INT 368

Project report on

Heart Disease Prediction using Machine Learning Algorithms

Bachelor of Technology

(Computer Science and Engineering)

Submitted to

LOVELY PROFESSIONAL UNIVERSITY

PHAGWARA, PUNJAB

Subject Teacher

Mr. Ajay Sharma

SUBMITTED BY

Name of Student: Godina Venkata Akhil Chandra

Registration Number: 12014751

Roll No.: RK20CHB45

DECLARATION STATEMENT

I hereby declare that the work reported in the project report proposal entitled "HEART DISEASE PREDICTION USING MACHINE LEARNING ALGORITHMS" in partial fulfilment of the requirement for the award of Degree for Bachelor of Technology in Computer Science and Engineering at Lovely Professional University, Phagwara, Punjab is an authentic work carried out under supervision of my project supervisor Mr. Ajay Sharma. I have not submitted this work elsewhere for any degree or diploma.

I understand that the work presented here with is in direct compliance with Lovely Professional University's Policy on plagiarism, intellectual property rights, and highest standards of moral and ethical conduct. Therefore, to the best of my knowledge, the content of this dissertation represents authentic and honest research effort conducted, in its entirety, by me. I am fully responsible for the contents of my dissertation work.

Signature of Candidate

Godina Venkata Akhil Chandra

R.N0.: RK20CHB45

ACKNOWLEDGEMENT

The outcome of learning Machine Learning project required a lot of guidance and assistance from the instructor and his team and I am happy to have got all along the completion of my course and few of the projects. All I have done is only due to such supervision and assistance and I would like to thank them. I respect and thank Lovely Professional University and upGrad Campus, for providing me an opportunity to do the course and project work and giving me all support and guidance, which made me complete the course. I am extremely thankful to the course instructor **Mr. Ajay Sharma** sir and fortunate enough to get constant encouragement, support, and guidance from the members of upGrad Campus which helped me in successfully completing my course and project work.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

OBJECTIVE AND SCOPE OF THE PROJECT

Background: One of the major causes of morbidity in the world's population is heart attack prediction. Cardiovascular disease is a very essential disease that is included in the clinical data analysis as one of the most crucial sections for the prediction. When predicting heart attacks, data science and machine learning (ML) can be very helpful by considering many risk factors such as high blood pressure, high cholesterol, an abnormal pulse rate, diabetes, etc. This work aims to improve machine learning (ML)-based cardiac disease prediction.

Methodology: In this research, we describe a machine learning-based heart attack prediction (ML-HAP) method that uses Logistic Regression and Naive Bayes Classifier ML methodologies to analyse various risk factors and predict heart attacks. From the UCI ML Repository, data on heart disease symptoms were gathered, and machine learning (ML) methods were used to analyse the data. Enhancing the prediction based on several parameters has been the focus.

Observation: Of the two, Logistic Regression offered the most accurate prediction. Naive Bayes and Logistic Regression both obtain an area under the curve of.85. With boosting techniques, the prediction with ML models for identifying heart attack symptoms is very effective. The prediction was made to assess the forecast's area under the curve, recall, accuracy, and precision. ML models are being developed to make the most accurate forecasts.

Conclusions: This prediction can be useful clinically in interpreting the patient scenario and analysing the illness risk variables. Increasing the algorithm produced good outcomes for predicting heart disease symptoms. By focusing on the conditions risk factors more, it can be improved still further.

INTRODUCTION

One of the most catastrophic conditions in the cardiovascular disease subset is a heart attack, also known as an acute myocardial infarction (AMI). It happens when the blood supply to the heart's muscle is cut off, harming the heart's muscle. Another vital task is making a heart disease diagnosis. Heart disease must be diagnosed using the signs and symptoms, physical exam, and knowledge of the various symptoms. Heart disease can be caused by a variety of causes, including excessive cholesterol, inherited heart disease, high blood pressure, a lack of physical activity, obesity, and smoking. Blood flow to the coronary arteries becoming blocked is the main cause of heart attacks.

When blood flow is decreased, the red blood cells (RBC) begin to decline; as a result, the body stops receiving the essential oxygen and a person loses consciousness. If the prediction is accurate enough, an early diagnosis using symptoms and signs may help patients avoid heart attacks. Different heart attack symptoms are depicted in Figure 1 features or attributes with numerical values are input into the work that is being presented. It has been said that making little lifestyle changes like giving up smoking, drinking, or using tobacco, eating healthily, and exercising regularly can help prevent heart attacks.

Any person who leads a healthy lifestyle and receives therapy quickly after being diagnosed can significantly improve the outcomes. However, it can be challenging to determine someone who has a high risk of developing heart disease when other concerns like diabetes, high blood pressure, and cholesterol issues are also present. ML can aid in the early detection of disease in these kinds of situations.

Following are some of the common reasons which leads to heart disease in a healthy human being.

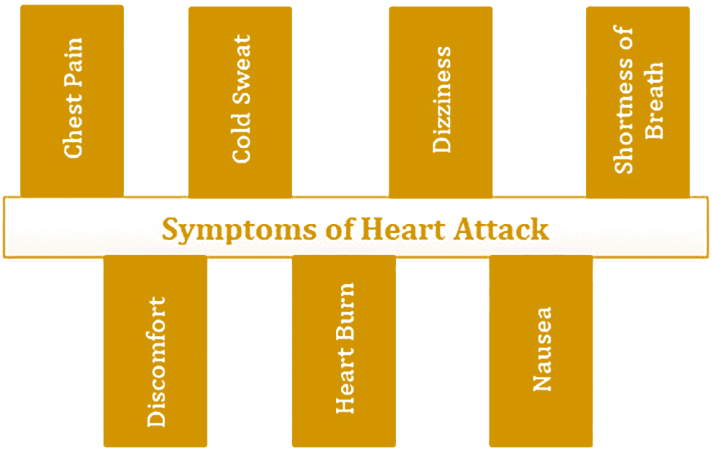


Fig 1.1: Symptoms of Heart Attack

About Dataset

The four databases in this data set are Cleveland, Hungary, Switzerland, and Long Beach V collected in 1988. It has 76 properties, including the one that was anticipated, however all published experiments only mention using a portion of 14 of these. The "target" field alludes to the patient's having heart illness. “0” means there is no disease, while “1” means there is a disease.

Attributes in the data set:

- age: Age in years
- sex: sex (1 = male; 0 = female)
- Cp: Chest pain type
  -- Value 1: typical angina
  -- Value 2: atypical angina
  -- Value 3: non-anginal pain
  -- Value 4: asymptomatic chest
- trestbps: Resting blood pressure (in mm Hg on admission to the hospital).
- chol: Serum cholesterol in mg/dl.
- fbs: (fasting blood sugar > 120 mg/dl) (1 = true; 0 = false).
- restecg: Resting electrocardiographic results
  -- Value 0: Normal
  -- Value 1: Having ST-T wave abnormality (T wave inversions and/or ST elevation or depression of > 0.05 mV).
  -- Value 2: Showing probable or definite left ventricular hypertrophy by Estes' criteria.
- thalach: Maximum heart rate achieved
- exang: Exercise induced angina (1 = yes; 0 = no)
- oldpeak = ST depression induced by exercise relative to rest
- slope: The slope of the peak exercise ST segment
  -- Value 1: upsloping
  -- Value 2: flat
  -- Value 3: down sloping
- ca: Number of major vessels (0-3) coloured by fluoroscopy.
- 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.

LITERATURE SURVEY

Using various techniques, a thorough search of earlier work in the field of heart disease has been conducted. In order to enhance our research, the preceding 21 years of effort have been taken into consideration and their flaws have been identified. After removing duplicates and publications with the same domain as others, a total of 50 papers from Web of Science, Science direct, and Scopus were obtained. Of these, 27 were chosen for the final study.

From January 1, 2021, through December 31, 2021, Scopus, Web of Science, and Science Direct were used to gather literature. The obtained publications underwent scrutiny. To comprehend the difficulties in the field of heart disease prediction, analysis is conducted. The collected papers were examined, and the work's advantages and disadvantages were noted based on the evaluation criteria, methodology, and application of algorithms.

The inclusion criteria were based on finding relevant publications, using the most recent machine learning techniques, and tough areas in the field of heart disease. "Machine learning based health disease prediction," "optimisation of Health disease prediction," and "Challenges in identifying health disease" are search terms for finding papers. Duplicate articles, papers that showed subpar work in terms of evaluation parameter values, and outdated work were among the exclusion criteria.

In one study, a neural network was used to construct an electronic health record (ehr) model based on sequential modelling. The EHR was used to conduct experiments and forecast heart disease. In this study, word vectors and hot encryption were utilised to simulate diagnostic scenarios and forecast heart failure. A network-based extended memory model was used in conjunction with the same methodology. According to the paper, using outcomes analysis to take care of the sequential nature of healthcare is quite important. The sequential nature of healthcare includes monitoring a person's behaviour, such as his or her activities related to their health, changes in their healthcare providers during illness, exercise routines, dietary routines, etc.

Cardiovascular (CV) disease is more common in diabetics. Both fasting glucose levels and glycosylated haemoglobin have been used for determining CV risk-assessment techniques. There is conflicting evidence to suggest that these components are being utilised. According to the cardiovascular heart study, there is only a minimal correlation between fasting blood glucose and CV risk. Like our findings, other investigations conducted by other researchers have demonstrated a relationship between postprandial glucose levels and CV risk as well as glycosylated haemoglobin levels.

