VISUALISING AND PREDICTING HEART DISEASES WITH AN INTERACTIVE DASHBOARD

USING CLOUD

NALAIYA THIRAN PROJECT REPORT

2022

A Project report submittedin partial fulfilment of 7th semester in degreeof

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

PROGRESS THROUGH KNOWLEDGE

Submitted by

TEAMID: PNT2022TMID44038

ROSHINI .K -723719104066

SINTHIYA.M -723719104073

SUGANTHARANI.F -723719104079

SWETHA.K -723719104083

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

V.S.B COLLEGE OF ENGINEERING TECHNICAL CAMPUS,COIMBATORE

BONAFIDE CERTIFICATE

Certified that this project report " **VISUALISING AND PREDICTING HEART DISEASES WITH AN INTERACTIVE DASHBOARD"** is the bonafiderecord work done by **ROSHINI .K (723719104066),SINTHIYA.M (723719104073), SUGANTHARANI.F(723719104079)** AND **SWETHA.K(723719104083)** for **IBM-NALAIYATHIRAN** in **VII** semester of **B.E.,** degree course **in Computer Science and Engineering** branch during the academic year of 2022 - 2023.

STAFF-IN CHARGE	EVALUATOR
Ms.Dhrisya.S	**Mrs. Subhashree.B**
HEAD OF THE DEPARTMENT	**PRINCIPAL**
Mr. Dinesh Kumar .P	**Dr.Velmurugan.V, M.E,Ph.D**

ACKNOWLEDGEMENT

We expressour breathless thanks to our **Dr.V.Velmurugan M.E,Ph.D, principal of V.S.B College of Engineering technical campus, Coimbatore** for giving constant motivation in succeeding in our goal.

We acknowledge our sincere thanks to Head of the Department (i/c) **Mr.P. Dinesh Kumar,** for giving us valuable suggestion and help towards us throughout thisProject.

We are highly grateful to thank our Project coordinator **Ms. Dhrisya** and our Project Evaluator **Mrs.B.Subhashree** Department of ComputerScience and Engineering, V.S.B College of Engineering technical campus Coimbatore, for the coordinating us throughout this Project.

We are very much indebted to thank all the faculty members of Department of Computer science and Engineering in our Institute, for their excellent moral supportand suggestions to completeour Project work successfully.

Finally our acknowledgment does our parents, sisters and friends those who had extended their excellent support and ideas to make our Projecta pledge one.

ROSHINI .K

SINTHIYA .M

SUGANTHARANI.F

SWETHA.K

CONTENT

1.INTRODUCTION

1.1PROJECT OVERVIEW

The leading cause of death in the developed world is heart disease. Therefore, there needs to be work done to help prevent the risks of having a heart attack or stroke. Use this dataset to predict which patients are most likely to suffer from a heart disease

1.2PURPOSE

-> Know fundamental concepts and can work on IBM Cognos Analytics
-> Gain a broad understanding of plotting different visualizations to provide a suitable solution.
-> Able to create meaningful Visualizations and Dashboard(s).

2.LITERATURE SURVEY

2.1 EXISTING PROBLEM

. Heart disease generally allows to some conditions that involve narrowed or blocked blood vessels which can lead to a heart attack, stroke or chest pain. Other heart conditions, such as those that affect your heart's muscle, valves or rhythm, also are considered forms of heart disease .There are various types of cardiovascular disease. The most similar types are heart failure (HF) and Coronary Artery Disease (CAD). The main root cause of heart failure (HF) is occur due to the blockade or narrowing down of coronary arteries. Coronary arteries also supply blood to the heart. Now-a-days heart disease is one of the most significant causes of fatality. The prediction of heart disease is a critical challenge in the clinical area. But time to time, several techniques are discovered to predict the heart disease in data mining. In this survey paper, many techniques were described for predicting the heart disease.

2.2 REFERENCE

PAPER 1

Published In: International Research Journal of Engineering and Technology

Date of Conference: 07/05/2020

Print ISSN: 2395-0072

Proposed Model: Predicting the Risk of Heart Failure With EHR Sequential Data Modeling

Proposed By: Bo Jin, Chao Che et al. IEEE Accession Year: 2018 Conference Location: China

Data analysis using IBM congnos analytics and IBM cloud

Data analysis, is a process for obtaining raw data, and subsequently converting it into information useful for decision-making by users.Data, is collected and analyzed to answer questions, test hypotheses, or disprove theories.

Statistician John Tukey, defined data analysis in 1961, as:

"Procedures for analyzing data, techniques for interpreting the results of such procedures, ways of planning the gathering of data to make its analysis easier, more precise or more accurate, and all the machinery and results of (mathematical) statistics which apply to analyzing data."

There are several phases that can be distinguished, described below. The phases are iterative, in that feedback from later phases may result in additional work in earlier phases. The CRISP framework, used in data mining, has similar steps.

IBM Cognos Business Intelligence is a web-based integrated business intelligence suite by IBM. It provides a toolset for reporting, analytics, scorecarding, and monitoring of events and metrics. The software consists of several components designed to meet the different information requirements in a company. IBM Cognos has components such as IBM Cognos Framework Manager, IBM Cognos Cube Designer, IBM Cognos Transformer.

PAPER 2

Published In: International Research Journal of Engineering and Technology

Date of Conference: 07/05/2020

Print ISSN: 2395-0072

Proposed Model: Effective Heart Disease Prediction Using Hybrid Machine Learning Techniques

Proposed By: Senthilkumar Mohan,Chandrasegar Thirumalai and Gautam Srivastava

IEEE Accession Year: 2019

Conference Location: India

Data Visualization for Health care

Data visualization in the healthcare industry is no longer an option— it’s a must-have for modern medical organizations. The global market of healthcare data analytics is estimated to grow 3.5 times in just six years, from $11.5 billion in 2019 to $40.8 billion in 2025. Meanwhile, more than half of the healthcare organizations worldwide name data integration as the first technology they plan to adopt by the end of 2021.

While many factors influence the boom in data analytics and visualization tools, the most recent and obvious one is the pandemic. The COVID-19 outbreak drove the health tech adoption, which naturally increased the volumes of data available in digital format. To bring relevant information into focus, healthcare organizations implement tools for data integration and visualization.

Interactive maps, sites, or widgets allow users to choose how they interact with the data and focus on what's relevant. For example, the Institute for Health Metrics and Evaluation offers an interactive website to analyze death rates and leading death causes worldwide. There, you can switch between maps and charts or choose a specific country, age, or gender group.

Healthcare data visualization tools allow everyone to view simplified information at a glance, resulting in better understanding and higher engagement, regardless of whether your audience is stakeholders or patients

PAPER 3

Published in: 2021 Second International Conference on Electronics and Sustainable Communication Systems (ICESC)

Date of Conference: 04-06 August 2021

Date Added to IEEE Xplore: 23 September 2021

ISBN Information:

INSPEC Accession Number: 21224734

DOI: 10.1109/ICESC51422.2021.9532790

Publisher: IEEE

Conference Location: Coimbatore, India

ISBN Information:

Electronic ISBN:978-1-6654-2867-5

MACHINE LEARNING

Heart Diseases Prediction With Machine learning

Artificial Intelligence can enable the computer to think.Computer ismade much more intelligent by AI. Machine learning is the subfield of AI study. Various researchers think that without learning, cannot be developed.Machine learning (ML) is causing quite the buzz in intelligence healthcare industry as a whole . Payers to healthcare companies around the world are taking advantage of ML today. In this post, I will demonstrate a use case and show how we can harness the power of ML and apply it real world problems. We'll walk through a very simple baseline model for predicting heart disease make some predictions. from patient data, how to load the data, andDiagnosis of Diseases by Using Different Machine Learning Algorithms

Heart Disease

Coronary artery disease is detected and monitored by this proposed system. Cleveland heart data set is taken from UCI. This data set consists of 303 cases and 76 attributes/features. 13 features are used out of 76 features. Two tests with three algorithms Bayes vector machine, and Functional Trees FT are performed purpose. WEKA tool is used for detection. Net, Support for detection

CRITICAL FINDING:

The below mentioned link is to show the existing solution of predicting heart diseases

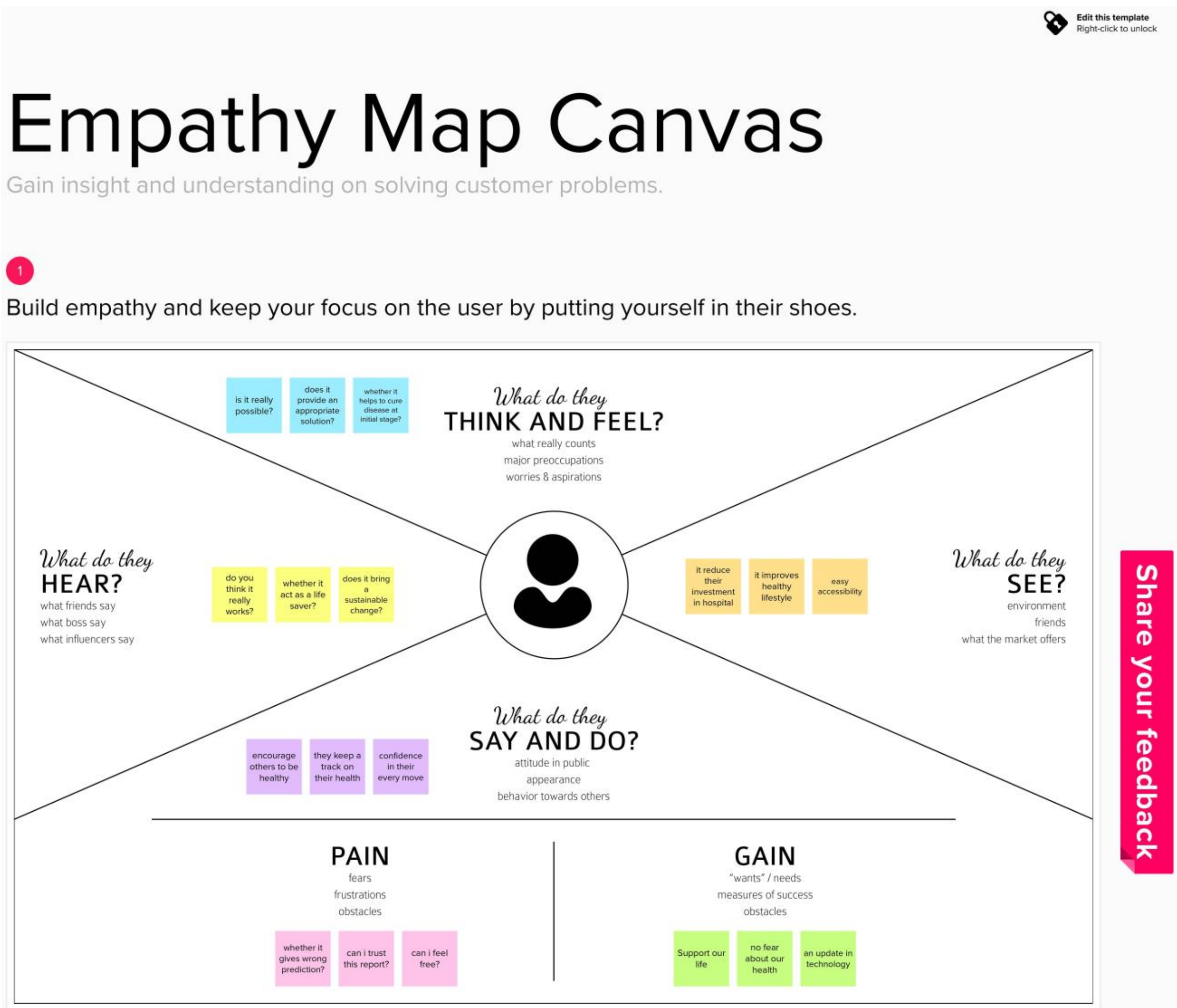
- https://www.readmyecg.co/
- https://www.fitbit.com/global/us/technology/health-metrics

2.3 PROBLEM STATEMENT

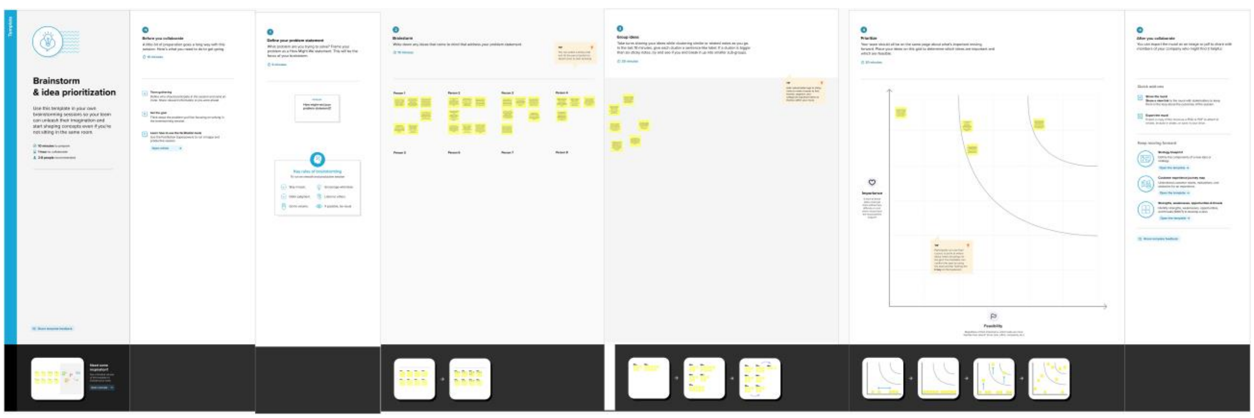
Heart related diseases or Cardiovascular Diseases (CVDs) are the main reason for a huge number of deaths in the world over the last few decades and has emerged as the most life-threatening disease, not only in India but in the whole world. Many researchers, in recent times, have been using several machine learning techniques to help the health care industry and the professionals in the diagnosis of heart related diseases. This indicates a need of reliable, accurate and feasible system to continuously monitor and diagnose for CVD for timely action and treatment. This work proposes a smartphone-based heart disease prediction system than can have both monitoring as well as prediction of heart disease. A system to monitor patients in real-time has been developed using Node MCU interfaced with temperature, humidity and pulse rate sensors. The developed system is capable to transmit the acquired sensor data to a cloud(firebase) every 10 seconds. An Android application is designed to display the sensor data. One best machine learning algorithm was ported to the Android application for heart disease prediction in real-time. The machine learning algorithms were trained and tested using two widely used open-access datasets. Five machine learning algorithms were checked for their performances using two different methods. ANN was found to be the best performing algorithm with an accuracy of 93.5%. This algorithm is deployed to the Android application and the heart disease is predicted in real-time. The proposed work is limited by use of single hidden layer for implementing Neural network. Data from few more sensors related to heart parameters should be experimented with. Trying out with increasing hidden layer size may increase the accuracy of the neural network. There is further scope in optimizing the Android application user interface.

3. IDEATION & PROPOSED SOLUTION

3.1 EMPATHY MAP CANVAS



3.2 IDEATION AND BRAINSTORMING



3.3 PROPOSED SOLUTION

NOVELTY:

The result of the data analysis to identify the necessary hidden patterns for predicting heart diseases are presented in this section. Here the variables considered to predict the heart disease are age, chest pain type, blood pressure, blood glucose level, ECG in rest, heart rate and four types of chest pain and exercise angina. The heart disease dataset is effectively pre-processed by eliminating unrelated records and given values to missing tuples. The pre-processed heart disease data set [10] is then composed by K-means algorithm. Here, four types of heart diseases are discussed namely asymptomatic pain, atypical angina pain, non-anginal pain and non-anginal pain. The results are computed using all the four types of chest pain with other deciding variables.

FEASIBILITY OF IDEA:

Healthcare industries generate enormous amount of data, so called big data that accommodates hidden knowledge or pattern for decision making. The huge volume of data is used to make decision which is more accurate than intuition. Exploratory Data Analysis (EDA) detects mistakes, finds appropriate data, checks assumptions and determines the correlation among the explanatory variables. In the context, EDA is considered as analysing data that excludes inferences and statistical modelling. Analytics is an essential technique for any profession as it forecast the future and hidden pattern. Data analytics is considered as a cost effective technology in the recent past and it plays an essential role in healthcare which includes new research findings, emergency situations and outbreaks of disease. The use of analytics in healthcare improves care by facilitating preventive care and EDA is a vital step while analysing data. In this paper, the risk factors that causes heart disease is considered and predicted using K-means algorithm and the analysis is carried out using a publicly available data for heart disease. The dataset holds 209 records with 8 attributes such as age, chest pain type, blood pressure, blood glucose level, ECG in rest, heart rate and four types of chest pain. To predict the heart disease, K-means clustering algorithm is used along with data analytics and visualization tool. The paper discusses the pre-processing methods, classifier performances and evaluation metrics. In the result section, the visualized data shows that the prediction is accurate.

BUSINESS MODEL:

Predicting and diagnosing heart disease is the biggest challenge in the medical industry and it is based on factors like physical examination, symptoms and signs of the patient . Factors which influence heart diseases are cholesterol level of the body, smoking habit, and obesity, family history of diseases, blood pressure and working environment. Machine learning algorithms play a vital and accurate role in predicting heart disease . The advancement of technologies allows machine language to pair with big data tools to handle unstructured and exponentially growing data . In the paper, K means clustering method is proposed in big data environment and the visualization is made with the tableau dashboard.

SOCIAL IMPACT:

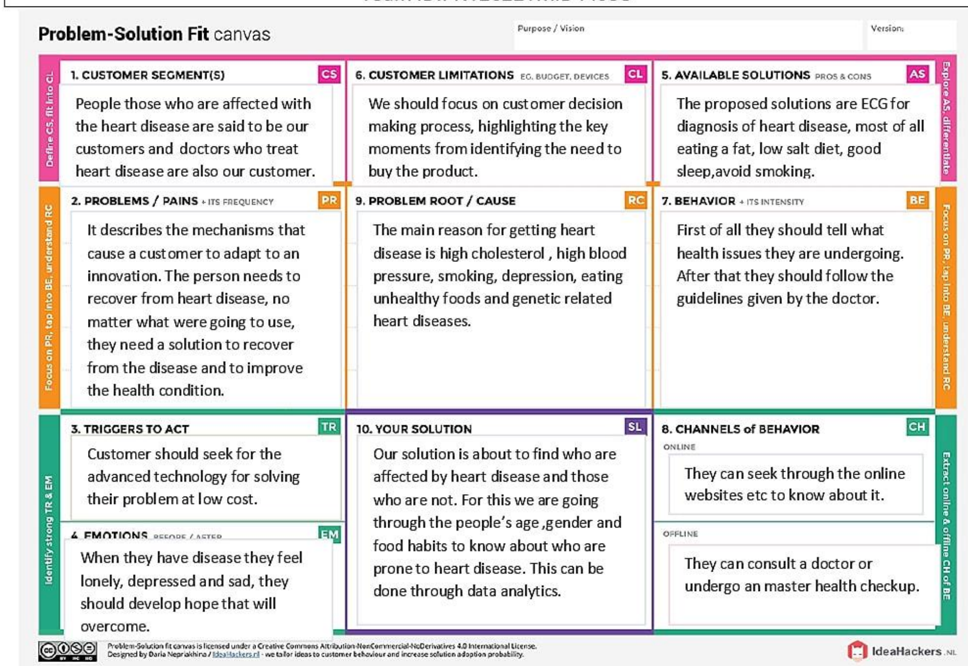
Recent advances in molecular genetics are making it increasingly feasible to construct individual genetic profiles predicting susceptibility to heart disease, cancer and respiratory disorders. This paper reviews current knowledge about the social and cultural impact of providing people with information relating to their risk for future disease, focusing not only on currently available genetic testing but also on hypertension, hyperlipidaemia and cancer screening. We highlight the importance of issues of probability and uncertainty, and the tension between collective and individual goals in the assessment of medical risk. We conclude with a proposed research agenda for studies of the social and cultural impact of predictive genetic testing, and argue that there is a pressing need for rigorous, empirical, social research in this area.

SCALABILITY OF THE SOLUTION:

As wearable medical sensors continuously generate enormous data, it is difficult to process and analyse. This paper focuses on developing scalable sensor data processing architecture in cloud computing to store and process body sensor data for healthcare applications. Proposed architecture uses big data technologies such as Apache Flume, Apache Pig and Apache HBase to collect and store huge sensor data in the Amazon web service. Apache Mahout implementation of MapReduce-based online stochastic gradient descent algorithm is used in the logistic regression to develop the scalable diagnosis model. Cleveland heart disease database (CHDD) is used to train the logistic regression model. Wearable body sensors are used

3.4 PROBLEM SOLUTON FIT

Team ID:PNT2022TMID44038



4. REQUIREMENT ANALYSIS

Solution Requirements (Functional & Non-functional)

4.1 FUNCTIONAL REQUIREMENTS:

Following are the functional requirements of the proposedsolution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through Form Registration through Gmail Registration throughLinkedIN
FR-2	User Confirmation	Confirmation viaEmail Confirmation via OTP
FR-3	User verification	Verification through CAPTCHAVerification through I'm not a robot
FR-4	User Authentication	Recognition of correct person Resending the code in case of forgot password
FR-5	User validation	Reconfirming the new password Sending a two digit number in (Google account) your Old devices, so thatyou can enterinto a new device By entering the two digit number
FR-6	User Submission	Submission through Google form Submission through Email.

4.2 NON-FUNCTIONAL REQUIREMENTS:

Following are the non-functional requirements of the proposed solution.

FR No.	Non-Functional Requirement	Description

NFR-1	**Usability**	The EHDPS predicts the likelihood of patients gettingheart disease. It enables significant knowledge, eg, relationships between medical factors related to heart disease and patterns, to be established.
NFR-2	**Security**	When it deals with(comes to)health factors, we should provide more security services. There shouldn’t be no errors, lagging , base of data of apatient profile, whileworking on the software or product.
NFR-3	**Reliability**	Reliability is said to be the measure of stability orconsistency of test scores shown in your product. Therefore your product will normal as a goodperformance one in the field of accuracy

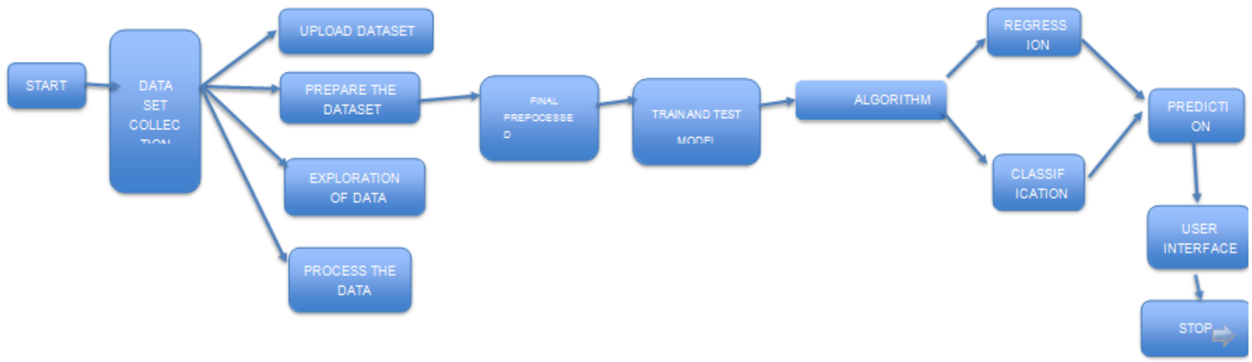
5. PROJECT DESIGN

5.1 DATA FLOW DIAGRAMS

Data Flow Diagrams:

A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict therightamount of the system requirement graphically. It shows how data enters and leaves the system, what changes the information, and where datais stored.

