A study of How to Detect Heart Disease

Name - Akesh Mark

Date - 09.03.2024

Abstract

Heart disease remains a significant public health concern globally, necessitating accurate detection methods for timely intervention. In this study, a predictive model was developed using an Artificial Neural Network (ANN) to identify individuals at risk of heart disease based on various clinical and demographic features. The model, implemented using the Keras library, achieved impressive accuracy, reaching 100% on both training and validation datasets. The architecture comprised three dense layers with dropout regularization to mitigate overfitting. Model training utilized the Adam optimizer with a binary cross-entropy loss function, demonstrating robust convergence and performance. Through comprehensive evaluation using metrics such as accuracy and confusion matrix analysis, the model exhibited exceptional predictive capability in distinguishing between individuals with and without heart disease. Additionally, feature scaling techniques using MinMaxScaler were applied to normalize input data, contributing to improved model performance. The study findings underscore the potential of deep learning approaches, particularly ANN models, in enhancing heart disease risk prediction and guiding preventive strategies. The developed model and associated preprocessing scalers have been saved for future deployment, facilitating widespread utilization in clinical settings. Overall, this research contributes to advancing predictive modeling techniques for heart disease detection, thereby aiding in early diagnosis and targeted interventions to mitigate cardiovascular risk factorts.

Contents

Abstract 2

Chapter 1: Introduction 6

1.1 Background 6

1.2 Research problem 7

1.3 Research questions 8

1.4 Objectives 8

1.5 Scope of the research 8

1.6 Justification of the Research 9

1.7 Limitations 9

Chapter 2: Literature Review 10

2.1 Introduction to the research theme 10

2.2 Theoretical explanation about the keywords in the topic 10

2.3 Findings by other researchers 11

2.4 Research gap 12

2.5 Variables 13

Chapter 3: Methodology 14

3.1 Introduction 14

3.2 Population Sample and sampling technique 14

3.3 Types of Data to be Collected and data sources 15

3.4 Data Preprocessing and data wrangling 15

3.5 Methods techniques and Tools 15

3.5.1 Artificial Neural Network in Deep Learning 15

3.5.2 Testing for statistical significance 16

3.5.3 Detecting outliers 16

3.5.4 Visualization of Data and Interpretations 16

Chapter 4: Data Analysis 16

4.1 Importing the Dataset 16

4.2 Data preprocessing 19

4.3 Exploratary Data Analysis (EDA) 26

4.3.1 Univariate Analysis 27

4.3.2 Bivariate Analysis 35

4.3.3 Statistical Analysis 41

Chapter 5: Training and Testing the Model 42
Chapter 6: Creating the web application 51
Chapter 7: Conclusion, Discussion and Recommendation 53
7.1 Conclusion 53
7.2 Discussion 54
7.3 Recommendation 54
Appendices 55
List of references 70

List of Figures

Figure 1 importing the dataset 17
Figure 2 head of the dataset. 17
Figure 3 tail of the dataset 18
Figure 4 information of the dataset 18
Figure 5 finding missing values of the dataset. 19
Figure 6 finding outliers of the datset 20
Figure 7 after removing outliers of the dataset 21
Figure 8 renamed the outliers removed columns 22
Figure 9 concatenated dataset 22
Figure 10 after droping the columns with outliers 23
Figure 11 finding missing values of the new dataset 23
Figure 12 after removing null values in the dataset 24
Figure 13 after removing null values from the dataset 24
Figure 14 after renaming and reordering the dataset 25
Figure 15 after converting the data types as before dataset 25
Figure 16 Importing the preprocessed dataset 26
Figure 17 summary of the dataset 26
Figure 18 histograms of the numerical variables 27
Figure 19 barchart of sex variable 28
Figure 20 barchart of cp variable 29
Figure 21 barchart of fbs variable 30
Figure 22 barchart of restecg variable 31
Figure 23 barchart of slope variable 32
Figure 24 barchart of ca variable 33
Figure 25 barchart of thal variable 34
Figure 26 barchart of target variable 35
Figure 27 heatmap of numerical variables 36
Figure 28 barchart of sex vs age 37
Figure 29 barchart of target vs age 38
Figure 30 barchart of fbs vs target 39
Figure 31 barchart of cp vs target 40
Figure 32 barchart of restecg vs target 41
Figure 33 importing the dataset to training 42

Figure 34 converted the dataset into numpy array **43**
Figure 35 dividing the dataset into data and target **44**
Figure 36 normalizing the data and target **45**
Figure 37 dividing the dataset into training and testing **45**
Figure 38 creating the model **46**
Figure 39 training the model **47**
Figure 40 loss functions of the model **47**
Figure 41 accuracy functions of the model **48**
Figure 42 predicting using test data **49**
Figure 43 accuracy of the model **49**
Figure 44 testing the model **49**
Figure 45 saving the model and scaled data and scaled target **50**
Figure 46 testing the model with random data **50**
Figure 47 picture of the web application **52**
Figure 48 picture of results of the web app **53**

Chapter 1: Introduction

1.1 Background

In today's world, a lot of people want a luxurious life with money so they are working like robots to achieve their dreams. While people work as robots they forget to care about their health and themselves. Because of that people's lifestyles had changed. There are so many risk factors such as chest pain, cholesterol, smoking, and unhealthy lifestyle that can affect heart disease. Some of the risks can be controlled by the person. High blood pressure and high sugar levels can affect a person even at a young age because of this lifestyle. These things can lead to cardiovascular diseases. Cardiovascular diseases are also known as heart disease.

It is the world leading cost of death and costs 18.6 million lives annually.

Cardiovascular disease patients have increased more than in the past.

Heart disease has become a major problem in low and middle-income countries. Heart disease is the leading cause of death for white people, black people and Hispanics. There are various types of heart disease such as Arrhythmia, Atherosclerosis, Cardiomyopathy, and Heart infections. Chest pain, slow pulse, coldness in the arms and legs, shortness of breath, and lightheadedness are some of the symptoms of heart disease. Over 17.9 million people died from heart disease in 2019 and most of them were due to heart attack and stroke.

About 80% of deaths from cardiovascular diseases are between the age of 30 and 70 and it can be prevented. It can be prevented by living a healthy lifestyle and avoiding tobacco.

1.2 Research problem

These days a lot of people are living an unhealthy lifestyles. Most people are addicted to smoking, and alcohol drinking and some people are addicted to drugs too. So these things are seriously injurious to health. This has become a major problem in heart disease. Heart disease patients are increasing today because of this. Physical activity is another problem that causes heart disease. There are also some other risk factors that we can't control but they also affect heart disease. Some of them are age, sex, and family history.

Heart disease can be controlled by living a good healthy lifestyle and with medicines and in some cases surgeries. In this study, I will show how risk factors affect heart disease. Some people don't know whether they have heart disease or not. So it is very important to know the symptoms of heart disease so the doctors can give the right treatments quickly.

In this research, I try to predict heart disease using Deep Learning by creating a model and developing an application so that can easily predict the risk level of the people. So this can be very helpful for the people to find whether they have a risk of heart disease. I think this research will help to save lot of people's lives by knowing their risk levels and living according to them. My aim is to reduce the number of heart disease patients in Sri Lanka and save a lot of lives so that can be good for the country's development too.

1.3 Research questions

- What are the most significant risk factors associated with the early diagnosis of heart disease, and how can these risk factors be identified for effective prevention and treatment strategies?

- What is the nature of the relationship between lifestyle factors, genetic predisposition, and environmental factors and how can this understanding inform the development of effective preventative measures for heart disease?

1.4 Objectives

Main objective – Detecting a person having heart disease or not
Other objectives
- Identifying risk factors of heart disease for early diagnosis.
- Understanding the relationship between various factors and heart disease to gain insights into preventative measures.

1.5 Scope of the research

This research is carried out to mainly analyze whether a person has heart disease or not. The dataset was collected via the Internet and it has been used by other researchers too. The dataset provides valuable information that may be associated with heart disease. Therefore Researchers can explore the relationship between these factors and heart disease incidence to identify the most significant risk factors. The research has used Deep Learning techniques like Artificial Neural Network model to predict heart disease. Furthermore, this research has been used to identify the relationship between various factors using statistical tests and how these factors can affect heart disease. The outcome of the predictions can be used to help doctors with their decisions. Sometimes doctors make wrong decisions by their experience and knowledge so this will help them to make correct decisions at

the right time because this model will good accurate results. Also, this research will give an idea for a person as to whether they have the risk of diagnosing heart disease.

1.6 Justification of the Research

Heart disease is one of the leading causes of death worldwide and it is responsible for a significant burden of morbidity and mortality. As previously mentioned the purpose of this research is to detect heart disease as early as possible. This type of research has been done in Europe countries, the USA, and some other countries like Australia. This type of research also have been done in Sri Lanka but I think this research is special because of the results the model predicts. So this research will be very important in Sri Lanka because heart disease has become a major problem in Sri Lanka too. In Sri Lanka, a lot of people are dying from heart disease because of their wrong lifestyle and alcohol and cigarettes. Sri Lanka is not a very good country with a healthcare system. That's why sometimes patients are transferred to different countries. So this research will help Sri Lankans to protect their lives early by looking at the predictions of this research and by the web application. This research will help people to protect themselves and this will help doctors to make correct decisions.

1.7 Limitations

Like most research projects this research has some limitations too. This dataset is from one year and it consists of four countries: Cleveland, Hungary, Switzerland, and Long Beach V. It's better to have a dataset with countries around the world. Also it is better to have dataset in Sri Lanka because I'm doing this research in Sri Lanka. Conclusions would be drawn from this dataset, which contains real-world data. The model is trained by using ANN in Deep Learning and Machine Learning models have not been used to train the model so that is a limitation too because this can be done in Machine Learning too.

Chapter 2: Literature Review

2.1 Introduction to the research theme

The theme of this research is to detect Cardiovascular diseases also known as heart diseases. This research seeks to determine how to reduce heart disease by knowing the patients. Cardiovascular diseases are the ones that cost the most human deaths in today's world. Over three-quarters of CVD deaths take place in low and middle-income countries. As a result, this chapter reviews the previous studies and tries to give the maximum to protect human lives from this heart disease.

2.2 Theoretical explanation about the keywords in the topic

Since this is a familiar topic to people this does not have many complex words.
There is an explanation of some of the important keywords below. Some other keywords might be added throughout this research.

Keyword	Theoretical explanation
Cardiovascular disease	This is a general term for conditions affecting the heart and blood vessels.
Diagnosis	The identification of the nature of an illness or other problem by examination of the symptoms
Thalach	The person's maximum heart rate achieved
Patients	A person registered to receive medical treatment for heart disease

Treatments	Medical care is given to the person for illness
Behaviours	A person's behaviour that can cause heart disease

2.3 Findings by other researchers

Numerous studies have been done that focus on the diagnosis of heart disease. They have applied different data mining techniques and machine learning algorithms for the diagnosis of heart disease and they achieved different probabilities.

(Voon Khai Tick, Ng Yung Meeng, Nur Farahiyah Mohammad, Nor Hazlyna Harun, Hiam Alquran and Mohamad Farhan Mohamad Mohsin 1997) has done a research using ANN in Deep Learning and they have normalized the data into floating points and they have divided the dataset into training 70% and testing 30%. And they have tested the training with different learning rates 0.25,0.5,0.75,1.0 and they have found that 0.25 is the best learning rate for them. They have trained this by 25 neurons and 1000 epochs. This model got 80.26% accuracy.

(Raniya R. Sarra, Ahmed Musa Dinar, Mazin Abed Mohammed 2023) has done a research using ANN. The dataset have been devided into 242 training and 61 testing. Then the dataset have been again dived the training set into 8:2 ration for training and validation. This model has got 93.44% accuracy. In this model they have used 2 layers and there were 30 units in the input layer, which was also the first hidden layer.

(Polaraju, Durga Prasad, Tech Scholar 2017) have done a prediction of heart disease using a multiple regression model. This work is done by using training data set consisting of 3000 instances with 13 different attributes. The data have been divided into two parts 70% of the data are training data and 30% data have been used for testing.

(Hlaudi Daniel Masthe,Mosima Anna Masthe 2014) have researched heart disease. The researchers used pattern recognition and data mining methods in the domain of cardiovascular diagnoses. The purpose of predictions in data mining is to help discover trends in patient data to improve their health. The researchers used data mining algorithms decision trees, naïve Bayes, neural networks, association classification and genetic algorithms for predicting and analyzing heart disease from the dataset.

(Purushottam, Saxena, and Sharma 2016) have done a heart prediction using data mining. This helps medical practitioner to make effective decision making based on certain parameters. Training and testing provide 86.3% accuracy in the testing phase and 87.3% in the training phase.

(Yangguang He, Xinlong Li, Ruixian Song) have done a heart disease project to predict whether a person has heart disease. Different methods have been used in this research such as logistic regression, SVM, naïve Bayes, random forest and artificial neural work. (Beyene and Kamat 2018) researched heart disease. This research targets one of the most common problems in medical centres. It is about all experts do not have equal knowledge to treat their patients so they give their own decision that may give poor results. Machine learning algorithms like decision trees, naïve Bayes, k-nearest neighbour, support vector machine and artificial neural networks have been used in this research.

2.4 Research gap

In this research I have used Artificial Neural Network in Deep Learning to predict heart disease using important features. That has been very successful because the model has 100% accuracy for the validation data and 99% accuracy for the training data. So I think this model will give

incredible results and this will give good help for the doctors. Also I have develop a heart risk level predictor application so I think that is also a plus point in this research. Also I have used statistical tests like t-test,chi-squared test,anova test to determine the relationship between the variables. Most of the research don't have used this statistical tests to identify relationships and that is an advantage in this research.

2.5 Variables

Variable	Description	Values
Age	Patient's age in years	Continuous value
Sex	Sex of patient	1=Male 0=Female
Cp	Chest pain	0:typical angina 1:atypical angina 2:non-angina pain 3:asymptomatic
Trestbps	Resting blood pressure	Continuous value in mm/Hg
Chol	Serum cholesterol in mg/dl	Continuous value in mg/dl
Fbs	Fasting blood sugar	1>=120mg/dl 0<=120mg/dl
Restcg	Resting electrocardiographic results	0=normal 1=having_ST_T wave abnormal 2=left ventricular hypertrophy
Thalach	Maximum heart rate achieved	Continuous value
Exang	Exercise induced angina	1:yes 0:no
Oldpeak	ST depression induced by exercise relative to rest	Continuous value
Slope	The slope of the peak exercise ST segment	0:upsloping 1:flat 2:down sloping
Ca	Number of major vessels colored by fluoroscopy	0-4 value

Thal	Defect type	0=normal 1=fixed defect 2=reversable defect 3=another effect
Target	Diagnosis of heart diseases	1:yes 0:no

Chapter 3: Methodology

3.1 Introduction

In this study, we're looking closely at information about heart disease that we found on Kaggle, a website where researchers share data. Our main goal is to detect people that are having heart disease and understand what things might be related to heart disease and how we can find it early to help stop it from getting worse. We're using a computer program called Python to sort through the data and figure out what's important. Then, we'll make graphs and pictures to help us see the information better. Our hope is that by doing this, we can learn more about heart disease and maybe find new ways to prevent it from happening in the first place, which would be really good for people's health.