The available risk-assessment tools are not applicable to everyone due to our differences in genetic make-up, cultures, food preferences, and social and behavioural traits. Researchers examined the global burden of CV disease and found that different populations have varying disease burdens as well as various primary Rheumatic fevers (RFs) that contribute to this burden. The Asia Pacific Cohort research aimed to compare the Framingham and Asian cohorts in terms of risk factors and illness incidence and found that the Asian cohort had higher smoking rates. The Framingham group also had higher systolic blood pressure, total cholesterol, and CV events. The risk-assessment instruments to be used in Asian populations for risk stratification have not been agreed upon.

Clinicians are confused as a result and are unable to use risk stratification to rank people for primary prevention interventions. As a result, it has been argued that it will be advantageous to create a predictive equation using population-based data that is both current and representative. Consideration has been given to the present mix of known and unidentified RF depending on genetic features. We must therefore be mindful of the limitations of each of these risk-assessment methodologies and exercise caution when interpreting the findings.

HARDWARE AND SOFTWARE USED

Hardware:

- CPU: 11th Gen Intel Core i5-11300H @ 3.10GHz 3.11 GHz
- GPU: Intel Iris Xe
- RAM: Kingston 8GB ram clocked at 3200Mhz
- Memory: 512 GB Seagate M.2 NVMe SSD

Software:

- OS: Microsoft Windows 10 (64-bit)
- Programming Language: Python 3.9
- Code editor: Jupyter Notebook
- Data set: Microsoft Excel

Python Libraries:

- Numpy
- Pandas
- Sci-kit
- Matplotlib
- Seaborn

METHODOLOGY AND ALGORITHMS USED

Research plan

Below is a list of all the steps in this research. Error detection, locating relevant information, and verifying the correlation between exploratory analysis variables are all accomplished using exploratory data analysis (EDA). The risk factors associated with heart disease are considered in this work, as well as the heart attack prediction. Logistic regression and naive Bayes are the machine learning classifiers that were used in this study. The past heart disease prediction tests have been considered in a thorough literature review, and the classifiers Logistic Regression and Naive Bayes are considered based on their performance characteristics. Below is a list of all the steps in this investigation.

1. The first phase is of data acquisition, often known as data collection. This involved assessing the physical conditions and considering the numerical data by converting the samples that the computer will use to alter.
   a. Data for studying the risk factors for heart disease was gathered from the UCI ML repository, as described in the section on data acquisition.

2. The second phase, termed "pre-processing," involves cleaning up the dataset by addressing issues with the data such missing values, outlier detection, and redundancy removal. For the uniform environment, predictive analysis has been carried out, moving the application closer to EDA.
   a. The gathered data has been cleaned using pre-processing techniques like duplicate removal, outlier identification, and missing value replacement.
   b. Mean values are used to replace any missing values.
   c. By comprehending the minimum, maximum, and interquartile ranges of the data, outliers in the dataset have been identified using Boxplots.
   d. In order to eliminate duplicates from the data, the function dict() was used to create a dictionary.

3. The third phase, called "integration," involves combining libraries and other subsets by importing separate Python modules and merging them to carry out the required tests.
   a. The experiment's first step was to have the pre-processed data.
   b. After the data had been cleansed, ML algorithms were merged.

4. The fourth phase is "analysis phase" involved using EDA to determine how various data qualities relate to one another.
   a. Analysis relies on the idea of learning from data, spotting patterns, and making conclusions with the least amount of human involvement.
   b. EDA is being used to understand how the attributes relate to one another.
   c. To understand the correlation, the variables were compared, and the same variables were examined using pair plot, boxplots and heatmaps.

5. The final phase was "intervention" to enter decision-making procedures, which involved using a search method to comprehend earlier experimental investigations to ascertain whether it was effective to apply models to real-world issues.
    a. To learn how ML models are used in the same domain and to identify the most promising ones for enhancing our outcomes, a thorough literature review was conducted. Based on their performance in previously implemented work in the related domains for heart disease, the most promising publications were chosen.

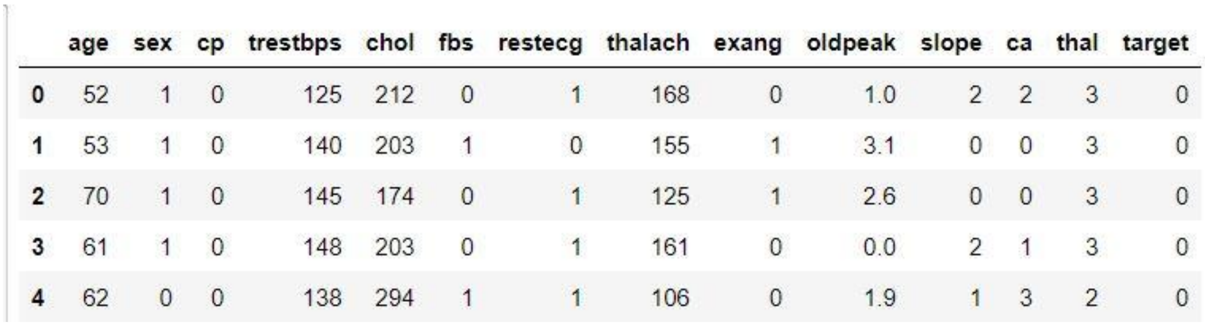
6. The 'application' of ML algorithms to the prediction process was the sixth phase. Four machine learning models, including Naive Bayes and Logistic Regression, were used in this study.
    a. In Python, the linear model class of sk-learn was used using logistic regression.
    b. The Scikit Learn library of neighbours is being used to apply the Naive Bayes classifier in Python.

The task is done in stages commencing with the collection of data. The data has undergone pre-processing, such as duplicate removal, outlier detection, and mean filling in for missing values, to make it cleaner. The outputs were then classified further using Logistic Regression and Naive Bayes, which are two of the four machine learning classifiers.

Data Collection

The dataset in use is divided into four sections, or sub-databases, namely Hungary, Switzerland, Cleveland, and Long Beach, and it comprises 76 distinct properties. A selection of 14 qualities is used in this study since the literature review's published experiments all referred to the 14 attributes that were chosen because they are crucial to understanding the main heart disease risk factors. This dataset is publicly accessible for use in experiments online in the UCI repository. The final column, called the target value, displays the patient's disease status as indicated by binary values of O or 1, respectively.

A sample of the data set is displayed here, and the prediction is being made on the entire dataset in order to highlight its characteristics and behaviour.



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

Table 3.1: Sample dataset

Modelling with ML

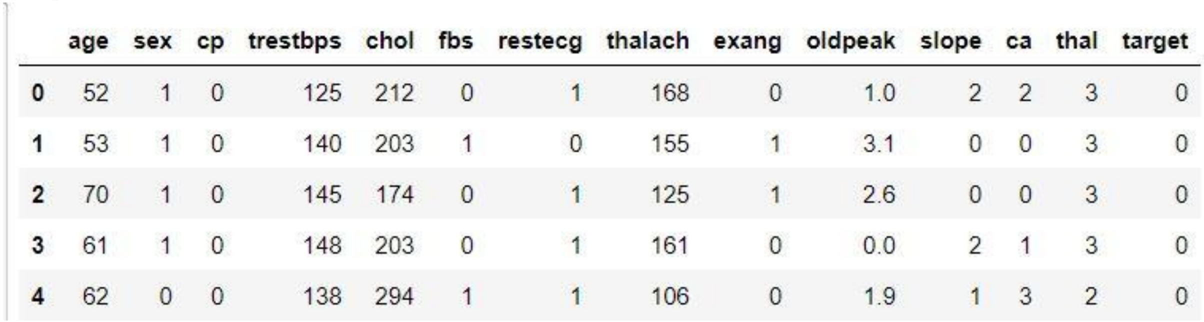
Two machine learning (ML) models were used to complete the study: logistic regression and naive Bayes.

Logistic regression: A supervised learning model and one of the most often used algorithms is logistic regression. The categorical predictions it makes can either be "true" or "false." Rather of exact values, this model offers probabilistic ones. Both continuous and discrete values can be used with this approach. A straightforward S-shaped curve can precisely elaborate the logistic regression.

Nave Bayes: Nave Bayes is a supervised learning model that uses the Bayes theorem and can make quick predictions. Given high dimensional data, our probabilistic classifier performs incredibly well.