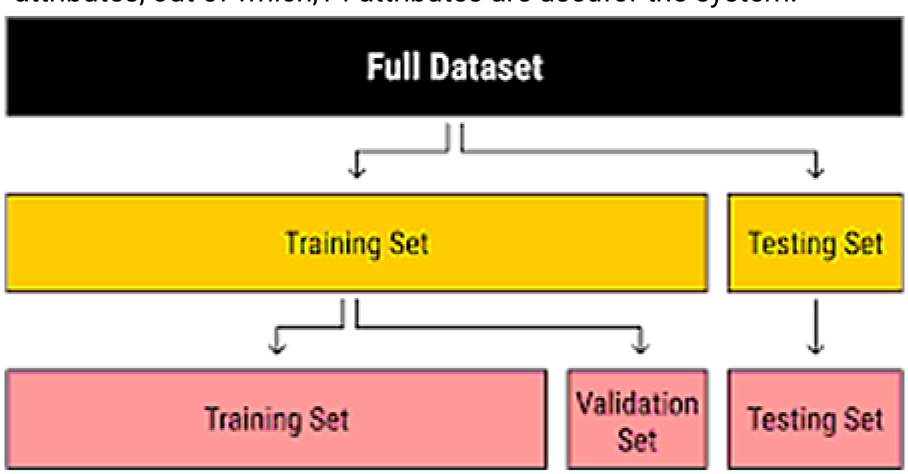
Data Flow Diagram:



5.2 SOLUTION AND TECHNICAL ARCHITECTURE

Collection of dataset:

Initially, we collect a dataset for our heart disease prediction system. After the collection of thedataset, we split the datasetinto training data and testingdata. The trainingdataset is used for prediction model learning and testing data is used for evaluating the prediction model. For this project, 70% of training data is used and 30% of data is used for testing. The dataset used for this project is Heart Disease UCI. The dataset consistsof 76 attributes; out of which,14 attributes are usedfor the system.



SELECTION OF ATTRIBUTES:

Attribute or Feature selection includes the selection of appropriate attributes for the predictionsystem. This is used to increase the efficiency of the system. Various attributes of the patient like gender, chest pain type, fasting blood pressure, serum cholesterol, exang, etc are selected for the prediction. The Correlation matrix is used for attribute selection for this model.

Figure: Correlation matrix

PRE-PROCESSING OF DATA:

Data pre-processing is an importantstep for the creation of a machinelearning model. Initially, data may not be clean or in the required format for the model which can cause misleading outcomes. In pre-processing of data, we transform data into our required format. It is used to deal with noises,duplicates, and missingvalues of the dataset. Data pre-processing has the activities like importing datasets,splitting datasets, attributescaling, etc. Preprocessing of data is required for improving the accuracy of the model.

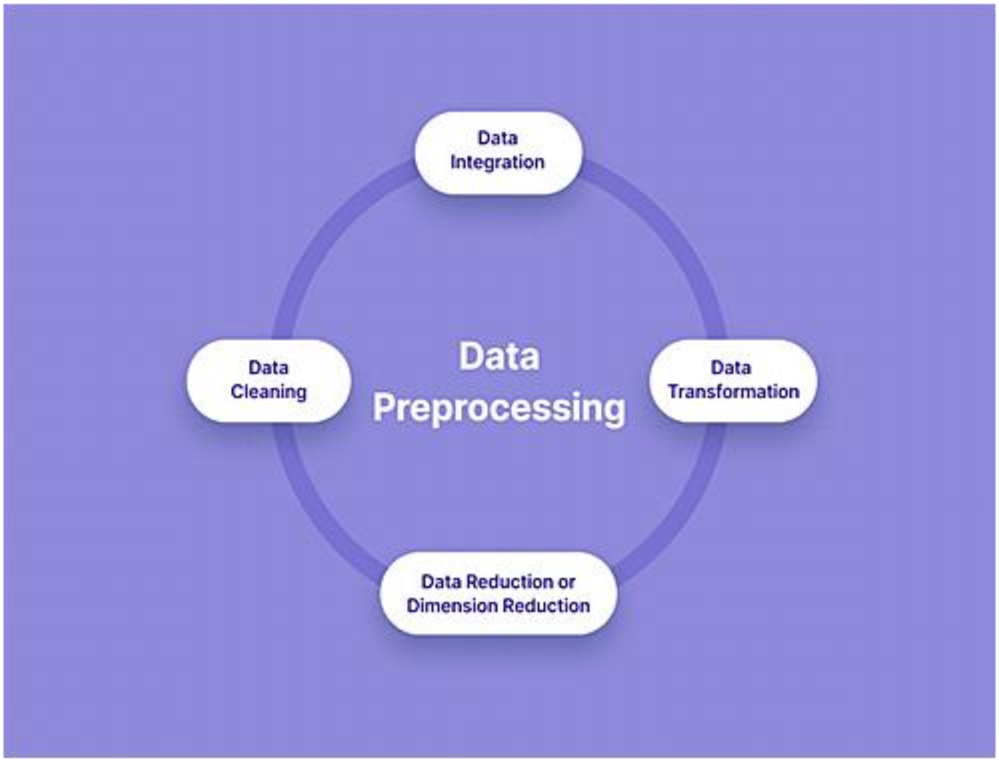


Figure: Data pre-processing

BALANCING OF DATA:

Imbalanced datasetscan be balanced in two ways. They are Under Sampling and Over Sampling

1. **Under Sampling**: In Under Sampling, dataset balance is done by the reductionof the size of theample class. This process is considered when the amount of data is adequate.

2. **Over Sampling**: In Over Sampling, dataset balance is done by increasing the size of the scarcesamples. This process is considered when the amount of data is inadequate.

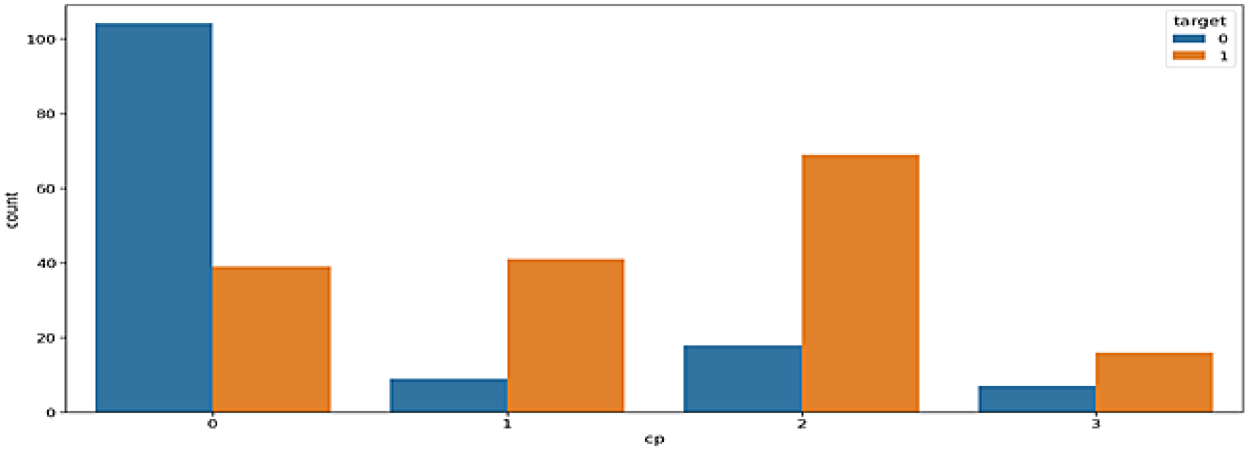


Figure: Data Balancing

PREDICTION OF DISEASE:

Various machine learningalgorithms like SVM, Naive Bayes, Decision Tree,Random Tree,Logistic Regression, Ada-boost, Xg-boostare used for classification. Comparative analysis is performed
among algorithms and the algorithm that gives the highest accuracyis used for heart disease prediction.

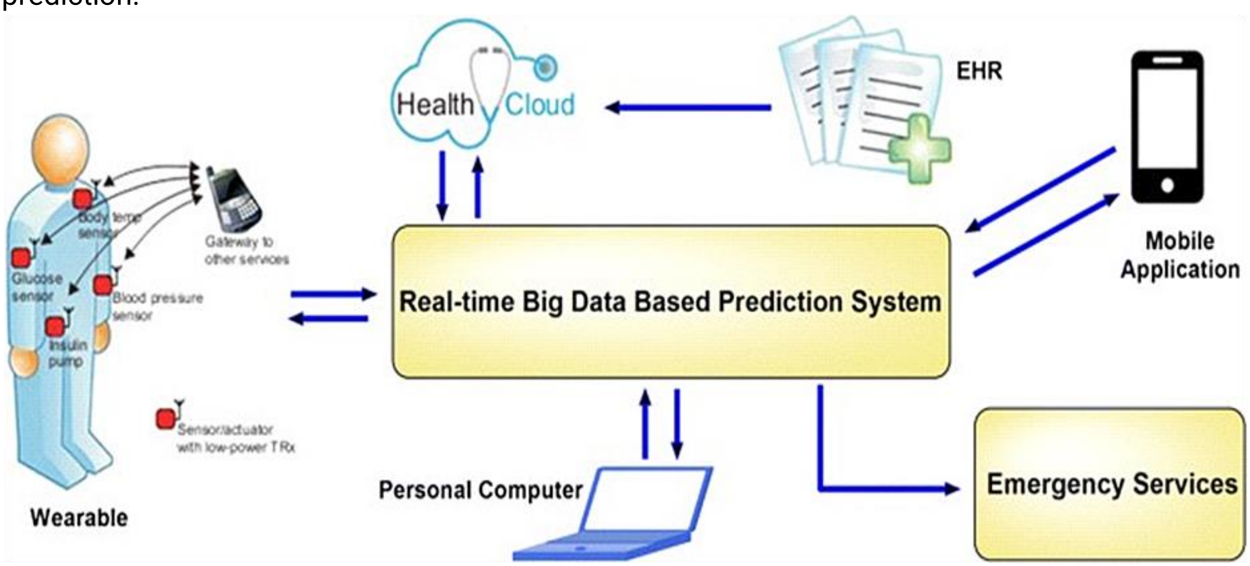


Figure: Prediction of Disease

Architecture Diagram:

Methodology of heart disease prediction using ensemble of machine learning algorithms

5.3 USER STORIES

User Type	Functional Requirement(Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Web user)	Registration	USN-1	As a user, I can register for the web page by entering my email, password, and confirming my password.	I can access my account / dashboard	High	Sprint-1
	Authentication	USN-2	As a user, I will receive confirmation email once I have registered for the application	I can receive confirmation email& click confirm	High	Sprint-1

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
		USN-3	As a user, I can register for the application through Gmail	I can login using the logincredentials	Medium	Sprint-1
	Login	USN-4	As a user,I can log into the applicationby entering email& password	I can check whether the login credentialsare correct	High	Sprint-1
	Dashboard	USN-5	If entering the registered email id, Ican access the dashboard	I will be able to view the dashboard of the user	High	Sprint-1
Customer Care Executive	Support	USN-6	If the I facesany issues,then I can reportit to their email address	Report and feedback optionwill be accessible	High	Sprint-2
	Accessing dashboard	USN-7	The user uses his/her personal emailid to accessthe dashboard	There is very less chance for other userstoaccess my details	High	Sprint-2
Administrator	Validation	USN-8	The administrator will be able to login with their login ids	He/she will be able to validate the user details	High	Sprint-3

6. PROJECT PLANNING & SCHEDULING

6.1 SPRINT PLANNING & ESTIMATION

Sprint	Functional Requirement (Epic)	User Story Number	User Story/ Task	Story Points	Priority	Team Members
Sprint-1	Datasets	USN-1	As a user,I can gather the details of thepatients.	2	High	2
Sprint-1		USN-2	As an Analyst, I will check the data set and clean the datasetto create an efficient model.	3	High	2
Sprint-1		USN-3	As an Analyst I will also correct the raw data andcreate a datamodule.	5	High	2
Sprint-2	Cleaning, exploring data and creating model	USN-4	As an AnalystI can create an Exploratory data analysis to identify the important factorsofpatient dataset	5	High	2
Sprint-2		USN-5	As a Dataanalyst, I create a predicted model by also preparing story card with usingexplored data	5	High	2
Sprint-3	Data Prediction	USN-6	As a Dataanalyst, I willcreate different typesof models in explored data to identify	5	Medium	1

Sprint	Functional Requirement (Epic)	User Story Number	User Story/ Task	Story Points	Priority	Team Members

			suitable model with effectively and efficiently.			
Sprint-3		USN-7	As a Data Analyst, I will analysis of the heartdisease patient's datasets.	5	High	1
Sprint-4	Creation of deployeddata UI	USN-8	As a Data analyst, I will create a heart disease prediction iterative dashboard.	5	High	2
Sprint-4		USN-9	As an Analyst, I will import my analysedmodel into suitable framework.	5	High	2

6.2 SPRINT DELIVERY SCHEDULE

Sprint	Total Story Points	Duration	Sprint StartDate	Sprint End Date (Planned)	Story PointsCompleted (as on PlannedEnd Date)	Sprint ReleaseDate (Actual)
Sprint-1	10	5 Days	24 Oct 2022	29 Oct 2022	10	29 Oct 2022
Sprint-2	10	5 Days	31 Oct 2022	05 Nov 2022	10	05 Nov 2022
Sprint-3	10	5 Days	07 Nov2022	12 Nov2022	10	12 Nov2022
Sprint-4	10	5 Days	14 Nov2022	19 Nov2022	10	19 Nov2022

Velocity:

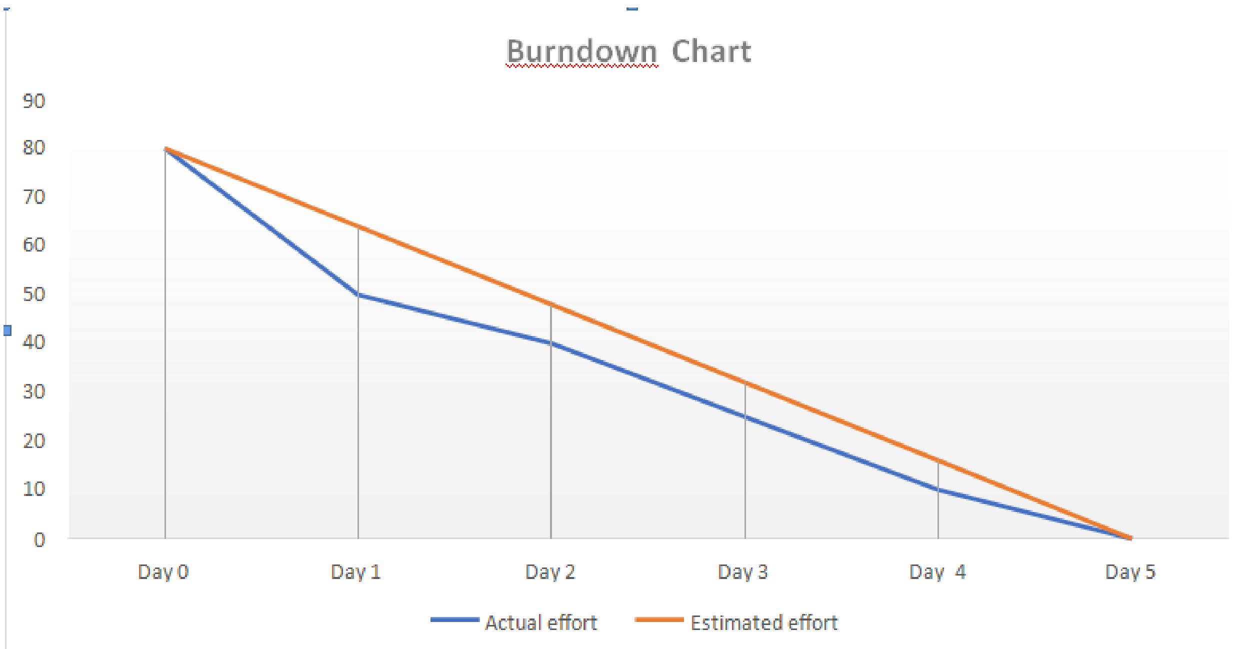
Imagine we have a 05-day sprint duration, and the velocity of the team is 10 (points per sprint). Let's calculate the team's average velocity (AV)per iteration unit (story points per day)

AV=Sprint Duration/Velocity=10/5=2

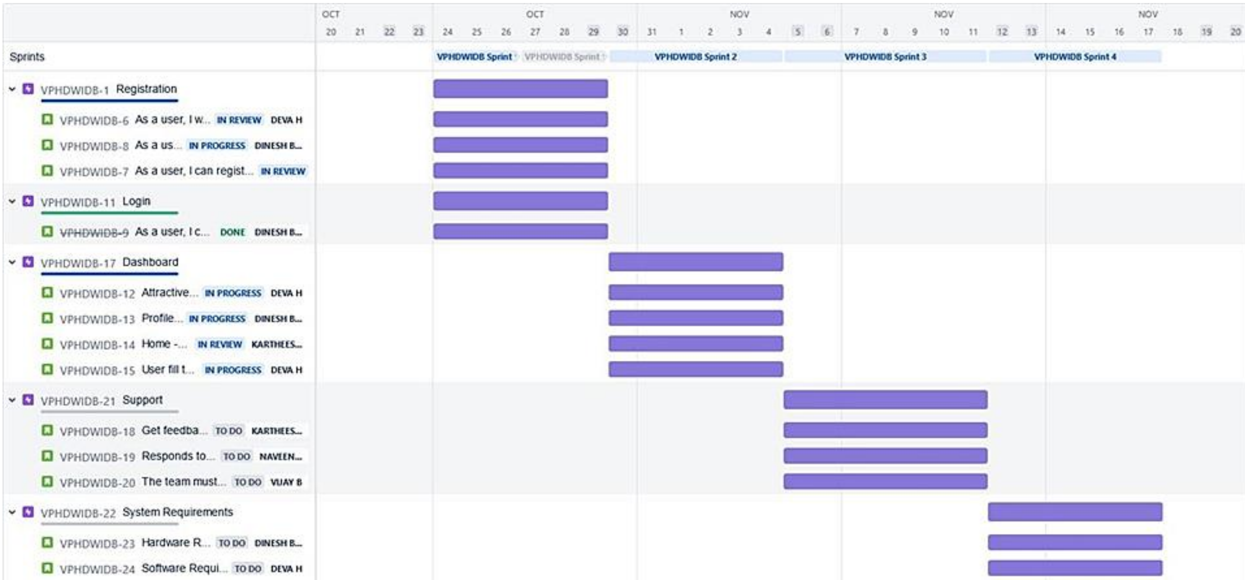
Burndown Chart:

A burn down chart is a graphical representation of work left to do versus time. It is often used in agile software development methodologiessuch as Scrum. However, burn down chartscan be applied to any project containing measurable progress over time.

Goal:60 hours in 5 days



<u>6.3 REPORTS FROM JIRA</u>



7. CODING & SOLUTIONING

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

In [3]:

```
df = pd.read_csv('Heart_Disease_Prediction.csv')
df.head()
```

Out[3]:

	Age	Sex	Chest pain type	BP	Cholesterol	FBS over 120	EKG results	Max HR	Exercise angina	ST depression	Slope of ST	Number of vessels fluro	Thallium	Heart Disease
0	70	1	4	130	322	0	2	109	0	2.4	2	3	3	Presence
1	67	0	3	115	564	0	2	160	0	1.6	2	0	7	Absence
2	57	1	2	124	261	0	0	141	0	0.3	1	0	7	Presence
3	64	1	4	128	263	0	0	105	1	0.2	2	1	7	Absence
4	74	0	2	120	269	0	2	121	1	0.2	1	1	3	Absence

In [4]:

df.describe()

Out[4]:

	Age	Sex	Chest pain type	BP	Cholesterol	FBS over 120	EKG results	Max HR	Exercise angina	d
count	270.000000	270.000000	270.000000	270.000000	270.000000	270.000000	270.000000	270.000000	270.000000	2
mean	54.433333	0.677778	3.174074	131.344444	249.659259	0.148148	1.022222	149.677778	0.329630	
std	9.109067	0.468195	0.950090	17.861608	51.686237	0.355906	0.997891	23.165717	0.470952	
min	29.000000	0.000000	1.000000	94.000000	126.000000	0.000000	0.000000	71.000000	0.000000	
25%	48.000000	0.000000	3.000000	120.000000	213.000000	0.000000	0.000000	133.000000	0.000000	
50%	55.000000	1.000000	3.000000	130.000000	245.000000	0.000000	2.000000	153.500000	0.000000	
75%	61.000000	1.000000	4.000000	140.000000	280.000000	0.000000	2.000000	166.000000	1.000000	
max	77.000000	1.000000	4.000000	200.000000	564.000000	1.000000	2.000000	202.000000	1.000000	

In [5]:

```
df.info()
RangeIndex: 270 entries, 0 to 269
Data columns (total 14 columns):
 #   Column                   Non-Null Count  Dtype  
---  ------                   --------------  -----  
 0   Age                      270 non-null    int64  
 1   Sex                      270 non-null    int64  
 2   Chest pain type          270 non-null    int64  
 3   BP                       270 non-null    int64  
 4   Cholesterol              270 non-null    int64  
 5   FBS over 120             270 non-null    int64  
 6   EKG results              270 non-null    int64  
 7   Max HR                   270 non-null    int64  
 8   Exercise angina          270 non-null    int64  
 9   ST depression            270 non-null    float64
 10  Slope of ST              270 non-null    int64  
 11  Number of vessels fluro  270 non-null    int64  
 12  Thallium                 270 non-null    int64  
 13  Heart Disease            270 non-null    object 
dtypes: float64(1), int64(12), object(1)
memory usage: 29.7+ KB
```

In [6]:

```
df.columns.values
```

Out[6]:

```
array(['Age', 'Sex', 'Chest pain type', 'BP', 'Cholesterol',
       'FBS over 120', 'EKG results', 'Max HR', 'Exercise angina',
       'ST depression', 'Slope of ST', 'Number of vessels fluro',
       'Thallium', 'Heart Disease'], dtype=object)
```

In [7]:

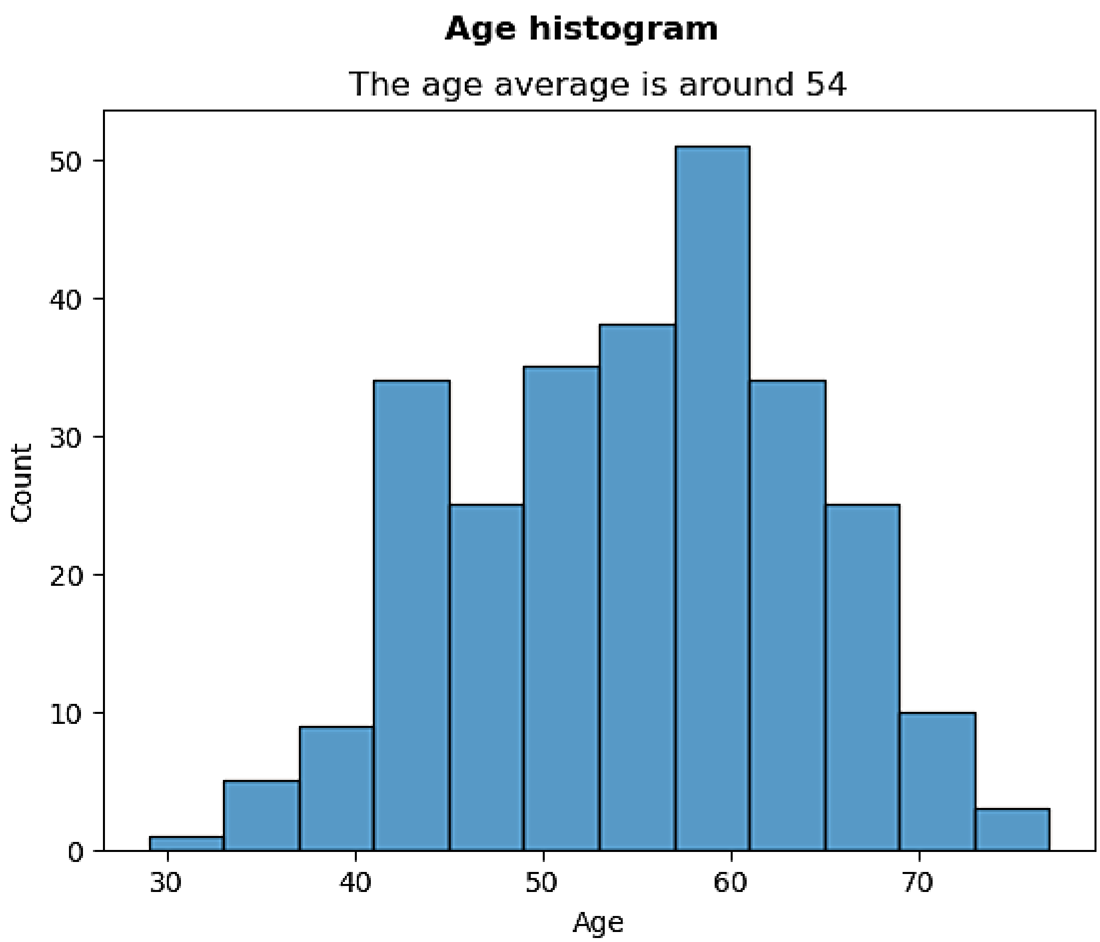
```
df.nunique()
```

Out[7]:

```
Age                         41
Sex                          2
Chest pain type              4
BP                          47
Cholesterol                144
FBS over 120                 2
EKG results                  3
Max HR                      90
Exercise angina              2
ST depression               39
Slope of ST                  3
Number of vessels fluro      4
Thallium                     3
Heart Disease                2
dtype: int64
```

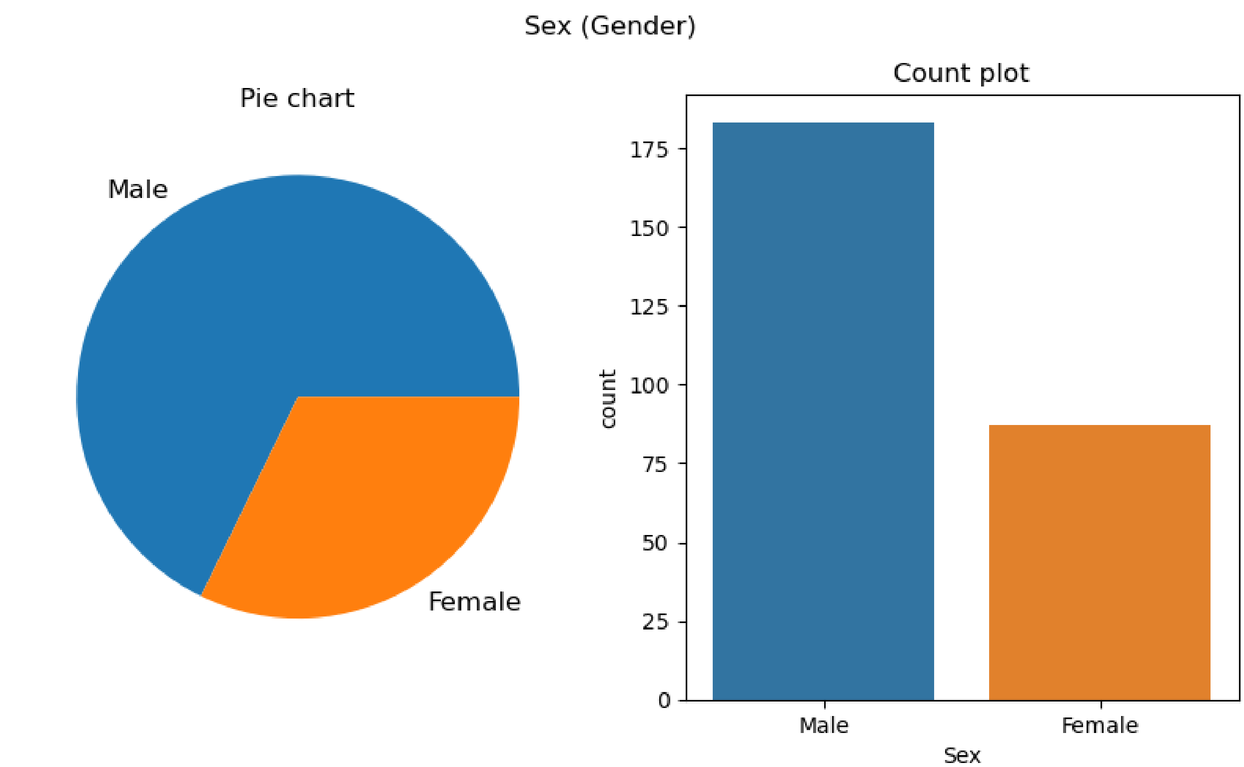
In [8]:

```
plt.suptitle('Age histogram', fontweight='heavy')
plt.title('The age average is around 54')
sns.histplot(data=df, x='Age')
plt.show()
```



In [9]:

```
labels = ['Male', 'Female']
order = df['Sex'].value_counts().index

plt.figure(figsize=(10,5))
plt.suptitle("Sex (Gender)")

plt.subplot(1,2,1)
plt.title('Pie chart')
plt.pie(df['Sex'].value_counts(), labels=labels, textprops={'fontsize':12})

plt.subplot(1,2,2)
plt.title('Count plot')
sns.countplot(x='Sex', data=df, order=order)
plt.xticks([0, 1], labels)

plt.show()

print(df['Sex'].value_counts())
print("It can be noticed that predictor (Gender) is imbalance")
```

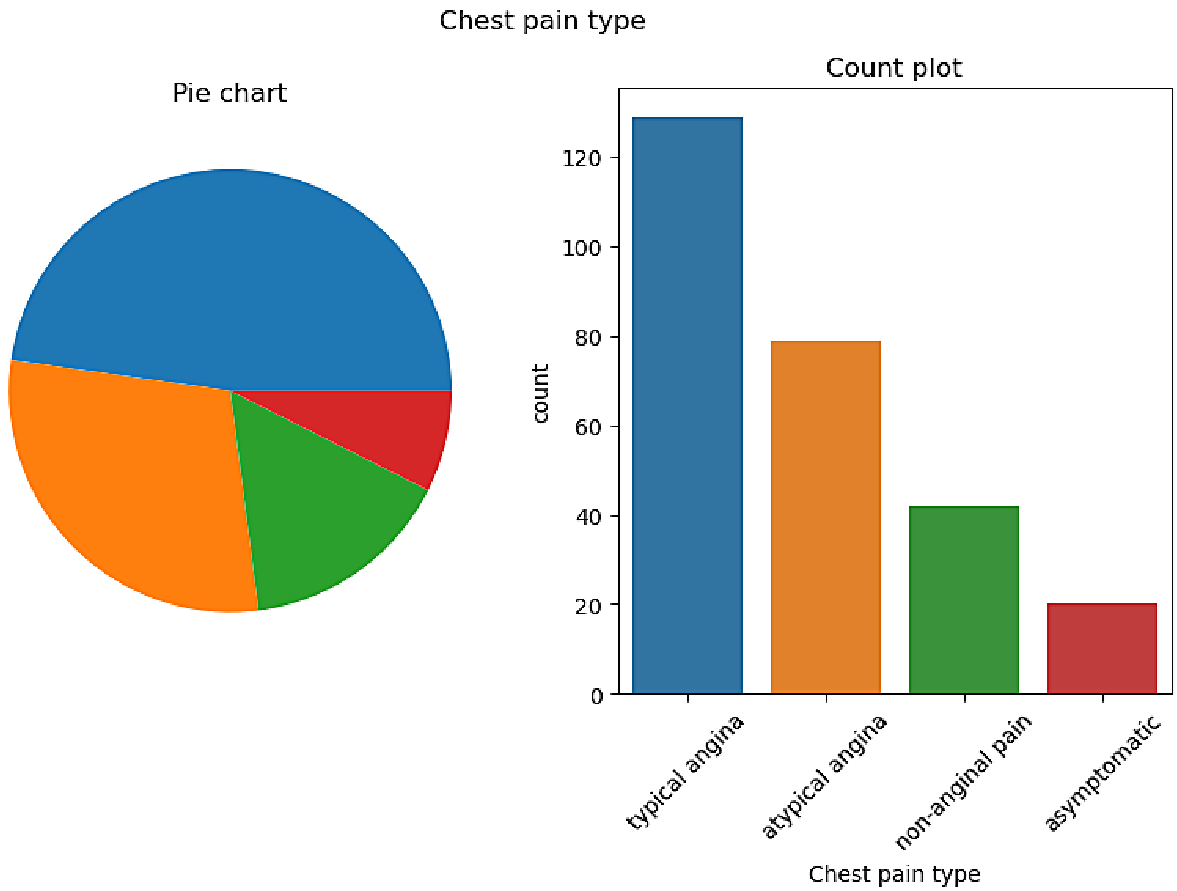



```
1    183
0     87
Name: Sex, dtype: int64
```

It can be noticed that predictor (Gender) is imbalance

In [10]:

```
labels = ["typical angina", "atypical angina", "non-anginal pain", "asymptomatic"]
order = df['Chest pain type'].value_counts().index

plt.figure(figsize=(10,5))
plt.suptitle("Chest pain type")

plt.subplot(1,2,1)
plt.title('Pie chart')
plt.pie(df['Chest pain type'].value_counts(), textprops={'fontsize':12})
plt.subplots_adjust(left=0.125)

plt.subplot(1,2,2)
plt.title('Count plot')
sns.countplot(x='Chest pain type', data=df, order=order)
plt.xticks([0,1,2,3], labels, rotation=45)

plt.show()

df['Chest pain type'].value_counts()
```

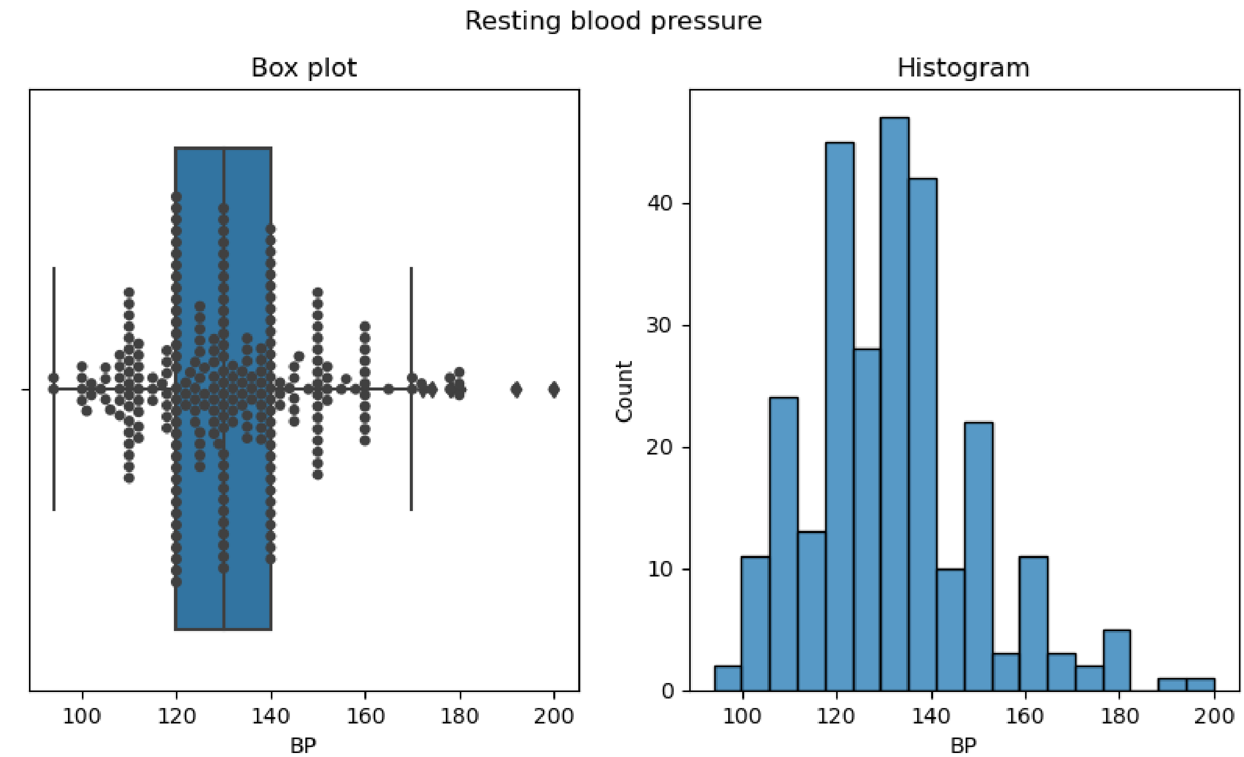


Out[10]:

```
4    129
3     79
2     42
1     20
Name: Chest pain type, dtype: int64
```

In [11]:

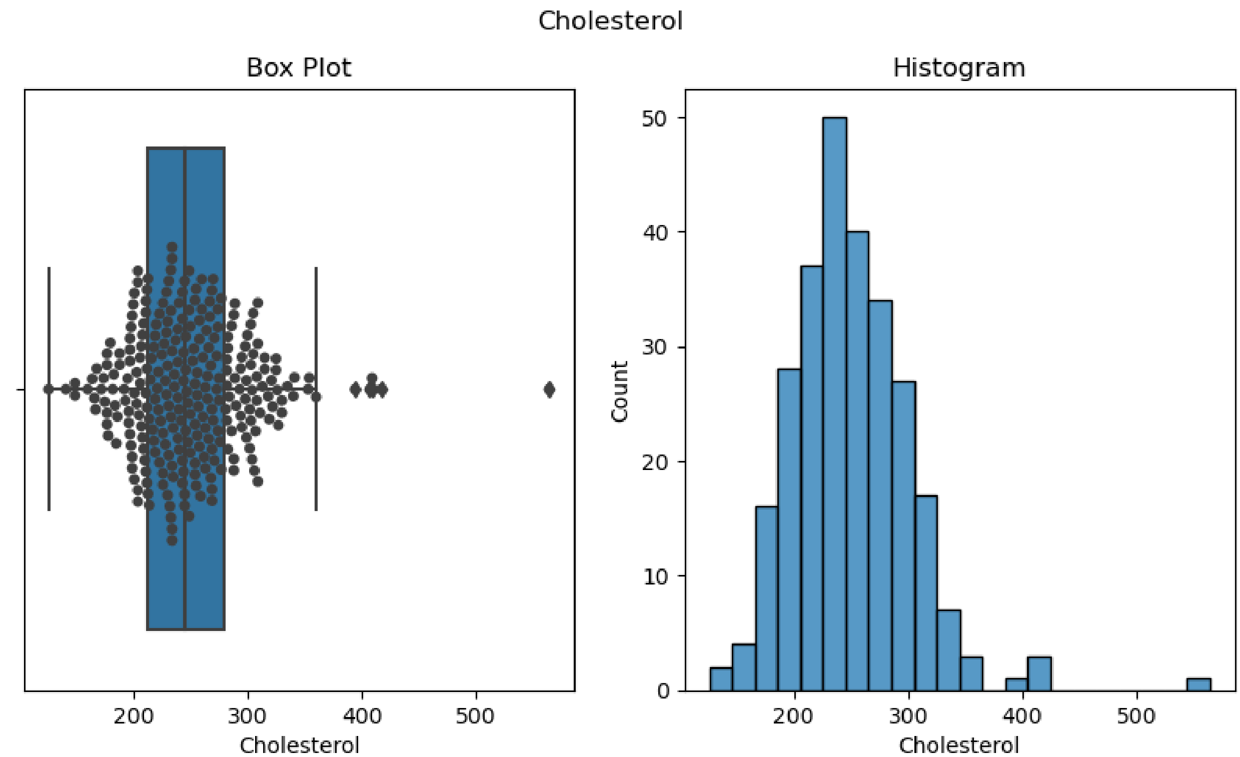
```
plt.figure(figsize=(10,5))
plt.suptitle("Resting blood pressure")

plt.subplot(1,2,1)
plt.title('Box plot')
sns.boxplot(x="BP", data=df)
sns.swarmplot(x="BP", data=df, color=".25")

plt.subplot(1,2,2)
plt.title('Histogram')
sns.histplot(x='BP', data=df)
plt.show()

print("The average resting heart rate: %2.2f It can be observed that histogram is skewed to right side"
% (df["BP"].mean()))
```



The average resting heart rate: 131.34 It can be observed that histogram is skewed to right side

In [12]:

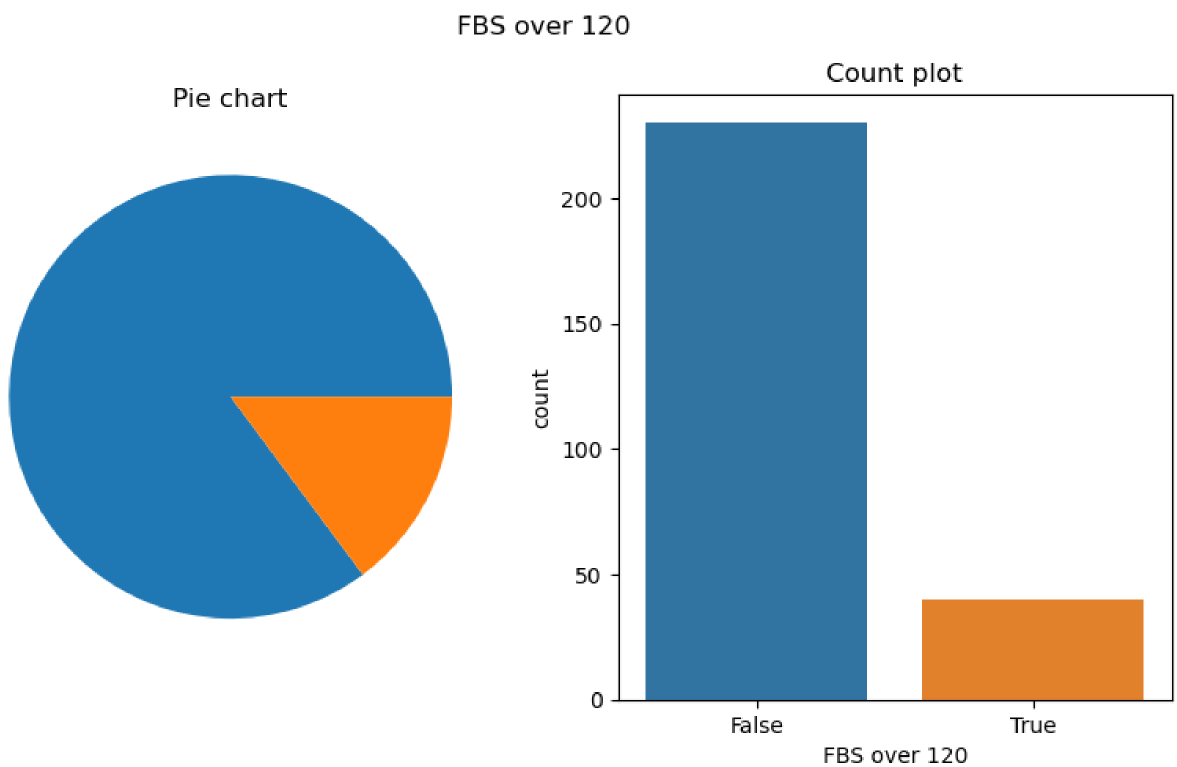
```
plt.figure(figsize=(10,5))
plt.suptitle("Cholesterol")

plt.subplot(1,2,1)
plt.title('Box Plot')
sns.boxplot(x="Cholesterol", data=df)
sns.swarmplot(x="Cholesterol", data=df, color=".25")

plt.subplot(1,2,2)
plt.title('Histogram')
sns.histplot(x='Cholesterol', data=df)
plt.show()

print("The average resting heart rate: %2.2f. The shape of histogram resamble a normal distribution" %
(df["Cholesterol"].mean()))
```



The average resting heart rate: 249.66. The shape of histogram resamble a normal distribution

In [13]:

```
labels = ["False", 'True']
order = df['FBS over 120'].value_counts().index

plt.figure(figsize=(10,5))
plt.suptitle("FBS over 120")

plt.subplot(1,2,1)
plt.title('Pie chart')
plt.pie(df['FBS over 120'].value_counts(), textprops={'fontsize':12})
plt.subplots_adjust(left=0.125)

plt.subplot(1,2,2)
plt.title('Count plot')
sns.countplot(x='FBS over 120', data=df, order=order)
plt.xticks([0,1], labels=labels)

plt.show()

df['FBS over 120'].value_counts()
```

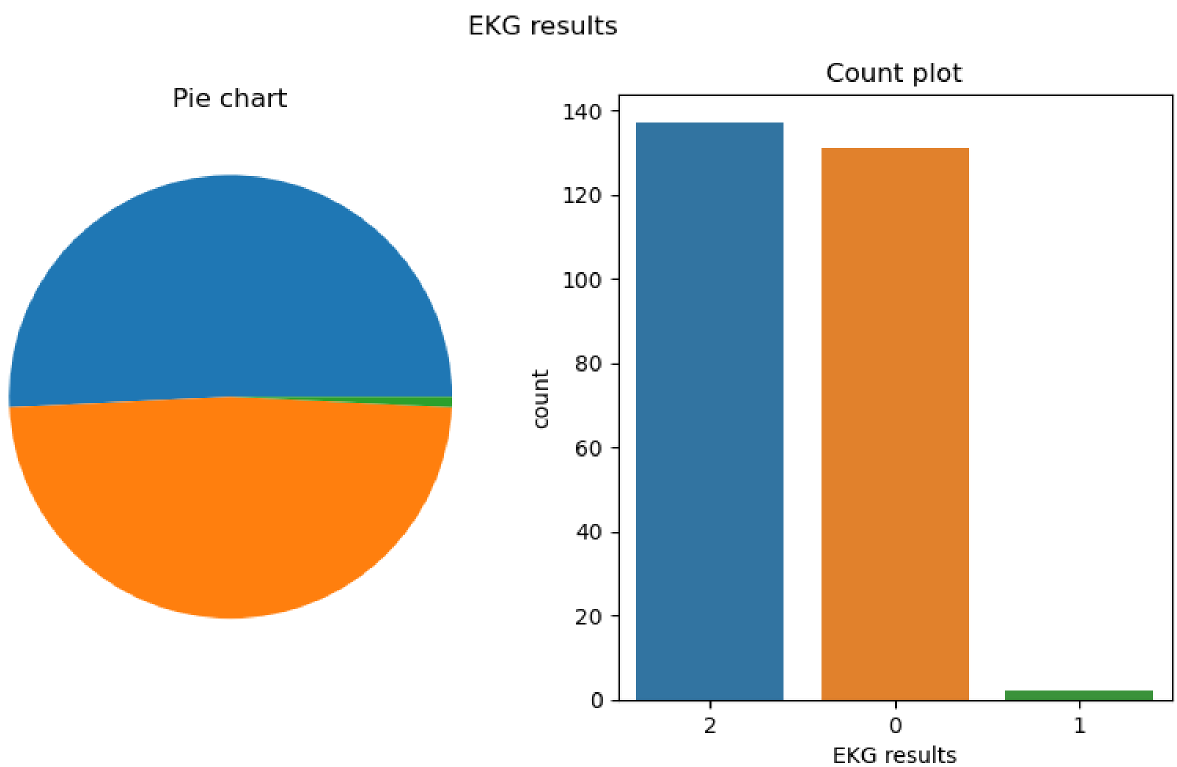


Out[13]:

```
0    230
1     40
Name: FBS over 120, dtype: int64
```

In [14]:

```
labels = ["normal", 'aving ST-T wave abnormality', "showing probable or definite left ventricular
hypertrophy by Estes' criteria"]
order = df['EKG results'].value_counts().index

plt.figure(figsize=(10,5))
plt.suptitle("EKG results")

plt.subplot(1,2,1)
plt.title('Pie chart')
plt.pie(df['EKG results'].value_counts(), textprops={'fontsize':12})
plt.subplots_adjust(left=0.125)

plt.subplot(1,2,2)
plt.title('Count plot')
sns.countplot(x='EKG results', data=df, order=order)
#plt.xticks([0,1,2], labels=labels, rotation=45)

plt.show()

df['EKG results'].value_counts()
```

