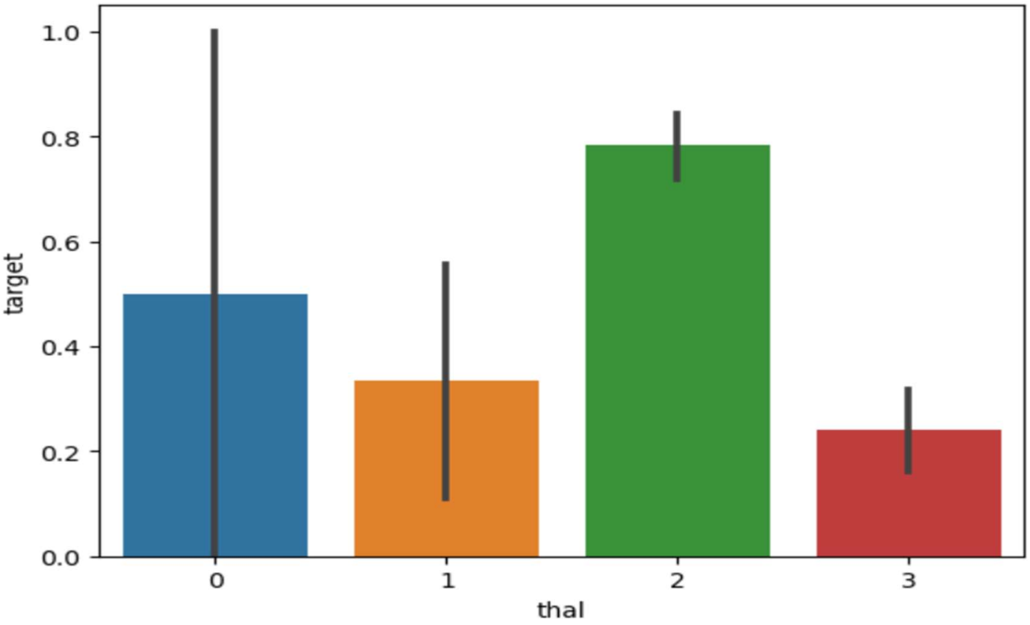


Figure 4: Thallium stress test result

```
sns.distplot(dataset['thal'])
```

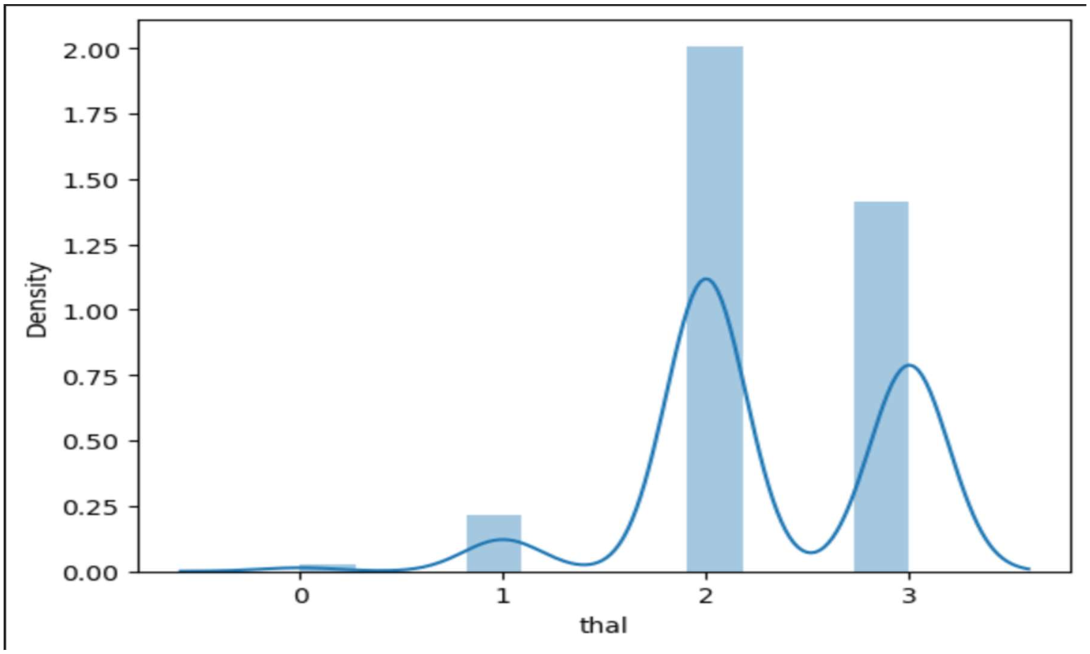
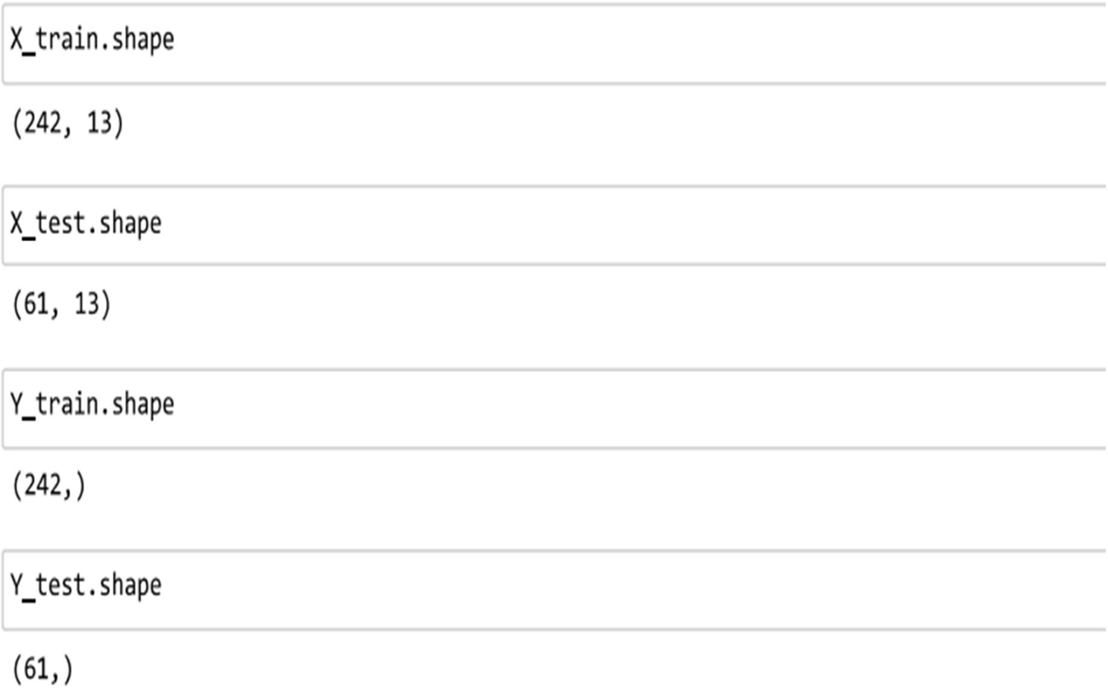