3.2 Population Sample and sampling technique

Out of the entire population, this dataset is a sample for this study.
Because the population for this study will be large and it is very hard to deal with population data. So systematic sampling procedures will be employed. As a result of sampling, precision must be at a high level.

3.3 Types of Data to be Collected and data sources

This dataset was obtained from Kaggle as mentioned earlier. Since this dataset is not collected by me this is not a primary dataset. It means that this is already available on the internet so this is secondary data. This dataset has numerical values and categorical values so it will be easy to analyze. The dataset is in CSV format.

3.4 Data Preprocessing and data wrangling

Data cleaning is very important in data preprocessing. Data cleaning techniques are used to clean the dataset, before going through with the analysis. Data cleaning is very crucial because it gives a high-quality research study. Some data-cleaning techniques would be used in this research to find missing and null values and remove duplicate data and removing outliers. It is important to mention that this is a secondary dataset that is collected by someone else. The file is in a CSV format, making it easy to understand. Normalization of the data has been done before training the model in ANN.

3.5 Methods techniques and Tools

3.5.1 Artificial Neural Network in Deep Learning

In our research on heart disease, we use a special type of computer program called Artificial Neural Networks (ANNs). These networks are part of a bigger system called deep learning, which helps us find hidden patterns in lots of data. By training these networks with information like age, cholesterol levels, and lifestyle habits, we can predict if someone might have heart disease. These programs learn from examples and get better at making predictions over time. With their help, we aim to understand and detect heart disease more accurately, which is the main goal of our research.

3.5.2 Testing for statistical significance

The first step in conducting a statistical significance is to state the null hypothesis and alternative hypothesis. The second step is selecting the alpha value. After that, we have to choose a test like a t-test, ANOVA test, or chi-square test to compute the statistical significance. In the final step, we can interpret the results by rejecting the null hypothesis or alternative hypothesis.

3.5.3 Detecting outliers

Inter quartile range is the method to identify an outlier. Calculating outliers is very important because descriptive statistics methods such as mean, correlation coefficient, and the standard deviation is sensitive to outliers.

3.5.4 Visualization of Data and Interpretations

The data will be visualized using charts and various methods. So the users of this study can get a good idea along with the interpretation. Matplotlib, seaborn, and other packages will be used to visualize this study.

Chapter 4: Data Analysis

4.1 Importing the Dataset

First, The Pandas library and the dataset have been imported into the the jupyter notebook. Jupyter notebook is a platform that is specially used for data analytics, Machine Learning, Deep Learning and python and for other data science related things. You can see a picture of the dataset below in Figure 1.

	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
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
1020	59	1	1	140	221	0	1	164	1	0.0	2	0	2	1
1021	60	1	0	125	258	0	0	141	1	2.8	1	1	3	0
1022	47	1	0	110	275	0	0	118	1	1.0	1	1	2	0
1023	50	0	0	110	254	0	0	159	0	0.0	2	0	2	1
1024	54	1	0	120	188	0	1	113	0	1.4	1	1	3	0

1025 rows × 14 columns

Figure 1 importing the dataset

Then we can see the first 10 rows and the last 10 rows of the dataset from the following Figure 2 and Figure 3

```
data.head(10)
```

	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
5	58	0	0	100	248	0	0	122	0	1.0	1	0	2	1
6	58	1	0	114	318	0	2	140	0	4.4	0	3	1	0
7	55	1	0	160	289	0	0	145	1	0.8	1	1	3	0
8	46	1	0	120	249	0	0	144	0	0.8	2	0	3	0
9	54	1	0	122	286	0	0	116	1	3.2	1	2	2	0

Figure 2 head of the dataset.

```
data.tail(10)
```

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
1015	58	1	0	128	216	0	0	131	1	2.2	1	3	3	0
1016	65	1	3	138	282	1	0	174	0	1.4	1	1	2	0
1017	53	1	0	123	282	0	1	95	1	2.0	1	2	3	0
1018	41	1	0	110	172	0	0	158	0	0.0	2	0	3	0
1019	47	1	0	112	204	0	1	143	0	0.1	2	0	2	1
1020	59	1	1	140	221	0	1	164	1	0.0	2	0	2	1
1021	60	1	0	125	258	0	0	141	1	2.8	1	1	3	0
1022	47	1	0	110	275	0	0	118	1	1.0	1	1	2	0
1023	50	0	0	110	254	0	0	159	0	0.0	2	0	2	1
1024	54	1	0	120	188	0	1	113	0	1.4	1	1	3	0

Figure 3 tail of the dataset

The information of the dataset variables can be seen in the following Figure 4.

```
data.info()
```

```
<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
```

Figure 4 information of the dataset

4.2 Data preprocessing

Data preprocessing is an important step in data analysis because that involves preparing the data for modeling or analysis. If the dataset is not properly preprocessed then the predictions will give errors. In this research, preprocessing is used to ensure that the data is consistent, complete, and accurate and that is in a format suitable for analysis.

First thing is to find if there are any missing values in this dataset.

```
: 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
```

Figure 5 finding missing values of the dataset.

According to the Figure 5 we can see that there are no missing values in this dataset. So there is no need to drop or fill missing values. Then I have created a boxplot for detecting outliers in the numerical variables.

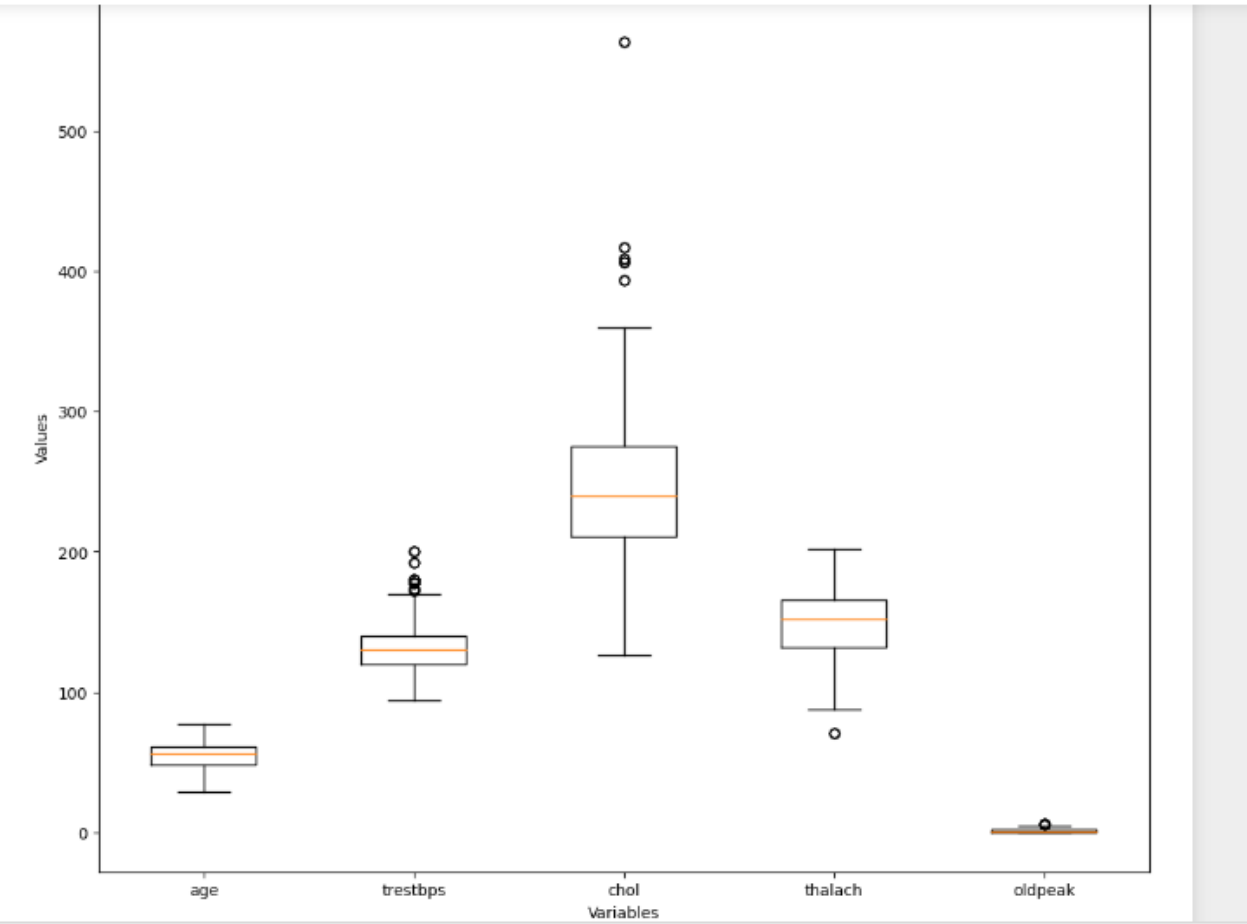

Figure 6 finding outliers of the datset

According to the Figure 6 we can see that there are some outliers in the numerical variables of the dataset. Then I have removed the outliers of the dataset. That can be seen in the following Figure 7.

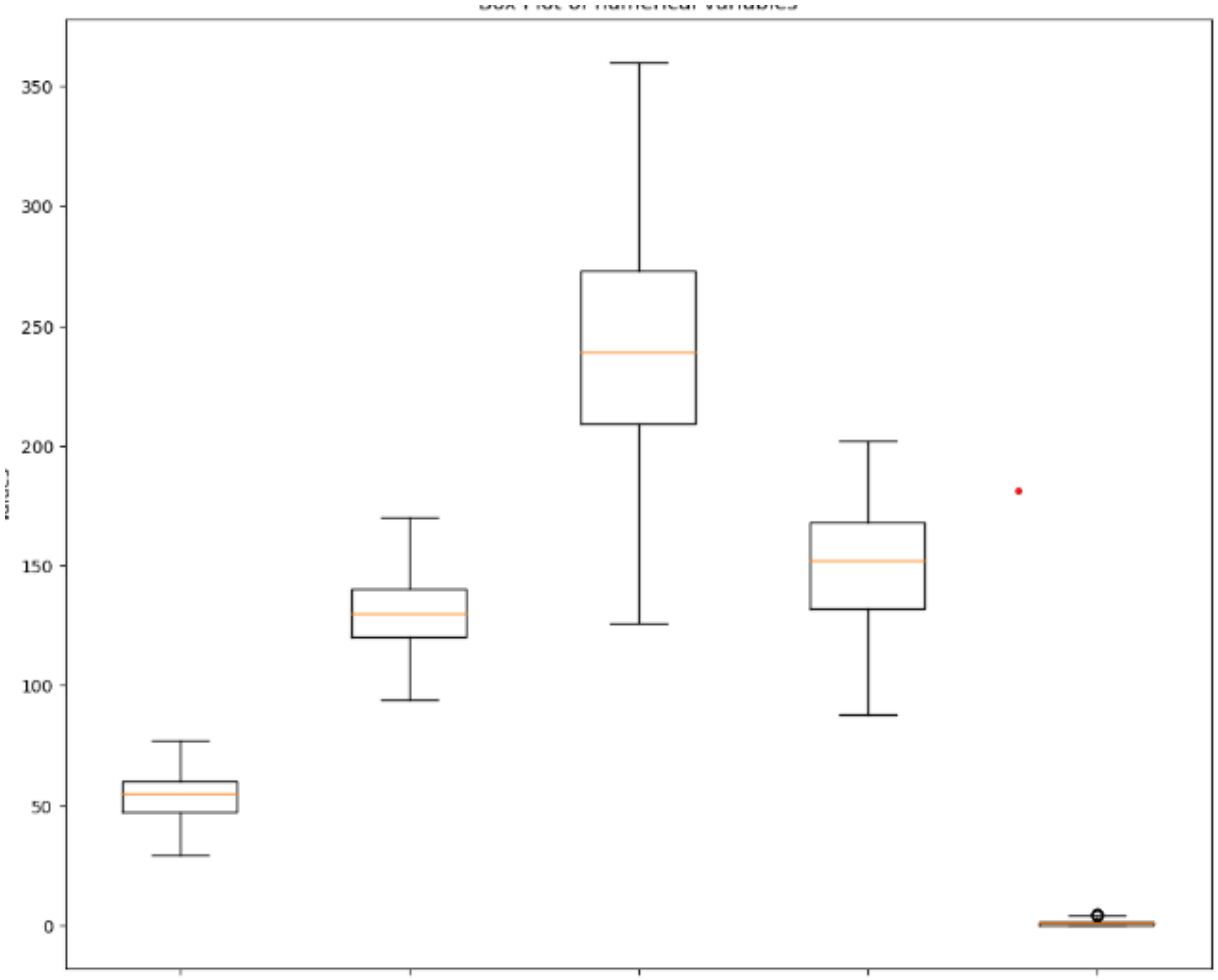

Figure 7 after removing outliers of the dataset

Then I have renamed the outliers removed columns to new names that can be seen in the below Figure 8.

	age_new	trestbps_new	chol_new	thalach_new	oldpeak_new
0	52	125	212	168	1.0
1	53	140	203	155	3.1
2	70	145	174	125	2.6
3	61	148	203	161	0.0
4	62	138	294	106	1.9
...	...	...	...	...	...
1020	59	140	221	164	0.0
1021	60	125	258	141	2.8
1022	47	110	275	118	1.0
1023	50	110	254	159	0.0
1024	54	120	188	113	1.4

968 rows × 5 columns

Figure 8 renamed the outliers removed columns

After that I have joined the outliers removed dataset with the original dataset by using the 'concat' function in the pandas.