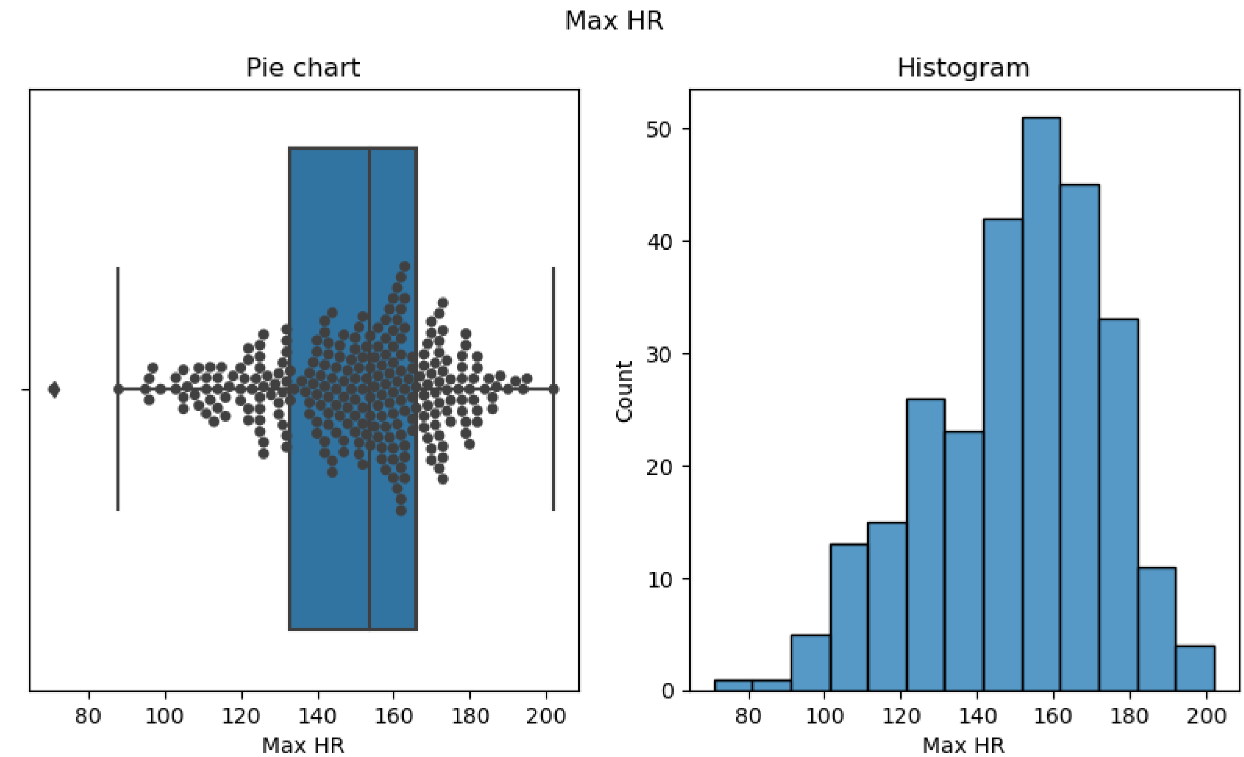


Out[14]:

```
2    137
0    131
1      2
Name: EKG results, dtype: int64
```

In [15]:

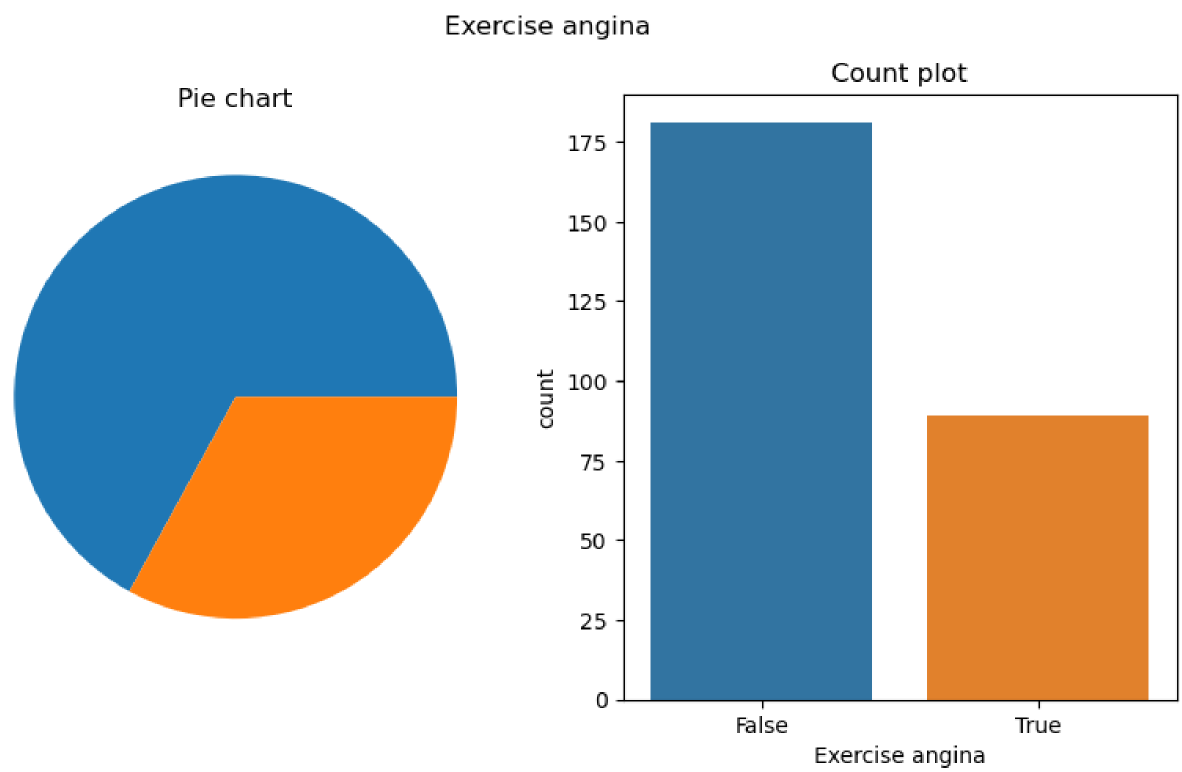
```
plt.figure(figsize=(10,5))
plt.suptitle("Max HR")

plt.subplot(1,2,1)
plt.title('Pie chart')
sns.boxplot(x="Max HR", data=df)
sns.swarmplot(x="Max HR", data=df, color=".25")

plt.subplot(1,2,2)
plt.title('Histogram')
sns.histplot(x='Max HR', data=df)
plt.show()

print("The max heart rate: %2.2f The histogram is slightly left skewed" % (df["Max HR"].mean()))
```



The max heart rate: 149.68 The histogram is slightly left skewed

In [16]:

```
labels = ["False", 'True']
order = df['Exercise angina'].value_counts().index

plt.figure(figsize=(10,5))
plt.suptitle("Exercise angina")

plt.subplot(1,2,1)
plt.title('Pie chart')
plt.pie(df['Exercise angina'].value_counts(), textprops={'fontsize':12})
plt.subplots_adjust(left=0.125)

plt.subplot(1,2,2)
plt.title('Count plot')
sns.countplot(x='Exercise angina', data=df, order=order)
plt.xticks([0,1], labels=labels)

plt.show()

df['Exercise angina'].value_counts()
```

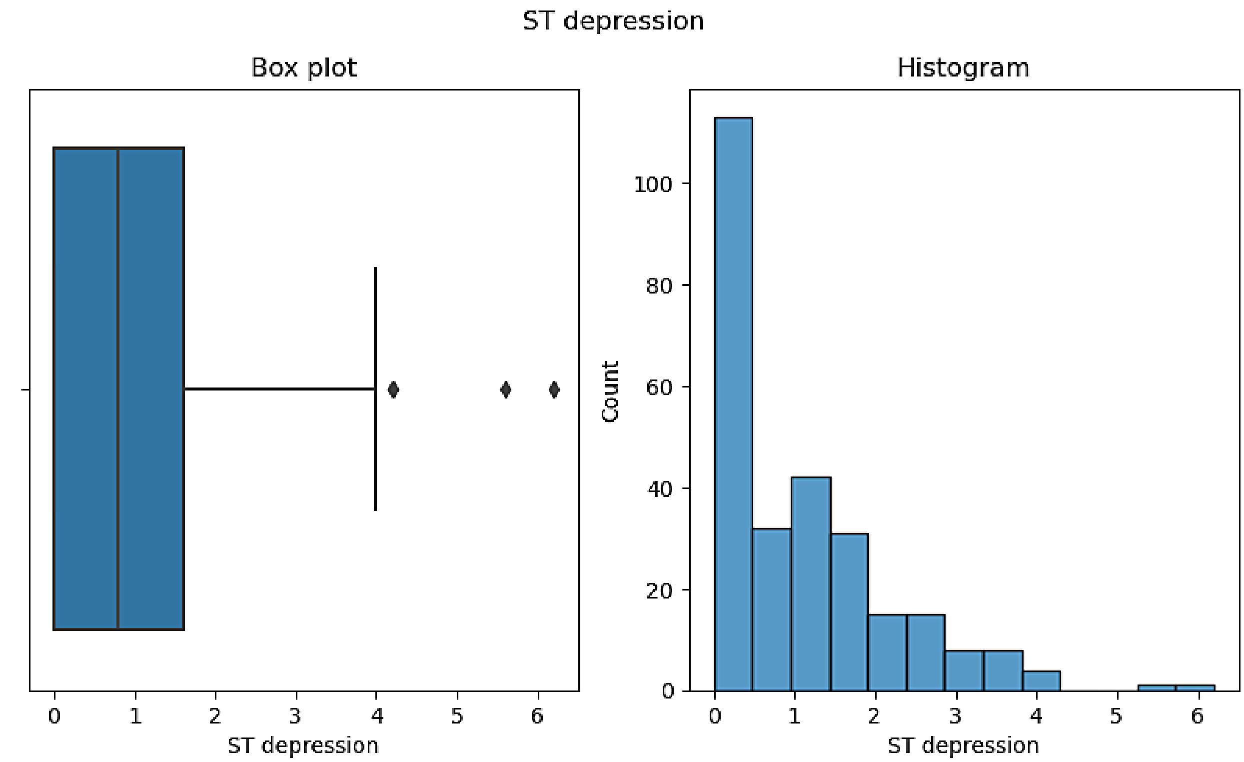


Out[16]:

```
0    181
1     89
Name: Exercise angina, dtype: int64
```

In [17]:

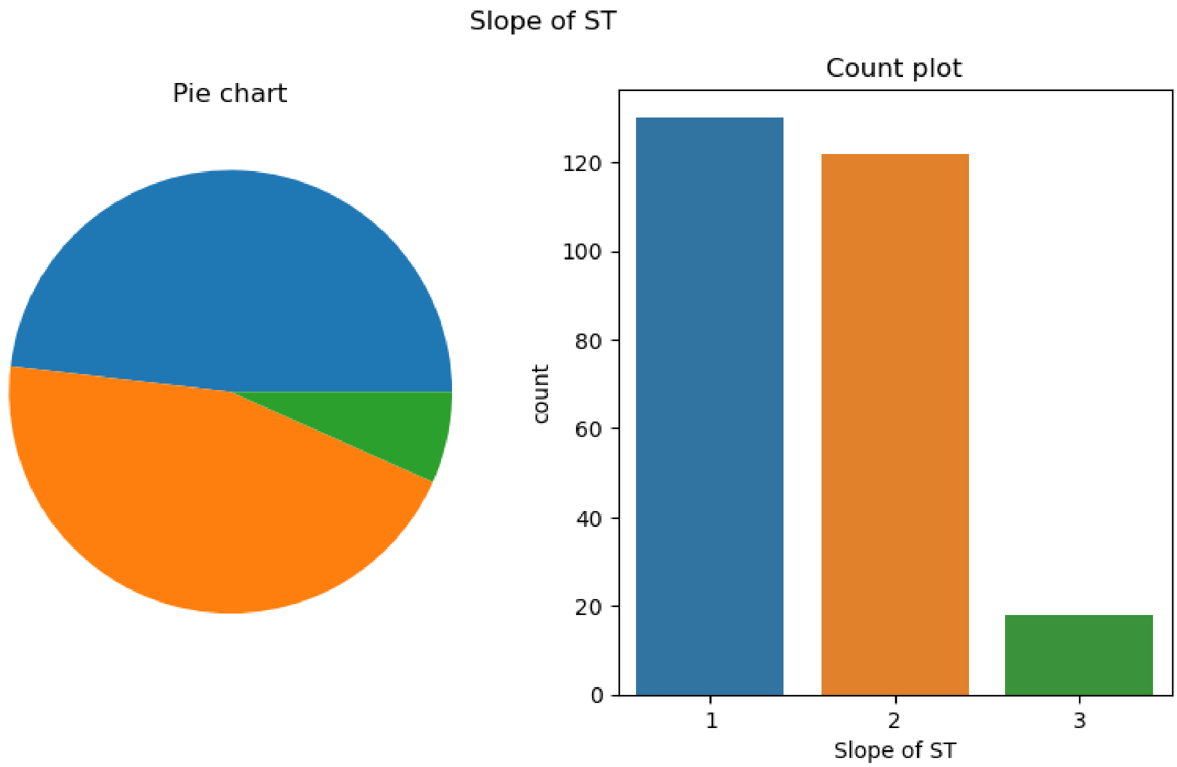
```
plt.figure(figsize=(10,5))
plt.suptitle("ST depression")

plt.subplot(1,2,1)
plt.title('Box plot')
sns.boxplot(x="ST depression", data=df)

plt.subplot(1,2,2)
plt.title('Histogram')
sns.histplot(x='ST depression', data=df)
plt.show()

print("The ST depression average: %2.2f The histogram is left skewed" % (df["ST depression"].mean()))
```



The ST depression average: 1.05 The histogram is left skewed

In [18]:

```
labels = ["1", '2', '3']
order = df['Slope of ST'].value_counts().index

plt.figure(figsize=(10,5))
plt.suptitle("Slope of ST")

plt.subplot(1,2,1)
plt.title('Pie chart')
plt.pie(df['Slope of ST'].value_counts(), textprops={'fontsize':12})
plt.subplots_adjust(left=0.125)

plt.subplot(1,2,2)
plt.title('Count plot')
sns.countplot(x='Slope of ST', data=df, order=order)
plt.xticks([0,1,2], labels=labels)

plt.show()

df['Slope of ST'].value_counts()
```

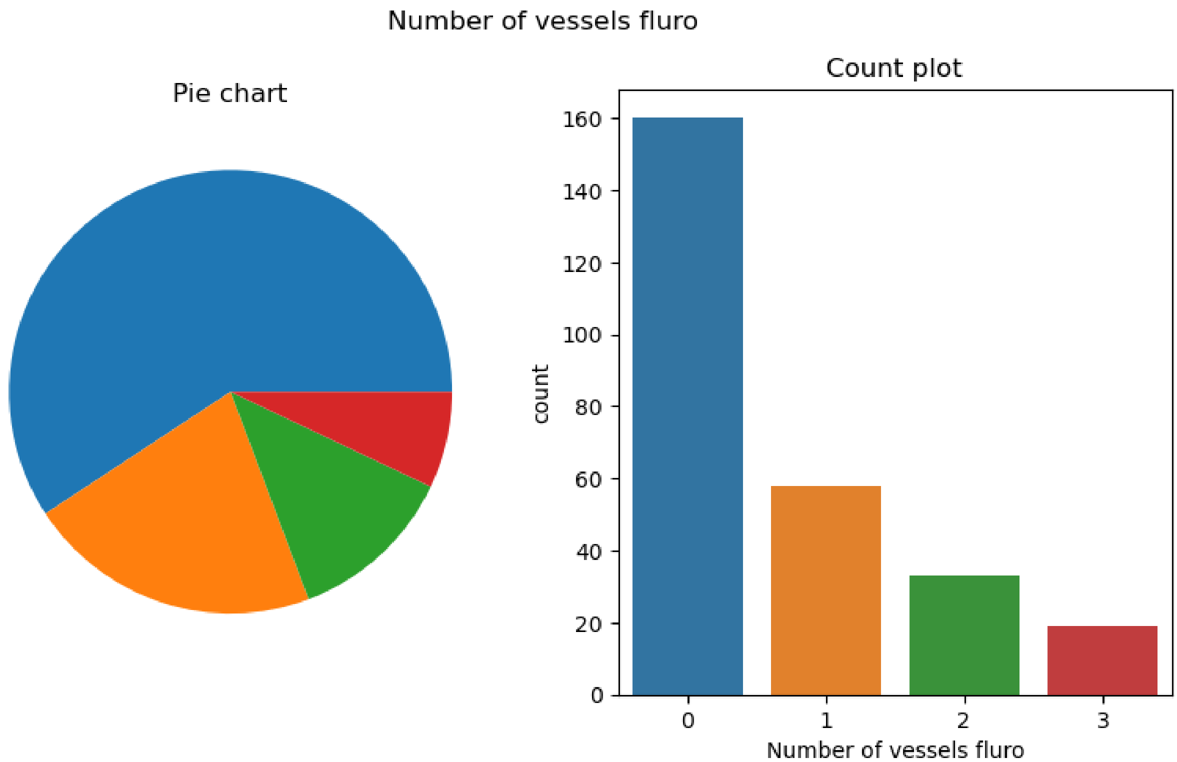


Out[18]:

```
1    130
2    122
3     18
Name: Slope of ST, dtype: int64
```

In [19]:

```
labels = ["0", '1', '2', "3"]
order = df['Number of vessels fluro'].value_counts().index

plt.figure(figsize=(10,5))
plt.suptitle("Number of vessels fluro")

plt.subplot(1,2,1)
plt.title('Pie chart')
plt.pie(df['Number of vessels fluro'].value_counts(), textprops={'fontsize':12})
plt.subplots_adjust(left=0.125)

plt.subplot(1,2,2)
plt.title('Count plot')
sns.countplot(x='Number of vessels fluro', data=df, order=order)
plt.xticks([0,1,2,3], labels=labels)

plt.show()

df['Number of vessels fluro'].value_counts()
```

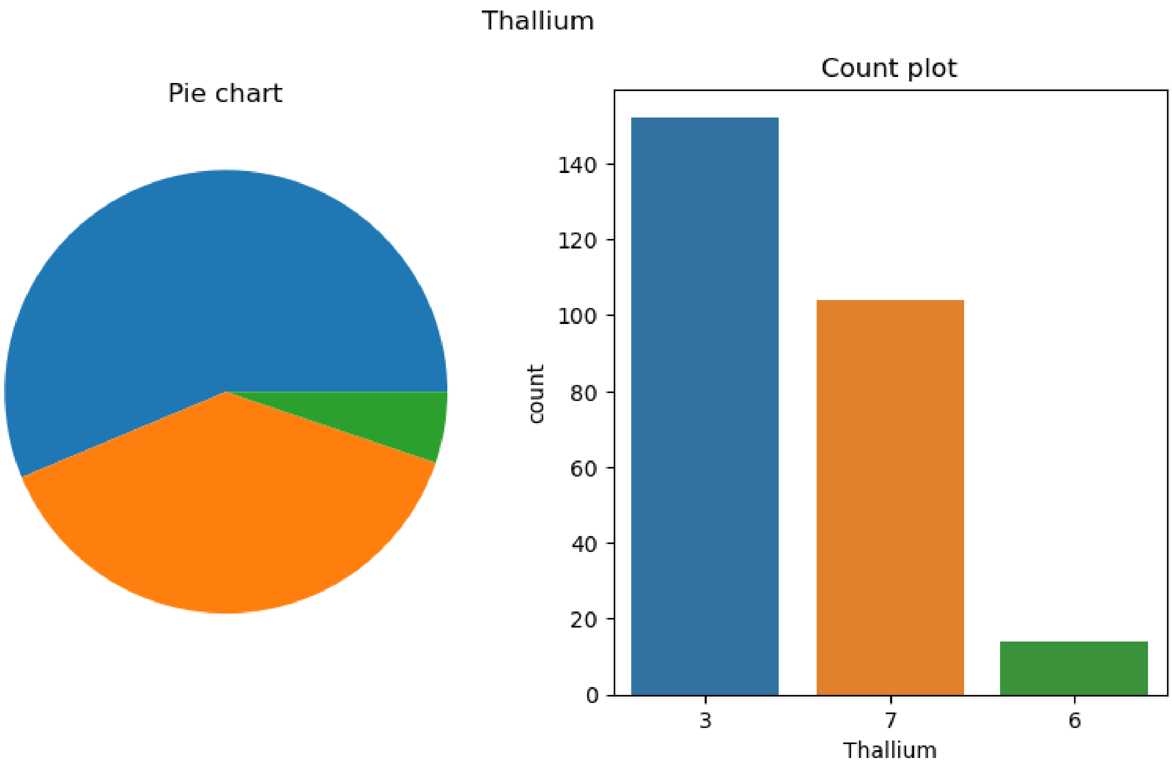


Out[19]:

```
0    160
1     58
2     33
3     19
Name: Number of vessels fluro, dtype: int64
```

In [20]:

```
labels = ["3", '7', '6']
order = df['Thallium'].value_counts().index

plt.figure(figsize=(10,5))
plt.suptitle("Thallium")

plt.subplot(1,2,1)
plt.title('Pie chart')
plt.pie(df['Thallium'].value_counts(), textprops={'fontsize':12})
plt.subplots_adjust(left=0.125)

plt.subplot(1,2,2)
plt.title('Count plot')
sns.countplot(x='Thallium', data=df, order=order)
plt.xticks([0,1,2], labels=labels)

plt.show()

df['Thallium'].value_counts()
```



Out[20]:

```
3    152
7    104
6     14
Name: Thallium, dtype: int64
```

EDA - Exploratory Data Analysis - PNT2022TMID09615

In [21]:

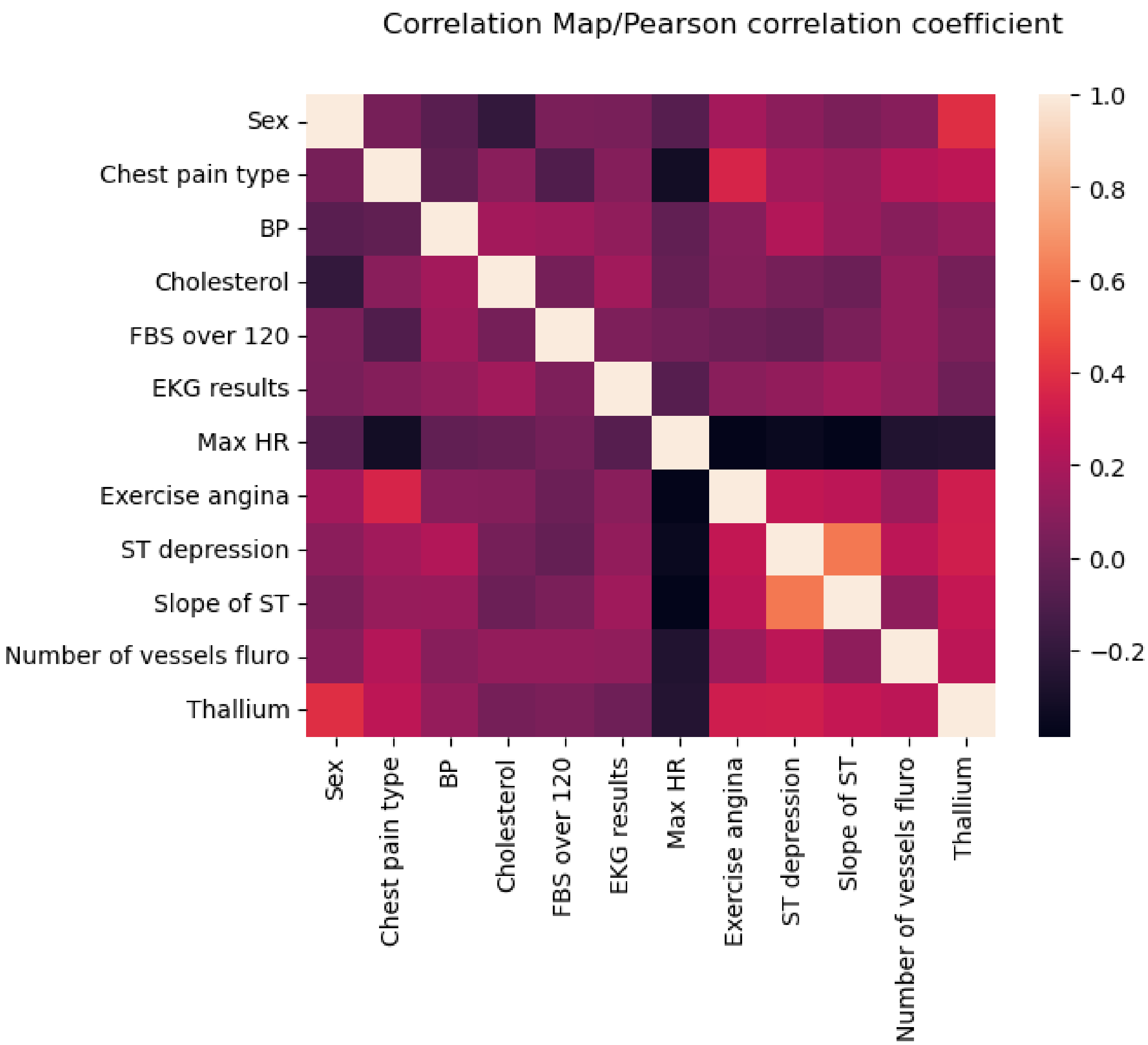
```
target = df['Heart Disease'].map({'Presence':1, 'Absence':0})
inputs = df.drop(['Heart Disease'], axis=1)
```

In [22]:

```
# Correlation matrix - The 'Slope of ST' is highly linearly correlated with 'ST depression'
```

In [23]:

```
plt.suptitle("Correlation Map/Pearson correlation coefficient")
sns.heatmap(df.iloc[:,1:-1].corr())

plt.show()
```

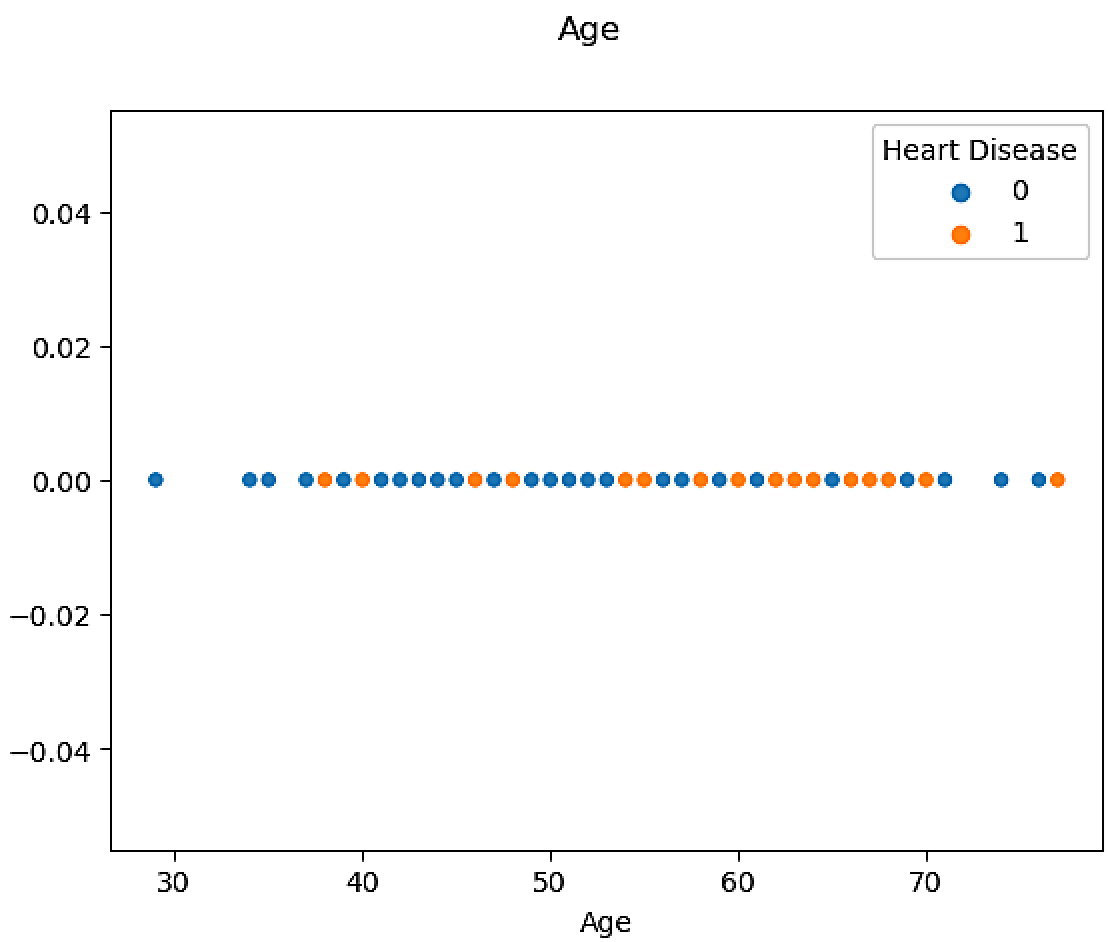


In [ ]:

```
# Age Analysis
```

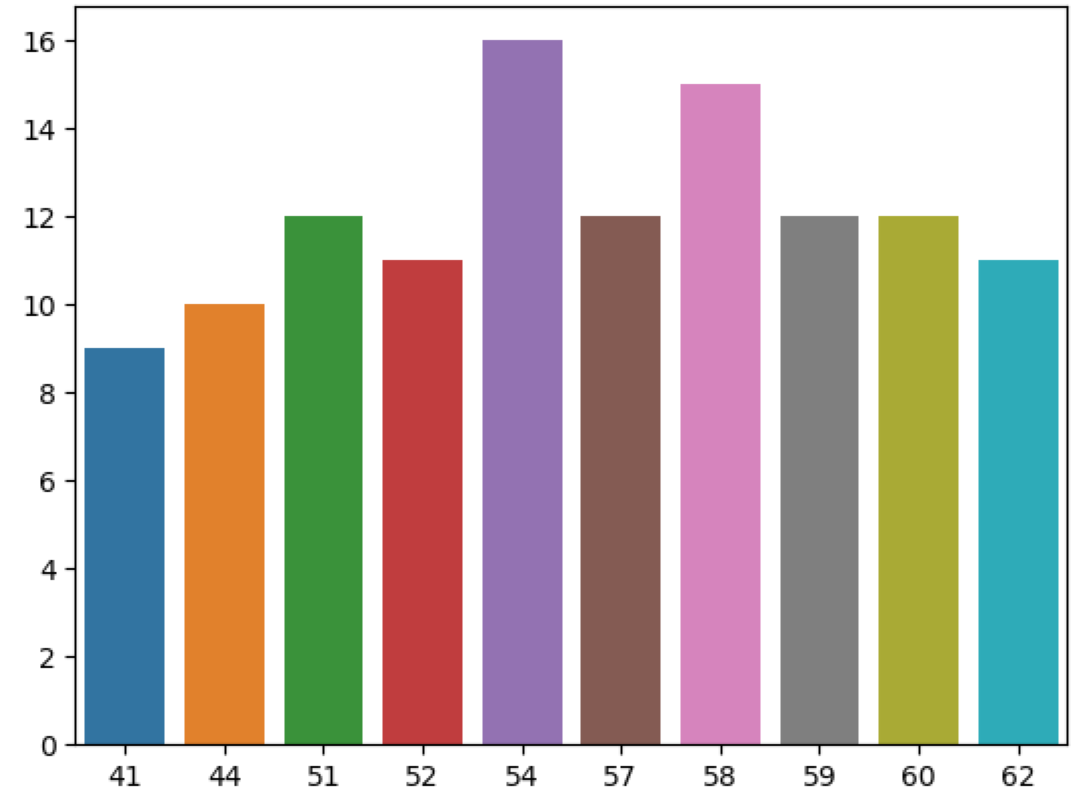
In [24]:

```
plt.suptitle("Age")
sns.scatterplot(data=df, x='Age', y=np.zeros(len(df['Age'])), hue=target)
plt.show()
```



In [25]:

```
data=df
sns.barplot(x=data.Age.value_counts()[:10].index,y=data.Age.value_counts()[:10].values)
plt.show()
```



In [26]:

```
minAge=min(data.Age)
maxAge=max(data.Age)
meanAge=data.Age.mean()
print('Min Age :',minAge)
print('Max Age :',maxAge)
print('Mean Age :',meanAge)
```

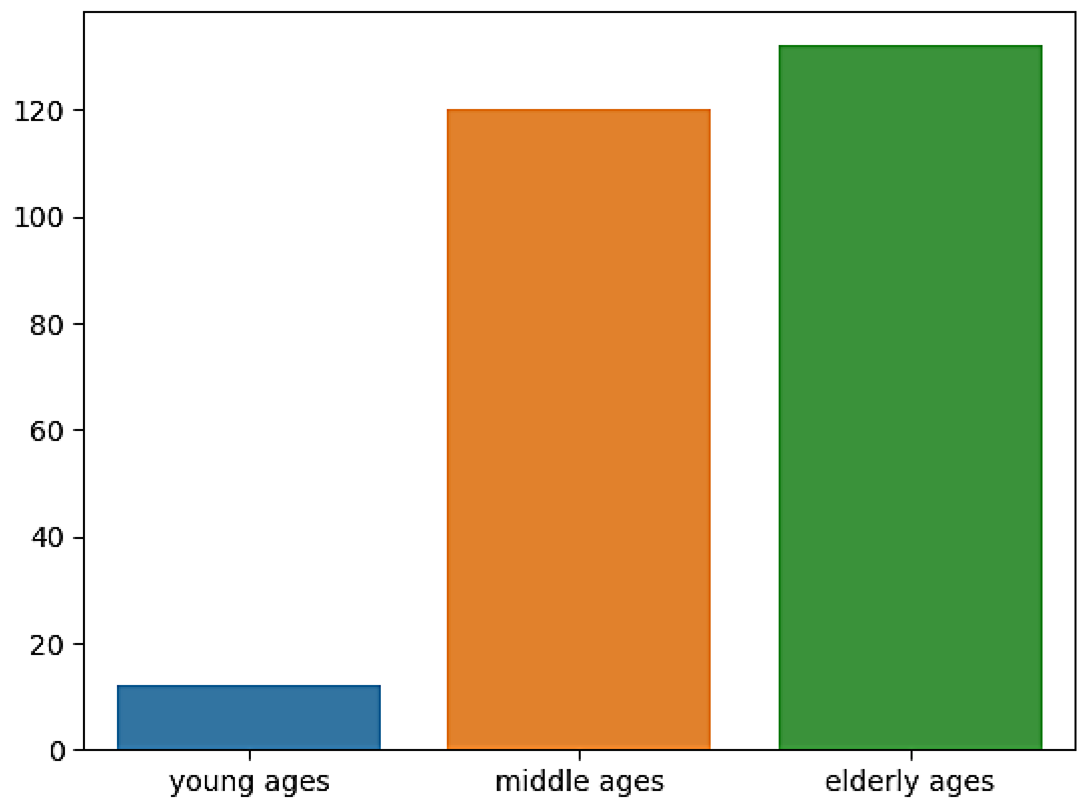
```
Min Age : 29
Max Age : 77
Mean Age : 54.43333333333333
```

In [ ]:

```
# Dividing the Age feature into three parts - "Young", "Middle" and "Elder"
```

In [27]:

```
Young = data[(data.Age>=29)&(data.Age<40)]
Middle = data[(data.Age>=40)&(data.Age<55)]
Elder = data[(data.Age>55)]


sns.set_context(font_scale = 1)
sns.barplot(x=['young ages','middle ages','elderly ages'],y=[len(Young),len(Middle),len(Elder)])
plt.show()
```



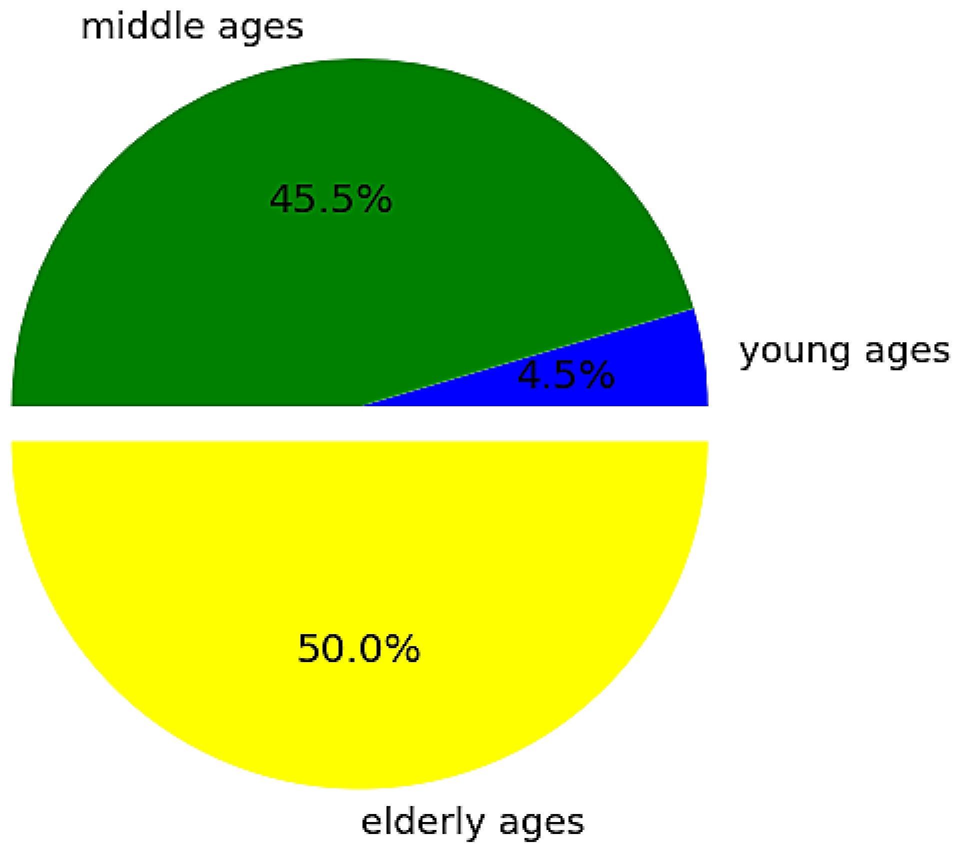
In [ ]:

 # A large proportion of dataset contains Elder people.

 # Elderly people are more likely to suffer from heart disease.

In [28]:

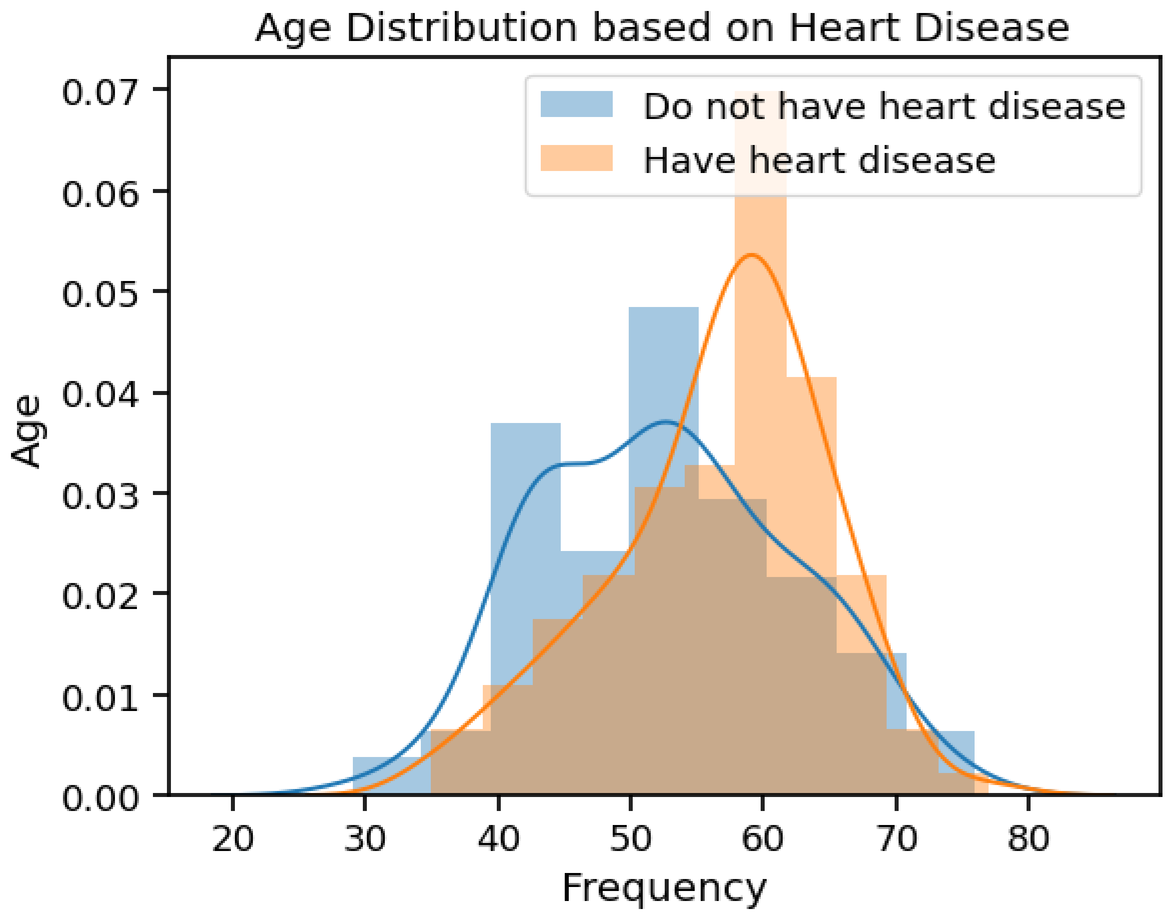
```
colors = ['blue','green','yellow']
explode = [0,0,0.1]

sns.set_context('notebook',font_scale = 1.2)
plt.pie([len(Young),len(Middle),len(Elder)],labels=['young ages','middle ages','elderly
ages'],explode=explode,colors=colors, autopct='%1.1f%%')
plt.tight_layout()
```



In [29]:

```
# Display age distribution based on heart disease
sns.distplot(data[data['Heart Disease'] == 'Absence']['Age'], label='Do not have heart disease')
sns.distplot(data[data['Heart Disease'] == 'Presence']['Age'], label = 'Have heart disease')
plt.xlabel('Frequency')
plt.ylabel('Age')
plt.title('Age Distribution based on Heart Disease')
plt.legend()
plt.show()
```

```
C:\Users\91904\anaconda3\lib\site-packages\seaborn\distributions.py:2619: FutureWarning: `distplot` is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
C:\Users\91904\anaconda3\lib\site-packages\seaborn\distributions.py:2619: FutureWarning: `distplot` is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

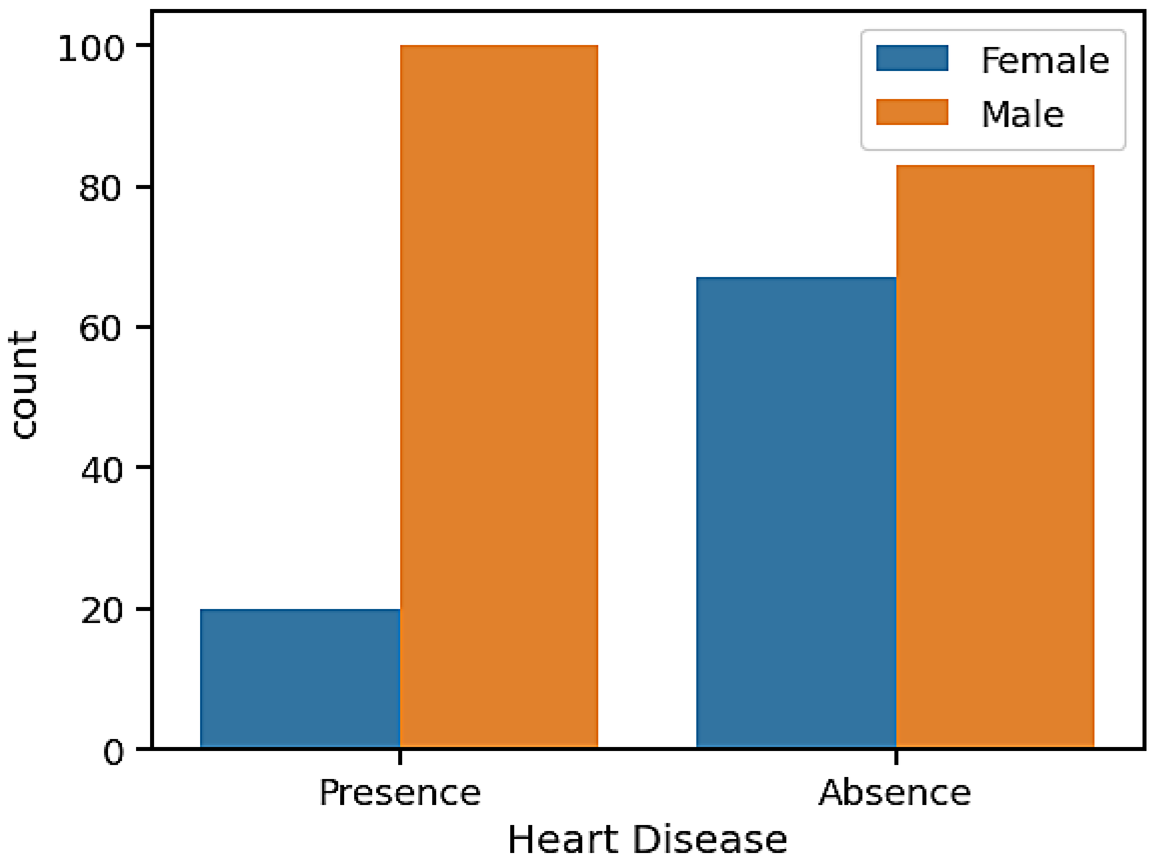


In [ ]:

# *Sex Feature Analysis*
sns.set_context('notebook',font_scale=1.5) sns.countplot(data['Sex']) plt.show()

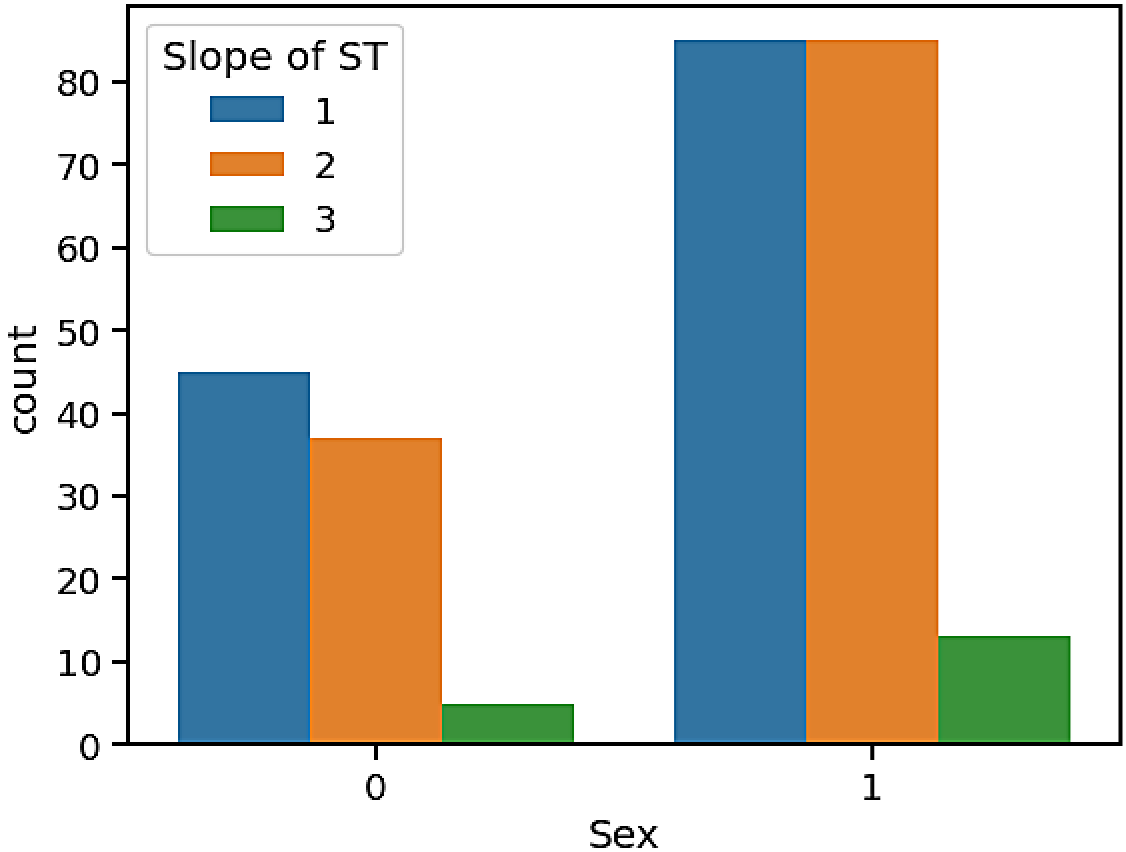
In [30]:

```
ax = sns.countplot(x='Heart Disease', hue='Sex', data=df)
legend_labels, _= ax.get_legend_handles_labels()
ax.legend(legend_labels, ['Female','Male'], bbox_to_anchor=(1,1))
plt.show()
```



In [31]:

```
sns.countplot(data['Sex'],hue=data["Slope of ST"])
plt.show()
C:\Users\91904\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variable as a keyword arg: x. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```



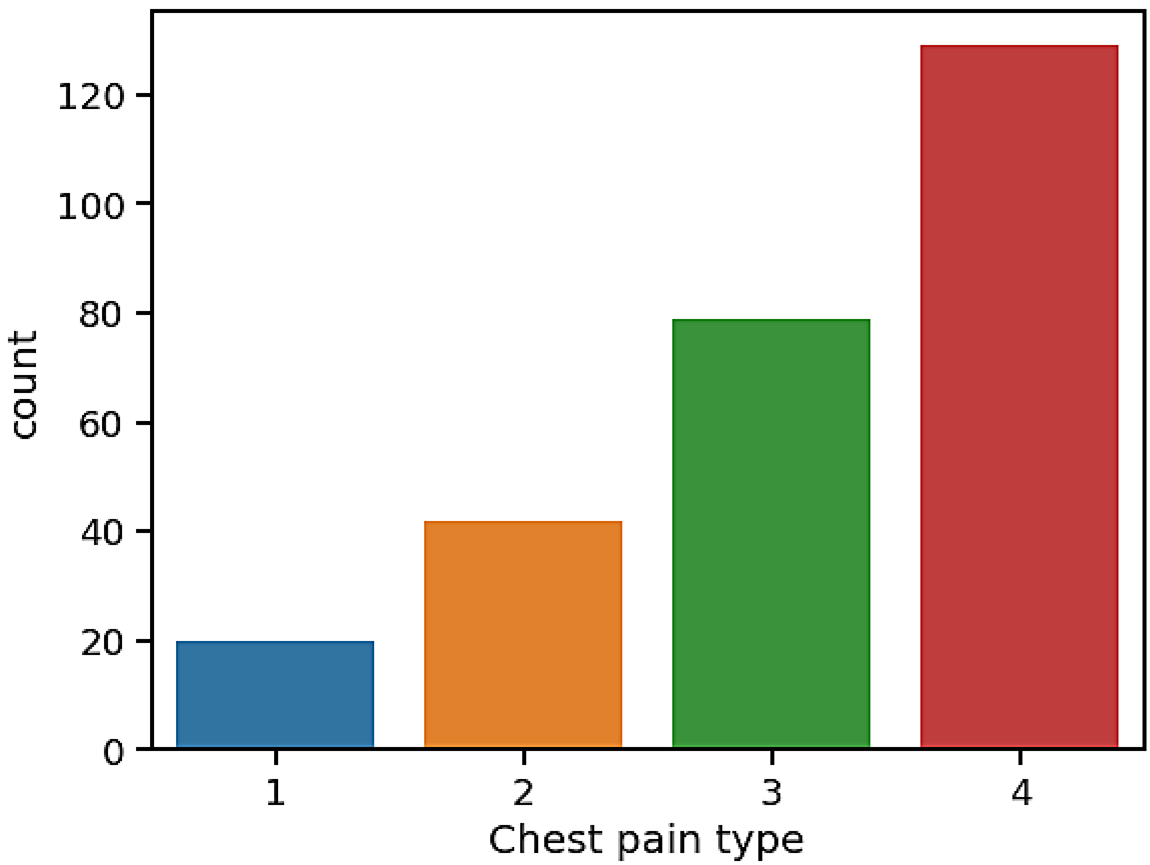
In [ ]:

Chest Pain Type Analysis

In [32]:

```
sns.countplot(data['Chest pain type'])
plt.show()
```

C:\Users\91904\anaconda3\lib\site-packages\seaborn_decorators.py:36: FutureWarning: Pass the following variable as a keyword arg: x. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(

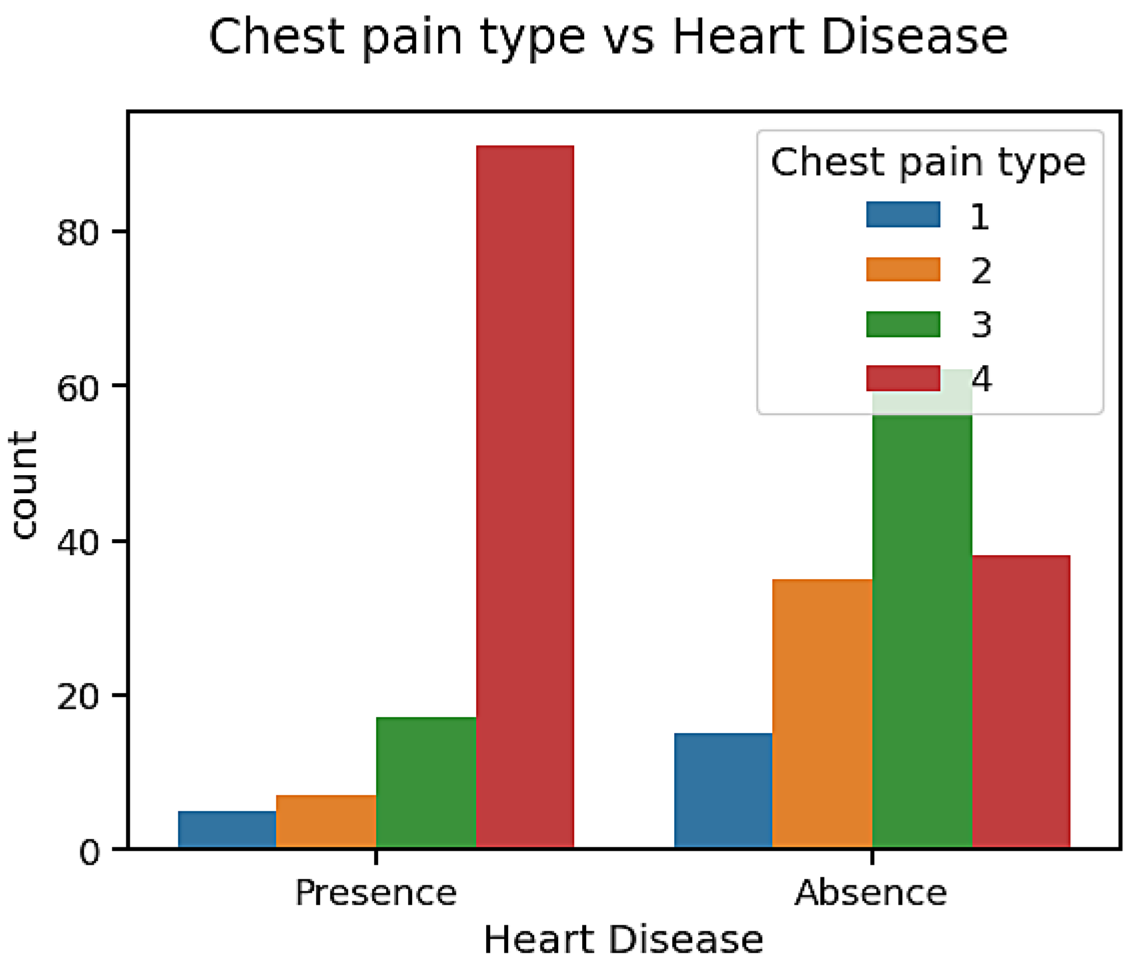


In [ ]:

 # Heart disease based on Chest pain type - 4th type of chest pain dominate in heart disease

In [33]:

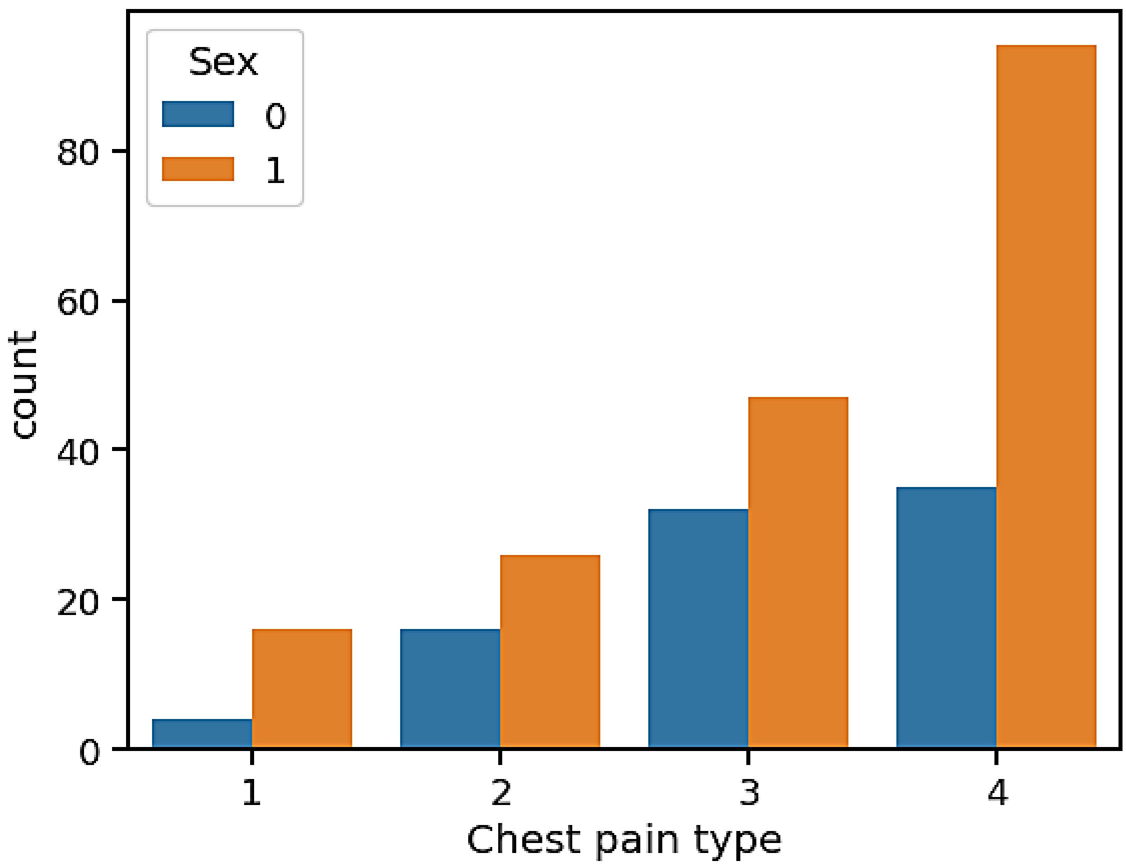
```
plt.suptitle('Chest pain type vs Heart Disease')
sns.countplot(data=df, x='Heart Disease', hue='Chest pain type')
plt.show()
```



In [34]:

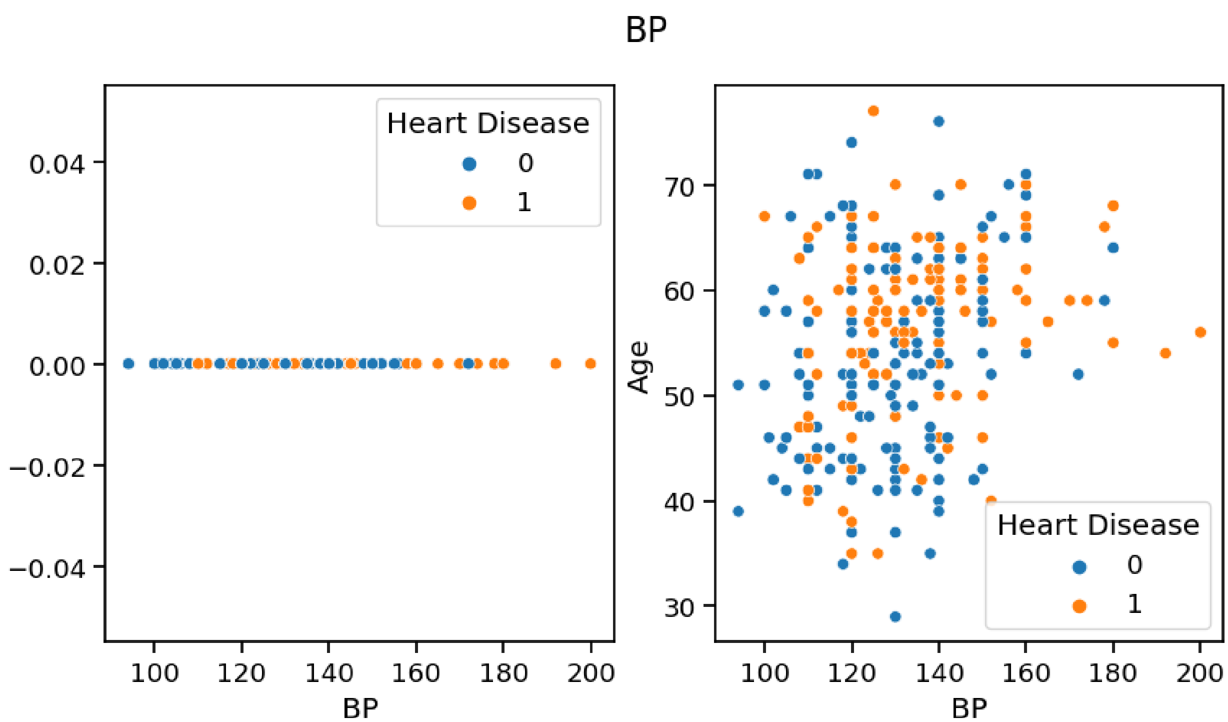
```
sns.countplot(data['Chest pain type'],hue=data["Sex"])
plt.show()
```

C:\Users\91904\anaconda3\lib\site-packages\seaborn_decorators.py:36: FutureWarning: Pass the following variable as a keyword arg: x. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(



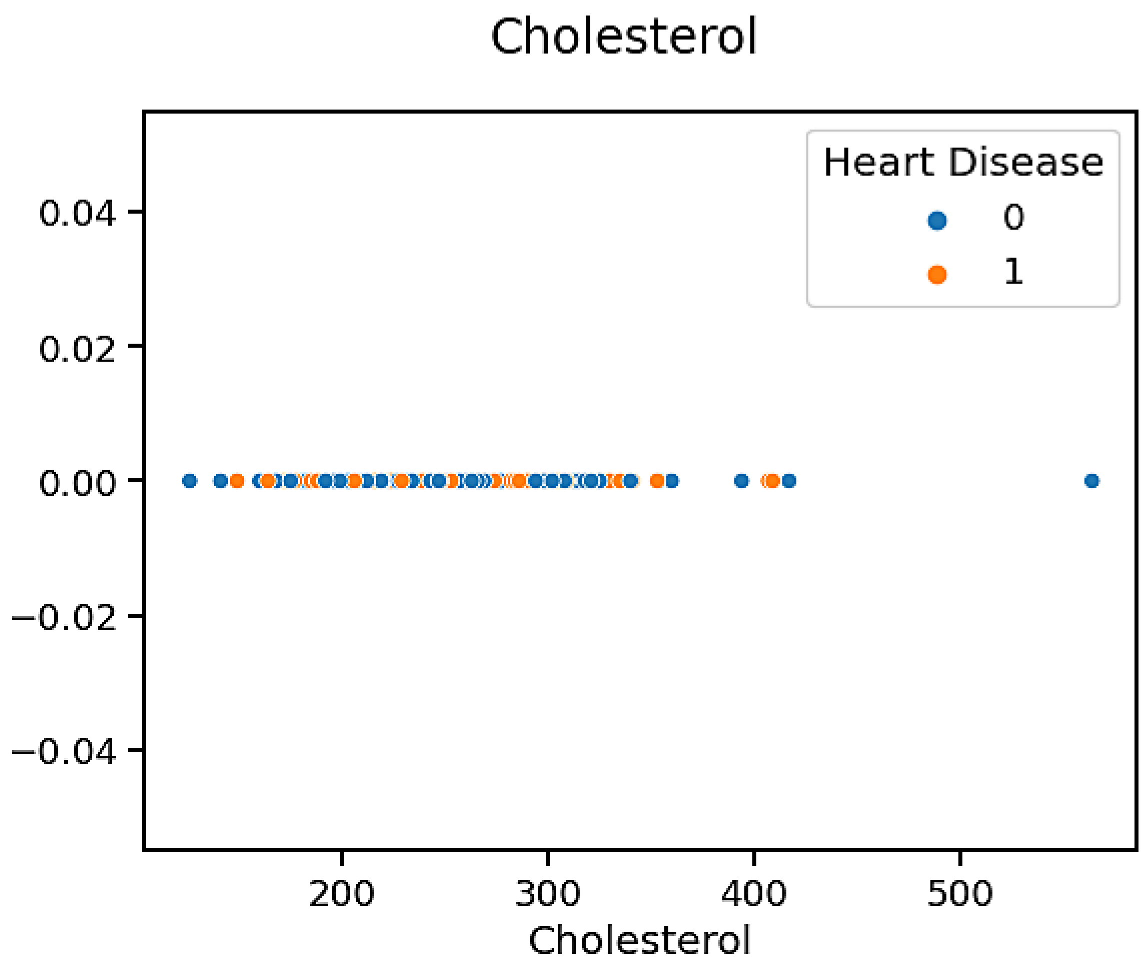
In [35]:

```
plt.figure(figsize=(10,5))

plt.subplot(1,2,1)
plt.suptitle("BP")
sns.scatterplot(data=df, x='BP', y=np.zeros(len(df['BP'])), hue=target)

plt.subplot(1,2,2)
sns.scatterplot(data=df, x='BP', y='Age', hue=target)


plt.show()
```



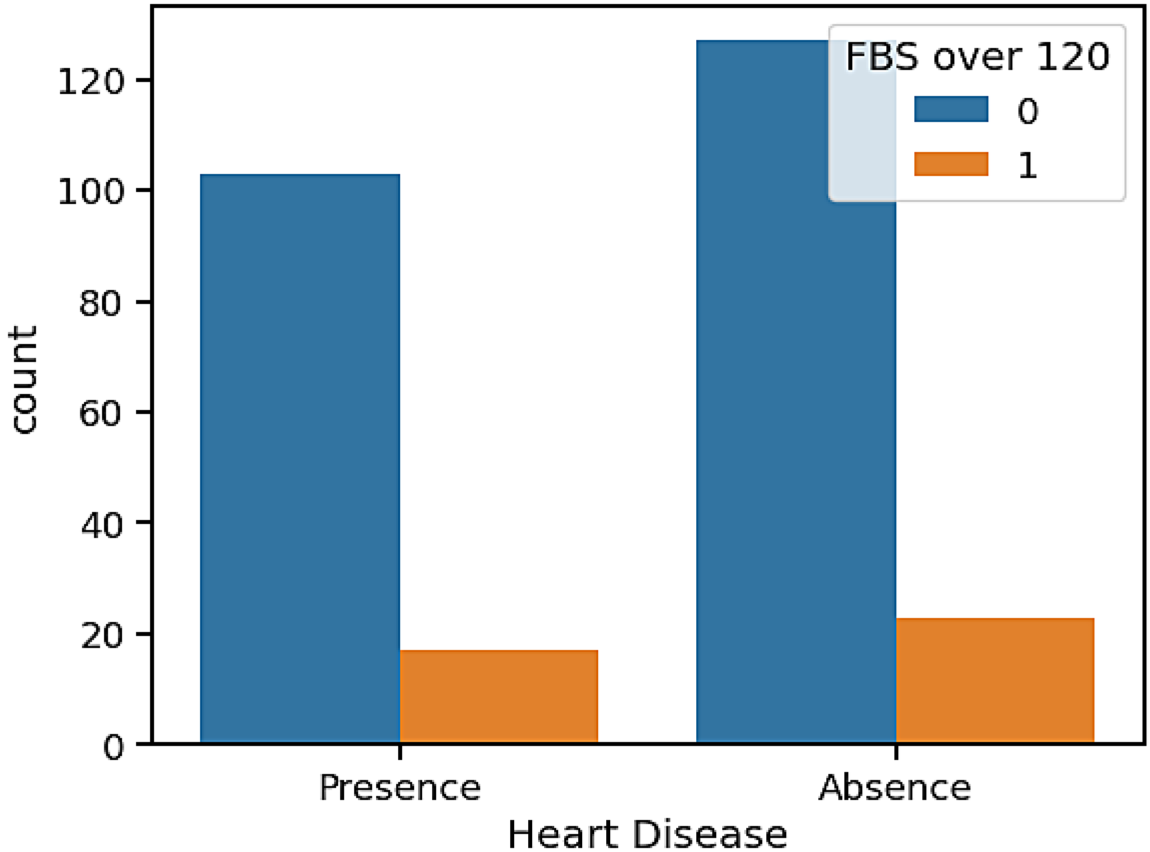
In [36]:

```
plt.suptitle("Cholesterol")
sns.scatterplot(data=df, x='Cholesterol', y=np.zeros(len(df['Cholesterol'])), hue=target)
plt.show()
```



In [37]:

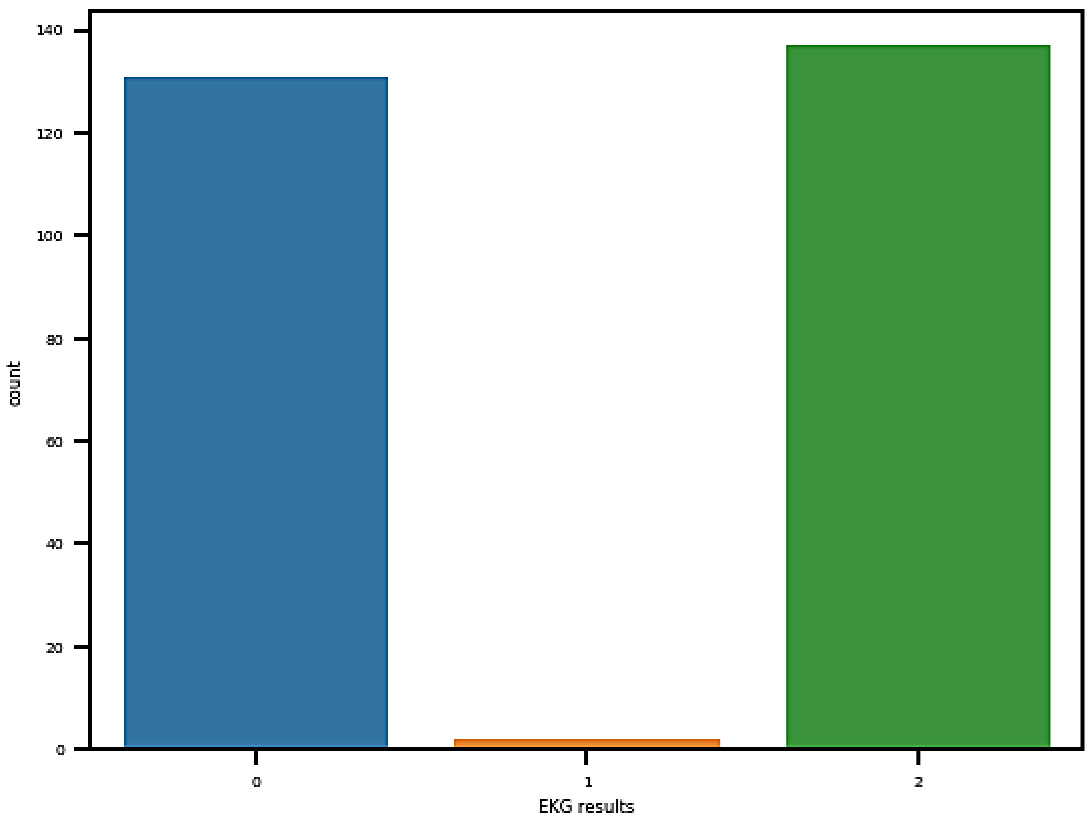
```
ax = sns.countplot(x='Heart Disease', hue='FBS over 120', data=df)
sns.set_context('notebook',font_scale = 0.5)
plt.show()
```



In [38]:

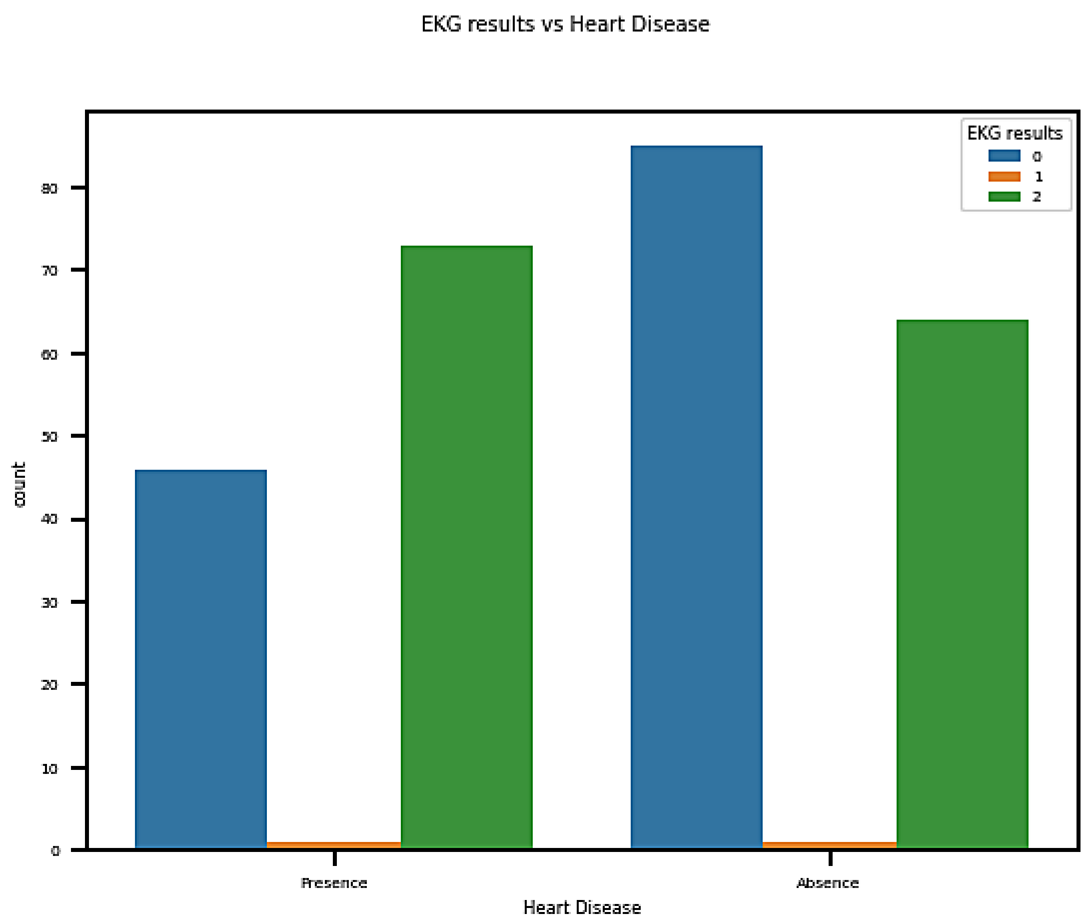
```
sns.countplot(data['EKG results'])
plt.show()
```

C:\Users\91904\anaconda3\lib\site-packages\seaborn_decorators.py:36: FutureWarning: Pass the following variable as a keyword arg: x. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(



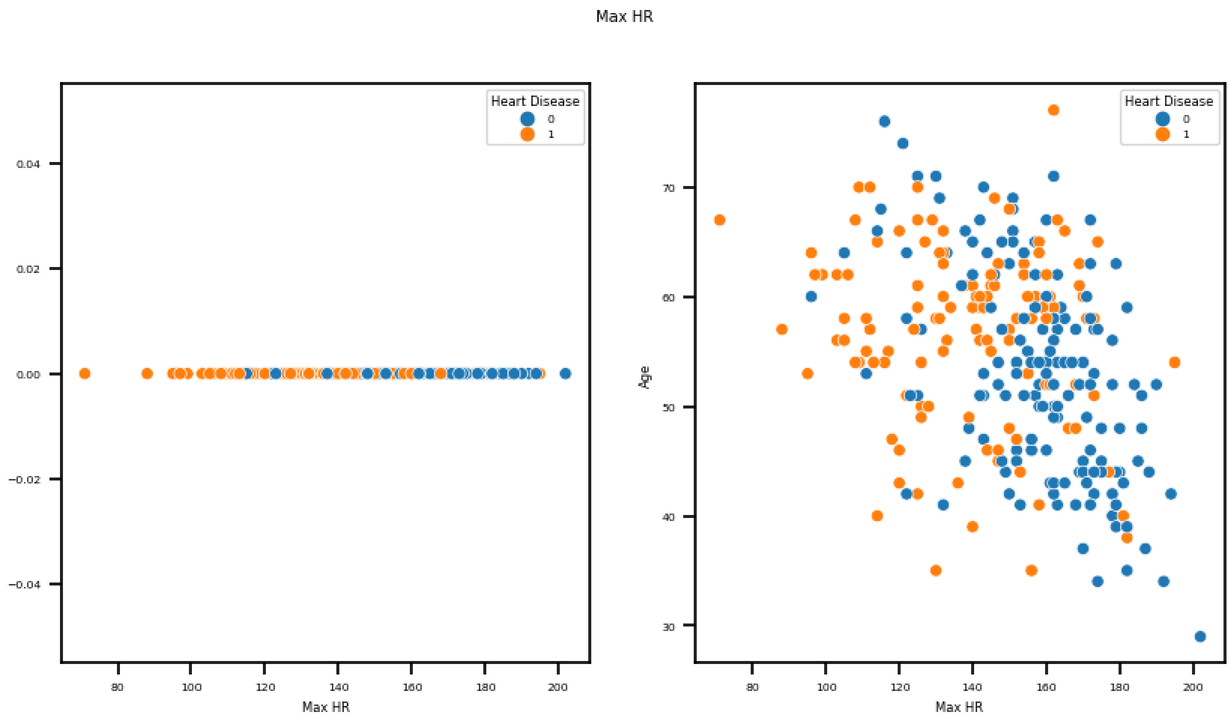
In [39]:

```
plt.suptitle('EKG results vs Heart Disease')
sns.countplot(data=df, x='Heart Disease', hue='EKG results')
plt.show()
```



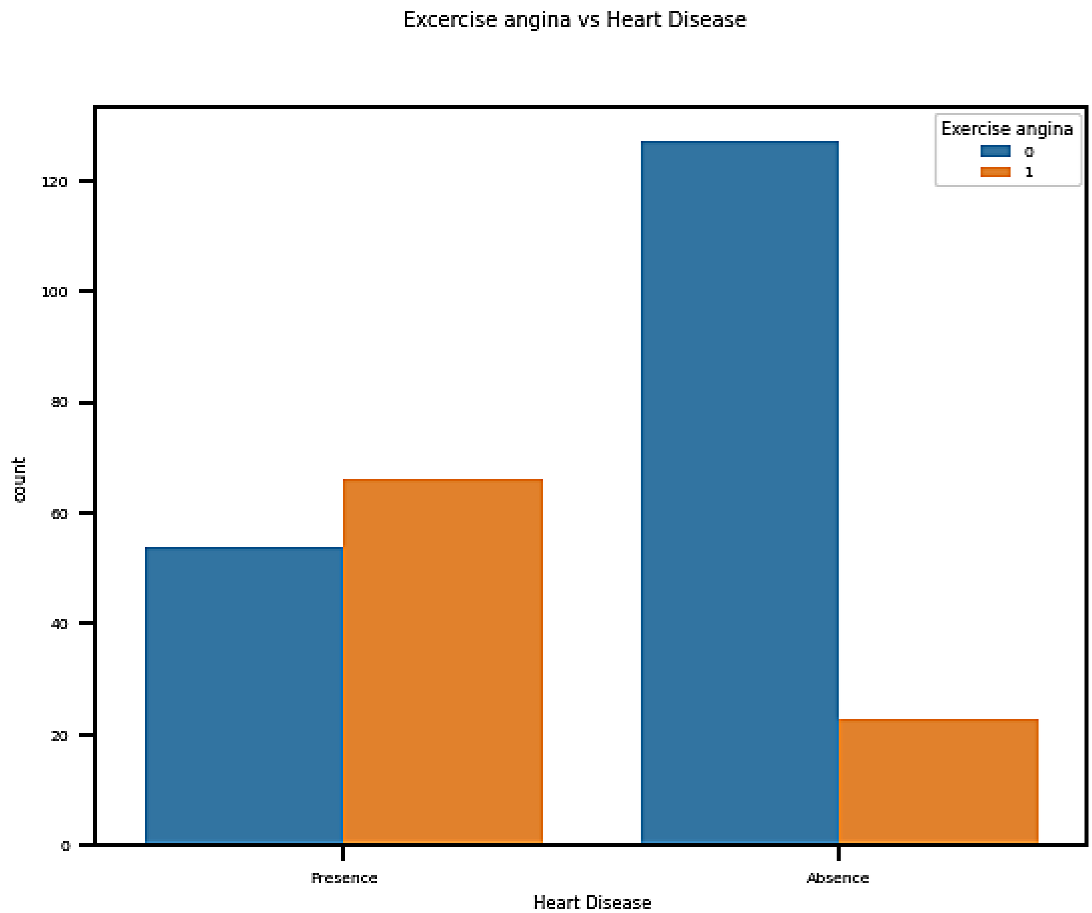
In [40]:

```
plt.figure(figsize=(10,5))

plt.subplot(1,2,1)
plt.suptitle("Max HR")
sns.scatterplot(data=df, x='Max HR', y=np.zeros(len(df['Max HR'])), hue=target)

plt.subplot(1,2,2)
sns.scatterplot(data=df, x='Max HR', y='Age', hue=target)

plt.show()
```



In [41]:

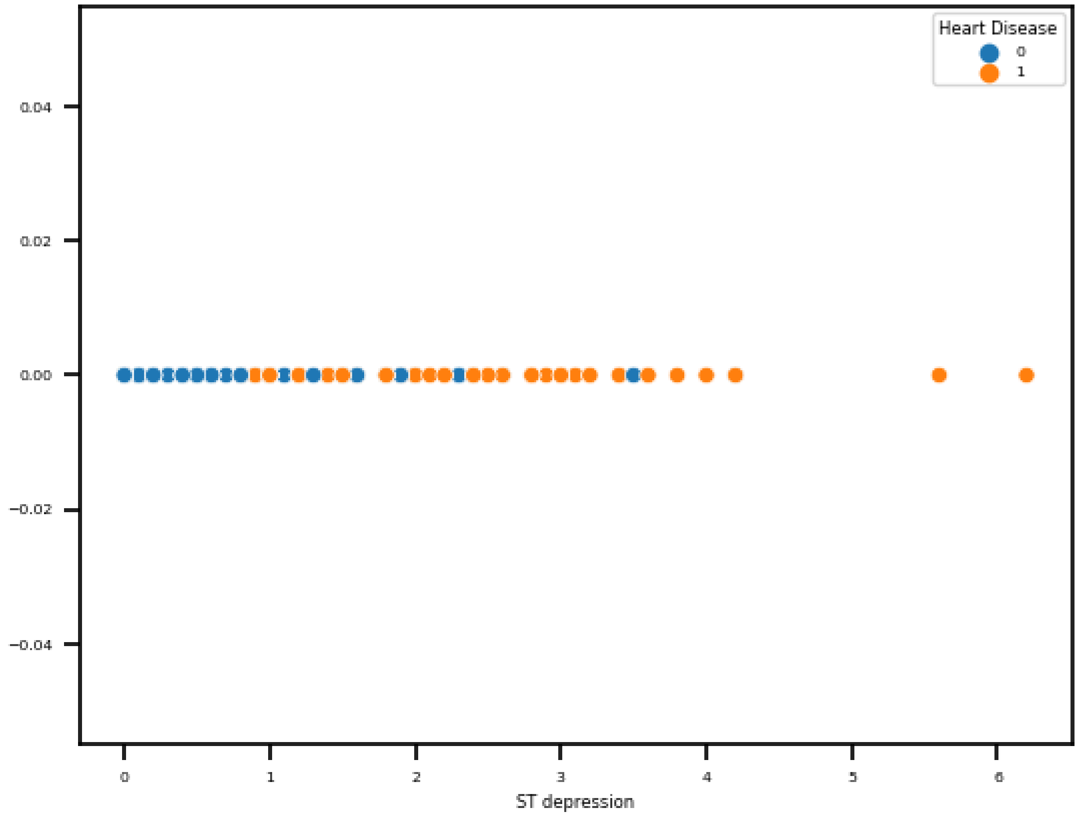
```
plt.suptitle('Excercise angina vs Heart Disease')
sns.countplot(data=df, x='Heart Disease', hue='Exercise angina')
plt.show()
```

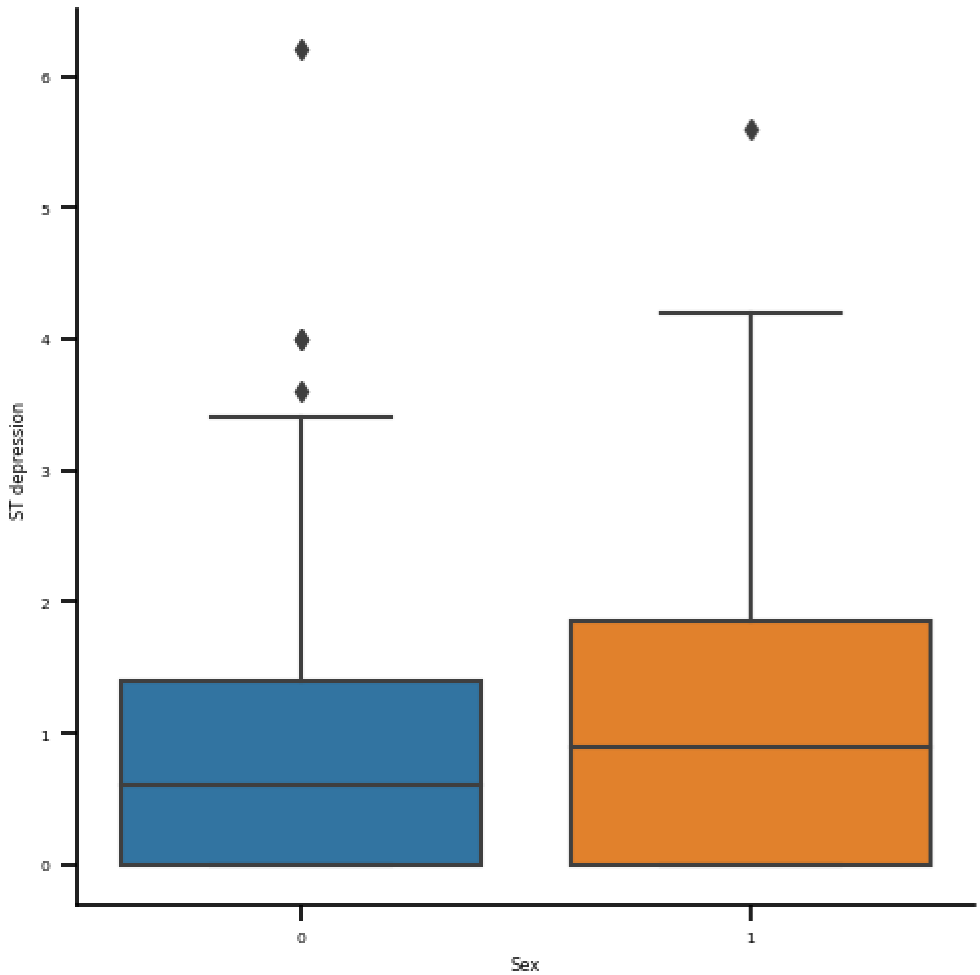


In [42]:

```
plt.suptitle("ST Depression")
sns.scatterplot(data=df, x='ST depression', y=np.zeros(len(df['ST depression'])), hue=target)


ax = sns.catplot(x='Sex', y='ST depression', kind='box',  data = df)
plt.show()
```

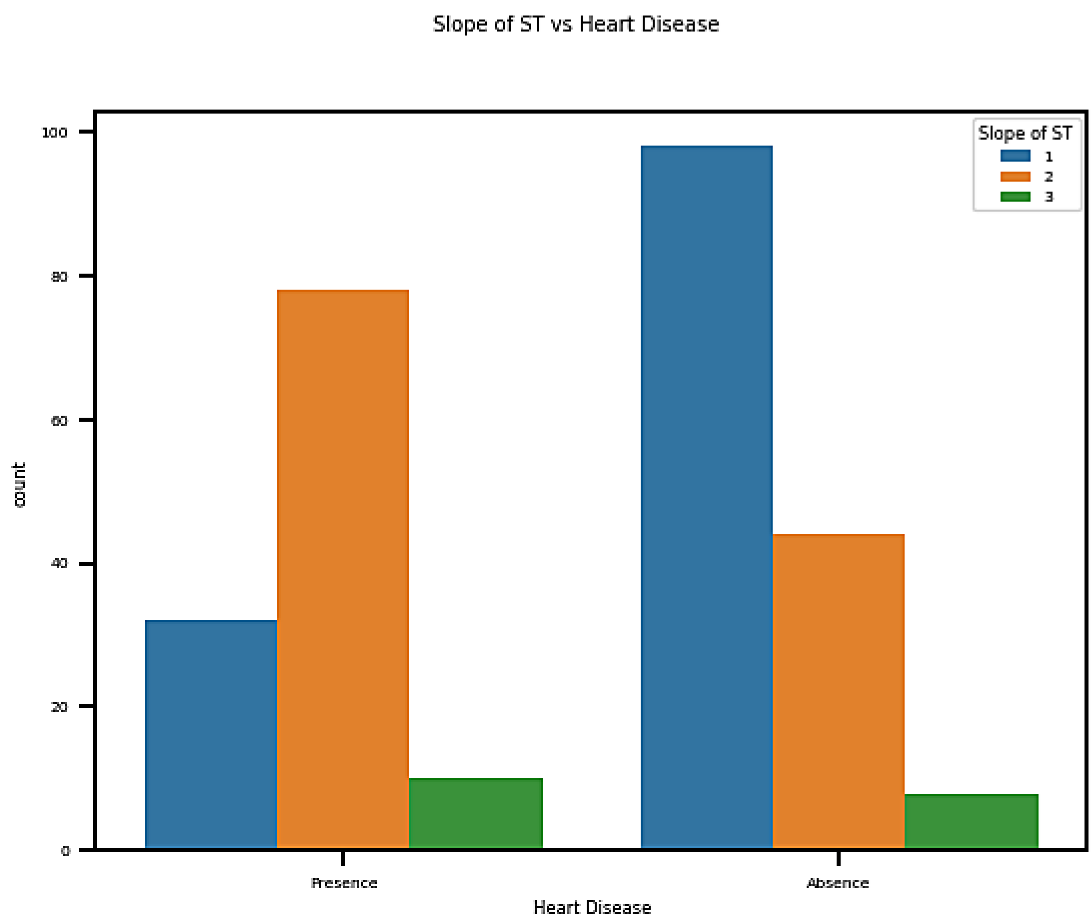
ST Depression





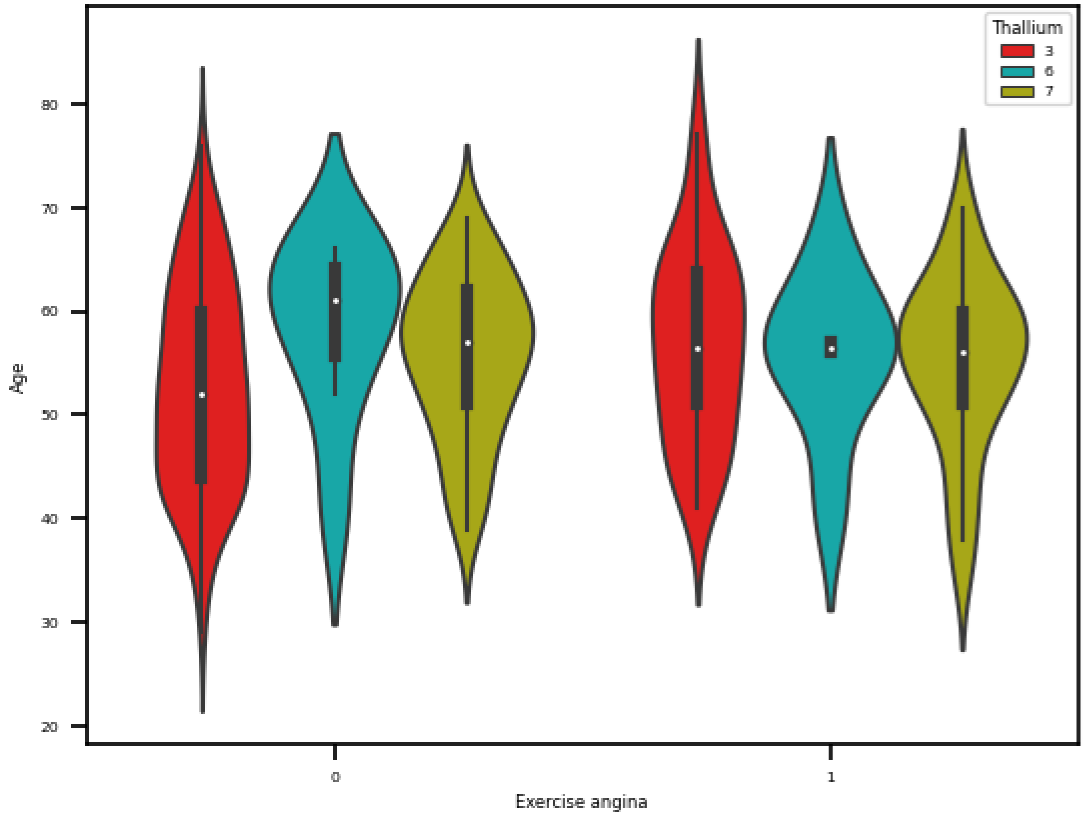
In [43]:

```
plt.suptitle('Slope of ST vs Heart Disease')
sns.countplot(data=df, x='Heart Disease', hue='Slope of ST')
sns.set_context(font_scale = 1)
plt.show()
```