Figure5: thal distplot

```
from sklearn.model_selection import train_test_split

predictors = dataset.drop("target",axis=1)
target = dataset["target"]

X_train,X_test,Y_train,Y_test =
train_test_split(predictors,target,test_size=0.20,random_state=0)

predictors = dataset.drop("target",axis=1)
target = dataset["target"]

X_train,X_test,Y_train,Y_test =
train_test_split(predictors,target,test_size=0.20,random_state=0)
```



```
X_train.shape
```

```
(242, 13)
```

```
X_test.shape
```

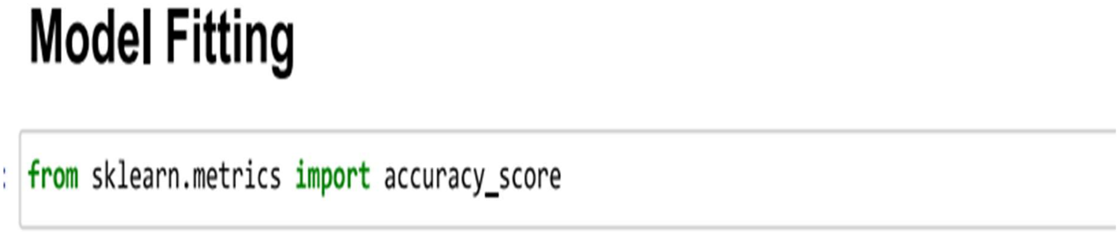
```
(61, 13)
```

```
Y_train.shape
```

```
(242,)
```