]:

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target	age_new	trestbps_new	chol_new	thalach_new	oldpeak_new
0	52	1	0	125	212	0	1	168	0	1.0	2	2	3	0	52.0	125.0	212.0	168.0	1.0
1	53	1	0	140	203	1	0	155	1	3.1	0	0	3	0	53.0	140.0	203.0	155.0	3.1
2	70	1	0	145	174	0	1	125	1	2.6	0	0	3	0	70.0	145.0	174.0	125.0	2.6
3	61	1	0	148	203	0	1	161	0	0.0	2	1	3	0	61.0	148.0	203.0	161.0	0.0
4	62	0	0	138	294	1	1	106	0	1.9	1	3	2	0	62.0	138.0	294.0	106.0	1.9
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
1020	59	1	1	140	221	0	1	164	1	0.0	2	0	2	1	59.0	140.0	221.0	164.0	0.0
1021	60	1	0	125	258	0	0	141	1	2.8	1	1	3	0	60.0	125.0	258.0	141.0	2.8
1022	47	1	0	110	275	0	0	118	1	1.0	1	1	2	0	47.0	110.0	275.0	118.0	1.0
1023	50	0	0	110	254	0	0	159	0	0.0	2	0	2	1	50.0	110.0	254.0	159.0	0.0
1024	54	1	0	120	188	0	1	113	0	1.4	1	1	3	0	54.0	120.0	188.0	113.0	1.4

025 rows × 19 columns

Figure 9 concatenated dataset

Now the dataset have duplicate columns. So we have to remove the numerical variables with outliers. So I have droped the columns that are having outliers. Then the dataset can be seen as according to the following Figure 10.

	sex	cp	fbs	restecg	exang	slope	ca	thal	target	age_new	trestbps_new	chol_new	thalach_new	oldpeak_new
0	1	0	0	1	0	2	2	3	0	52.0	125.0	212.0	168.0	1.0
1	1	0	1	0	1	0	0	3	0	53.0	140.0	203.0	155.0	3.1
2	1	0	0	1	1	0	0	3	0	70.0	145.0	174.0	125.0	2.6
3	1	0	0	1	0	2	1	3	0	61.0	148.0	203.0	161.0	0.0
4	0	0	1	1	0	1	3	2	0	62.0	138.0	294.0	106.0	1.9
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
1020	1	1	0	1	1	2	0	2	1	59.0	140.0	221.0	164.0	0.0
1021	1	0	0	0	1	1	1	3	0	60.0	125.0	258.0	141.0	2.8
1022	1	0	0	0	1	1	1	2	0	47.0	110.0	275.0	118.0	1.0
1023	0	0	0	0	0	2	0	2	1	50.0	110.0	254.0	159.0	0.0
1024	1	0	0	1	0	1	1	3	0	54.0	120.0	188.0	113.0	1.4

1025 rows × 14 columns

Figure 10 after droping the columns with outliers

After that I have checked whether there are null values in the new dataset because there can be missing values.

```
sex              0
cp               0
fbs              0
restecg          0
exang            0
slope            0
ca               0
thal             0
target           0
age_new         57
trestbps_new    57
chol_new        57
thalach_new     57
oldpeak_new     57
dtype: int64
```

Figure 11 finding missing values of the new dataset

According the above Figure 11 we can see that there are now missing values in the dataset. That is because I have removed outliers in the numerical variables and the I have concatenated those variables with the original dataset. Now we have to drop the null values in this dataset.

```
]: sex              0
   cp               0
   fbs              0
   restecg          0
   exang            0
   slope            0
   ca               0
   thal             0
   target           0
   age_new          0
   trestbps_new     0
   chol_new         0
   thalach_new      0
   oldpeak_new      0
   dtype: int64
```

Figure 12 after removing null values in the dataset

According to the above Figure 12 we can see that the null values from the dataset have been removed. Then I have reordered the columns as to the order of the previous dataset and I have renamed the numerical variables also according to the previous dataset and changed the data types of them to previous dataset. That can be seen in the following Figures.

	age_new	sex	cp	trestbps_new	chol_new	fbs	restecg	thalach_new	exang	oldpeak_new	slope	ca	thal	target
0	52.0	1	0	125.0	212.0	0	1	168.0	0	1.0	2	2	3	0
1	53.0	1	0	140.0	203.0	1	0	155.0	1	3.1	0	0	3	0
2	70.0	1	0	145.0	174.0	0	1	125.0	1	2.6	0	0	3	0
3	61.0	1	0	148.0	203.0	0	1	161.0	0	0.0	2	1	3	0
4	62.0	0	0	138.0	294.0	1	1	106.0	0	1.9	1	3	2	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
1020	59.0	1	1	140.0	221.0	0	1	164.0	1	0.0	2	0	2	1
1021	60.0	1	0	125.0	258.0	0	0	141.0	1	2.8	1	1	3	0
1022	47.0	1	0	110.0	275.0	0	0	118.0	1	1.0	1	1	2	0
1023	50.0	0	0	110.0	254.0	0	0	159.0	0	0.0	2	0	2	1
1024	54.0	1	0	120.0	188.0	0	1	113.0	0	1.4	1	1	3	0

968 rows × 14 columns

Figure 13 after removing null values from the dataset

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
0	52.0	1	0	125.0	212.0	0	1	168.0	0	1.0	2	2	3	0
1	53.0	1	0	140.0	203.0	1	0	155.0	1	3.1	0	0	3	0
2	70.0	1	0	145.0	174.0	0	1	125.0	1	2.6	0	0	3	0
3	61.0	1	0	148.0	203.0	0	1	161.0	0	0.0	2	1	3	0
4	62.0	0	0	138.0	294.0	1	1	106.0	0	1.9	1	3	2	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
1020	59.0	1	1	140.0	221.0	0	1	164.0	1	0.0	2	0	2	1
1021	60.0	1	0	125.0	258.0	0	0	141.0	1	2.8	1	1	3	0
1022	47.0	1	0	110.0	275.0	0	0	118.0	1	1.0	1	1	2	0
1023	50.0	0	0	110.0	254.0	0	0	159.0	0	0.0	2	0	2	1
1024	54.0	1	0	120.0	188.0	0	1	113.0	0	1.4	1	1	3	0

968 rows × 14 columns

Figure 14 after renaming and reordering the dataset

	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
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
1020	59	1	1	140	221	0	1	164	1	0.0	2	0	2	1
1021	60	1	0	125	258	0	0	141	1	2.8	1	1	3	0
1022	47	1	0	110	275	0	0	118	1	1.0	1	1	2	0
1023	50	0	0	110	254	0	0	159	0	0.0	2	0	2	1
1024	54	1	0	120	188	0	1	113	0	1.4	1	1	3	0

968 rows × 14 columns

Figure 15 after converting the data types as before dataset

After doing the data preprocessing part I have saved the new dataset in a csv format so that can be easy for the future use. After that I have imported the preprocessed dataset into the jupyter notebook.

	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
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
963	59	1	1	140	221	0	1	164	1	0.0	2	0	2	1
964	60	1	0	125	258	0	0	141	1	2.8	1	1	3	0
965	47	1	0	110	275	0	0	118	1	1.0	1	1	2	0
966	50	0	0	110	254	0	0	159	0	0.0	2	0	2	1
967	54	1	0	120	188	0	1	113	0	1.4	1	1	3	0

968 rows × 14 columns

Figure 16 Importing the preprocessed dataset

4.3 Exploratary Data Analysis (EDA)

In this section I have done a summary of the dataset variables, univariate analysis, bivariate analysis, and statistical analysis. In the following figure shows the summary of the dataset variables.

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	
count	968.000000	968.000000	968.000000	968.000000	968.000000	968.000000	968.000000	968.000000	968.000000	968.000000	968.000000	968.000000	968.00
mean	54.073347	0.715909	0.960744	129.891529	242.448347	0.140496	0.530992	149.466942	0.327479	1.000310	1.410124	0.747934	2.30
std	9.127036	0.451213	1.029306	15.323725	45.264264	0.347680	0.521582	22.682671	0.469536	1.072853	0.605200	1.025273	0.62
min	29.000000	0.000000	0.000000	94.000000	126.000000	0.000000	0.000000	88.000000	0.000000	0.000000	0.000000	0.000000	0.00
25%	47.000000	0.000000	0.000000	120.000000	209.000000	0.000000	0.000000	132.000000	0.000000	0.000000	1.000000	0.000000	2.00
50%	55.000000	1.000000	1.000000	130.000000	239.000000	0.000000	1.000000	152.000000	0.000000	0.800000	1.000000	0.000000	2.00
75%	60.250000	1.000000	2.000000	140.000000	273.000000	0.000000	1.000000	168.000000	1.000000	1.600000	2.000000	1.000000	3.00
max	77.000000	1.000000	3.000000	170.000000	360.000000	1.000000	2.000000	202.000000	1.000000	4.400000	2.000000	4.000000	3.00

Figure 17 summary of the dataset

According to the above Figure 17 we can see the mean, median, max value and other things related to the variables.

4.3.1 Univariate Analysis

4.3.1.1 Numerical variables

In this study univariate analysis have been done for the numerical variables. Histograms have been used for this analysis.

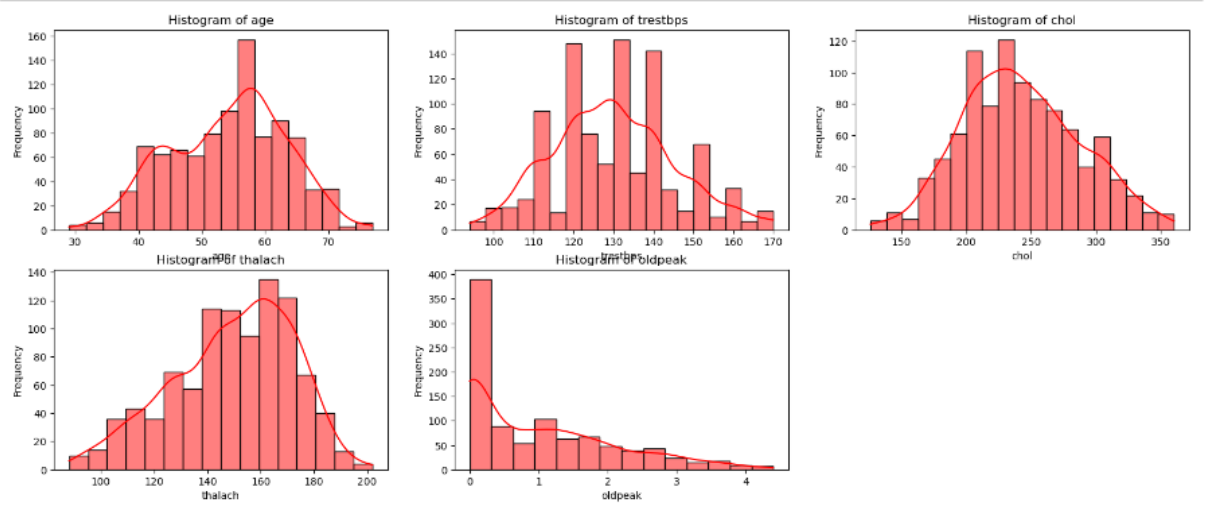

Figure 18 histograms of the numerical variables

According to the above Figure 18 shows the distribution of the numerical variables of the dataset. The histogram of the age shows that a lot of people are in the age of 55-58 and a small distribution after the age of 70. According to the trestbps histogram most of the people have a trestbps level of 130. A small distribution for the trestbps level of below 100. According to the chol histogram most of the people are in the cholesterol level of 200-250. After the 250 chol level the distribution is going down. Most of the people having heart rate of 160-170 according to the thalach histogram chart. A lot of people are in oldpeak=0 level and that is a good sign and that can be seen in the oldpeak histogram.

4.3.1.2 Categorical variables

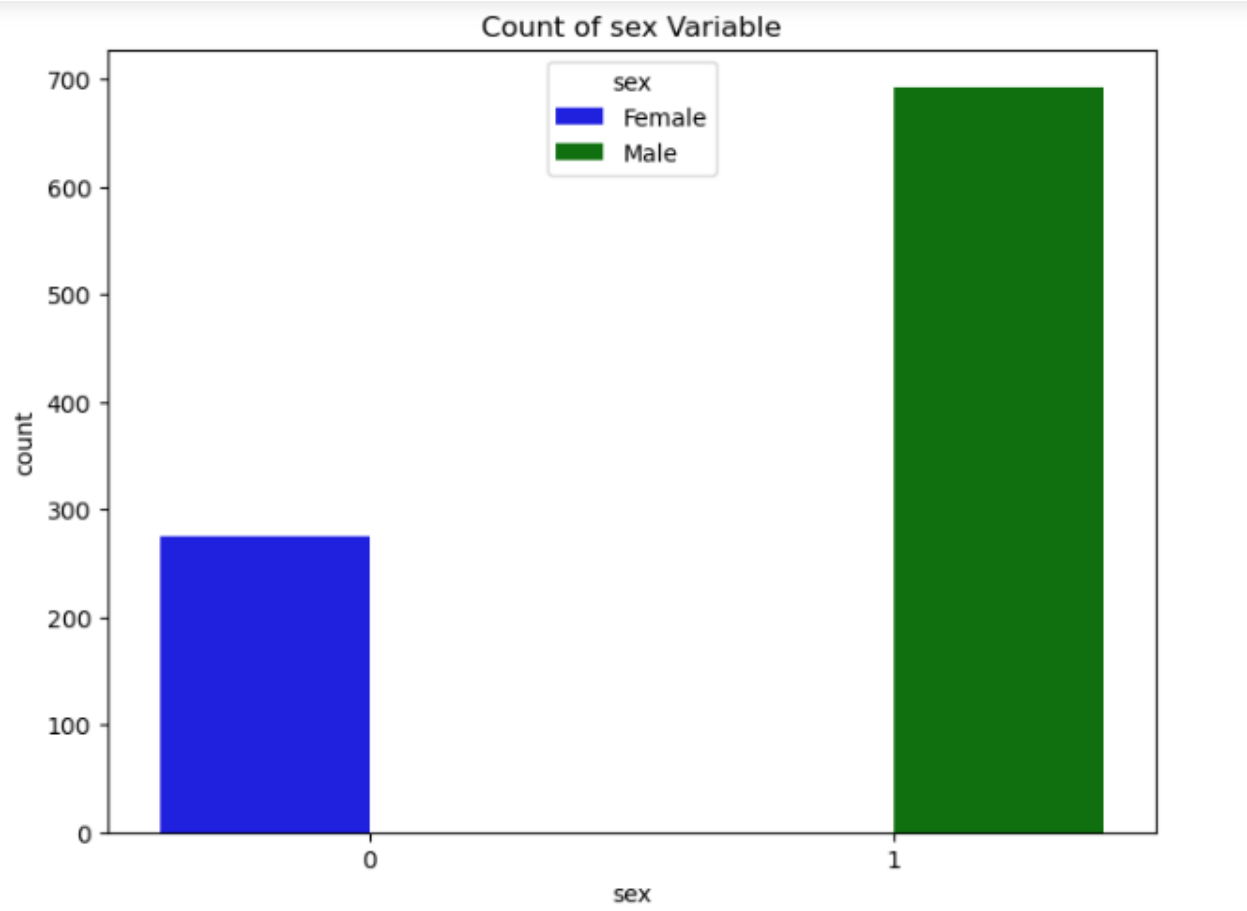

Figure 19 barchart of sex variable

The above Figure 19 shows the distribution of the sex variable. Here 0 represents female and 1 represents 1. So we can see there are lot of males in this dataset according to the above Figure 19.