In [44]:

```
sns.violinplot(x="Exercise angina",y="Age",data=data,palette=["r", "c", "y"],hue="Thallium")
plt.show()
```



In [45]:

```
Chest_pain_type = pd.get_dummies(df['Chest pain type'], prefix='Chest pain type', drop_first=True)
EKG_results = pd.get_dummies(df['EKG results'], prefix='EKG results', drop_first=True)
Number_of_vessels_fluro = pd.get_dummies(df['Number of vessels fluro'], prefix='Number of vessels
fluro', drop_first=True)
Thallium = pd.get_dummies(df['Thallium'], prefix='Thallium', drop_first=True)

frames = [df, Chest_pain_type, EKG_results, Number_of_vessels_fluro, Thallium]
df = pd.concat(frames, axis=1)

df.drop(columns = ['Chest pain type', 'EKG results', 'Number of vessels fluro', 'Thallium', 'Slope of ST'])

target = df['Heart Disease'].map({'Presence':1, 'Absence':0})
inputs = df.drop(['Heart Disease'], axis=1)

df.describe().T
```

Out[45]:

	count	mean	std	min	25%	50%	75%	
Age	270.0	54.433333	9.109067	29.0	48.0	55.0	61.0	77.0
Sex	270.0	0.677778	0.468195	0.0	0.0	1.0	1.0	1.0
Chest pain type	270.0	3.174074	0.950090	1.0	3.0	3.0	4.0	4.0

BP	270.0	131.344444	17.861608	94.0	120.0	130.0	140.0	200.0
Cholesterol	270.0	249.659259	51.686237	126.0	213.0	245.0	280.0	564.0
FBS over 120	270.0	0.148148	0.355906	0.0	0.0	0.0	0.0	1.0
EKG results	270.0	1.022222	0.997891	0.0	0.0	2.0	2.0	2.0
Max HR	270.0	149.677778	23.165717	71.0	133.0	153.5	166.0	202.0
Exercise angina	270.0	0.329630	0.470952	0.0	0.0	0.0	1.0	1.0
ST depression	270.0	1.050000	1.145210	0.0	0.0	0.8	1.6	6.2
Slope of ST	270.0	1.585185	0.614390	1.0	1.0	2.0	2.0	3.0
Number of vessels fluro	270.0	0.670370	0.943896	0.0	0.0	0.0	1.0	3.0
Thallium	270.0	4.696296	1.940659	3.0	3.0	3.0	7.0	7.0
Chest pain type_2	270.0	0.155556	0.363107	0.0	0.0	0.0	0.0	1.0
Chest pain type_3	270.0	0.292593	0.455798	0.0	0.0	0.0	1.0	1.0
Chest pain type_4	270.0	0.477778	0.500434	0.0	0.0	0.0	1.0	1.0
EKG results_1	270.0	0.007407	0.085906	0.0	0.0	0.0	0.0	1.0
EKG results_2	270.0	0.507407	0.500874	0.0	0.0	1.0	1.0	1.0
Number of vessels fluro_1	270.0	0.214815	0.411456	0.0	0.0	0.0	0.0	1.0
Number of vessels fluro_2	270.0	0.122222	0.328151	0.0	0.0	0.0	0.0	1.0
Number of vessels fluro_3	270.0	0.070370	0.256245	0.0	0.0	0.0	0.0	1.0
Thallium_6	270.0	0.051852	0.222140	0.0	0.0	0.0	0.0	1.0
Thallium_7	270.0	0.385185	0.487543	0.0	0.0	0.0	1.0	1.0

In [46]:

```
one_target = int(np.sum(target))
zero_counter = 0
indices_to_remove = []

for i in range(target.shape[0]):
  if target[i] == 0:
    zero_counter += 1
  if zero_counter > one_target:
    indices_to_remove.append(i)
```

```
print("Indices before balancing data:", target.shape[0])
print("Idices to delete:", len(indices_to_remove))
```

Indices before balancing data: 270
Idices to delete: 54

In [47]:

```
balanced_inputs = inputs.drop(indices_to_remove, axis=0)
balanced_targets = target.drop(indices_to_remove, axis=0)

#reset indices
reset_inputs = balanced_inputs.reset_index(drop=True)
reset_targets = balanced_targets.reset_index(drop=True)

print("Inputs after balancing data:", reset_inputs.shape[0])
print("Targets after balancing data:", reset_targets.shape[0])

balanced_inputs.head()
```

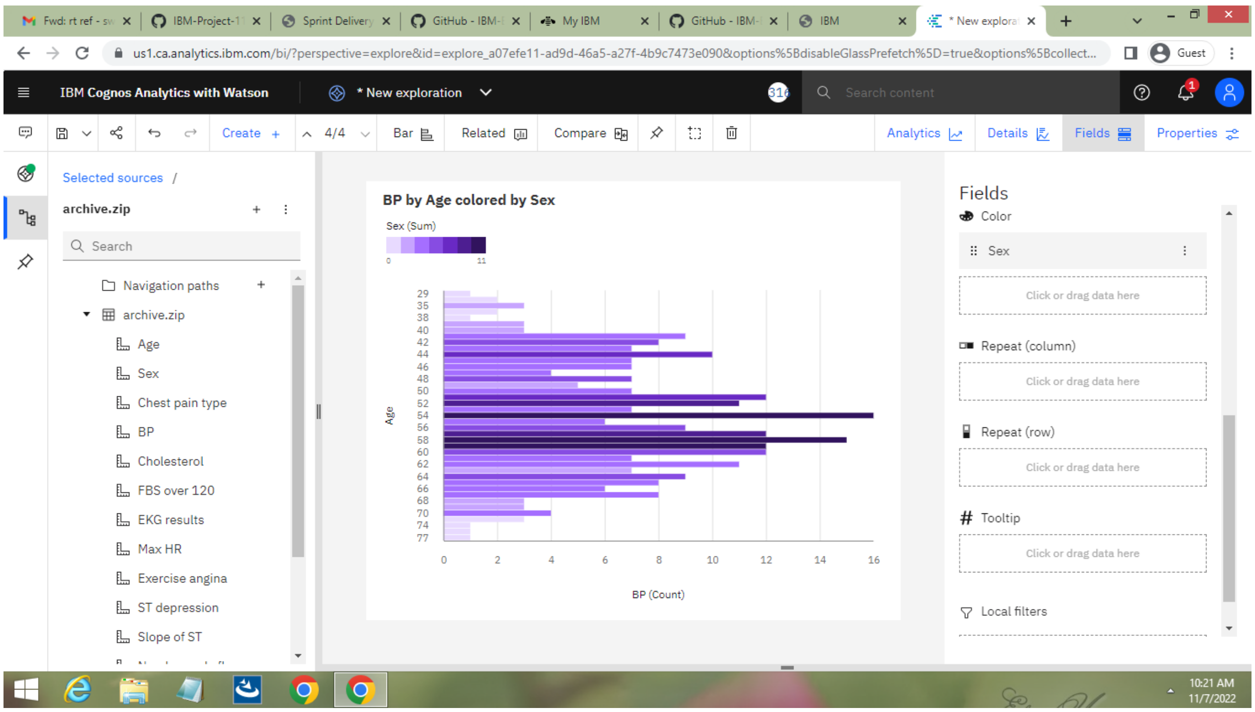
Inputs after balancing data: 216
Targets after balancing data: 216

Out[47]:

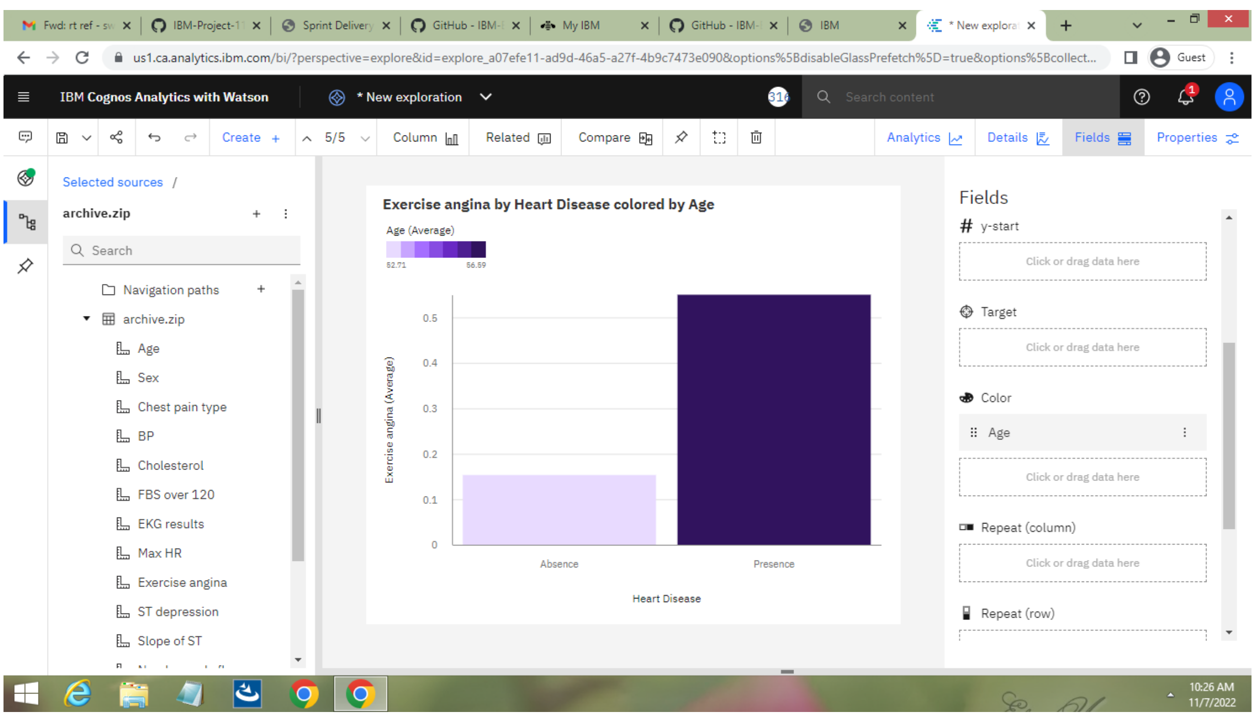
	Age	Sex	Chest pain type	BP	Cholesterol	FBS over 120	EKG results	Max HR	Exercise angina	ST depression	...	Chest pain type_2	Chest pain type_3	Chest pain type_4	EKG results_1
0	70	1	4	130	322	0	2	109	0	2.4	...	0	0	1	0
1	67	0	3	115	564	0	2	160	0	1.6	...	0	1	0	0
2	57	1	2	124	261	0	0	141	0	0.3	...	1	0	0	0
3	64	1	4	128	263	0	0	105	1	0.2	...	0	0	1	0
4	74	0	2	120	269	0	2	121	1	0.2	...	1	0	0	0

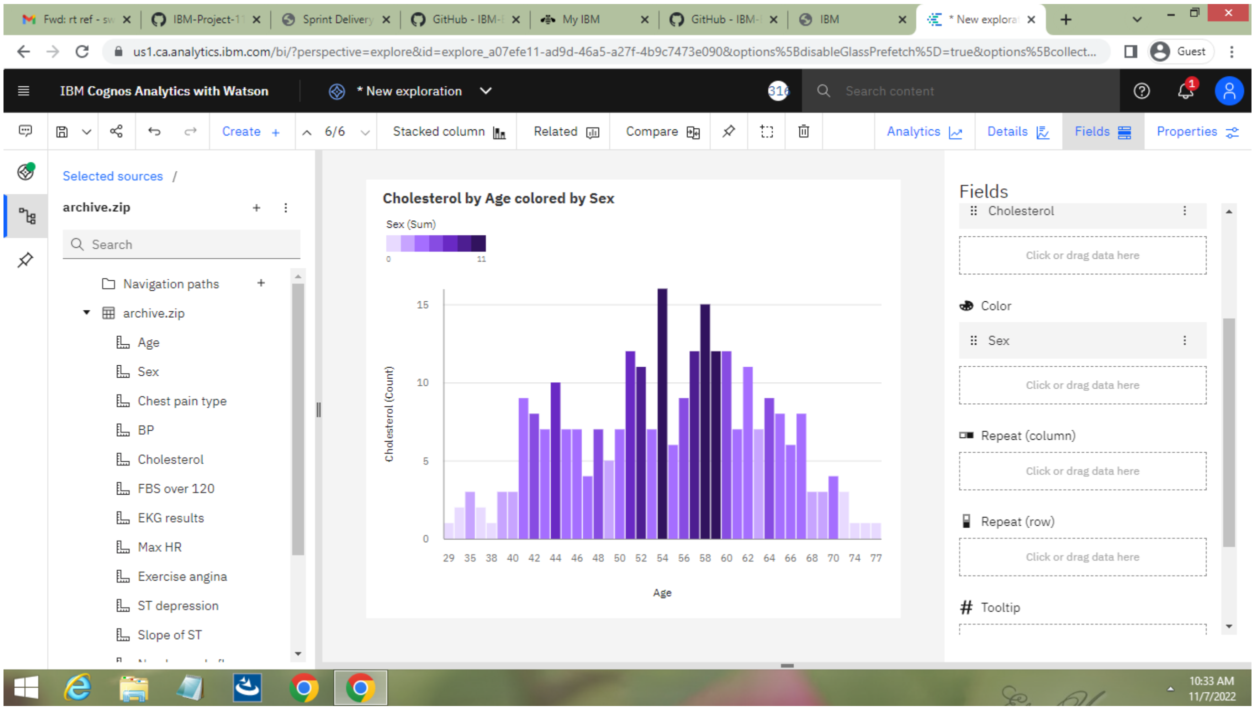
5 rows × 23 columns

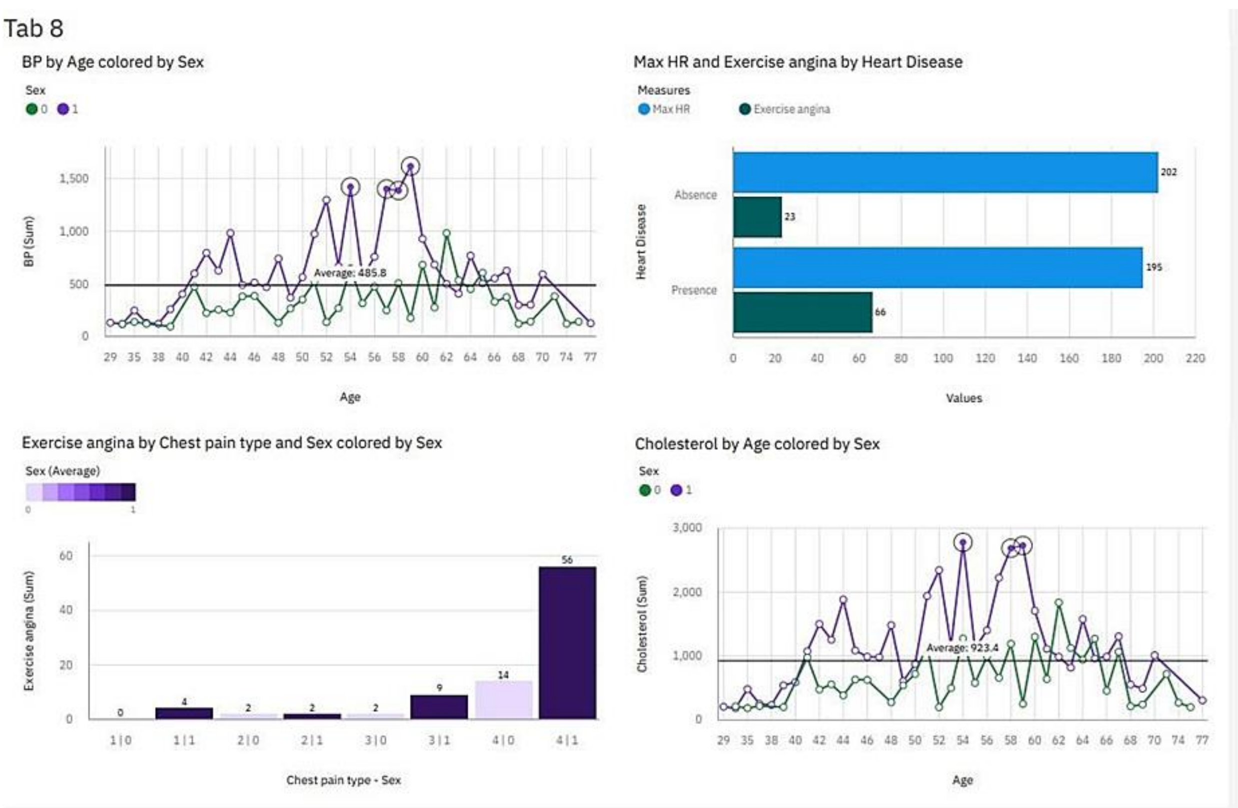
DATABASE SCHEMA




IBM Cognos Analytics with Watson
* New exploration
Selected sources /
archive.zip
Navigation paths
Age
Sex
Chest pain type
BP
Cholesterol
FBS over 120
EKG results
Max HR
Exercise angina
ST depression
Slope of ST
Exercise angina by Chest pain type colored by Sex
Sex (Sum)
Exercise angina (Average)
Chest pain type
Fields
Color
Sex
Repeat (column)
Repeat (row)
Tooltip
Local filters
10:04 AM
11/7/2022


IBM Cognos Analytics with Watson
* New exploration
Selected sources /
archive.zip
Navigation paths
Age
Sex
Chest pain type
BP
Cholesterol
FBS over 120
EKG results
Max HR
Exercise angina
ST depression
Slope of ST
Exercise angina by Heart Disease colored by Age
Age (Average)
Exercise angina (Average)
Absence
Presence
Heart Disease
Fields
y-start
Target
Color
Age
Repeat (column)
Repeat (row)
10:26 AM
11/7/2022





dashboard showing different types of visuals

<u>8. TESTING</u>

<u>8.1 TESTCASES</u>

Testing the data model forvarious input values.

```
In [ ]: from sklearn.metrics import accuracy_score
        input=(63,1,3,145,200,150,98,0,0,0,0,0,0)
        input_as_numpy=np.asarray(input)
        input_reshaped=input_as_numpy.reshape(1,-1)
        pre1=tree_model.predict(input_reshaped)
        print(pre1)
        a1 = accuracy_score(pre1,model1.predict(input_reshaped)) * 100
        print(a1)

        ['Absence']
        100.0

In [ ]: from sklearn.metrics import accuracy_score
        input=(70,1,4,130,322,0,2,109,0,2.4,2,3,3)
        input_as_numpy=np.asarray(input)
        input_reshaped=input_as_numpy.reshape(1,-1)
        pre1=tree_model.predict(input_reshaped)
        print(pre1)
        a1 = accuracy_score(pre1,model1.predict(input_reshaped)) * 100
        print(a1)

        ['Presence']
        100.0
```

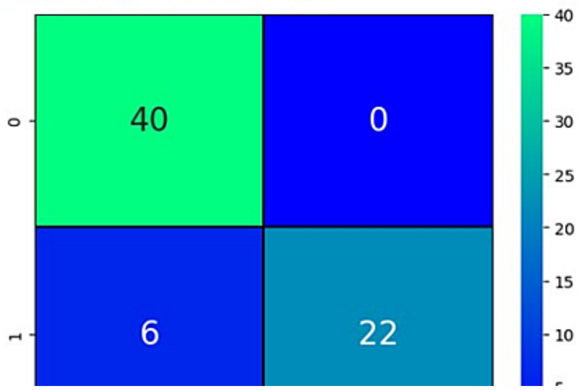
9. RESULT

The confusion matrix below shows the performance metrics of the machine learning model.

9.1PERFORMANCE METRICES

```
from sklearn.model_selection import RandomizedSearchCV
from sklearn.tree import DecisionTreeClassifier

tree_model = DecisionTreeClassifier(max_depth=5,criterion='entropy')
cv_scores = cross_val_score(tree_model, x, y, cv=10, scoring='accuracy')
m=tree_model.fit(x, y)
prediction=m.predict(X_test)
cm= confusion_matrix(y_test,prediction)
sns.heatmap(cm, annot=True,cmap='winter',linewidths=0.3, linecolor='black',annot_kws={"size": 20})
print(classification_report(y_test, prediction))

TP=cm[0][0]
TN=cm[1][1]
FN=cm[1][0]
FP=cm[0][1]
print('Testing Accuracy for Decision Tree:',(TP+TN)/(TP+TN+FN+FP))
print('Testing Sensitivity for Decision Tree:',(TP/(TP+FN)))
print('Testing Specificity for Decision Tree:',(TN/(TN+FP)))
print('Testing Precision for Decision Tree:',(TP/(TP+FP)))
```

```
              precision    recall  f1-score   support

     Absence       0.87      1.00      0.93        40
    Presence       1.00      0.79      0.88        28

    accuracy                           0.91        68
   macro avg       0.93      0.89      0.91        68
weighted avg       0.92      0.91      0.91        68

Testing Accuracy for Decision Tree: 0.9117647058823529
Testing Sensitivity for Decision Tree: 0.8695652173913043
Testing Specificity for Decision Tree: 1.0
Testing Precision for Decision Tree: 1.0
```



10. ADVANTAGES AND DISADVANTAGES

ADVANTAGES

- ✓ User can search for doctor’s help at any point of time.
- ✓ User can talk about their Heart Disease and get instant diagnosis.
- ✓ Doctors get more clients online.
- ✓ Very useful in case of emergency.

DISADVANTAGES

- ✓ Accuracy Issues: A computerized system alone does not ensure accuracy, and the warehouse data is only as good as the data entry that created it.
- ✓ The system is not fully automated, it needs data from user for full diagnosis.

11. CONCLUSION

Complications of heart disease include heart attack and stroke. You can reduce the risk of complications with early diagnosis and treatment.

So the suggestion that we get from the websitemighthelp save patients. It is always to get treated in the early stages of heart disease

12. FUTURE SCOPE

Like the saying goes “Prevention is better than cure”. We have to look into methods to prevent heart diseases altogether other than just predicting it in early stages.

To use this website we need to take a lot of tests beforehand. So it would be better if we require less attributes and still give an effective result

13. APPENDIX

SOURCE CODE

https://github.com/IBM-EPBL/IBM-Project-32959-1660213172/tree/main/FINAL%20DELIVERABLES/DATASET

DEMO LINK:

https://drive.google.com/file/d/1PxfcKR7mt43KsAKosrjweUISrbMaZA8H/view?usp=share_link