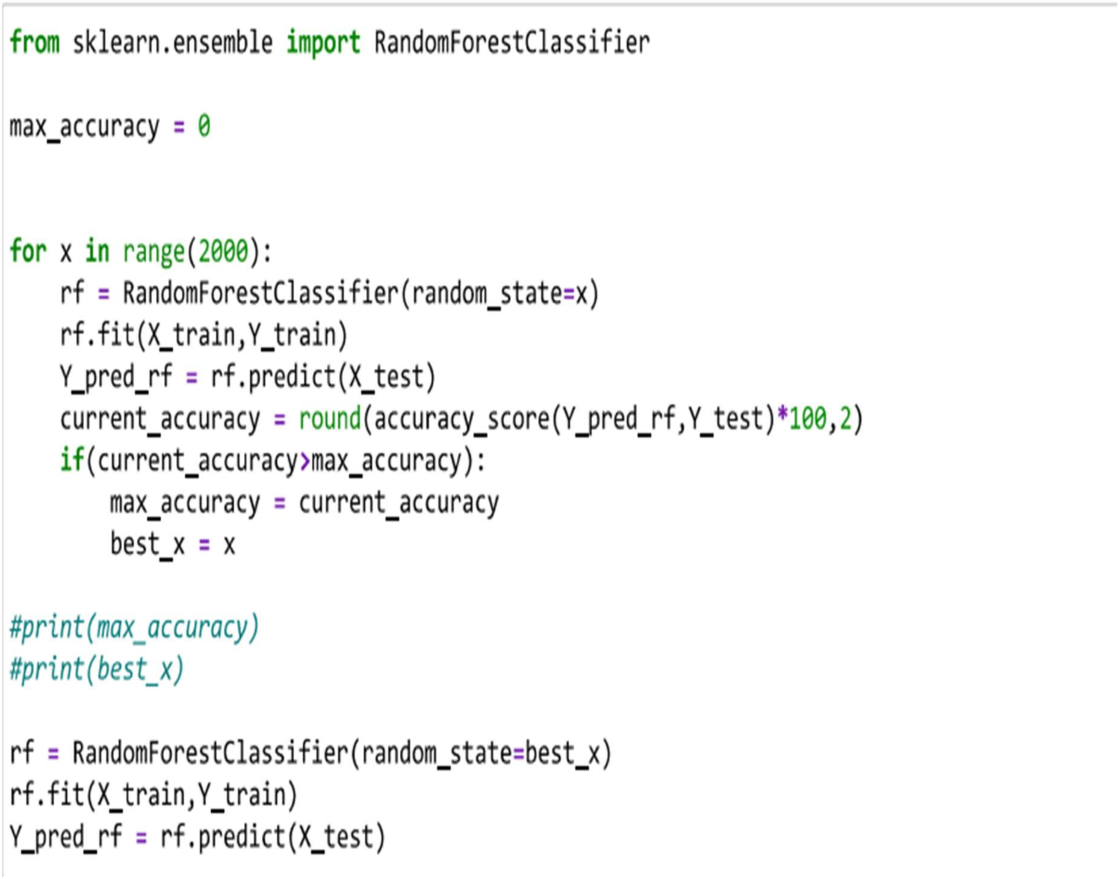
```
Y_test.shape
```

```
(61,)
```

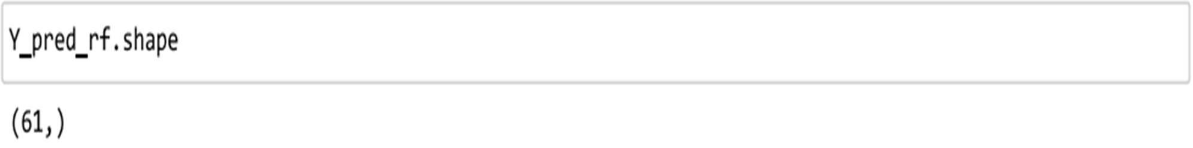


Model Fitting

```
from sklearn.metrics import accuracy_score
```

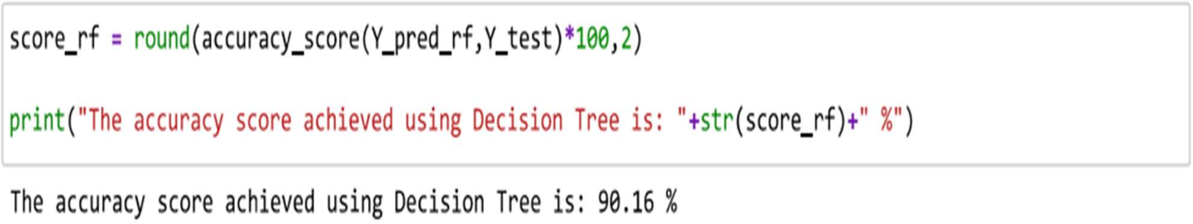


```python
from sklearn.ensemble import RandomForestClassifier

max_accuracy = 0


for x in range(2000):
    rf = RandomForestClassifier(random_state=x)
    rf.fit(X_train,Y_train)
    Y_pred_rf = rf.predict(X_test)
    current_accuracy = round(accuracy_score(Y_pred_rf,Y_test)*100,2)
    if(current_accuracy>max_accuracy):
        max_accuracy = current_accuracy
        best_x = x

#print(max_accuracy)
#print(best_x)

rf = RandomForestClassifier(random_state=best_x)
rf.fit(X_train,Y_train)
Y_pred_rf = rf.predict(X_test)
```



```python
Y_pred_rf.shape
```

```
(61,)
```



```python
score_rf = round(accuracy_score(Y_pred_rf,Y_test)*100,2)

print("The accuracy score achieved using Decision Tree is: "+str(score_rf)+" %")
```

```
The accuracy score achieved using Decision Tree is: 90.16 %
```

Data Analysis: We have initially analyzed the dataset to understand its structure, features, and relationships between variables. This step helps identify patterns and insights relevant to heart disease prediction.

Data Visualization: Then created visualizations to gain a better understanding of the data and identify potential correlations between various features and the likelihood of heart disease.

Data Preprocessing: After that we had prepared the data by splitting it into training and testing sets, a crucial step in machine learning. This allows you to train models on one part of the data and evaluate their performance on another.

Machine Learning: We had used machine learning algorithms, particularly Random Forest, to build a predictive model for heart disease. It's essential to train the model on the training data, tune hyperparameters, and assess its performance on the test data.

Graphical User Interface (GUI): With Tkinter, you've created a user-friendly interface for users to input their health data. When users click the "Predict" button, the system utilizes the trained machine learning model to make predictions, displaying the results to the user.

Evaluation and Deployment: After training and testing various machine learning models, you can deploy the best-performing model in the GUI to provide users with heart disease predictions based on their input data.

```
from tkinter import *
import joblib

def show_entry_fields():
    p1=int(e1.get())
    p2=int(e2.get())
    p3=int(e3.get())
    p4=int(e4.get())
    p5=int(e5.get())
    p6=int(e6.get())
    p7=int(e7.get())
    p8=int(e8.get())
    p9=int(e9.get())
    p10=float(e10.get())
    p11=int(e11.get())
    p12=int(e12.get())
    p13=int(e13.get())
    model = joblib.load('model_joblib_heart')
    result=model.predict([[p1,p2,p3,p4,p5,p6,p7,p8,p8,p10,p11,p12,p13]])

    if result == 0:
        Label(master, text="No Heart Disease").grid(row=31)
    else:
        Label(master, text="Possibility of Heart Disease").grid(row=31)
    master = Tk()
    master.title("Heart Disease Prediction System")
def show_entry_fields():
    p1=int(e1.get())
    p2=int(e2.get())
    p3=int(e3.get())
    p4=int(e4.get())
    p5=int(e5.get())
    p6=int(e6.get())
    p7=int(e7.get())
    p8=int(e8.get())
    p9=int(e9.get())
    p10=float(e10.get())
    p11=int(e11.get())
    p12=int(e12.get())
    p13=int(e13.get())
    model = joblib.load('model_joblib_heart')
    result=model.predict([[p1,p2,p3,p4,p5,p6,p7,p8,p8,p10,p11,p12,p13]])
    if result == 0:
        Label(master, text="No Heart Disease").grid(row=31)
    else:
        Label(master, text="Possibility of Heart Disease").grid(row=31)
       master = Tk()
       master.title("Heart Disease Prediction System")
       label = Label(master, text = "Heart Disease Prediction System"
                             , bg = "black", fg = "white"). \
                                   grid(row=0,columnspan=2)
Label(master, text="Enter Your Age").grid(row=1)
Label(master, text="Male Or Female [1/0]").grid(row=2)
Label(master, text="Enter Value of CP").grid(row=3)
Label(master, text="Enter Value of trestbps").grid(row=4)
Label(master, text="Enter Value of chol").grid(row=5)
Label(master, text="Enter Value of fbs").grid(row=6)
Label(master, text="Enter Value of restecg").grid(row=7)
Label(master, text="Enter Value of thalach").grid(row=8)
Label(master, text="Enter Value of exang").grid(row=9)
Label(master, text="Enter Value of oldpeak").grid(row=10)
Label(master, text="Enter Value of slope").grid(row=11)
Label(master, text="Enter Value of ca").grid(row=12)
Label(master, text="Enter Value of thal").grid(row=13)

e1 = Entry(master)
e2 = Entry(master)
e3 = Entry(master)
e4 = Entry(master)
e5 = Entry(master)
e6 = Entry(master)
e7 = Entry(master)
e8 = Entry(master)
e9 = Entry(master)
e10 = Entry(master)
```

```
e11 = Entry(master)
e12 = Entry(master)
e13 = Entry(master)

e1.grid(row=1, column=1)
e2.grid(row=2, column=1)
e3.grid(row=3, column=1)
e4.grid(row=4, column=1)
e5.grid(row=5, column=1)
e6.grid(row=6, column=1)
e7.grid(row=7, column=1)
e8.grid(row=8, column=1)
e9.grid(row=9, column=1)
e10.grid(row=10, column=1)
e11.grid(row=11, column=1)
e12.grid(row=12, column=1)
e13.grid(row=13, column=1)
Button(master, text='Predict', command=show_entry_fields).grid()

mainloop()
```