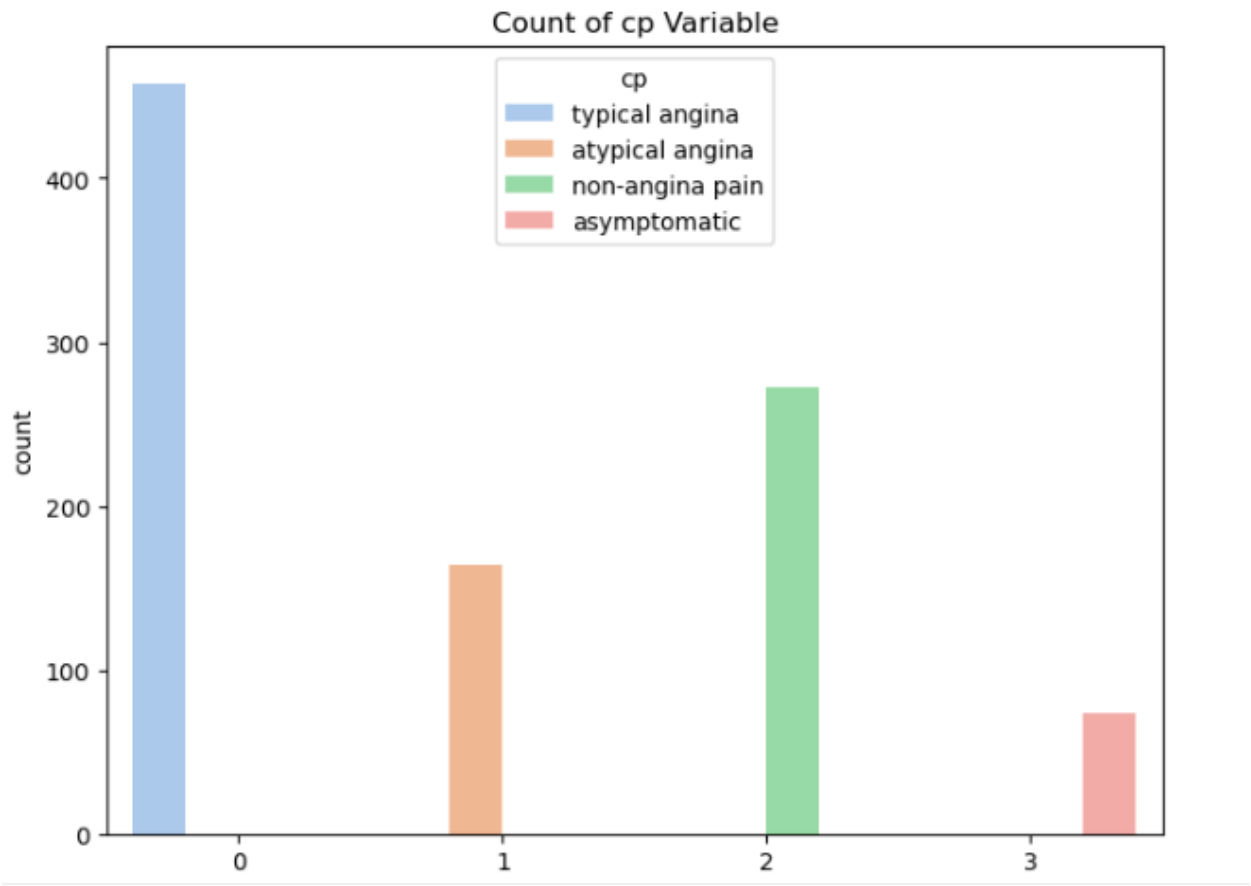

Figure 20 barchart of cp variable

The above Figure 20 shows the distribution of the cp variable. Most of the the people are in the cp=0 category and less number of people are in the cp=3 category.

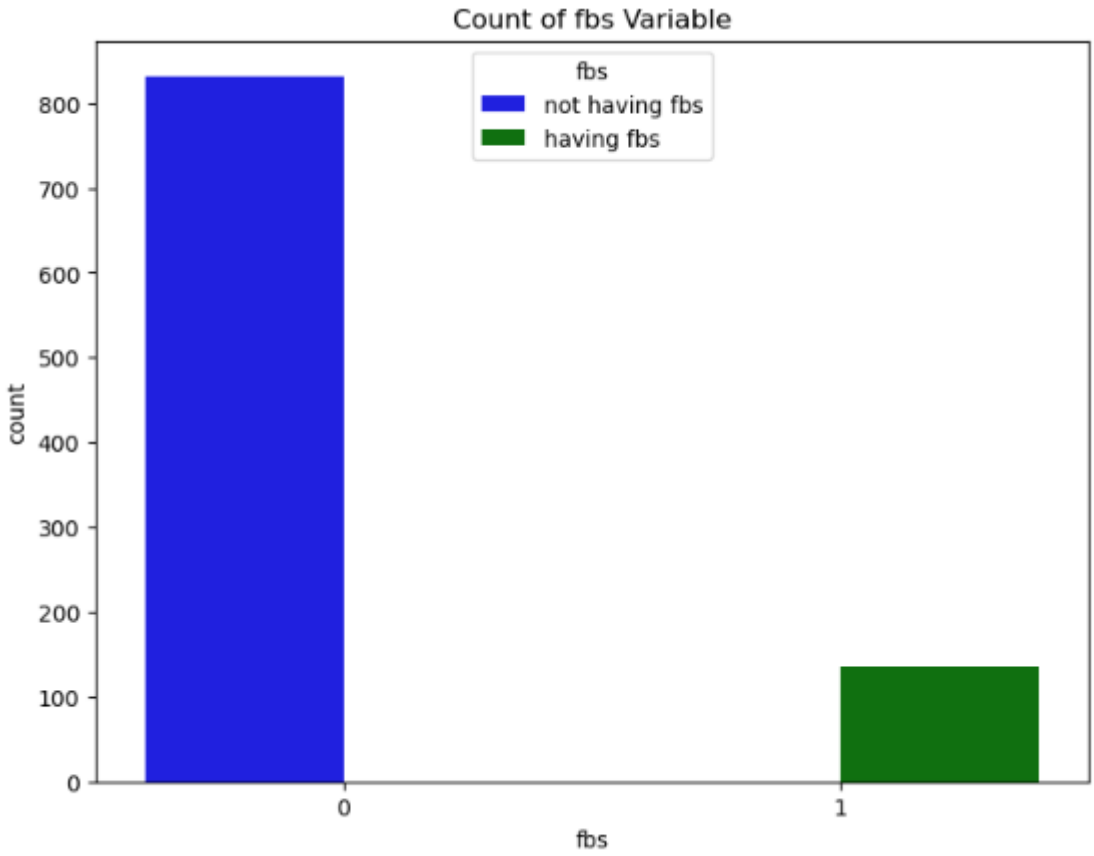

Figure 21 barchart of fbs variable

The above Figure 21 shows that most of the people in this dataset don't have fasting blood sugar.

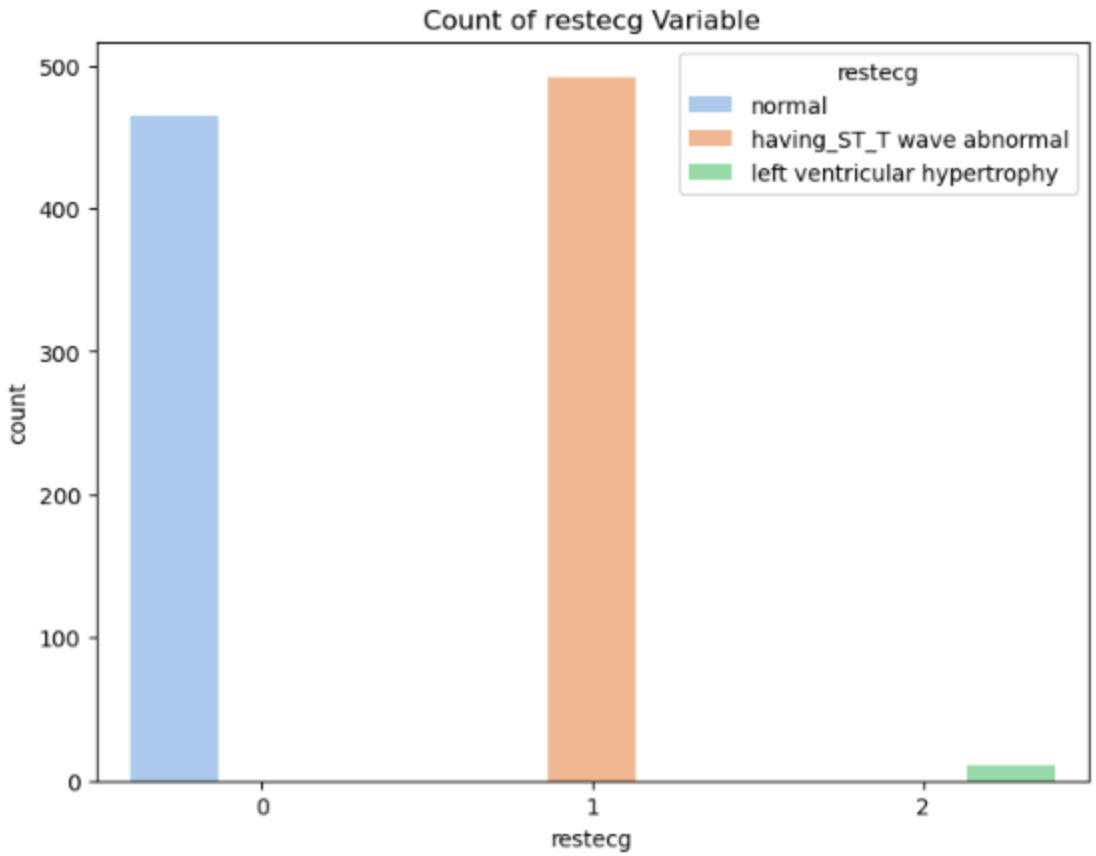

Figure 22 barchart of restecg variable

According to the above Figure 22 shows that there are a small number of people have left ventricular hypertrophy.

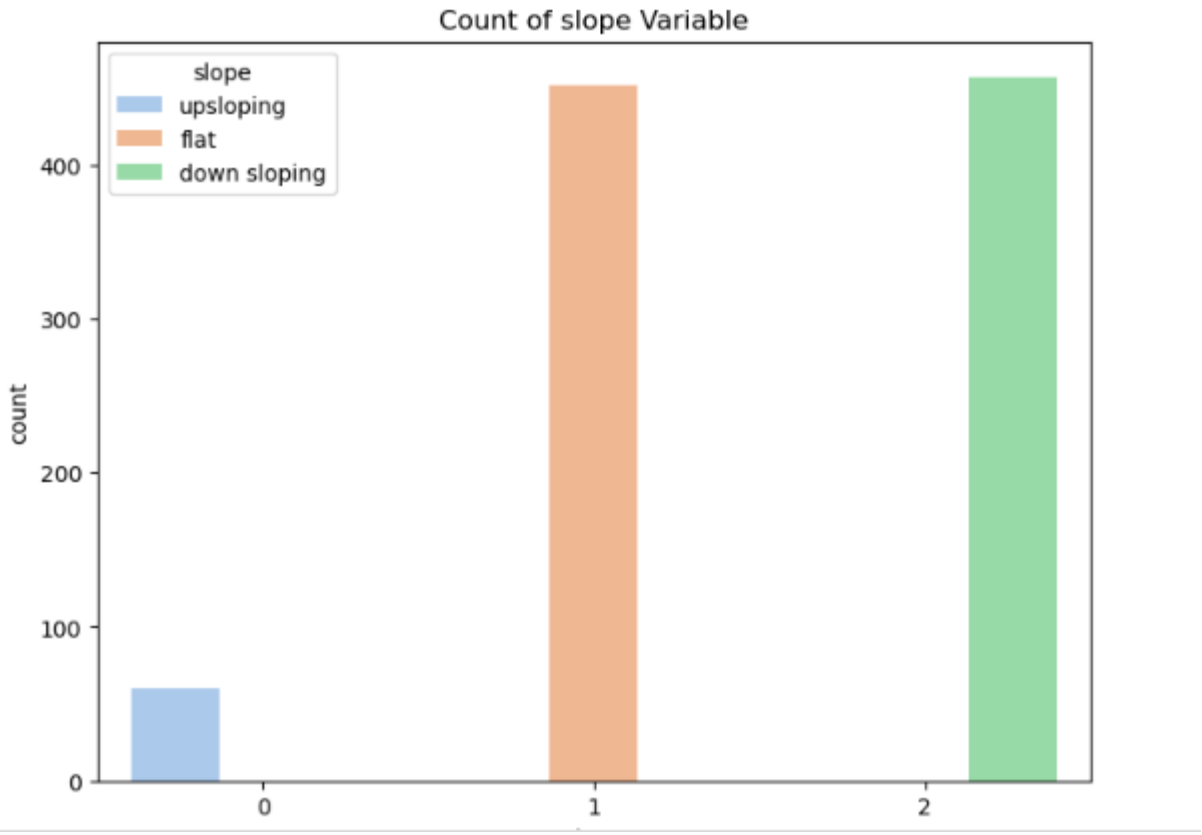

Figure 23 barchart of slope variable

Slope=1 and slope=2 are having a similar distribution and slope=0 is a small distribution according to others. That can be seen in the above Figure 23.

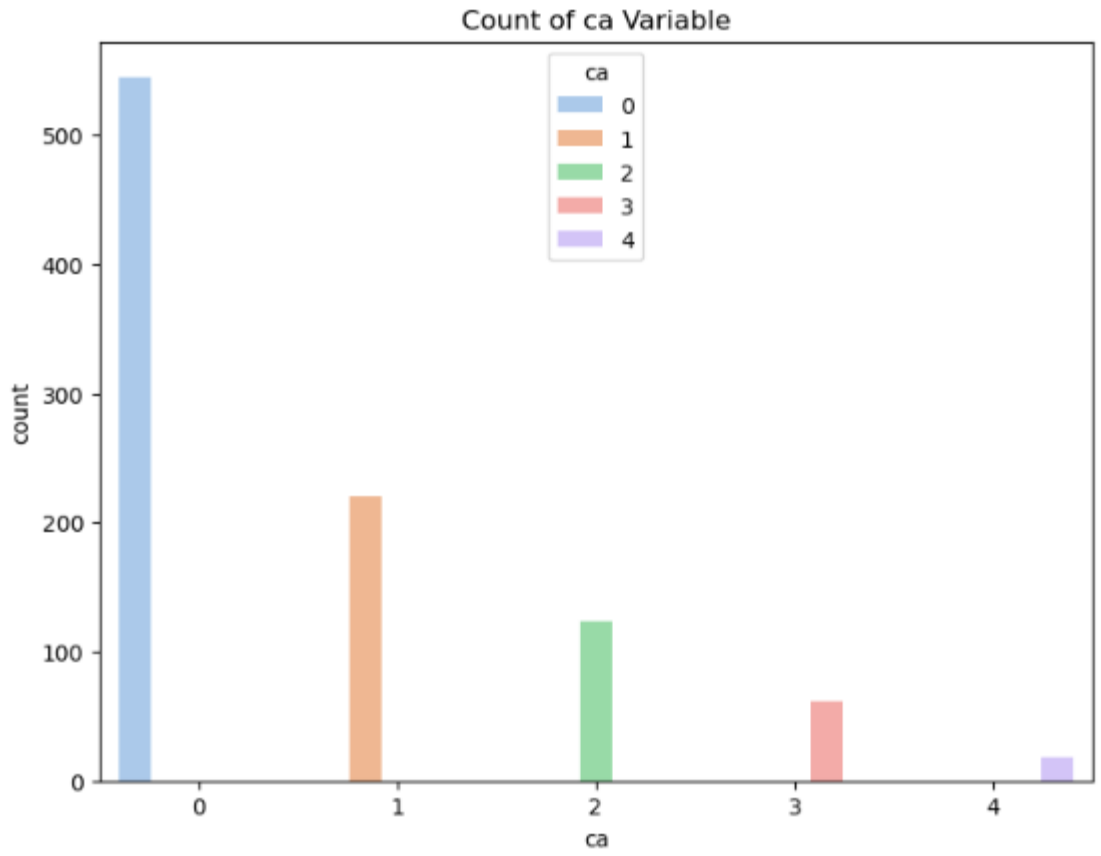

Figure 24 barchart of ca variable

This Figure 24 shows the the distribution of the ca variable. This is about the number of major vessels colored by fluoroscopy. Ca =0 has the highest distribution and the ca=4 has the lowest distribution.