WORKING CODE

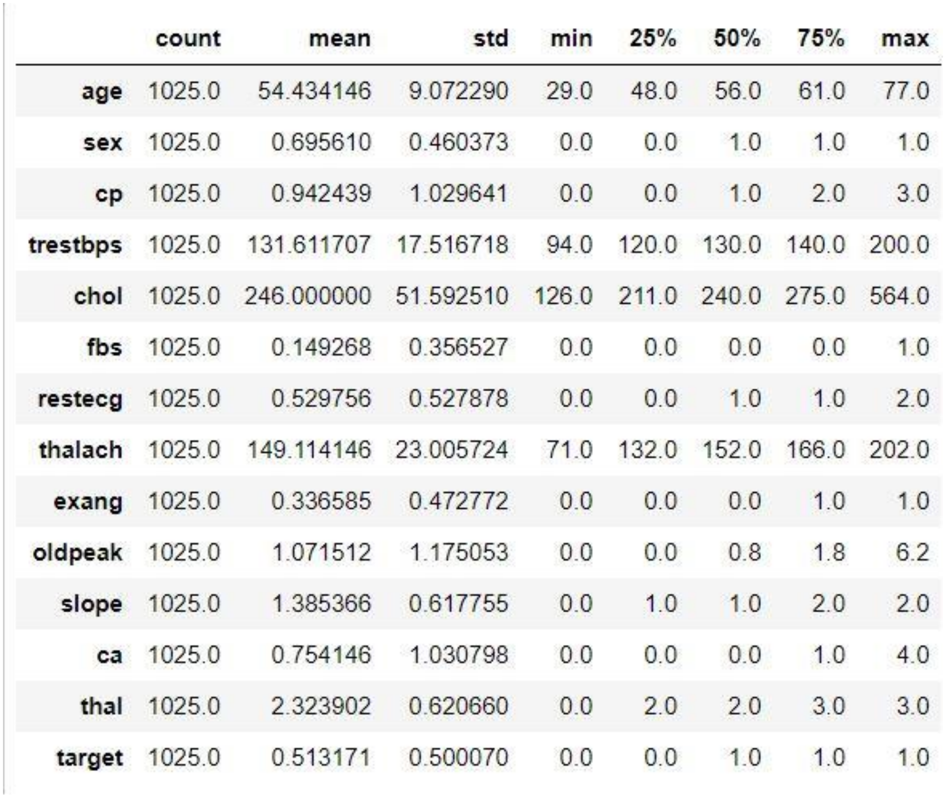
```
# importing required libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

import warnings
warnings.filterwarnings('ignore')

# creating dataframe using csv
data = pd.read_csv("heart.csv")

# creating copy of main dataframe
df = data.copy()

# shape of dataframe
print("Data frame has", df.shape[0], "rows.")
print("Data frame has", df.shape[1], "columns.")
output:
Data frame has 1025 rows.
Data frame has 14 columns.

# printing first 5 rows of dataframe
df.head()
output:
```



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

```
# Number of non-null values and data type of each attribute
df.info()
output:
<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
```

```
# basic statistical features of each attribute
df.describe().transpose()
Output:
```



	count	mean	std	min	25%	50%	75%	max
age	1025.0	54.434146	9.072290	29.0	48.0	56.0	61.0	77.0
sex	1025.0	0.695610	0.460373	0.0	0.0	1.0	1.0	1.0
cp	1025.0	0.942439	1.029641	0.0	0.0	1.0	2.0	3.0
trestbps	1025.0	131.611707	17.516718	94.0	120.0	130.0	140.0	200.0
chol	1025.0	246.000000	51.592510	126.0	211.0	240.0	275.0	564.0
fbs	1025.0	0.149268	0.356527	0.0	0.0	0.0	0.0	1.0
restecg	1025.0	0.529756	0.527878	0.0	0.0	1.0	1.0	2.0
thalach	1025.0	149.114146	23.005724	71.0	132.0	152.0	166.0	202.0
exang	1025.0	0.336585	0.472772	0.0	0.0	0.0	1.0	1.0
oldpeak	1025.0	1.071512	1.175053	0.0	0.0	0.8	1.8	6.2
slope	1025.0	1.385366	0.617755	0.0	1.0	1.0	2.0	2.0
ca	1025.0	0.754146	1.030798	0.0	0.0	0.0	1.0	4.0
thal	1025.0	2.323902	0.620660	0.0	2.0	2.0	3.0	3.0
target	1025.0	0.513171	0.500070	0.0	0.0	1.0	1.0	1.0

Table 4.01: statistical description of dataset

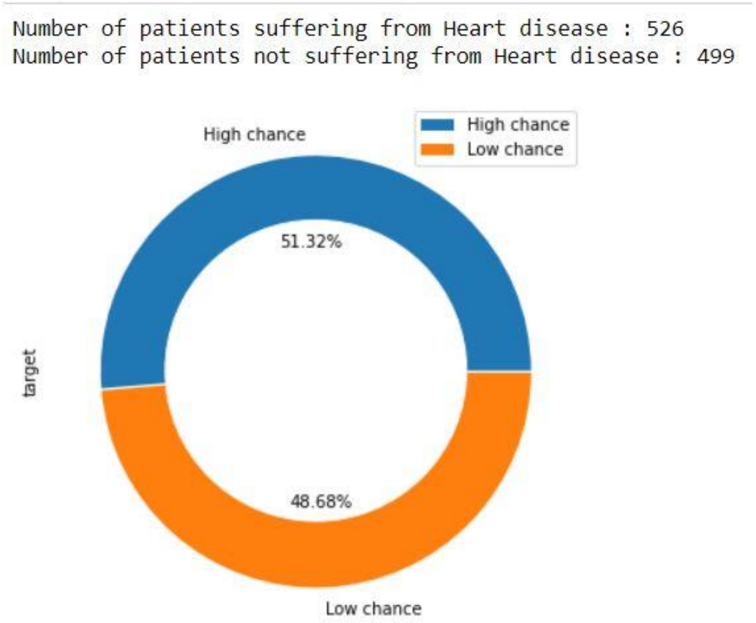
```
# number of unique values in each attribute
df.nunique()
Output:
age          41
sex           2
cp            4
trestbps     49
chol        152
fbs           2
restecg       3
thalach      91
exang         2
oldpeak      40
slope         3
ca            5
thal          4
target        2
dtype: int64
```

Data Manipulation

```
# sum of null values in each attribute
df.isnull().sum()
Output:
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


# total sum of null values
df.isnull().sum().sum()
Output:
0
```

Exploratory Data Analysis

```
# number of patients suffering and not suffering from heart disease
print("Number of patients suffering from Heart disease :",
df['target'].value_counts()[1])
print("Number of patients not suffering from Heart disease :",
df['target'].value_counts()[0])

# piechart to visualize the patients with and without heart disease
plt.figure(figsize=(6,6))
df['target'].value_counts().plot.pie(labels=['High chance', 'Low chance'],
explode=(0,0.01), autopct='%1.2f%%')

circle = plt.Circle( (0,0), 0.7, color='white')
p=plt.gcf() # get current figure
p.gca().add_artist(circle) # get current axis
plt.legend()
plt.show()
Output:
```



```
Number of patients suffering from Heart disease : 526
Number of patients not suffering from Heart disease : 499
```

Fig: 6.01 number of patients suffering and not suffering from heart disease

```
# number of male and female patients and their age distribution
print("Percentage of MALE patients:
"+str(round(df.sex.value_counts()[0]*100/1025,2)) + "%")
print("Percentage of FEMALE patients:
"+str(round(df.sex.value_counts()[1]*100/1025,2)) + "%")

plt.figure(figsize=(18,5))

plt.subplot(1,2,1)
df.sex.value_counts().plot(kind='barh', color=sns.color_palette('rainbow'),
edgecolor='black')
plt.title("Nummber of patients according to gender")

plt.subplot(1,2,2)
sns.kdeplot(data=df, x="age", hue="sex", fill=True, common_norm=False,
palette="mako", alpha=.5)
plt.title("Age distribution according to gender")
plt.show()
```

Output:

```
Percentage of MALE patients: 30.44%
Percentage of FEMALE patients: 69.56%
```

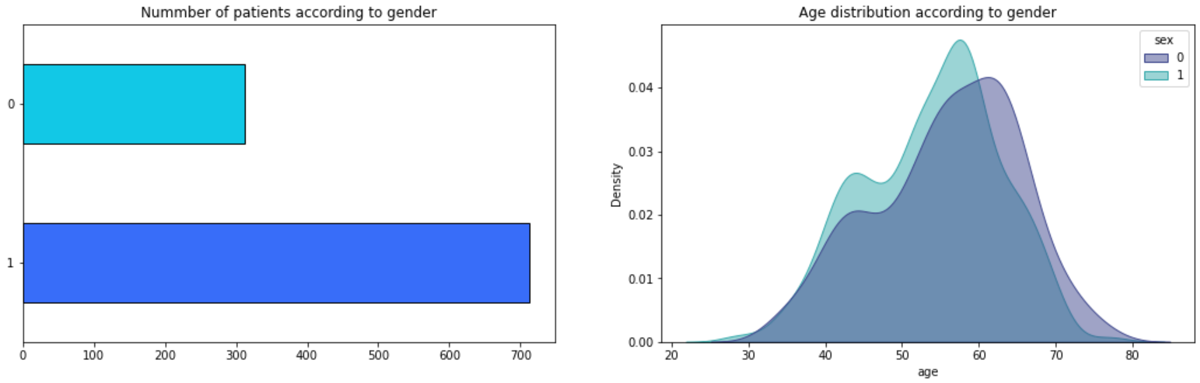


Fig:6.02 number of male and female patients and their age distribution

```
# number of patients having 0,1,2 type of slope
sns.barplot(df['slope'], df['slope'].value_counts())
plt.title("Number of patients per slope type")
plt.show()
```

Output:

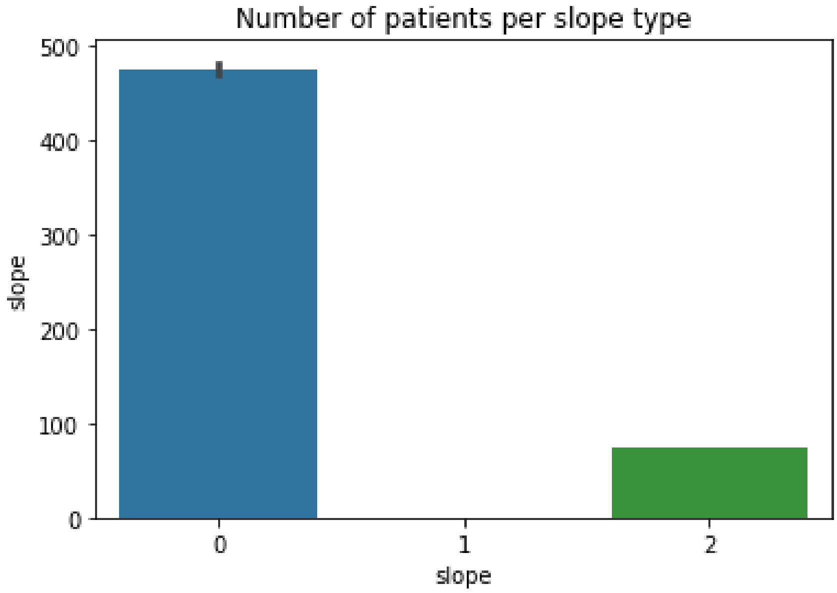


Fig:6.03 number of patients having 0,1,2 type of slope

```
# Age, blood pressure, cholesterol distribution
print("Median age of patients:", df.age.median())
print("Median resting blood pressure of patients:", df.trestbps.median())
print("Median Serum cholesterol in mg/dl. of patients:", df.chol.median())
plt.figure(figsize=(18,5))

plt.subplot(1,3,1)
sns.kdeplot(df.age, fill=True, alpha=0.6, color='red', edgecolor='red')
plt.axvline(df['age'].median(), c='black', linestyle='dotted')

plt.subplot(1,3,2)
sns.kdeplot(df.trestbps, fill=True, alpha=0.3, color='blue',
edgecolor='blue')
plt.axvline(df['trestbps'].median(), c='black', linestyle='dotted')

plt.subplot(1,3,3)
sns.kdeplot(df.chol, fill=True,
alpha=1,color='lawngreen',edgecolor='green')
plt.axvline(df['chol'].median(), c='black', linestyle='dotted')

plt.show()
```

Output:

```
Median age of patients: 56.0
Median resting blood pressure of patients: 130.0
Median Serum cholesterol in mg/dl. of patients: 240.0
```

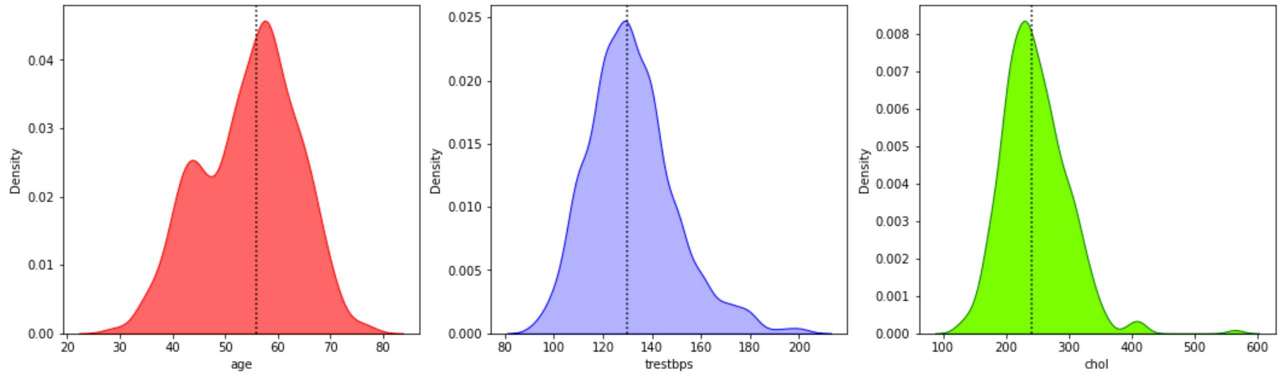


Fig:6.04 Age, blood pressure, cholesterol distribution

```
# fbs, thalach, old peak distribution
print("Median thalach of patients:", df.thalach.median())
print("Median oldpeak of patients:", df.oldpeak.median())
plt.figure(figsize=(18,5))

plt.subplot(1,3,1)
plt.xlabel('fbs')
df['fbs'].value_counts().plot(kind='bar', color='whitesmoke',
edgecolor='black')

plt.subplot(1,3,2)
sns.kdeplot(df.thalach, fill=True, alpha=0.3, color='blue',
edgecolor='blue')
plt.axvline(df['thalach'].median(), c='black', linestyle='dotted')

plt.subplot(1,3,3)
sns.kdeplot(df.oldpeak, fill=True,
alpha=1,color='lawngreen',edgecolor='green')
plt.axvline(df['oldpeak'].median(), c='black', linestyle='dotted')

plt.show()
```

Output:

```
Median thalach of patients: 152.0
Median oldpeak of patients: 0.8
```

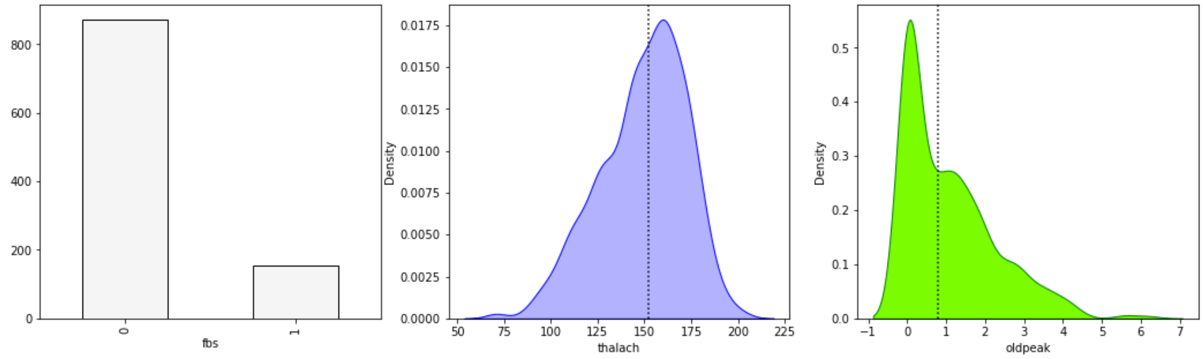


Fig:6.05 fbs, thalach, old peak distribution

```
# type of chest pain, restecg and slope percentage wise
plt.figure(figsize=(18,6))

plt.subplot(1,3,1)
myexplode = [0, 0.025, 0.05, 0.075]
```

```
df['cp'].value_counts().plot.pie(explode=myexplode, autopct='%1.2f%%',
colors=sns.color_palette('rainbow'))
plt.title("Percentage wise distribution of 'cp'")

plt.subplot(1,3,2)
myexplode = [0, 0.05, 0.075]
df['restecg'].value_counts().plot.pie(autopct='%1.2f%%',
colors=sns.color_palette('rainbow_r'))
plt.title("Percentage wise distribution of 'restecg'")

plt.subplot(1,3,3)
df['slope'].value_counts().plot.pie(explode=myexplode, autopct='%1.2f%%',
colors=sns.color_palette('rainbow'))
plt.title("Percentage wise distribution of 'slope'")
plt.show()
```

Output:

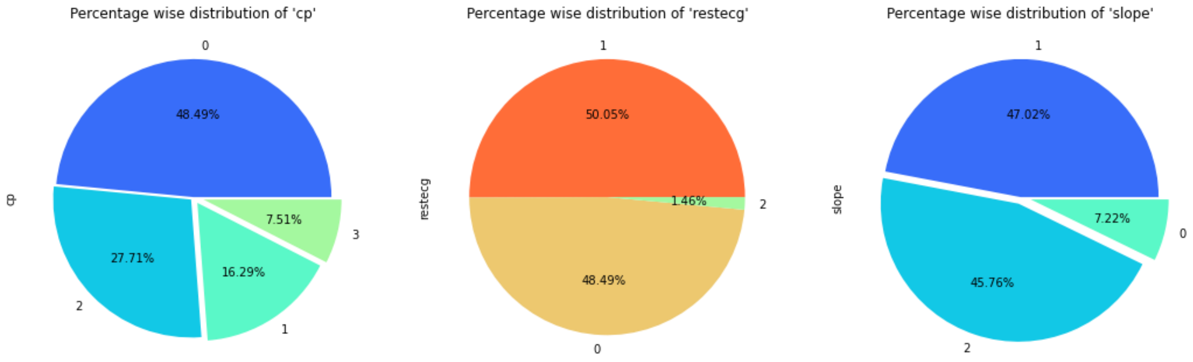


Fig:6.06 type of chest pain, restecg and slope percentage wise

```
# gender wise presence of heart disease
pd.crosstab(df.sex, df.target).plot(kind="bar", figsize=(10,5),
color=['lime','red' ], edgecolor='black')
plt.title('Heart Disease Frequency with respect to gender')
plt.xlabel('Sex (0 = Female, 1 = Male)')
plt.xticks(rotation=0)
plt.legend(["Heart disease not present", "Heart disease present"])
plt.ylabel('Frequency')
plt.show()
```

Output:

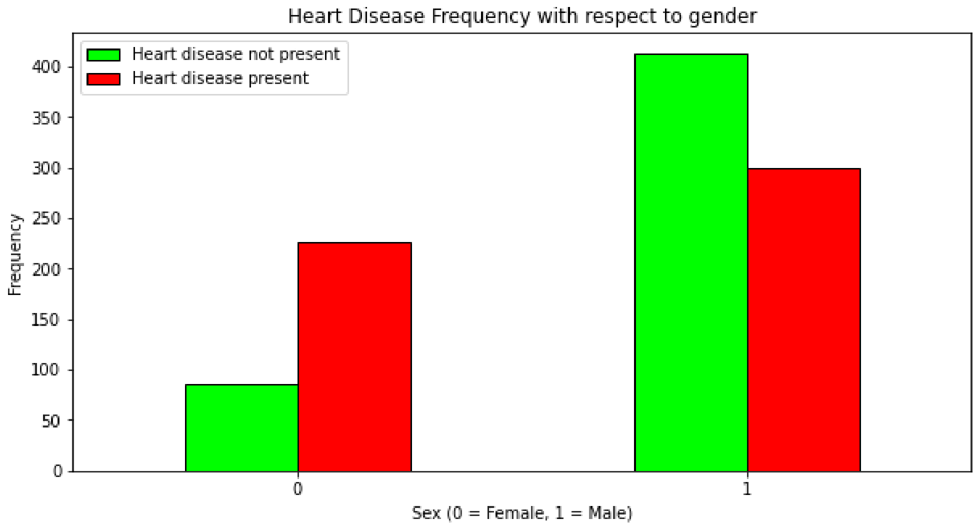


Fig:6.07: Heart Disease Frequency with respect to gender

Preview:

```
# distribution of patients age with respect to heart disease
plt.figure(figsize=(10, 5))
sns.kdeplot(data=df, x="age", hue="target", fill=True, common_norm=False,
palette="mako", alpha=.5)
plt.title("Distribution of patients age with respect to heart disease")
plt.show()
```

Output:

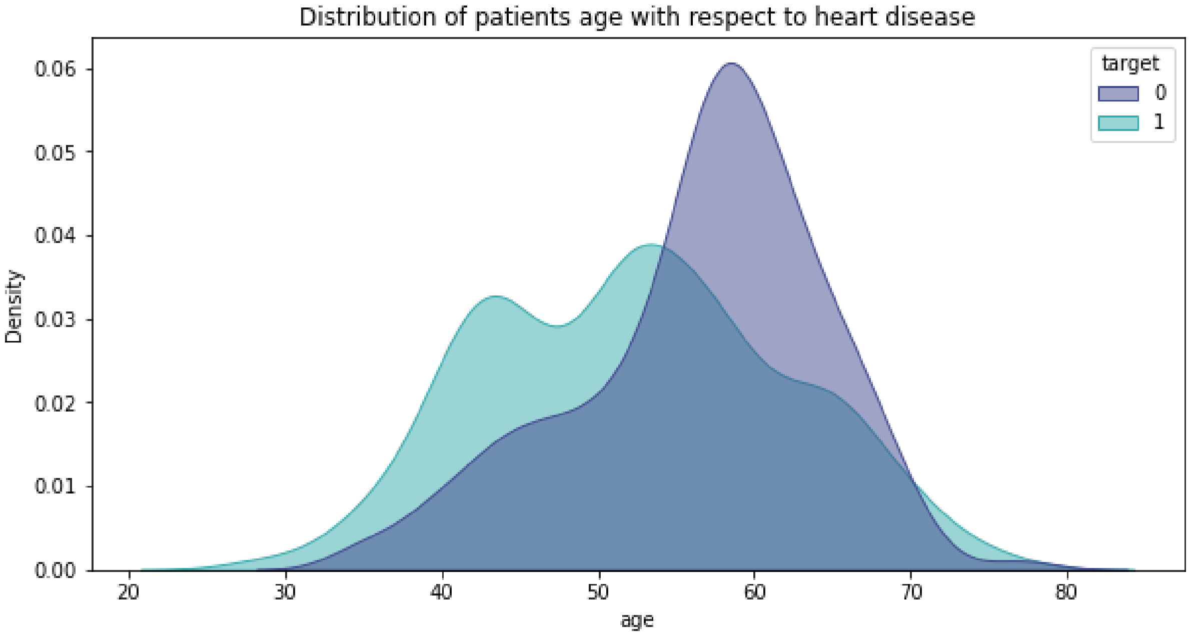


Fig:6.08: Distribution of patients age with respect to heart disease

```
# age wise presence of heart disease representation
pd.crosstab(df.age, df.target).plot(kind="bar",figsize=(18,6))
plt.title('Heart Disease Frequency for Ages')
plt.xlabel('Age')
plt.ylabel('Frequency')
plt.show()
```

Output:

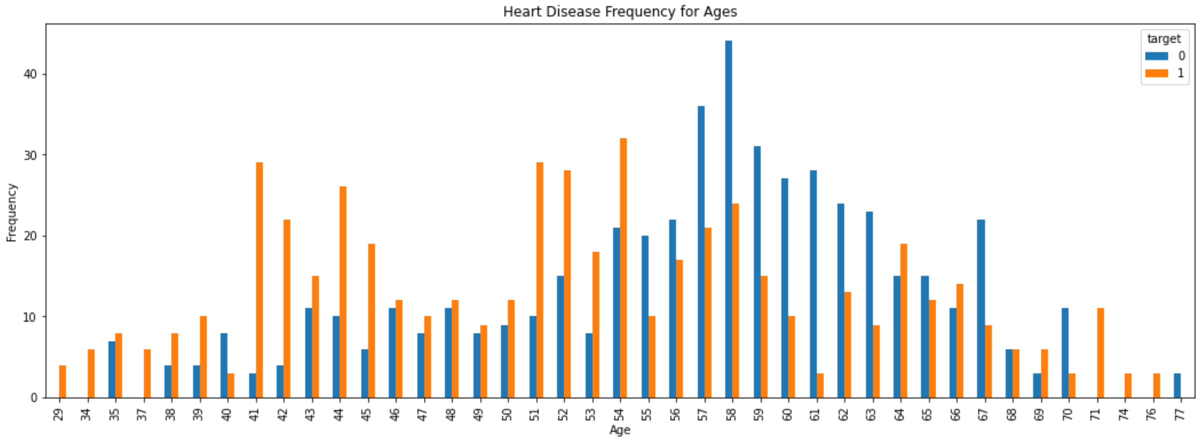


Fig:6.09: Heart Disease Frequency for Ages

```
# distribution of patients trest with respect to heart disease
plt.figure(figsize=(10, 5))
sns.kdeplot(data=df, x="trestbps", hue="target", fill=True,
common_norm=False, palette="rocket_r", alpha=.5)
plt.title("Distribution of patients trest with respect to heart disease ")
plt.show()
```

Output:

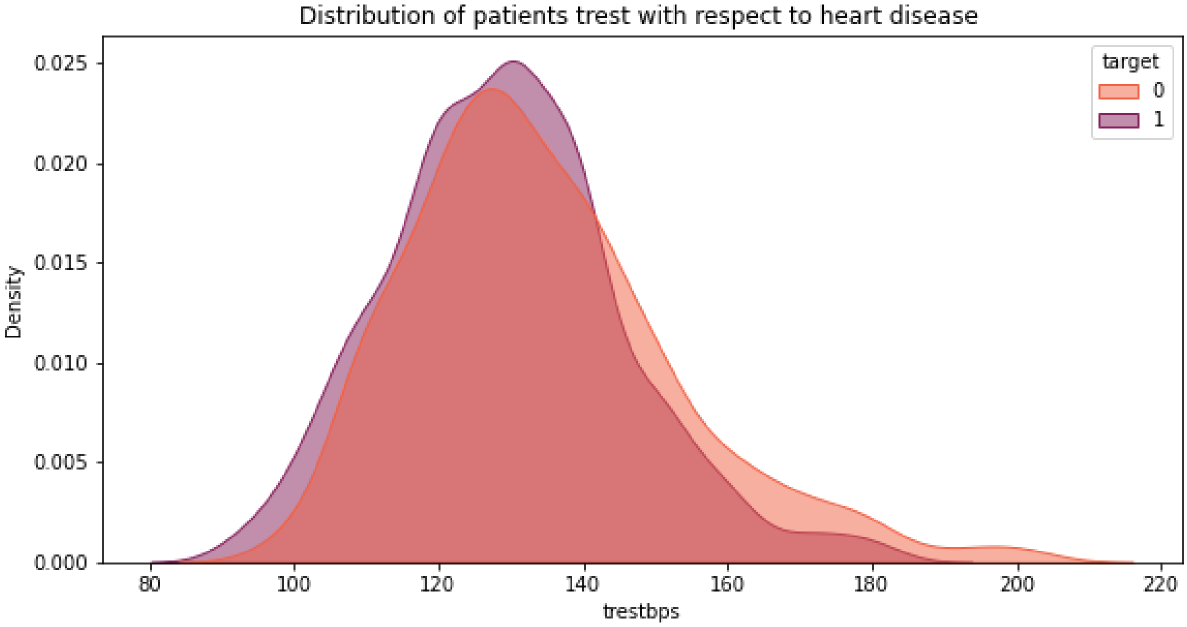


Fig:6.10: Distribution of patients trest with respect to heart disease

```
# distribution of patients chol with respect to heart disease
plt.figure(figsize=(10, 5))
sns.kdeplot(data=df, x="chol", hue="target", fill=True, common_norm=False,
palette="viridis", alpha=.5)
plt.title("Distribution of patients chol with respect to heart disease")
plt.show()
```

Output:

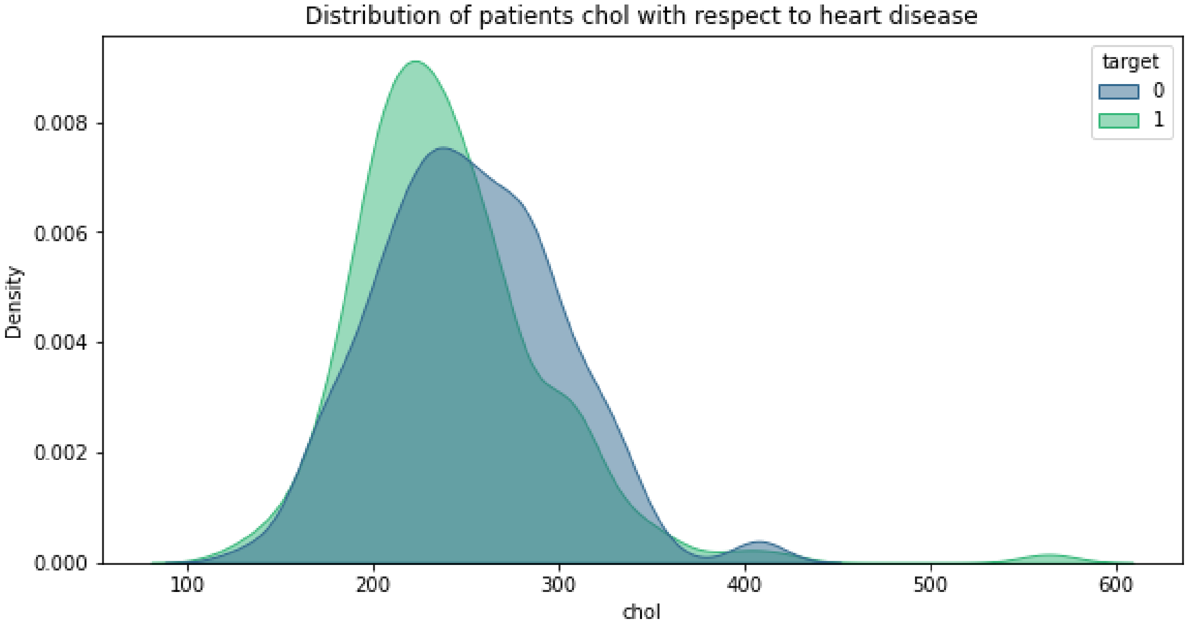


Fig:6.11: Distribution of patients chol with respect to heart disease

```
# distribution of patients oldpeak with respect to heart disease
plt.figure(figsize=(10, 5))
sns.kdeplot(data=df, x="oldpeak", hue="target", fill=True,
common_norm=False, palette="magma", alpha=.5)
plt.title("Distribution of patients oldpeak with respect to heart disease")
plt.show()
```

Output:

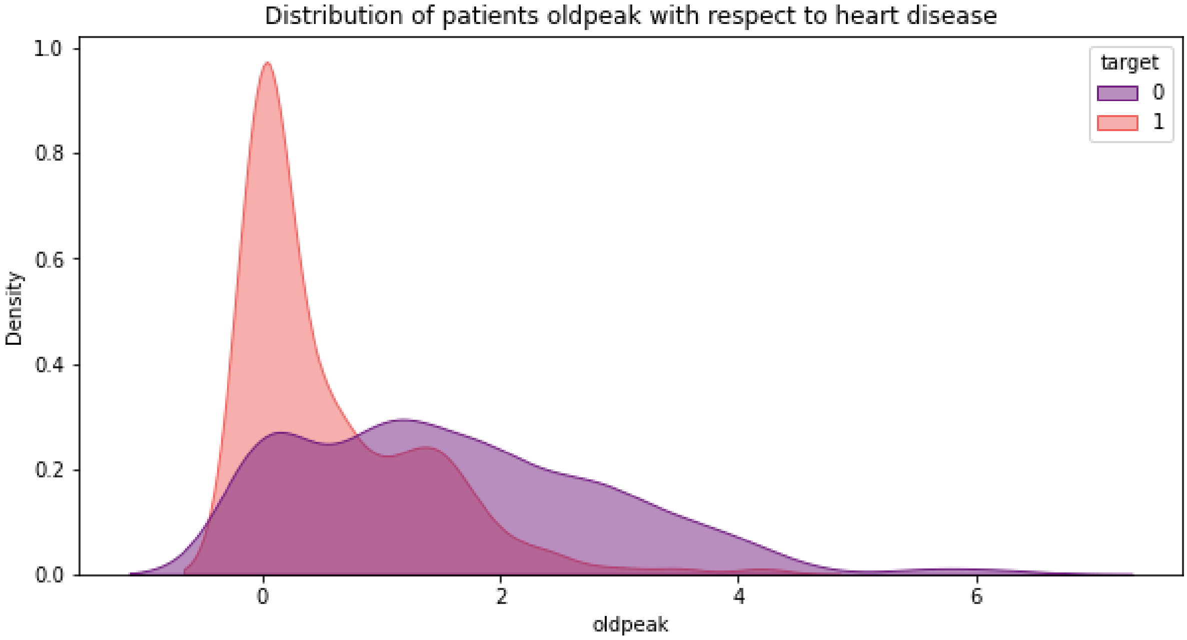


Fig:6.12: Distribution of patients oldpeak with respect to heart disease

```
# distribution of patients thalach with respect to heart disease
plt.figure(figsize=(10, 5))
sns.kdeplot(data=df, x="thalach", hue="target", fill=True,
common_norm=False, palette="rocket_r", alpha=.5)
plt.title("Distribution of patients thalach with respect to heart disease")
plt.show()
```

Output:

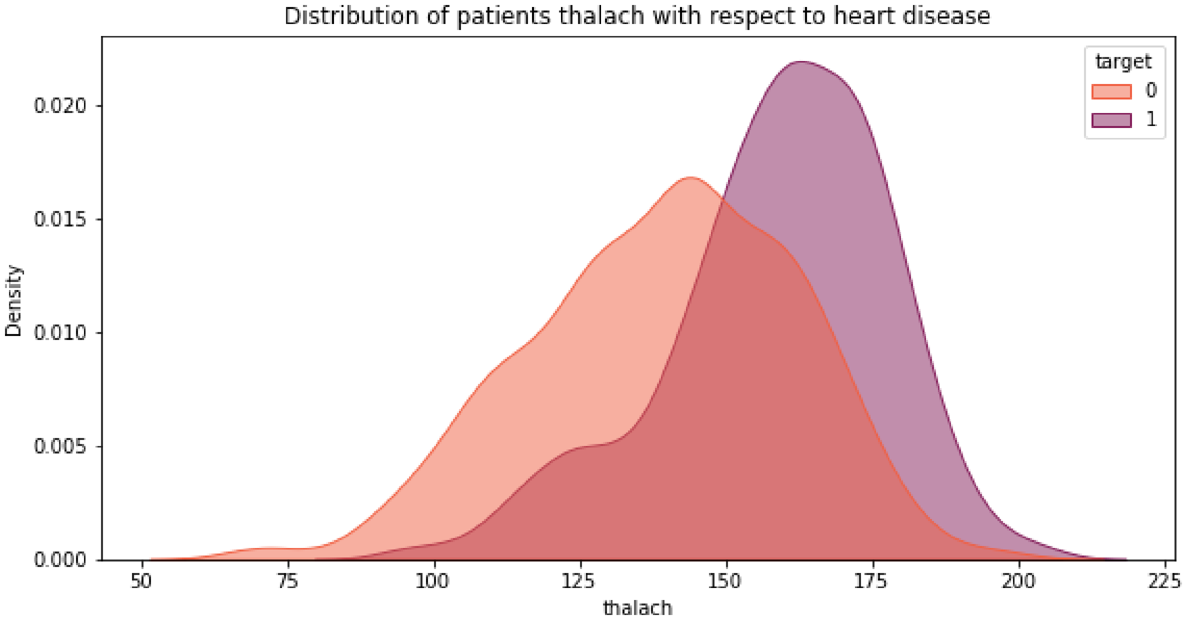


Fig:6.13: Distribution of patients thalach with respect to heart disease

```
# distribution of type of chest pain with respect of presence of heart
disease
print("Cp: Chest pain type \n 1: typical angina \n 2: atypical angina \n 3:
non-anginal pain \n 4: asymptomatic chest")

plt.figure(figsize=(11,5))
sns.kdeplot(data=df, x="cp", hue="target", fill=True, common_norm=False,
palette="viridis_r", alpha=.5)
plt.title("Distribution of patients having the type of chest pain with
respect to presence of heart disease")
plt.show()
```

Output:

```
Cp: Chest pain type
 1: typical angina
 2: atypical angina
 3: non-anginal pain
 4: asymptomatic chest
```