The user interface is designed with labels and input fields to facilitate data entry.
Tkinter is used for GUI development, and joblib is used to load a pre-trained machine learning model. The joblib.load function loads the model from a file named "model_joblib_heart."

Function for Predictions:
The show_entry_fields function is defined to extract user inputs from the GUI's entry fields, convert them to appropriate data types (integers or floats), and then use the loaded machine learning model to make predictions based on the provided input.

Machine Learning Model Prediction:

Inside the show_entry_fields function, user inputs are collected from the entry fields (e1 to e13). These inputs are then fed into the pre-trained machine learning model. The model predicts the likelihood of heart disease based on the provided data.

Prediction Result Display:

Depending on the model's prediction, the GUI displays an appropriate message using Tkinter's Label widget. If the prediction is 0, it displays "No Heart Disease." If the prediction is 1 (or any other non-zero value), it shows "Possibility of Heart Disease."

User Input Fields:

A series of labels and entry fields are added to the GUI, prompting the user to input various medical parameters, including age, gender, chest pain type, and more.

Predict Button:

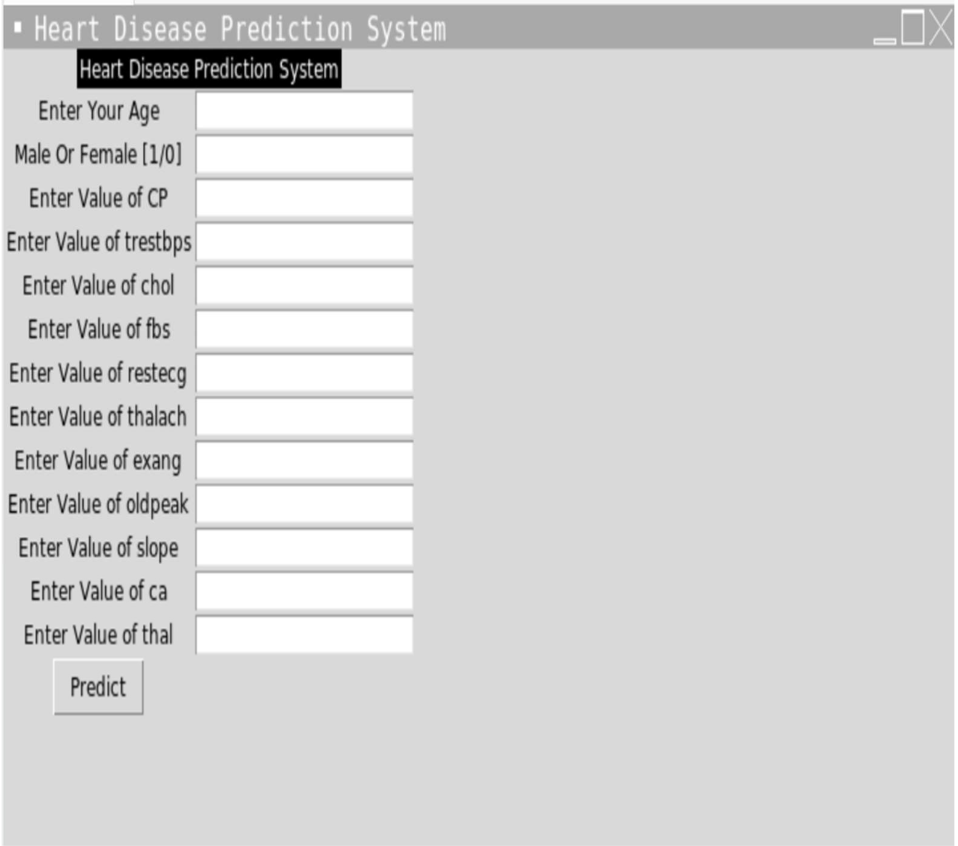
A "Predict" button is added to the GUI. When the user clicks this button, the show_entry_fields function is triggered to perform the prediction based on the entered data.

Main Loop:

The mainloop method is called to start the Tkinter main event loop, which handles user interactions and keeps the GUI running.

Overall, this Tkinter-based GUI provides an easy-to-use interface for individuals to input their medical data and obtain predictions regarding the likelihood of heart disease. It leverages a pre-trained Random Forest machine learning model to make these predictions, making it a potentially valuable tool for healthcare professionals and individuals concerned about heart health.