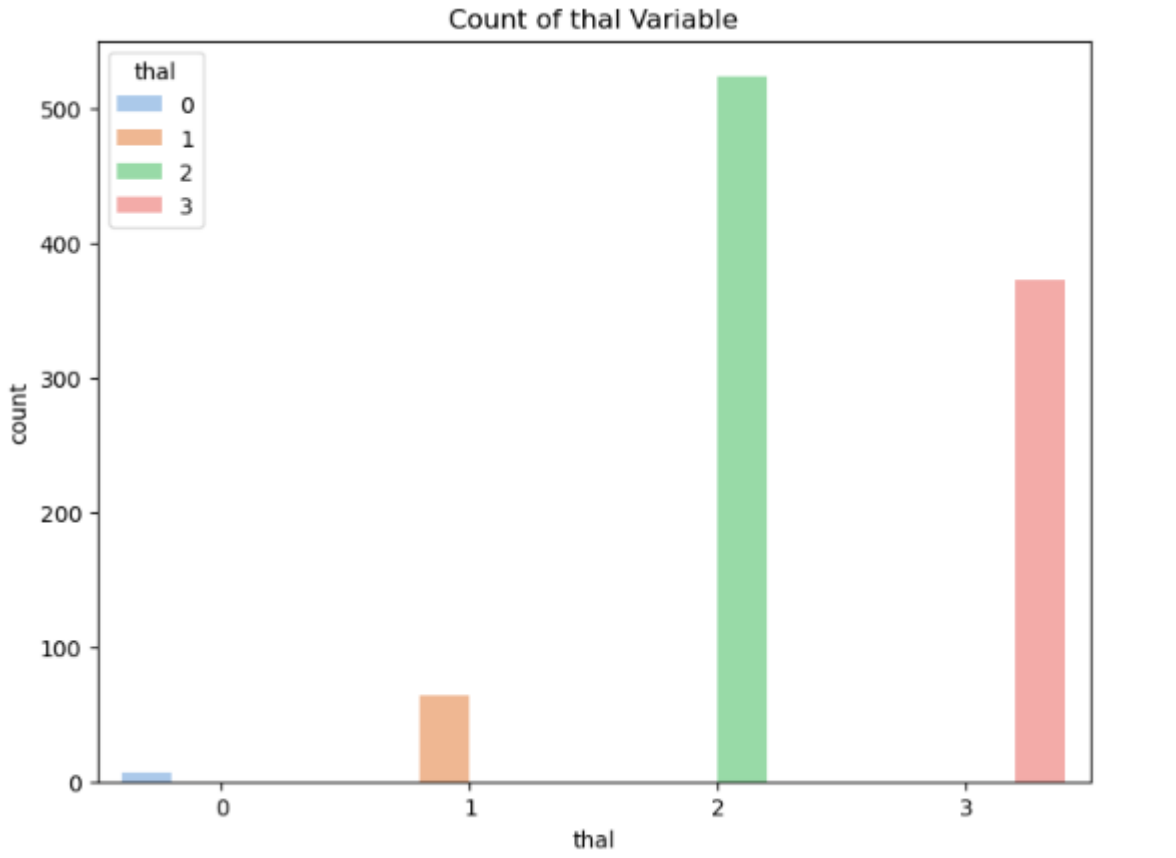

Figure 25 barchart of thal variable

According to the above Figure 25 shows the distribution of the thal variable. Thal=0 means normal thal=2 means reversable defect and that have the highest distribution.

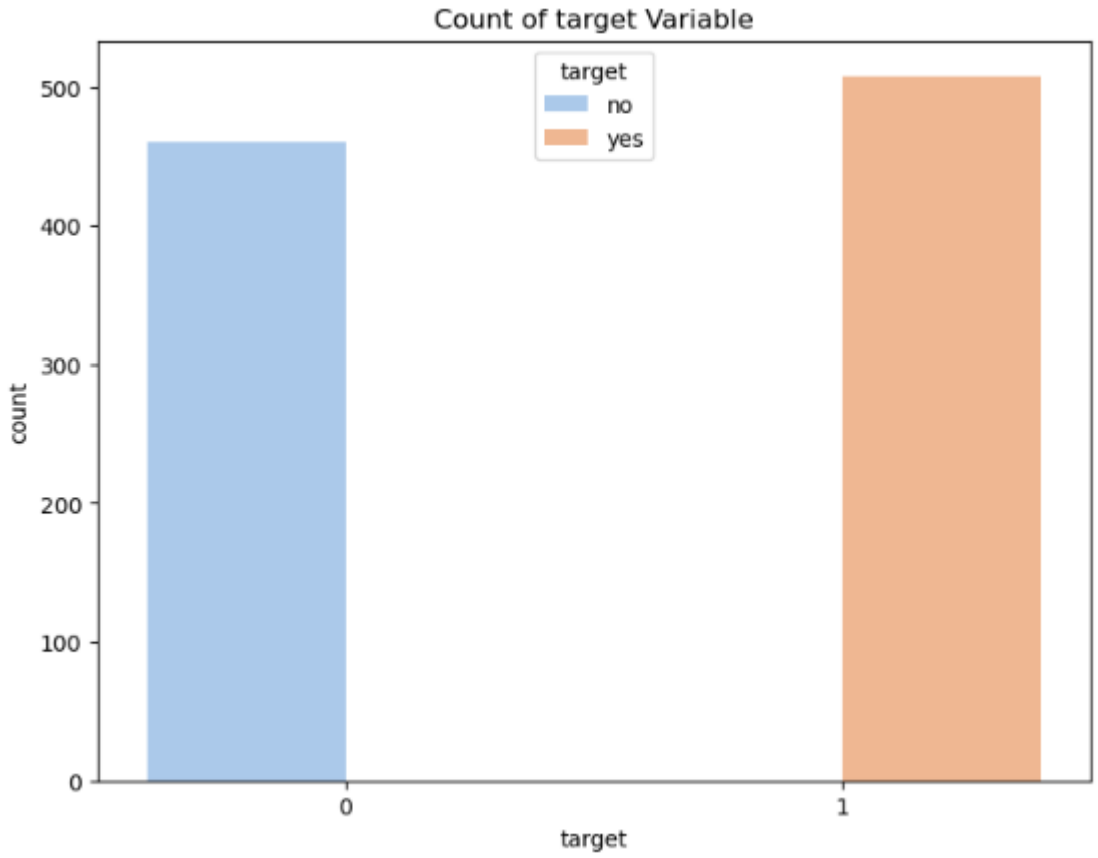

Figure 26 barchart of target variable

This above Figure 26 shows that there are more people having heart disease.

4.3.2 Bivariate Analysis

In this part I have done analysis of numerical vs numerical variables and categorical vs categorical variables and numerical vs categorical variables.

4.3.2.1 Numerical variables vs Numerical variables

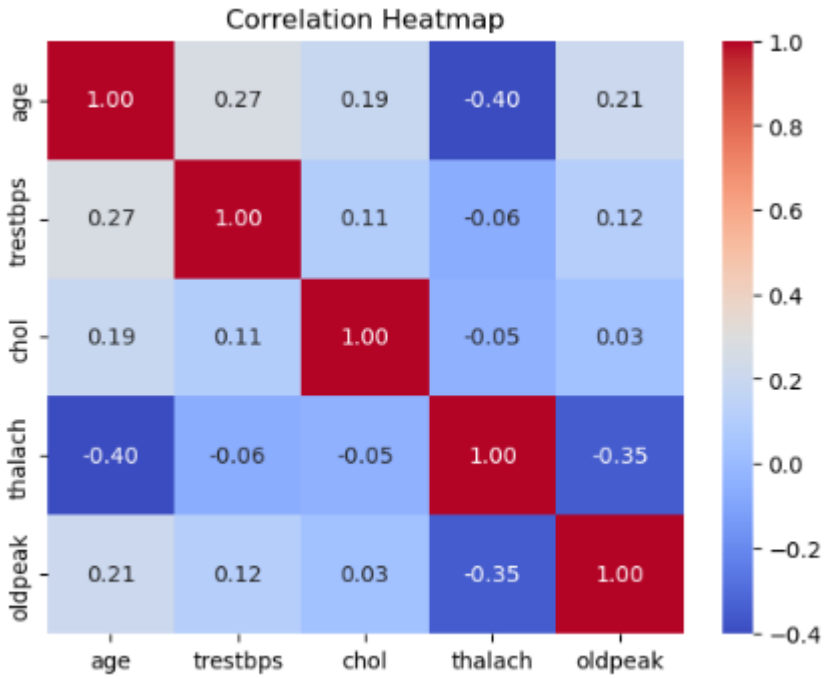

Figure 27 heatmap of numerical variables

The above Figure 27 shows the correlation plot of the numerical variables. There is a higher correlation between the age and the thalach variable and that is a negative relationship and lower correlation between the oldpeak and chol variables and that is a positive relationship. Thalach variable have negative relationship with chol variable and the trestbps variable. Other variables are having positive relationships.

4.3.2.2 Numerical variables vs Categorical variables

In this part I have used bar charts to plot the numerical and categorical variables.

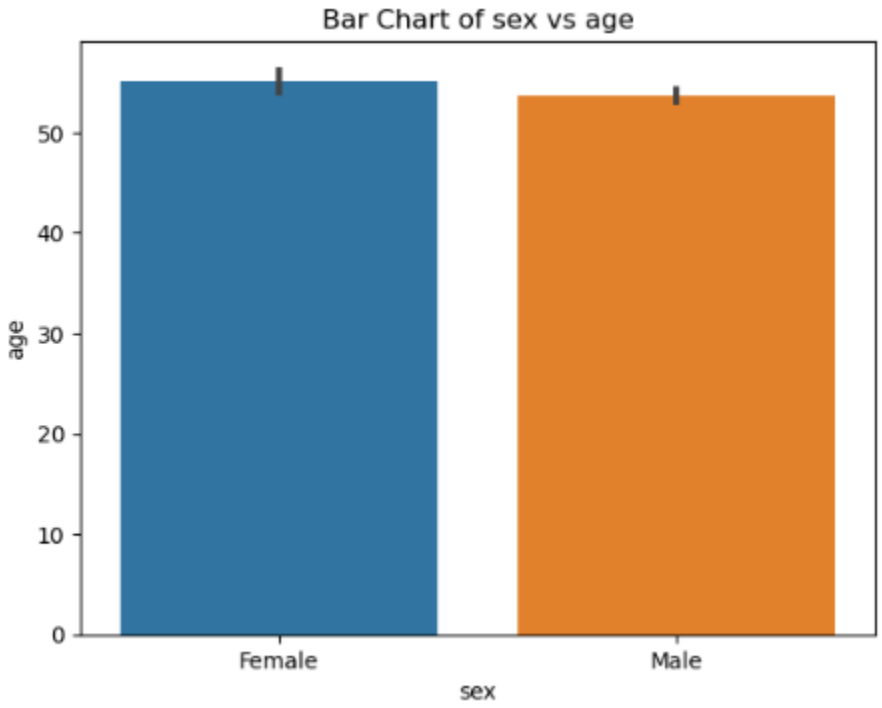

Figure 28 barchart of sex vs age

The above Figure 28 shows the bar chart of the sex variable with the age variable. In this dataset there are more females than the males and that can be seen above.

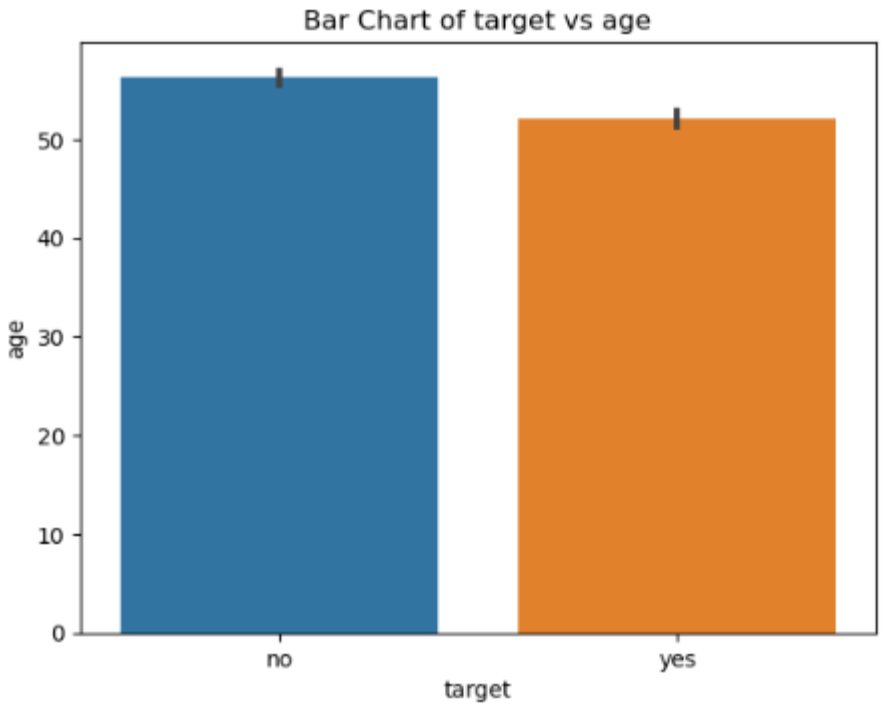

Figure 29 barchart of target vs age

The above Figure 29 shows the distribution of the age and the target variable.