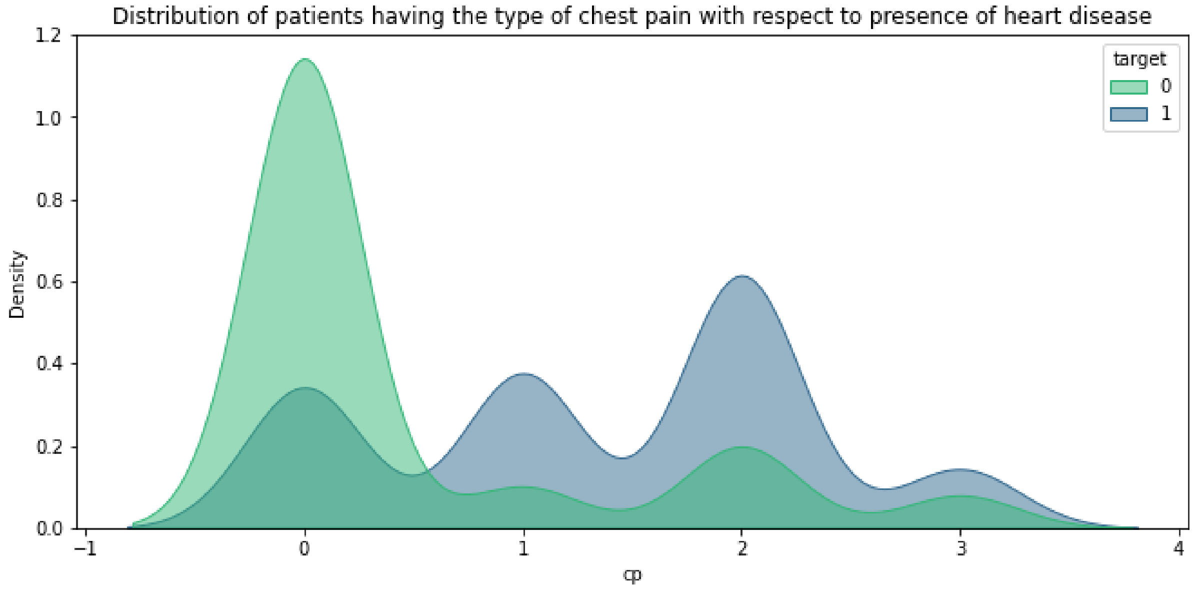


Fig:6.14: patients having the type of chest pain having of heart disease

```
# Excercise induced angina with respect to age of patient
plt.figure(figsize=(12,5))
sns.stripplot(data=df, x='exang', y='age', hue='target')
plt.title("Exercise induced angina with respect to age of patient")
plt.show()
```

Output:

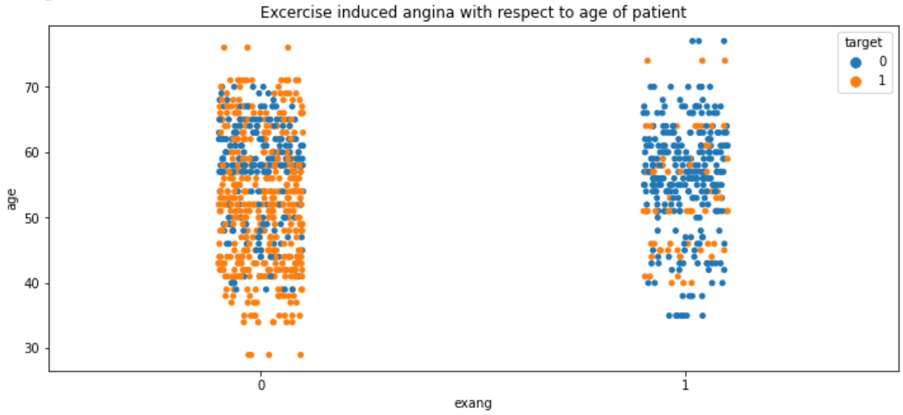


Fig:6.15: Exercise induced angina with respect to age of patient

```
# distribution of patients thal with respect to heart disease
print("thal:\n 0 = normal\n 1 = fixed defect\n 2 = reversable defect")
plt.figure(figsize=(10, 5))
sns.kdeplot(data=df, x="thal", hue="target", fill=True, common_norm=False,
palette="magma", alpha=.5)
plt.title("Distribution of patients thal with respect to heart disease")
plt.show()
```

Output:

```
thal:
 0 = normal
 1 = fixed defect
 2 = reversable defect
```

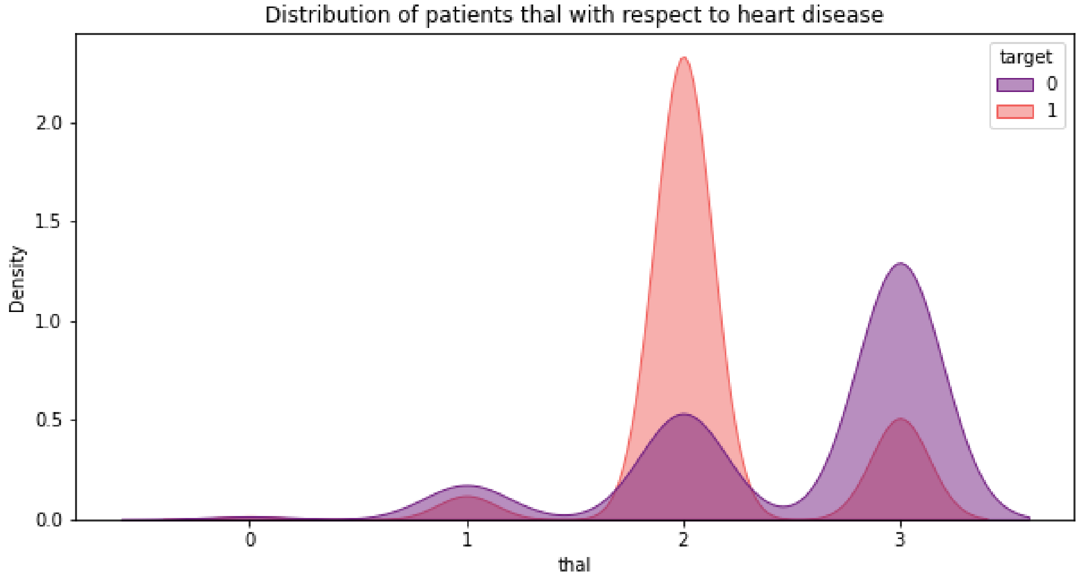


Fig:6.16: Distribution of patients thal with respect to heart disease

```
# Number of patients suffering and not suffering from heart disease with
respect to gender
plt.figure(figsize=(10,5))
sns.histplot(data=df, x='target', hue='sex')
plt.title("Number of patients suffering and not suffering from heart
disease with respect to gender")
plt.show()
```

Output:

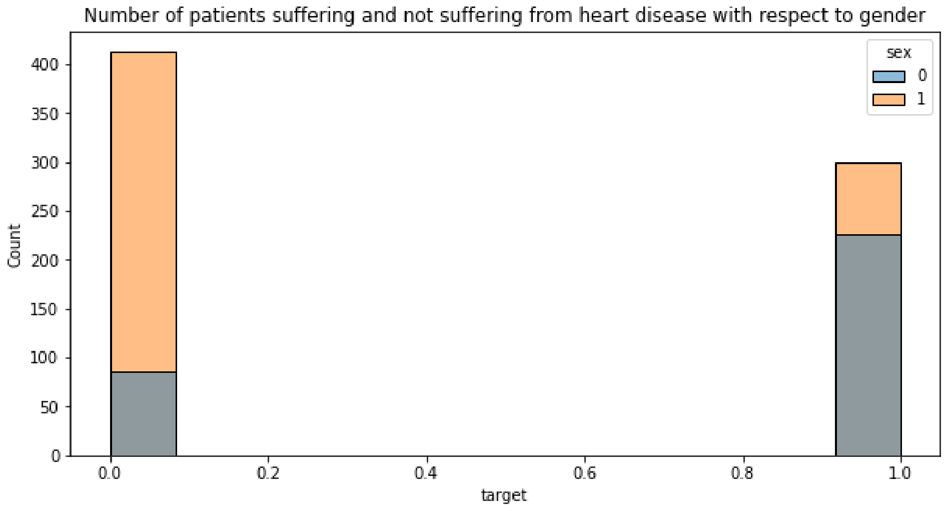


Fig:6.17: No. of patients suffering from heart disease respect to gender

```
# heat map to represent correlation between each attribute
plt.figure(figsize=(18,8))
sns.heatmap(data=df.corr(), linewidth=.1, annot=True)
plt.show()
```

Output:

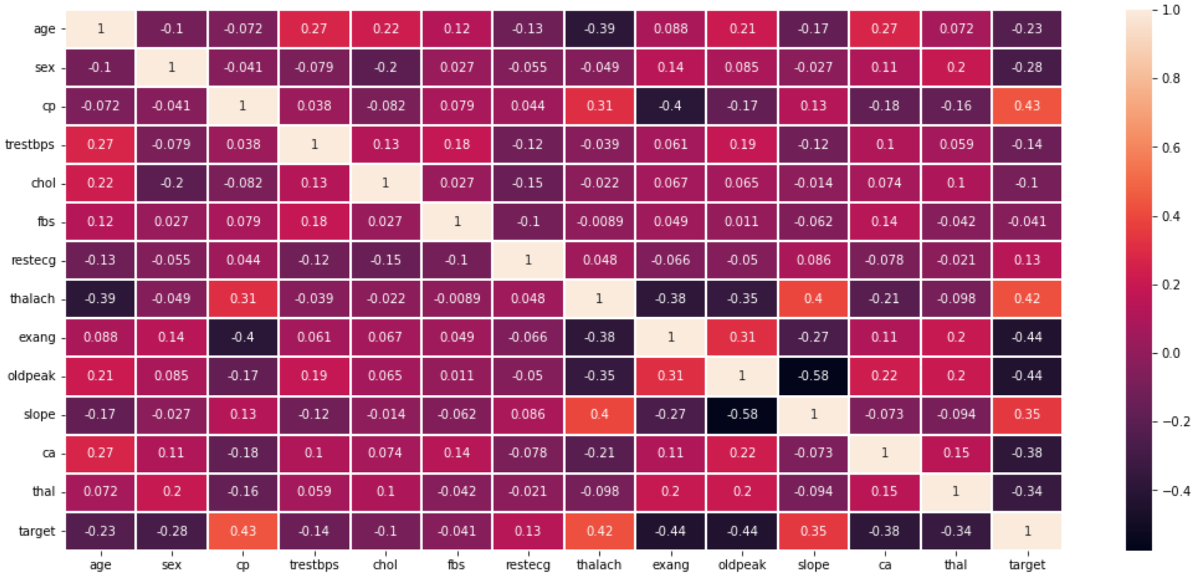


Fig:6.18: heat map to represent correlation between each attribute

Train-Test split

```
from sklearn.model_selection import train_test_split

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

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

# shape of train slip
X_train.shape
```