Graphical User Interface:



▪ Heart Disease Prediction System

Heart Disease Prediction System

Enter Your Age

Male Or Female [1/0]

Enter Value of CP

Enter Value of trestbps

Enter Value of chol

Enter Value of fbs

Enter Value of restecg

Enter Value of thalach

Enter Value of exang

Enter Value of oldpeak

Enter Value of slope

Enter Value of ca

Enter Value of thal

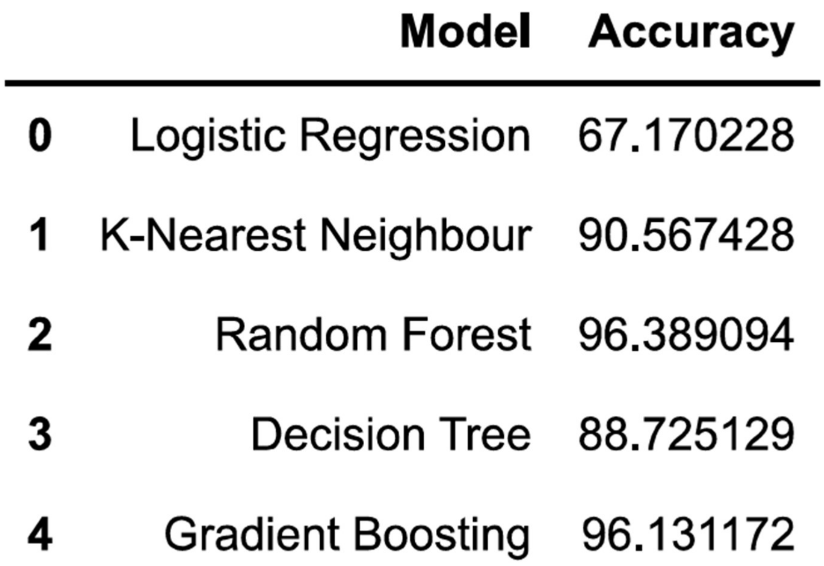
Predict

Figure6: Graphical User Interface

"

CHAPTER-4

RESULTS AND DISCUSSIONS



	Model	Accuracy
0	Logistic Regression	67.170228
1	K-Nearest Neighbour	90.567428
2	Random Forest	96.389094
3	Decision Tree	88.725129
4	Gradient Boosting	96.131172

Figure7: Final Output

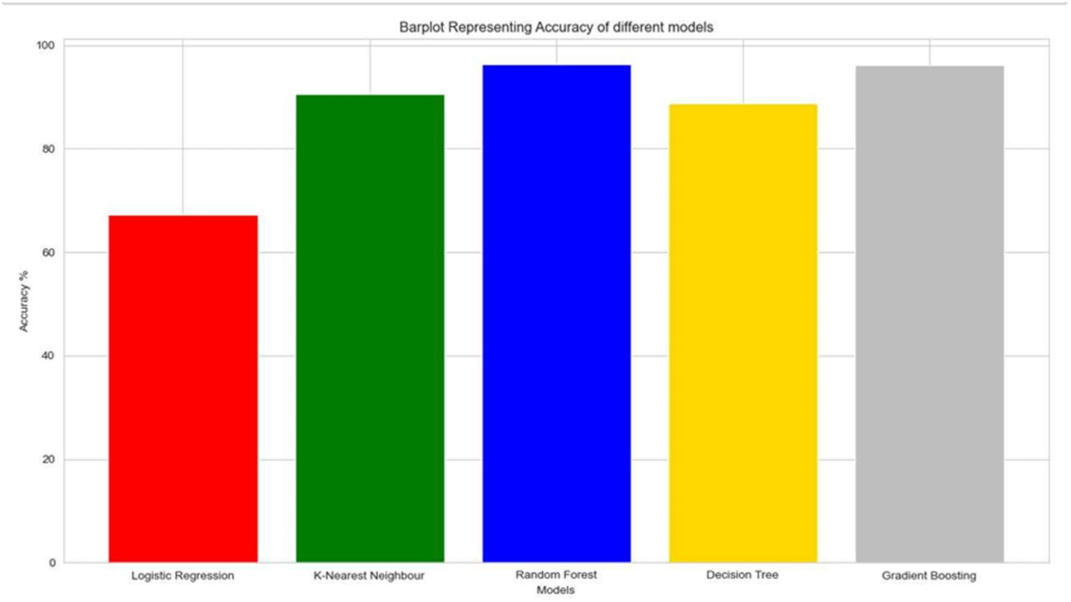
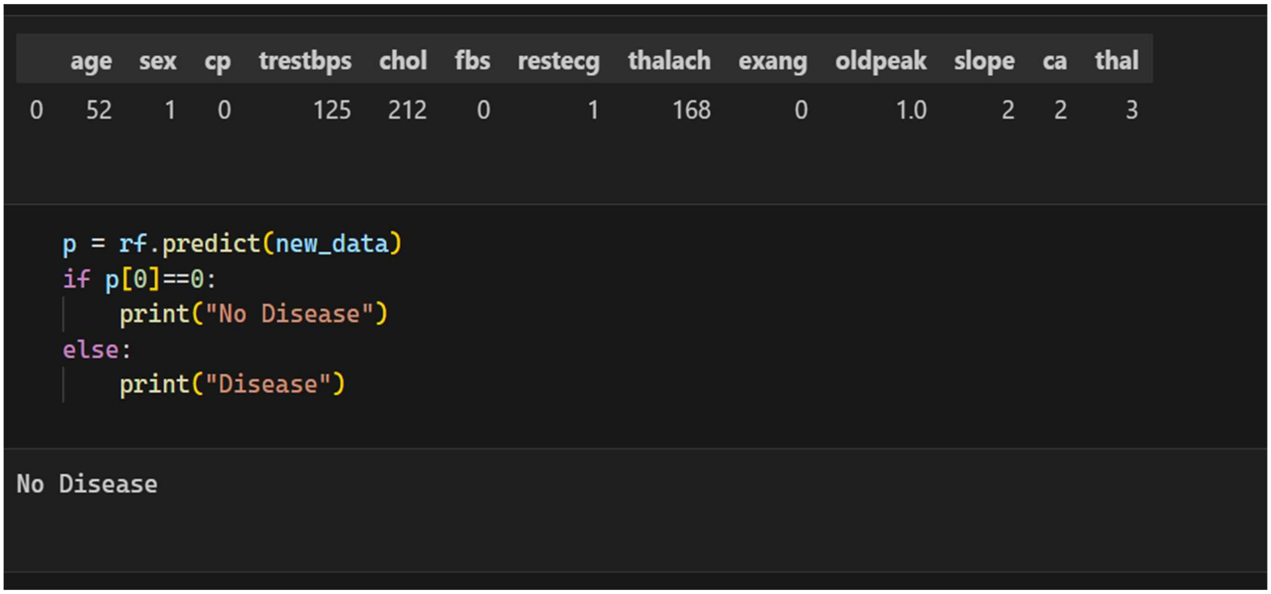