4.3.2.3 Categorical variables vs Categorical variables

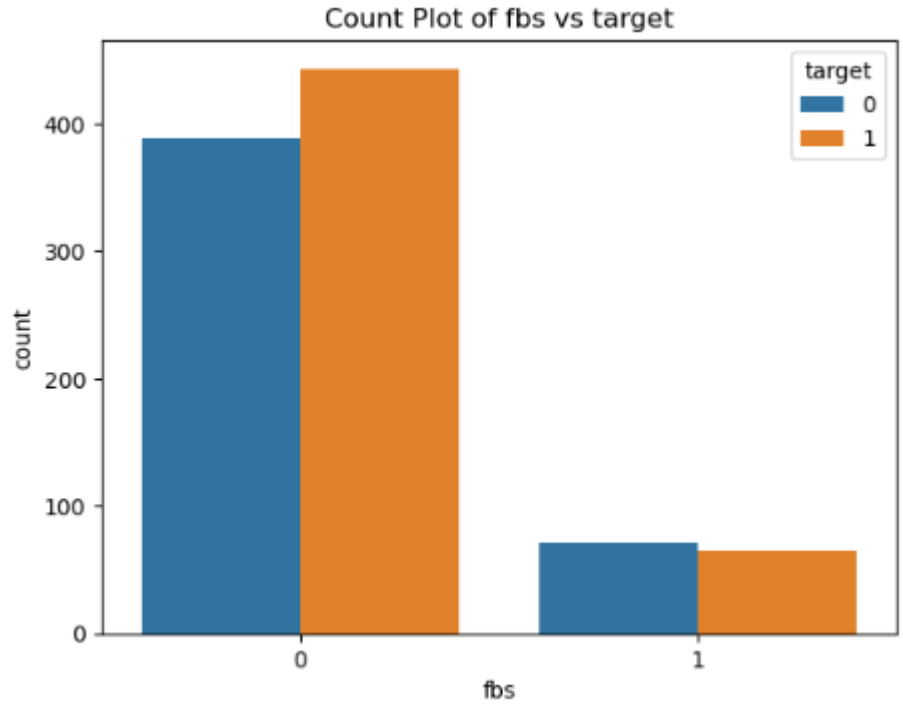

Figure 30 barchart of fbs vs target

The above Figure 30 shows that the people not having fbs is high and having fbs is very low. The people having fbs and having heart disease is low than the people with fbs and not having heart disease. So it seems like fbs is not seriously affecting the heart disease.

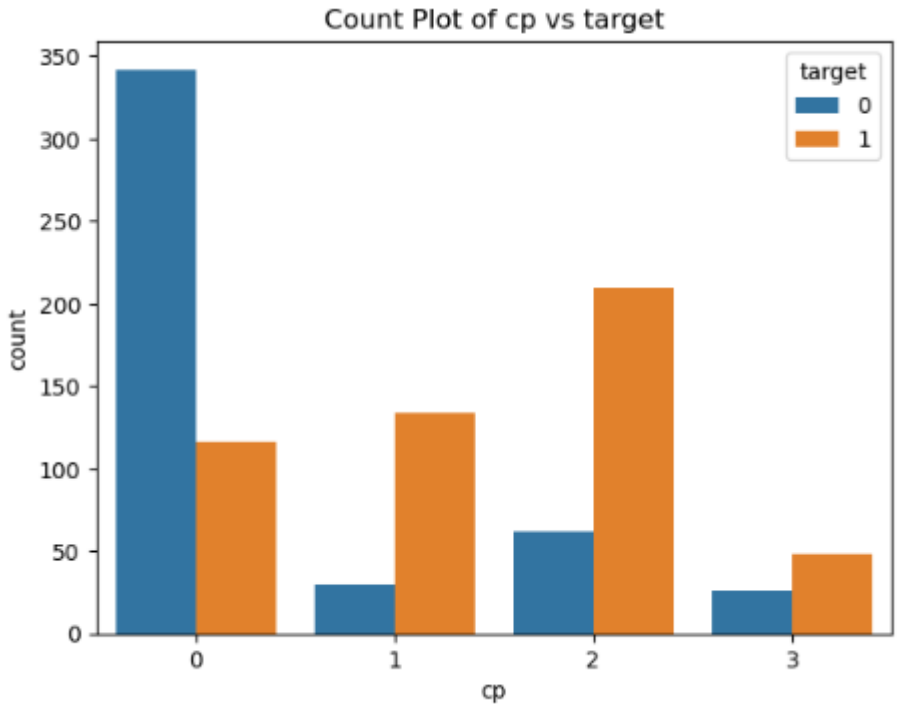

Figure 31 barchart of cp vs target

The above Figure 31 shows how cp variable affects the target variable. When cp=0, a lot of people don't have heart disease. When cp=2, a lot of people are having heart disease. Cp=2 means non -angina pain so that is not a good sign for the people because that is a risk. When cp=1, also lot of people are having heart disease. So cp=1,cp=2,cp=3 are risky for the people and they can be affected by heart disease.

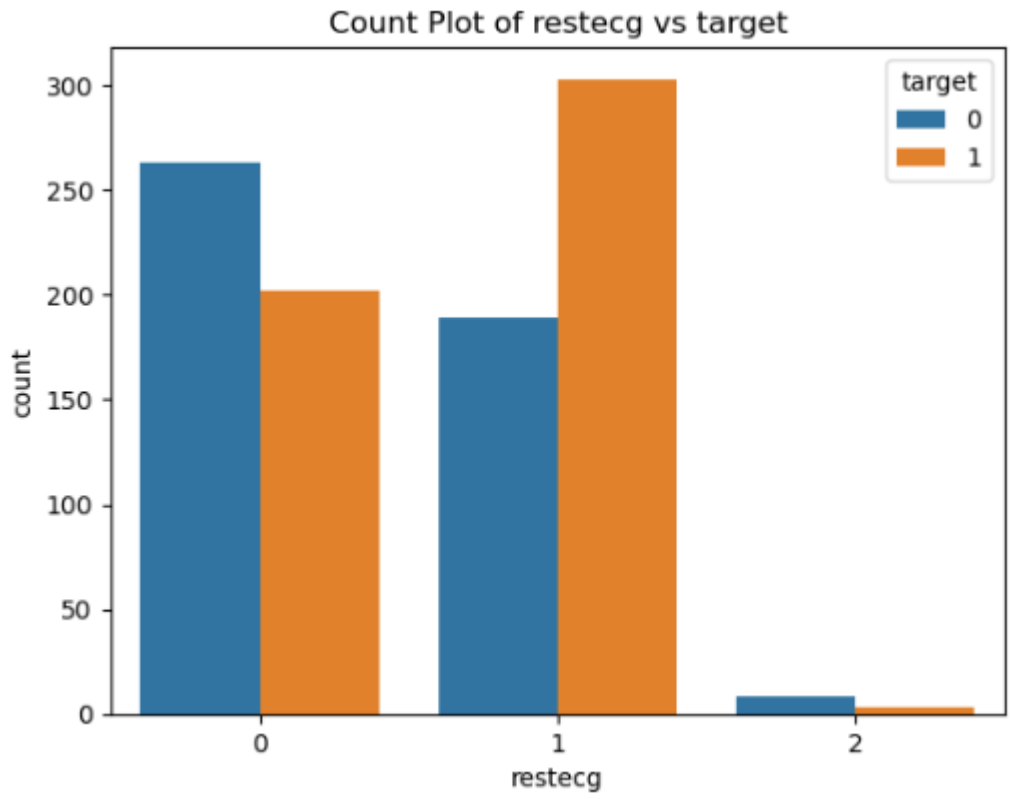

Figure 32 barchart of restecg vs target

This Figure 32 shows that restecg=1 are having a risk of heart disease. restecg=1 means having ST_T wave abnormal. So people with restecg=1 have to take care of them. Restecg=2 have a small distribution in this dataset. In restecg=0 not having heart disease people are higher than the people having heart disease.

4.3.3 Statistical Analysis

In this study, I have used t-test to identify the relationship between numerical variables and anova tests to identify the relationships between categorical and numerical variables and chi -squared tests to identify the relationships between categorical and categorical.

I have used t-tests for age and thalach, age and chol, age and trestbps, chol and trestbps. The results of the t-tests says that there is a relationship between those variables. The anova tests have been used for age and fbs, age and target, trestbps and target, chol and target. All these variable are having relationships with each other according to the anova tests. The chi-squared tests have

been used for exang and ca, sex and target, cp and target, fbs and target, restecg and target, exang and target, slope and target, ca and target, thal and target. So theare are relationships between the variables except the fbs and target because there is no relationship between them. So I have droped the fbs column because that is not affecting to heart disease. After that I have saved the dataset to a csv file for the training process.

Chapter 5: Training and Testing the Model

First, the saved dataset has been imported to a new jupyter notebook. That can be seen in the following Figure 33.

```
: import numpy as np
  import pandas as pd

  df=pd.read_csv('data.csv')
  df
```

:

	age	sex	cp	trestbps	chol	restecg	thalach	exang	oldpeak	slope	ca	thal	target
0	52	1	0	125	212	1	168	0	1.0	2	2	3	0
1	53	1	0	140	203	0	155	1	3.1	0	0	3	0
2	70	1	0	145	174	1	125	1	2.6	0	0	3	0
3	61	1	0	148	203	1	161	0	0.0	2	1	3	0
4	62	0	0	138	294	1	106	0	1.9	1	3	2	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...
963	59	1	1	140	221	1	164	1	0.0	2	0	2	1
964	60	1	0	125	258	0	141	1	2.8	1	1	3	0
965	47	1	0	110	275	0	118	1	1.0	1	1	2	0
966	50	0	0	110	254	0	159	0	0.0	2	0	2	1
967	54	1	0	120	188	1	113	0	1.4	1	1	3	0

968 rows × 13 columns

Figure 33 importing the dataset to training

Then the dataset has been converted into a numpy array by using the following code in the Figure 34.

```
df=df.values
```

```
df
```

```
array([[52.,  1.,  0., ...,  2.,  3.,  0.],
       [53.,  1.,  0., ...,  0.,  3.,  0.],
       [70.,  1.,  0., ...,  0.,  3.,  0.],
       ...,
       [47.,  1.,  0., ...,  1.,  2.,  0.],
       [50.,  0.,  0., ...,  0.,  2.,  1.],
       [54.,  1.,  0., ...,  1.,  3.,  0.]])
```

Figure 34 converted the dataset into numpy array

After that the dataset has been divided into data and target to separate the features and labels. That can be seen in the following Figure 35.

```
data=df[:,0:12]
target=df[:,12]
```

```
data
```

```
array([[52.,  1.,  0., ...,  2.,  2.,  3.],
       [53.,  1.,  0., ...,  0.,  0.,  3.],
       [70.,  1.,  0., ...,  0.,  0.,  3.],
       ...,
       [47.,  1.,  0., ...,  1.,  1.,  2.],
       [50.,  0.,  0., ...,  2.,  0.,  2.],
       [54.,  1.,  0., ...,  1.,  1.,  3.]])
```

```
target
```

```
array([0., 0., 0., 0., 0., 1., 0., 0., 0., 0., 1., 0., 1., 0., 0., 1., 1.,
       0., 1., 1., 0., 1., 1., 1., 1., 0., 1., 0., 0., 0., 1., 0., 0., 1.,
       0., 1., 1., 1., 0., 1., 1., 0., 0., 1., 1., 1., 1., 0., 1., 0., 1.,
       0., 1., 1., 0., 0., 1., 1., 0., 1., 1., 0., 1., 0., 1., 0., 0., 0.,
       0., 0., 1., 1., 0., 1., 1., 0., 0., 0., 1., 1., 1., 1., 0., 0., 1.,
       1., 0., 0., 1., 1., 1., 0., 0., 1., 1., 1., 1., 1., 1., 0., 0., 0.,
       0., 0., 0., 1., 0., 0., 0., 0., 0., 0., 1., 1., 1., 0., 0., 0., 1.,
       1., 1., 1., 1., 1., 1., 1., 1., 1., 0., 1., 1., 1., 0., 0., 0., 1.,
       1., 0., 1., 0., 1., 1., 0., 0., 1., 0., 1., 1., 1., 1., 0., 1., 0.,
       0., 0., 0., 0., 1., 1., 1., 1., 0., 1., 1., 0., 0., 0., 0., 0., 1.,
       0., 1., 1., 0., 0., 0., 0., 0., 1., 1., 1., 0., 1., 0., 1., 1., 0.,
       1., 1., 1., 1., 1., 1., 0., 1., 1., 0., 1., 0., 0., 1., 1., 1., 0.,
       1., 0., 0., 0., 0., 1., 1., 1., 1., 0., 1., 1., 0., 1., 0., 1., 1.,
       1., 0., 0., 0., 0., 1., 0., 1., 0., 1., 1., 0., 1., 1., 0., 1., 0.
```

Figure 35 dividing the dataset into data and target

After that normalizing the data and the target have been done using the MinMaxScaler function in the sklearn library. Scaling the data is very important because it will give faster convergence in the training. Also it will improve the performance of the model. The normalization part can be seen in the following Figure 36.

```
from sklearn.preprocessing import MinMaxScaler

target=np.reshape(target, (-1,1))

scaler_data = MinMaxScaler(feature_range=(0,1))
scaler_target = MinMaxScaler()

data_scaled=scaler_data.fit_transform(data)
target_scaled=scaler_target.fit_transform(target)
```

Figure 36 normalizing the data and target

After the normalization part the the scaled data and the scaled target have been splited into training and testing using the train_test_split function from the sklearn library. I have used 90% for training and 10% for testing. That can be seen in the following Figure 37.

```
from sklearn.model_selection import train_test_split

train_data, test_data, train_target, test_target = train_test_split(data_scaled, target_scaled,test_size=0.1)
```

Figure 37 dividing the dataset into training and testing

After this the ANN model has been created. The following figure is about the ANN model.

```
from keras.models import Sequential
from keras.layers import Dense,Dropout
from keras.optimizers import Adam

model = Sequential()
model.add(Dense(64,activation='relu',input_shape=(train_data.shape[1],)))
model.add(Dropout(0.2))
model.add(Dense(32, activation='relu'))
model.add(Dropout(0.2))
model.add(Dense(1, activation='sigmoid'))

optimizer = Adam(learning_rate=0.001)
model.compile(optimizer=optimizer,loss='binary_crossentropy',metrics=['accuracy'])

model.summary()
```

```
Model: "sequential_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense_3 (Dense)             (None, 64)                832

 dropout_2 (Dropout)         (None, 64)                0

 dense_4 (Dense)             (None, 32)                2080

 dropout_3 (Dropout)         (None, 32)                0

 dense_5 (Dense)             (None, 1)                 33

=================================================================
Total params: 2945 (11.50 KB)
Trainable params: 2945 (11.50 KB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

Figure 38 creating the model

According to the above Figure 38 I have used Dense layers, Dropout layers and relu activation functions, sigmoid activation function. For the learning rate I have used Adam optimizer and the staring learning rate has been scheduled to 0.001. The binary_crossentropy loss function has been used because this is a binary classification problem. The output of the code can be seen according to the above figure. After that I have trained the model and saved the best model that can be viewed from the following Figure 39.