Output:

```
(820, 13)
```

```
# shape of test split
X_test.shape
```

Output:

```
(205, 13)
```

```
# importing multiple perforamnce parameter
from sklearn.metrics import accuracy_score
from sklearn.metrics import confusion_matrix
from sklearn.metrics import classification_report
from sklearn.metrics import confusion_matrix
from sklearn.metrics import precision_score
from sklearn.metrics import recall_score

# importing machine learning models
from sklearn.linear_model import LogisticRegression
from sklearn.naive_bayes import GaussianNB
```

```
# building logistic regression model
lr = LogisticRegression()

lr.fit(X_train, Y_train)

y_pred_lr = lr.predict(X_test)

# accuracy score
score_lr = round(accuracy_score(y_pred_lr,Y_test)*100,2)
print("The accuracy score achieved using Logistic Regression is: 
"+str(score_lr)+" %")
```

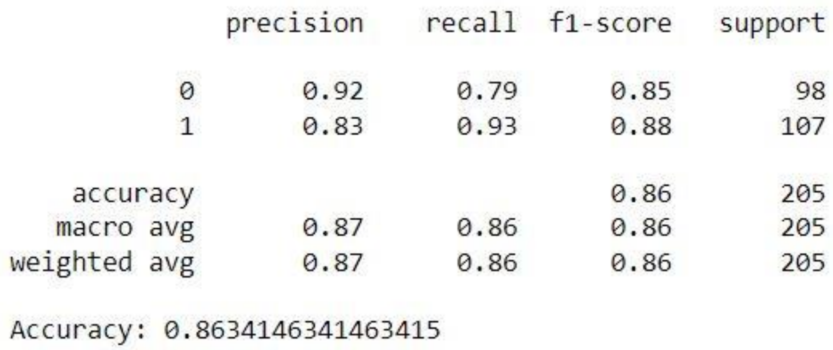
Output:

```
The accuracy score achieved using Logistic Regression is: 86.34 %
```

```
# confusion matrix
print(confusion_matrix(Y_test,y_pred_lr))
[[ 77  21]
 [  7 100]]
```

```
# classification report
print(classification_report(Y_test,y_pred_lr))
print("Accuracy:",accuracy_score(Y_test, y_pred_lr))
```

Output:



```
              precision    recall  f1-score   support

           0       0.92      0.79      0.85        98
           1       0.83      0.93      0.88       107

    accuracy                           0.86       205
   macro avg       0.87      0.86      0.86       205
weighted avg       0.87      0.86      0.86       205

Accuracy: 0.8634146341463415
```

Table: 6.02: Logistic Regression classification report

```
# precision score
precision = precision_score(Y_test, y_pred_lr)
print("Precision: ",precision)
```

Output:

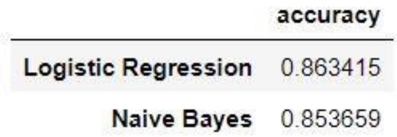
```
Precision:  0.8264462809917356
```

```
# building naive babyes classifer model
nb = GaussianNB()

nb.fit(X_train,Y_train)

y_pred_nb = nb.predict(X_test)
print(y_pred_nb)

# accuracy score
score_nb = round(accuracy_score(Y_pred_nb,Y_test)*100,2)
print("The accuracy score achieved using Naive Bayes is: "+str(score_nb)+"
%")
```

Output:

```
The accuracy score achieved using Naive Bayes is: 85.37 %
```

```
# confusion matrix
confusion_matrix(Y_test, y_pred_nb)
Output:
array([[79, 19],
       [11, 96]], dtype=int64)

# initialize an empty list
accuracy = []

# list of algorithms names
classifiers = ['Logistic Regression', 'Naive Bayes']

# list of algorithms with parameters
models = [LogisticRegression(), GaussianNB()]

# loop through algorithms and append the score into the list
for i in models:
    model = i
    model.fit(X_train, Y_train)
    score = model.score(X_test, Y_test)
    accuracy.append(score)

# create a dataframe from accuracy results
summary = pd.DataFrame({'accuracy':accuracy}, index=classifiers)
summary
Output:
```



	accuracy
Logistic Regression	0.863415
Naive Bayes	0.853659

Table:6.03: Comparison of accuracy of ml model

RESULTS AND DISCUSSION

It has been demonstrated in this work that tiny datasets are necessary for the deployment of ML classifiers, contrary to a previous research proposal. Additionally, there was a large decrease in computing time, which is important after the model has been put into use. During the research, it became clear that the dataset needed to be normalised and that overfitting might occur when the model is being trained. Evaluation of the real-world problem-based data has yielded minimal accuracy. There are numerous ways to normalise the data, and the results can be compared. There may be more ways to combine heart-disease trained ML models with multimedia for the benefit of patients and clinicians.

The optimized results have been achieved in the presented work and Logistic Regression provided best results when it came on to accuracy as 86 % and Area under the curve as 87%. Future work will be on optimizing the performance of algorithms with Newton-CG for the prediction of heart disease.

CONCLUSION

In this work, two machine learning algorithms for the prediction of cardiac disease were compared and evaluated, with encouraging results. The performance of ML techniques has improved in this work. For the 13 features in the dataset, Logistic Regression fared better in the ML technique when data pre-processing was utilised. The logistic regression achieved the greatest training and test scores, 86% and 89%, respectively. With the Naive Bayes classifier, results of 85% accuracy and an AUC value of 0.86 were obtained.

Future work on this project will add to the 76 existing features of heart disease by identifying and incorporating additional ones. In order to improve the prediction, it also plans to use additional classification techniques, such deep learning. The objective is to analyse and combine more datasets to provide a more pertinent dataset that covers a variety of population kinds. For the prediction of heart disease, the feature selection can produce more useful characteristics and productive outcomes.

BIBLIOGRAPHY

1. Zhang D, Zou L, Zhou X, *et al.*: Integrating feature selection and feature extraction methods with deep learning to predict clinical outcome of breast cancer. *IEEE Access.* 2018; **6**: 28936–28944.
2. Janosi A, Steinbrunn W, Pfisterer M, *et al.*: Heart Disease. UCI Machine Learning Repository.1988.
3. Mohan S, Thirumalai C, Srivastava G: Effective Heart Disease Prediction Using Hybrid Machine Learning Techniques. *IEEE Access.* 2019; **7**: 81542–81554.
4. Conti AA, Minelli M, Gensini GF: Global management of high-risk patients: integrated primary cardiovascular prevention in diabetics. *Int. Congr. Ser.* 2003; **207**: 10–20.
5. Beyene C, Kamat P: Survey on Prediction and Analysis the Occurrence of Heart Disease Using Data Mining Techniques. *Int. J. Pure Appl. Math.* 2018.
6. Jindal H, Agrawal S, Khera R, *et al.*: *IOP Conf. Ser.: Mater. Sci. Eng.* 2021; **1022**: 012072.
7. Liu J, Hong Y, Ralph B, *et al.*: Predictive value for the Chinese population of the Framingham CHD risk assessment tool compared with the Chinese Multi-provincial Cohort Study. *JAMA.* 2004; **291**: 2591–2599.
8. Fatima M, Pasha M: Survey of machine learning algorithms for disease diagnostic. *J. Intell. Learn. Syst. Appl.* 2017; **09**: 1–16.
9. Guyon I, Gunn S, Nikravesh M, *et al.*: *Feature Extraction: Foundations and Applications.* Cham, Switzerland: Springer;2008.
10. Ratnasari NR, Susanto A, Soesanti I, *et al.*: Thoracic X-ray features extraction using thresholding-based ROI template and PCA-based features selection for lung TB classification purposes. *Proceedings of the 2013 3rd International Conference on Instrumentation, Communications, Information Technology and Biomedical Engineering (ICICI-BME); Bandung, Indonesia. IEEE* November 2013; pp. 65–69.
11. Neha N: nandalneha/heart_disease: (heart.csv). Zenodo. Software.2022.
12. Salhi DE, Tari A, Kechadi MT:Using Machine Learning for Heart Disease Prediction.Senouci MR, Boudaren MEY, Sebbak F, *et al.*, editors. *Advances in Computing Systems and Applications. CSA 2020. Lecture Notes in Networks and Systems.* Cham.: Springer; vol. 199.
13. Tonkin AM, Lim SS, Schirmer H: Cardiovascular risk factors: when should we treat?. *Med. J. Aust.* 2003; **178**: 101–102.