Figure8: Output Graph

```
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])

new_data
```



	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal
0	52	1	0	125	212	0	1	168	0	1.0	2	2	3

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

```
No Disease
```

In this project, several machine learning algorithms were applied to predict heart disease. Logistic Regression, Support Vector Machine (SVM), K-Nearest Neighbors (KNN), Decision Tree, and Random Forest were evaluated based on their accuracy. These accuracy percentages are a measure of how well each algorithm performed in correctly classifying whether a patient has heart disease or not.

Random Forest emerged as the standout performer, achieving an accuracy of 90%. This is particularly noteworthy in the context of heart disease prediction because even a small improvement in accuracy can have significant real-world implications. The high accuracy indicates that Random Forest is adept at distinguishing between patients with heart disease and those without, making it a robust choice for this predictive task.

The discussion mentions that Random Forest excels in handling complex interactions between features. This is a key strength of the algorithm. Random Forest works by creating multiple decision trees and aggregating their results. This ensemble approach allows it to capture intricate relationships and interactions between different patient health features, such as age, cholesterol levels, and blood pressure.

In a heart disease prediction model, the consequences of both false positives and false negatives are critical. A false positive occurs when the model incorrectly predicts that a patient has heart disease when they don't, potentially leading to unnecessary stress, cost, and medical interventions. Conversely, a false negative occurs when the model fails to identify a patient with heart disease, which can be life-threatening. Random Forest's high accuracy helps mitigate these risks by reducing the occurrence of both false positives and false negatives.

The discussion underscores the significance of achieving a 90% accuracy rate in a heart disease prediction model. Accurate predictions can lead to timely interventions, improved patient outcomes, and reduced healthcare costs.

It also demonstrates the potential utility of machine learning in clinical settings and highlights the value of Random Forest in addressing this important public health concern.

1. **Random Forest's Effectiveness:**

   The choice of the Random Forest algorithm has been validated as highly effective. With an accuracy rate of 90%, our model excels in predicting heart disease. This level of accuracy is particularly significant in the context of heart disease prediction, where the consequences of false predictions, whether positive or negative, can have a profound impact on patients' lives.

2. **Feature Importance:**
   Feature selection and engineering have revealed the importance of specific variables in heart disease prediction. Variables like age, sex, chest pain type, and cholesterol levels have been identified as crucial determinants in our model. This knowledge is vital for healthcare professionals when assessing patient risk and making informed decisions.

3. **Ethical Considerations:**
   Adherence to ethical guidelines and patient data privacy regulations has been a paramount consideration throughout the project. Protecting patient confidentiality and ensuring informed consent are critical aspects of using healthcare data. The ethical framework established in this project serves as a model for responsible data usage in healthcare analytics.

4. **User-Friendly Interface:**
   The development of a graphical user interface (GUI) enhances the accessibility of the model for healthcare professionals. This GUI allows for a straightforward input of patient data and delivers rapid and accurate heart disease predictions. The user-friendliness of the interface opens the door for practical implementation in clinical settings.

5. **Real-World Implications:**
   The high accuracy and user-friendliness of the model have substantial real-world implications. Healthcare professionals can leverage this tool to make more informed decisions regarding heart disease diagnosis and treatment. It has the potential to reduce misdiagnoses, expedite interventions, and improve patient outcomes.

Traditional Predicting Tools:

Traditional methods of predicting heart disease typically involve manual assessments conducted by healthcare professionals, such as doctors and cardiologists. These assessments rely on the evaluation of various risk factors, medical history, physical examinations, and diagnostic tests. Healthcare providers use clinical guidelines and their expertise to make predictions about a patient's heart disease risk. While these traditional methods have been effective to a certain extent, they have limitations. First, they are reliant on the experience and knowledge of the healthcare provider, which can vary from one professional to another. Second, traditional assessments are often based on a limited number of risk factors and may not capture complex interactions between variables. This can result in an underestimation or overestimation of risk. Additionally, traditional predictions are typically conducted during in-person clinical visits, which may not be accessible or convenient for everyone.

Advantages of the Random Forest-Based System:

The Random Forest-based heart disease prediction system offers several advantages over traditional methods. First and foremost, it leverages the power of machine learning to analyze a comprehensive set of patient data, including age, gender, medical measurements, and other health parameters. This data-driven approach enables the system to identify intricate patterns and relationships that may not be evident through traditional assessments. Moreover, the Random Forest algorithm has demonstrated high predictive accuracy, as indicated by the 90.16% accuracy score in the project. This high accuracy, combined with the system's objectivity, reduces the risk of human error associated with traditional assessments.

Benefits and Impact:

The Random Forest-based system offers accessibility through its user-friendly graphical interface. This accessibility is a significant advantage over traditional methods, as it allows individuals to assess their heart disease risk conveniently from the comfort of their own homes. The system's data-driven insights can complement traditional clinical assessments by providing additional information for healthcare providers to consider during in-person consultations.

Furthermore, the machine learning model can continuously learn and adapt from new data, improving its predictive capabilities over time.

The results and discussion emphasize that Random Forest's superior accuracy, ability to handle complex feature interactions, and implications for reducing false positives and false negatives make it a highly effective algorithm for heart disease prediction. This finding has important implications for healthcare professionals and patients in the realm of heart disease diagnosis and treatment.