```
: from keras.callbacks import ModelCheckpoint,EarlyStopping

  checkpoint = ModelCheckpoint('models/model-{epoch:03d}.model',monitor='val_loss',save_best_only=True,mode='auto')

: model.fit(train_data, train_target, epochs=500, batch_size=16, validation_split=0.1,callbacks=[checkpoint])
Epoch 1/500
28/49 [================>.............] - ETA: 0s - loss: 0.6639 - accuracy: 0.6250 INFO:tensorflow:Assets written to: models
\model-001.model\assets

INFO:tensorflow:Assets written to: models\model-001.model\assets

49/49 [==============================] - 1s 16ms/step - loss: 0.6365 - accuracy: 0.6731 - val_loss: 0.5477 - val_accuracy: 0.
7273
Epoch 2/500
26/49 [==============>...............] - ETA: 0s - loss: 0.5092 - accuracy: 0.8173INFO:tensorflow:Assets written to: models\m
odel-002.model\assets

INFO:tensorflow:Assets written to: models\model-002.model\assets

49/49 [==============================] - 1s 13ms/step - loss: 0.4848 - accuracy: 0.8148 - val_loss: 0.4288 - val_accuracy: 0.
7273
Epoch 3/500
46/49 [===========================>..] - ETA: 0s - loss: 0.4296 - accuracy: 0.8084INFO:tensorflow:Assets written to: models\m
odel-003.model\assets
```

Figure 39 training the model

In the training I have used batch_size of 16 and 500 epochs and validation_split of 0.1.

Then the training loss and the validation loss have been plotted.

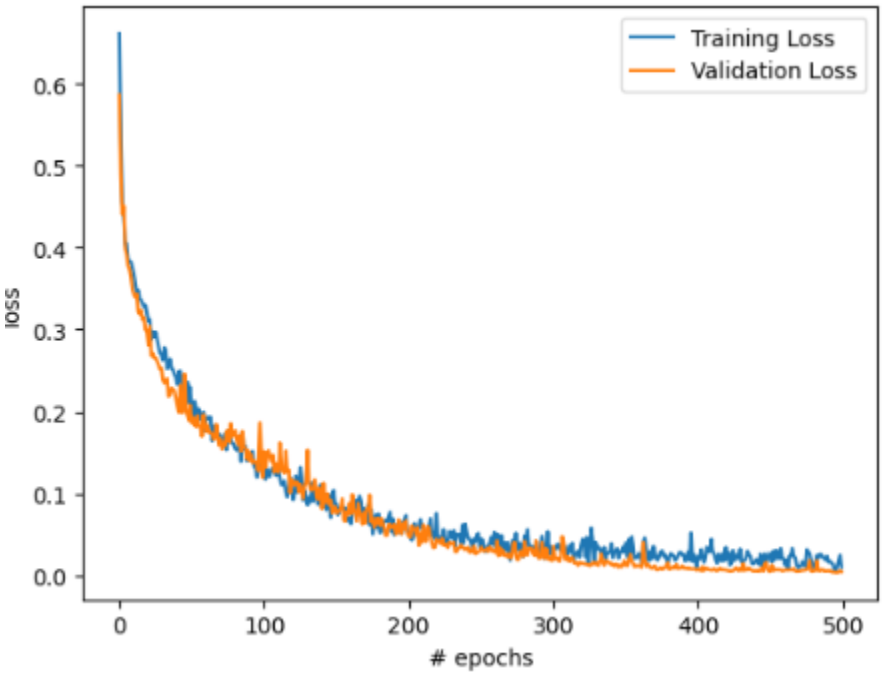

Figure 40 loss functions of the model

According to this Figure 40 the model is giving extremely good results on the training and the validation. So it is very important when predicting to the unseen data and this model is very good at it because the validation loss curve is excellent in this case.

After that the accuracy curves have been plotted.

```
plt.plot(model.history.history['accuracy'], label='Training Accuracy')
plt.plot(model.history.history['val_accuracy'], label='Validation Accuracy')
plt.xlabel('# epochs')
plt.ylabel('accuracy')
plt.legend()
plt.show()
```

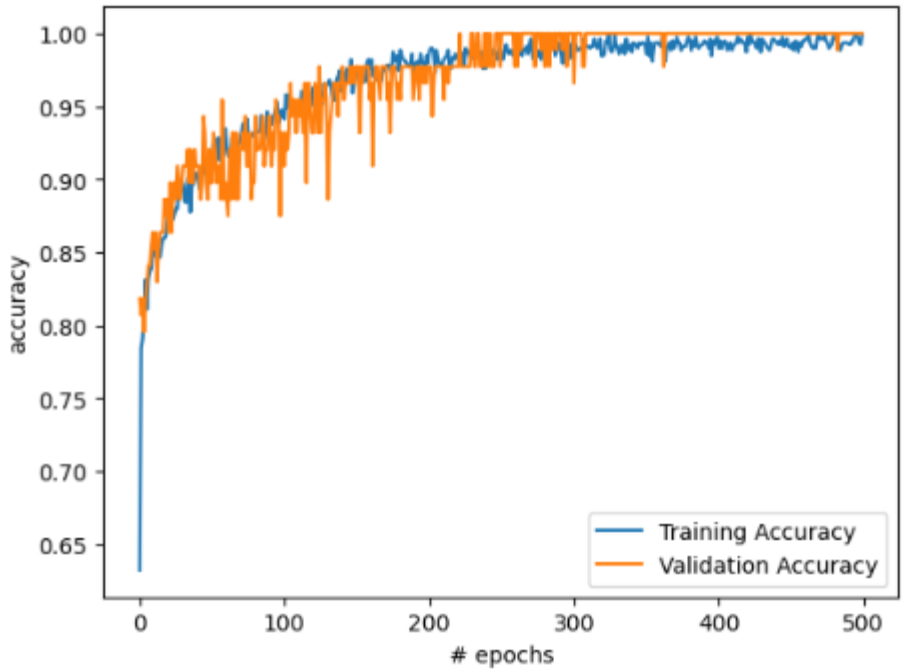

Figure 41 accuracy functions of the model

According to the above Figure 41 shows that this is giving good results for the training and validation by looking at the accuracy. The validation accuracy has been converged and I think that will give extreamly good results for the unseen data.

The following Figure 42 shows predicting the target using test data.

```
]: predicted_target = model.predict(test_data)
   predicted_target

   4/4 [==============================] - 0s 2ms/step

]: array([[1.92798194e-10],
          [9.27626118e-08],
          [1.00000000e+00],
          [1.00000000e+00],
          [1.00000000e+00],
          [1.57239898e-11],
          [1.46714016e-03],
          [3.34367741e-07],
          [4.64924099e-03],
          [3.77544275e-05],
          [9.99755561e-01],
          [9.99751866e-01],
          [9.99995708e-01],
          [1.00000000e+00],
          [8.73044257e-07],
          [9.99996722e-01],
          [1.00000000e+00],
```

Figure 42 predicting using test data

After this the accuracy of the model has been found and that is very excellent. That can be seen in the following Figure 43.

```
predicted_labels=np.argmax(predicted_target,axis=1)
actual_labels=np.argmax(test_target,axis=1)
```

```
from sklearn.metrics import accuracy_score

acc=accuracy_score(actual_labels,predicted_labels)
print('acc:',acc)
```

```
acc: 1.0
```

Figure 43 accuracy of the model

After that I have tested whether the model is giving correct results or what by the following Figure 44.

```
print('actual:',test_target[:10].T)
print('predicted:',predicted_target[:10].T)
```

```
actual: [[0. 0. 1. 1. 1. 0. 0. 0. 0. 0.]]
predicted: [[1.9279819e-10 9.2762612e-08 1.0000000e+00 1.0000000e+00 1.0000000e+00
  1.5723990e-11 1.4671402e-03 3.3436774e-07 4.6492410e-03 3.7754427e-05]]
```

Figure 44 testing the model

After training the model then I have saved the scaled data and scaled target into dump files from the joblib library and I have saved the model too. Those things can be viewed in the following Figure 45.

```
import joblib

joblib.dump(scaler_data,'scaler_data.sav')
joblib.dump(scaler_target,'scaler_target.sav')
```

```
['scaler_target.sav']
```

```
model.save("model.h5")
```

Figure 45 saving the model and scaled data and scaled target

Then I have imported the model into another jupyter notebook and then tested the model using some random data. The following Figure 46 shows the code and the output of that.

```
from keras.models import load_model
import numpy as np

model = load_model('model.h5')

import joblib

scaler_data = joblib.load('scaler_data.sav')
scaler_target = joblib.load('scaler_target.sav')

input_data = np.array([[60, 0, 0, 125, 258, 0, 141, 1, 2.8, 1, 1, 3],
                       [59, 1, 1, 140, 221, 1, 164, 1, 0.0, 2, 0, 2]])
input_data_scaled = scaler_data.transform(input_data)

predictions = model.predict(input_data_scaled)

print(predictions)
```

```
WARNING:tensorflow:From C:\Users\Acer\anaconda3\Lib\site-packages\keras\src\losses.py:2976: The name tf.losses.sparse_softmax_c
ross_entropy is deprecated. Please use tf.compat.v1.losses.sparse_softmax_cross_entropy instead.

WARNING:tensorflow:From C:\Users\Acer\anaconda3\Lib\site-packages\keras\src\backend.py:1398: The name tf.executing_eagerly_outs
ide_functions is deprecated. Please use tf.compat.v1.executing_eagerly_outside_functions instead.

1/1 [==============================] - 0s 118ms/step
[[8.9581636e-07]
 [9.9995965e-01]]
```

Figure 46 testing the model with random data

Chapter 6: Creating the web application

In addition to developing a heart disease detection model using Artificial Neural Networks (ANN), a web application was created using Flask to facilitate the interaction with the model. This web application comprises two distinct sections represented by separate HTML templates: one for inputting patient details and another for displaying the results of the heart disease prediction. The first template allows users to input relevant patient information, such as age, gender, cholesterol levels, and blood pressure, which are essential features for predicting heart disease risk. Upon submission, the input data is passed through the trained ANN model for prediction. The second template serves as the output interface, presenting the prediction results obtained from the model. This approach enhances accessibility and usability by providing a user -friendly interface for both inputting patient details and visualizing the corresponding heart disease prediction outcomes. Integrating these web application functionalities into the report serves to showcase the practical implementation and usability of the heart disease detection model developed in this project, emphasizing its potential real-world applications and impact. The following Figure 47 and 48 will show the interface and the results of the web application.

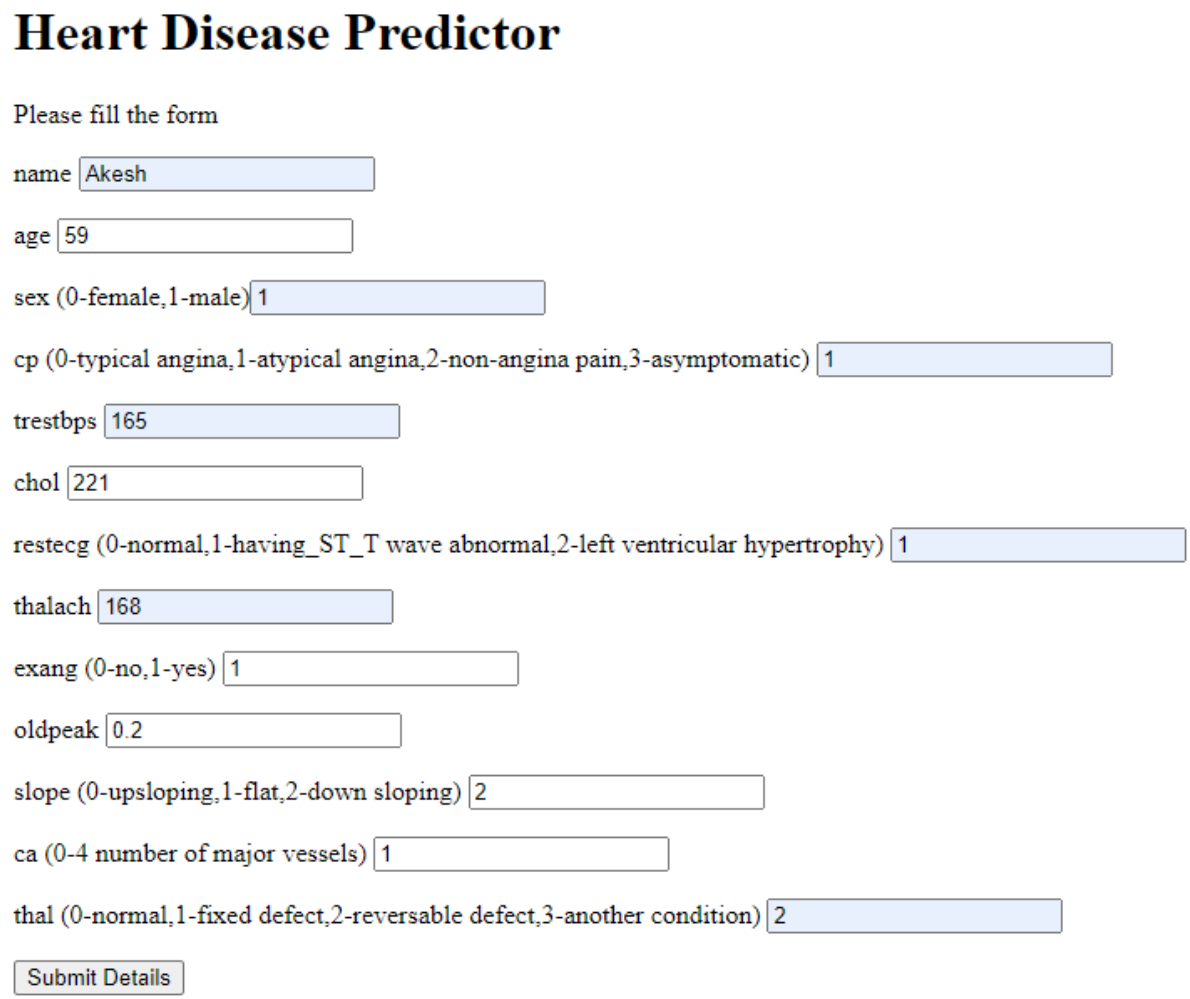

Figure 47 picture of the web application

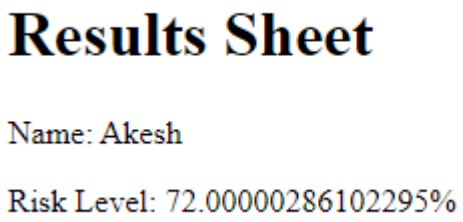

Figure 48 picture of results of the web app

Chapter 7: Conclusion, Discussion and Recommendation

7.1 Conclusion

In conclusion, this project has successfully completed a comprehensive analysis of heart disease prediction using a dataset obtained and preprocessed in Jupyter Notebook. The initial steps involved importing the dataset followed by preprocessing to ensure data quality and consistency. Subsequently, an Exploratory Data Analysis (EDA) was conducted to gain insights into the dataset's characteristics, including summarizing variables, analyzing relationships, and visualizing data distributions. The statistical analysis performed during EDA further enhanced our understanding of the dataset and identified potential risk factors associated with heart disease. Building on these insights, an Artificial Neural Network (ANN) model was trained using Deep Learning techniques to predict heart disease based on input variables. The model's performance was evaluated, demonstrating its capability to accurately predict heart disease outcomes. Furthermore, the development of a web application utilizing this trained model enhances accessibility and usability, allowing for real-time predictions based on individual input.

Overall, this project not only showcases the technical proficiency in data analysis and model training but also highlights the practical application of these findings in healthcare decision -making and risk assessment for heart disease prevention.

7.2 Discussion

In this study, we have successfully achieved our main objective of detecting heart disease by developing an Artificial Neural Network (ANN) model that accurately predicts heart disease based on several key risk factors. Through comprehensive data analysis, we identified significant risk factors such as age, sex, chest pain type (cp), cholesterol levels (chol), resting blood pressure (trestbps), and resting electrocardiographic results (restecg). Notably, we found that fasting blood sugar (fbs) did not exhibit a relationship with heart disease and was consequently dropped from our analysis. Leveraging statistical analysis tests, including ANOVA and chi-squared tests, we elucidated the relationships between various variables and heart disease. Specifically, using a heatmap, we visualized the correlations between numerical variables, revealing notable associations between age, trestbps, chol, and restecg with the target variable. Further, ANOVA tests confirmed significant relationships between age, trestbps, and chol with heart disease, while chi-squared tests revealed associations between categorical variables such as sex, chest pain type, and exercise-induced angina (exang) with the target variable. The accuracy of the ANN model is 1 and that is very good in making predictions. Moreover, we successfully implemented a web application to facilitate real-time heart disease predictions based on our ANN model. Overall, our findings contribute to a deeper understanding of heart disease risk factors and provide valuable insights for personalized prevention and treatment strategies.

7.3 Recommendation

Based on the findings of this project, it is recommended to further explore and validate the identified risk factors associated with the early diagnosis of heart disease through larger-scale studies and diverse populations. Additionally, incorporating advanced machine learning

techniques and incorporating more comprehensive datasets may enhance the accuracy and generalizability of heart disease prediction models. Furthermore, it is crucial to prioritize public health initiatives aimed at promoting lifestyle modifications, such as healthy diet and regular physical activity, to reduce the burden of heart disease. Collaborative efforts between healthcare professionals, researchers, policymakers, and community stakeholders are essential to implement effective prevention and treatment strategies tailored to individual risk profiles. Moreover, continued research into the interplay between genetic predisposition, environmental factors, and lifestyle behaviors is warranted to inform the development of personalized interventions for heart disease prevention. Overall, leveraging multidisciplinary approaches and harnessing emerging technologies hold promise in addressing the complex challenges associated with heart disease and improving cardiovascular health outcomes on a global scale.

Appendices