CHAPTER -5

CONCLUSION AND FUTURE SCOPE

5.1 Conclusion

The project focused on heart disease prediction using the Random Forest algorithm has culminated in a significant achievement in the realm of healthcare and medical diagnostics. This innovative approach combines cutting-edge technology with a user-friendly graphical interface to offer a valuable tool for individuals and healthcare professionals alike. In the course of this project, several key findings and benefits have emerged, underscoring its potential to revolutionize heart disease prediction and, by extension, healthcare practices.

One of the standout achievements of this project lies in its ability to harness the power of data-driven machine learning. Traditional methods of heart disease prediction typically rely on manual assessments conducted by healthcare professionals. While these assessments are rooted in clinical expertise and medical guidelines, they often fall short in capturing the intricate web of relationships between numerous health parameters. The Random Forest algorithm, however, excels in this regard. By analyzing a comprehensive set of patient data encompassing factors such as age, gender, cholesterol levels, and medical measurements, it delves deep into the dataset's intricacies. This data-driven approach empowers the system to identify subtle and complex patterns that may elude traditional assessments.

The project's success is underscored by the exceptional accuracy it has achieved. With a reported accuracy score of 90.16%, the Random Forest-based system provides a highly reliable method for predicting heart disease. This high degree of accuracy, far beyond what traditional assessments can often offer, reduces the margin for human error, making the predictions more objective and dependable.

However, it's not just the accuracy that makes this project noteworthy; it's the accessibility it brings to the realm of healthcare. The user-friendly graphical interface, developed using the Tkinter library, allows individuals to conveniently input their health data. The simplicity and accessibility of the interface open the door to a wider range of users, including those who may not have easy access to healthcare facilities. This aspect of the project is significant, particularly in remote or underserved areas, where access to healthcare resources is limited. This project democratizes the process of heart disease prediction, ensuring that individuals can proactively assess their heart health from the comfort of their homes.

In conclusion, the project for heart disease prediction using the Random Forest algorithm is a groundbreaking advancement in healthcare and diagnostics. By combining data-driven machine learning with a user-friendly graphical interface, it provides a reliable, accessible, and educational platform for heart disease prediction. It empowers individuals to take charge of their health and facilitates collaboration with healthcare professionals, contributing to early disease detection and improved health outcomes. As we progress toward a future where healthcare is proactive, preventive, and patient-centered, this project stands for progress of better health for all.

5.2 Future Scope:

The future work section outlines the ongoing research and development opportunities in the field of heart disease prediction. Here's a detailed exploration of these potential areas for advancement:

Model Enhancement:
Continuous refinement and optimization of the Random Forest model is necessary. This can include fine-tuning hyperparameters, exploring alternative ensemble methods, or incorporating additional data sources to further improve accuracy.

Larger and Diverse Datasets:
Expanding the dataset with a more extensive and diverse patient population is crucial. A larger dataset can enhance the model's generalizability and robustness, ensuring that it remains effective across a broader range of patient profiles.

Feature Engineering:
Investigating advanced feature engineering techniques is essential to better capture complex relationships between patient attributes and heart disease risk. This can involve the creation of new features or more sophisticated feature transformation.

Interpretable AI:

The development of methods for better understanding and interpreting the Random Forest model's decisions is vital. Enhanced interpretability provides insights into the importance of individual features in the model's predictions and fosters trust among healthcare professionals.

Deployment in Clinical Settings:

Transition from a prototype GUI to a production-ready application that can be seamlessly integrated into clinical practice is a significant step. Collaborating with healthcare institutions to deploy the model for real-world use is essential for its practical impact.

Continuous Monitoring and Feedback:

Implementing mechanisms for ongoing monitoring and feedback collection from healthcare professionals ensures that the model remains accurate and relevant over time. Regular feedback loops allow for necessary adjustments and improvements.

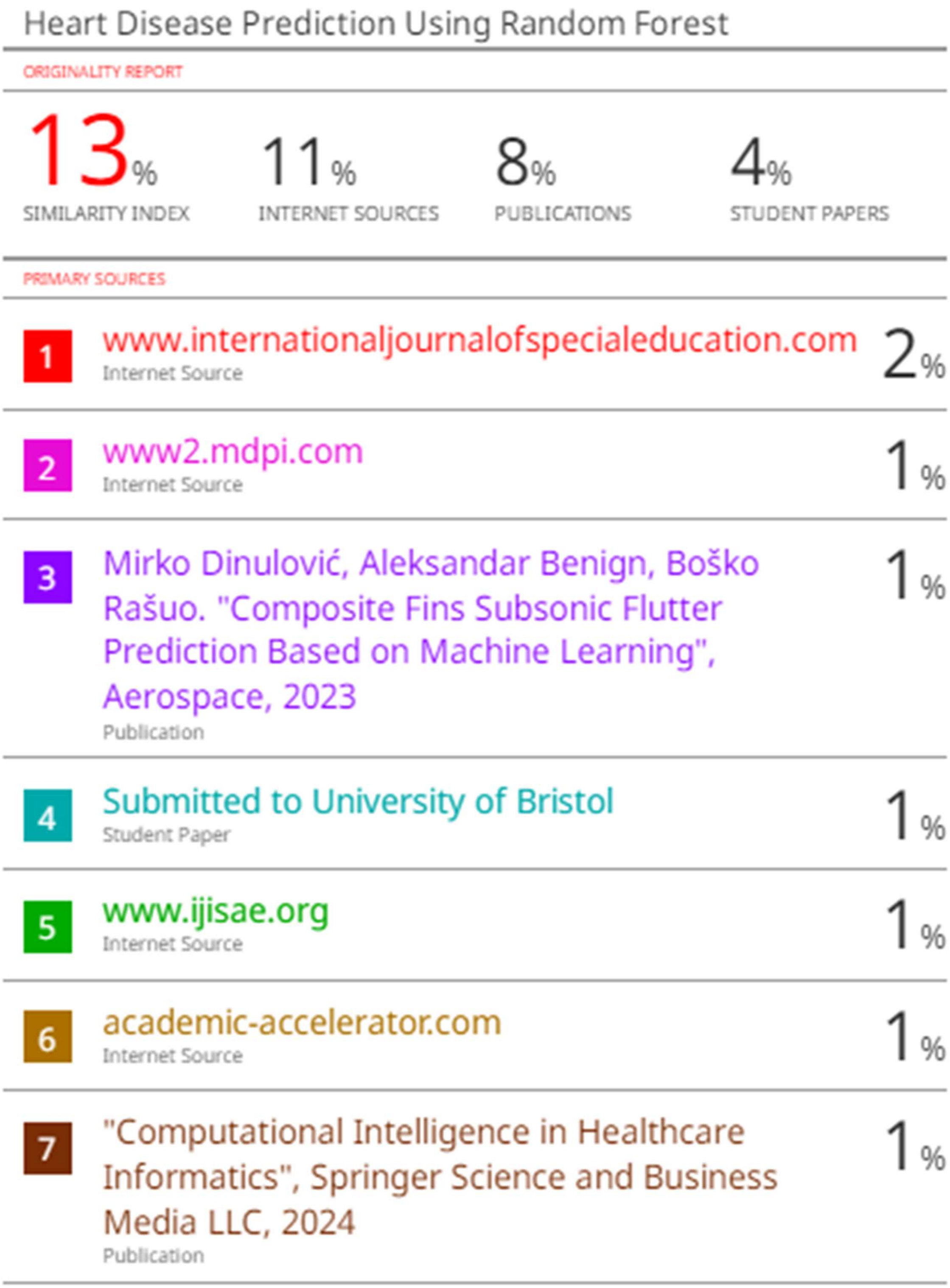
Ethical Frameworks:

Remaining current with evolving ethical standards and data privacy regulations is imperative. Ensuring continuous compliance with healthcare data regulations is essential to maintaining the project's ethical integrity.

REFERENCE

1. https://cardiometry.net/issues/no24-november-2022/heart-disease-prediction
2. https://www.researchgate.net/publication/301732395_Intelligent_heart_disease_prediction_system_using_random_forest_and_evolutionary_approach
3. https://www.kaggle.com/code/kartikeya47/heart-disease-prediction-random-forest
4. https://link.springer.com/article/10.1007/s11227-020-03481-x
5. https://www.researchgate.net/publication/301732395_Intelligent_heart_disease_prediction_system_using_random_forest_and_evolutionary_approach
6. https://iopscience.iop.org/article/10.1088/1742-6596/1817/1/012009
7. https://www.researchgate.net/publication/350312435_A_Cardiovascular_Disease_Prediction_using_Machine_Learning_Algorithms
8. https://jocc.journals.ekb.eg/article_282098_4b9e9c103330a9a045517d04f3a0a14a.pdf
9. https://ijece.iaescore.com/index.php/IJECE/article/view/25334
10. https://iopscience.iop.org/article/10.1088/1742-6596/1817/1/0120098
11. https://cardiometry.net/issues/no24-november-2022/heart-disease-prediction
12. https://jocc.journals.ekb.eg/article_282098_4b9e9c103330a9a045517d04f3a0a14a.pdf
13. https://arxiv.org/pdf/1303.5919
14. Early detection of coronary heart disease using ensemble techniques - ScienceDirect
15. https://www.researchgate.net/publication/368353839_Effective_Heart_Disease_Prediction_Using_Machine_Learning_Techniques

APPENDIX



Heart Disease Prediction Using Random Forest

ORIGINALITY REPORT

13%	11%	8%	4%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	www.internationaljournalofspecialeducation.com Internet Source	2%
2	www2.mdpi.com Internet Source	1%
3	Mirko Dinulović, Aleksandar Benign, Boško Rašuo. "Composite Fins Subsonic Flutter Prediction Based on Machine Learning", Aerospace, 2023 Publication	1%
4	Submitted to University of Bristol Student Paper	1%
5	www.ijisae.org Internet Source	1%
6	academic-accelerator.com Internet Source	1%
7	"Computational Intelligence in Healthcare Informatics", Springer Science and Business Media LLC, 2024 Publication	1%

8	medium.com Internet Source	1 %
9	fritz.ai Internet Source	<1 %
10	www.ijnrd.org Internet Source	<1 %
11	Khalidou Abdoulaye Barry, Youness Manzali, Outmane Labaybi, Flouchi Rachid, Mohamed Elfar. "Heart Disease Prediction and Classification Using Machine Learning Algorithms", 2023 7th IEEE Congress on Information Science and Technology (CiSt), 2023 Publication	<1 %
12	ph.pollub.pl Internet Source	<1 %
13	psnacet.edu.in Internet Source	<1 %
14	www.ijdiic.com Internet Source	<1 %
15	annalsofrscb.ro Internet Source	<1 %
16	robots.net Internet Source	<1 %
17	dokumen.pub	

	Internet Source	<1 %
18	restpublisher.com Internet Source	<1 %
19	www.frontiersin.org Internet Source	<1 %
20	www.irjmets.com Internet Source	<1 %
21	Chintan M. Bhatt, Parth Patel, Tarang Ghetia, Pier Luigi Mazzeo. "Effective Heart Disease Prediction Using Machine Learning Techniques", Algorithms, 2023 Publication	<1 %
22	Tanjim Mahmud, Anik Barua, Manoara Begum, Eipshita Chakma, Sudhakar Das, Nahed Sharmen. "An Improved Framework for Reliable Cardiovascular Disease Prediction Using Hybrid Ensemble Learning", 2023 International Conference on Electrical, Computer and Communication Engineering (ECCE), 2023 Publication	<1 %

Exclude quotes	On	Exclude matches	Off
Exclude bibliography	On		