```
import numpy as np
import pandas as pd

data=pd.read_csv('heart.csv')
data
```

```
data.head(10)
```

```
data.tail(10)
```

```
data.info()
```

```
#finding missing values
data.isnull().sum()
```

```
import matplotlib.pyplot as plt
import pandas as pd

df=data[['age','trestbps','chol','thalach','oldpeak']]

plt.figure(figsize=(12, 10))
plt.boxplot(df,labels=df.columns)
plt.title('Box Plot of numerical variables')
plt.ylabel('Values')
plt.xlabel('Variables')
plt.show()
```

```
#detecting outliers
q1=df.quantile(0.25)
q3=df.quantile(0.75)

IQR=q3-q1

outliers= ((df < (q1-1.5*IQR)) | (df > (q3+1.5*IQR))).any(axis=1)
outliers
```

```
df=df[~outliers]
```

```
df
```

```
plt.figure(figsize=(12, 10))
plt.boxplot(df,labels=df.columns)
plt.title('Box Plot of numerical variables')
plt.ylabel('Values')
plt.xlabel('Variables')
plt.show()
```

```
df = df.rename(columns={'age': 'age_new','trestbps':'trestbps_new','chol':'chol_new','thalach':'thalach_new','oldpeak':'oldpeak_
df
```

```
data = pd.concat([data,df], axis=1)
```

```
data
```

```
data.drop(['age','trestbps','chol','thalach','oldpeak'],axis=1,inplace=True)
```

```
data
```

```
data.isnull().sum()
```

```
data=data.dropna()
```

```
data.isnull().sum()
```

```
data[:60]
```

```
re_order = ['age_new','sex','cp','trestbps_new','chol_new','fbs','restecg','thalach_new','exang','oldpeak_new','slope','ca','thal

data= data[re_order]
data
```

```
data = data.rename(columns={'age_new':'age','trestbps_new':'trestbps','chol_new':'chol','thalach_new':'thalach','oldpeak_new':'ol
data
```

```
data['age'] = data['age'].astype(int)
data['trestbps'] = data['trestbps'].astype(int)
data['chol'] = data['chol'].astype(int)
data['thalach'] = data['thalach'].astype(int)

data
```

```
data.to_csv('preprocessed_data.csv', index=False)
```

```
data=pd.read_csv('preprocessed_data.csv')
data
```

```
#summary of the dataset
data.describe()
```

```
import seaborn as sns
numerical_columns = data[['age','trestbps','chol','thalach','oldpeak']]

plt.figure(figsize=(20,12))
for i, col in enumerate(numerical_columns):
    plt.subplot(3, 3, i + 1)
    sns.histplot(data[col], kde=True, color='red')
    plt.title(f'Histogram of {col}')
    plt.xlabel(col)
    plt.ylabel('Frequency')
plt.show()
```

```
plt.figure(figsize=(8,6))
sns.countplot(data=data, x='sex', hue='sex', palette=['blue','green'])
plt.title('Count of sex Variable')
plt.xlabel('sex')
plt.ylabel('count')
plt.legend(title='sex', loc='upper center', labels=['Female', 'Male'])
plt.show()
```

```
plt.figure(figsize=(8,6))
sns.countplot(data=data, x='cp', hue='cp', palette='pastel')
plt.title('Count of cp Variable')
plt.xlabel('cp')
plt.ylabel('count')
plt.legend(title='cp', loc='upper center', labels=['typical angina','atypical angina','non-angina pain','asymptomatic'])
plt.show()
```

```
plt.figure(figsize=(8,6))
sns.countplot(data=data, x='fbs', hue='fbs', palette=['blue','green'])
plt.title('Count of fbs Variable')
plt.xlabel('fbs')
plt.ylabel('count')
plt.legend(title='fbs', loc='upper center', labels=['not having fbs','having fbs'])
plt.show()
```

```
plt.figure(figsize=(8,6))
sns.countplot(data=data, x='restecg', hue='restecg', palette='pastel')
plt.title('Count of restecg Variable')
plt.xlabel('restecg')
plt.ylabel('count')
plt.legend(title='restecg', loc='upper right', labels=['normal','having_ST_T wave abnormal','left ventricular hypertrophy'])
plt.show()
```

```
plt.figure(figsize=(8,6))
sns.countplot(data=data, x='slope', hue='slope', palette='pastel')
plt.title('Count of slope Variable')
plt.xlabel('slope')
plt.ylabel('count')
plt.legend(title='slope', loc='upper left', labels=['upsloping','flat','down sloping'])
plt.show()
```

```
plt.figure(figsize=(8,6))
sns.countplot(data=data, x='ca', hue='ca', palette='pastel')
plt.title('Count of ca Variable')
plt.xlabel('ca')
plt.ylabel('count')
plt.legend(title='ca', loc='upper center')
plt.show()
```

```
plt.figure(figsize=(8,6))
sns.countplot(data=data, x='thal', hue='thal', palette='pastel')
plt.title('Count of thal Variable')
plt.xlabel('thal')
plt.ylabel('count')
plt.legend(title='thal', loc='upper left')
plt.show()
```

```
plt.figure(figsize=(8,6))
sns.countplot(data=data, x='target', hue='target', palette='pastel')
plt.title('Count of target Variable')
plt.xlabel('target')
plt.ylabel('count')
plt.legend(title='target', loc='upper center', labels=['no','yes'])
plt.show()
```

```
numerical_cols = data[['age', 'trestbps', 'chol', 'thalach', 'oldpeak']]

sns.heatmap(numerical_cols.corr(), annot=True, cmap='coolwarm',fmt=".2f")
plt.title('Correlation Heatmap')
plt.show()
```

```
sns.barplot(data=data, x='sex', y='age')
plt.title('Bar Chart of sex vs age')
plt.xlabel('sex')
plt.ylabel('age')
plt.xticks(ticks=[0, 1], labels=['Female', 'Male'])
plt.show()
```

```
sns.barplot(data=data, x='target', y='age')
plt.title('Bar Chart of target vs age')
plt.xlabel('target')
plt.ylabel('age')
plt.xticks(ticks=[0, 1], labels=['no', 'yes'])
plt.show()
```

```
sns.barplot(data=data, x='target', y='trestbps')
plt.title('Bar Chart of target vs trestbps')
plt.xlabel('target')
plt.ylabel('trestbps')
plt.xticks(ticks=[0, 1], labels=['no', 'yes'])
plt.show()
```

```
sns.countplot(data=data, x='fbs', hue='target')
plt.title('Count Plot of fbs vs target')
plt.xlabel('fbs')
plt.ylabel('count')
plt.legend(title='target')
plt.show()
```

```
sns.countplot(data=data, x='cp', hue='target')
plt.title('Count Plot of cp vs target')
plt.xlabel('cp')
plt.ylabel('count')
plt.legend(title='target')
plt.show()
```

```
sns.countplot(data=data, x='restecg', hue='target')
plt.title('Count Plot of restecg vs target')
plt.xlabel('restecg')
plt.ylabel('count')
plt.legend(title='target')
plt.show()
```

```
# H0- There is no significanct difference between the age and thalach variables
# H1- There is a significant difference between the age and thalach variables

#alpha=0.05
from scipy import stats

t_statistic, p_value = stats.ttest_ind(data['age'], data['thalach'])

print("T-statistic:", t_statistic)
print("P-value:", p_value)
```

```
T-statistic: -121.38824569684586
P-value: 0.0
```

Since the p-value is less than the alpha value of 0.05 we have to reject the null hypothesis and accept the alternative hypothesis. This means that there is a significant difference between the age and thalach variables.

```
# H0- There is no significanct difference between the age and chol variables
# H1- There is a significant difference between the age and chol variables

#alpha=0.05
from scipy import stats

t_statistic, p_value = stats.ttest_ind(data['age'], data['chol'])

print("T-statistic:", t_statistic)
print("P-value:", p_value)
```

```
T-statistic: -126.92624086723133
P-value: 0.0
```

Since the p-value is less than the alpha value of 0.05 we have to reject the null hypothesis and accept the alternative hypothesis. This means that there is a significant difference between the age and chol variables.

```
# H0- There is no significanct difference between the age and trestbps variables
# H1- There is a significant difference between the age and trestbps variables

#alpha=0.05
from scipy import stats

t_statistic, p_value = stats.ttest_ind(data['age'], data['trestbps'])

print("T-statistic:", t_statistic)
print("P-value:", p_value)
```

```
T-statistic: -132.25618983537555
P-value: 0.0
```

Since the p-value is less than the alpha value of 0.05 we have to reject the null hypothesis and accept the alternative hypothesis. This means that there is a significant difference between the age and trestbps variables.

```
# H0- There is no significanct difference between the chol and trestbps variables
# H1- There is a significant difference between the chol and trestbps variables

#alpha=0.05
from scipy import stats

t_statistic, p_value = stats.ttest_ind(data['chol'], data['trestbps'])

print("T-statistic:", t_statistic)
print("P-value:", p_value)
```

```
T-statistic: 73.28123831189451
P-value: 0.0
```

Since the p-value is less than the alpha value of 0.05 we have to reject the null hypothesis and accept the alternative hypothesis. This means that there is a significant difference between the chol and trestbps variables.

```
# H0- There is no significanct difference between the age and fbs variables
# H1- There is a significant difference between the age and fbs variables

#alpha=0.05
from scipy import stats

a = data[data['fbs']==0]['age']
b = data[data['fbs']==1]['age']

f_statistic, p_value = stats.f_oneway(a,b)

print("F-statistic:", f_statistic)
print("P-value:", p_value)
```

```
F-statistic: 11.513383711296209
P-value: 0.0007190317710372662
```

Since the p-value is less than the alpha value of 0.05 we have to reject the null hypothesis and accept the alternative hypothesis. This means that there is a significant difference between the age and fbs variables.

```
# H0- There is no significanct difference between the age and target variables
# H1- There is a significant difference between the age and target variables

#alpha=0.05
from scipy import stats

a = data[data['target']==0]['age']
b = data[data['target']==1]['age']

f_statistic, p_value = stats.f_oneway(a,b)

print("F-statistic:", f_statistic)
print("P-value:", p_value)
```

```
F-statistic: 53.39709373550737
P-value: 5.702655984191439e-13
```

Since the p-value is less than the alpha value of 0.05 we have to reject the null hypothesis and accept the alternative hypothesis. This means that there is a significant difference between the age and target variables.

```
# H0- There is no significanct difference between the trestbps and target variables
# H1- There is a significant difference between the trestbps and target variables

#alpha=0.05
from scipy import stats

a = data[data['target']==0]['trestbps']
b = data[data['target']==1]['trestbps']

f_statistic, p_value = stats.f_oneway(a,b)

print("F-statistic:", f_statistic)
print("P-value:", p_value)
```

```
F-statistic: 10.755009985642246
P-value: 0.0010771326612174677
```

Since the p-value is less than the alpha value of 0.05 we have to reject the null hypothesis and accept the alternative hypothesis. This means that there is a significant difference between the trestbps and target variables.

```
# H0- There is no significanct difference between the chol and target variables
# H1- There is a significant difference between the chol and target variables

#alpha=0.05
from scipy import stats

a = data[data['target']==0]['chol']
b = data[data['target']==1]['chol']

f_statistic, p_value = stats.f_oneway(a,b)

print("F-statistic:", f_statistic)
print("P-value:", p_value)
```

```
F-statistic: 17.481809883299384
P-value: 3.164180710668812e-05
```

Since the p-value is less than the alpha value of 0.05 we have to reject the null hypothesis and accept the alternative hypothesis. This means that there is a significant difference between the chol and target variables.

```
# H0- There is no significanct difference between the thalach and target variables
# H1- There is a significant difference between the thalach and target variables

#alpha=0.05
from scipy import stats

a = data[data['target']==0]['thalach']
b = data[data['target']==1]['thalach']

f_statistic, p_value = stats.f_oneway(a,b)

print("F-statistic:", f_statistic)
print("P-value:", p_value)
```

```
F-statistic: 215.65172520072178
P-value: 3.2229021107950706e-44
```

Since the p-value is less than the alpha value of 0.05 we have to reject the null hypothesis and accept the alternative hypothesis. This means that there is a significant difference between the thalach and target variables.

```
# H0- There is no significanct difference between the oldpeak and target variables
# H1- There is a significant difference between the oldpeak and target variables

#alpha=0.05
from scipy import stats

a = data[data['target']==0]['oldpeak']
b = data[data['target']==1]['oldpeak']

f_statistic, p_value = stats.f_oneway(a,b)

print("F-statistic:", f_statistic)
print("P-value:", p_value)
```

```
F-statistic: 245.2802767664218
P-value: 1.95509984627003e-49
```

Since the p-value is less than the alpha value of 0.05 we have to reject the null hypothesis and accept the alternative hypothesis. This means that there is a significant difference between the oldpeak and target variables.

```
# H0- There is no significanct difference between the chol and restecg variables
# H1- There is a significant difference between the chol and restecg variables

#alpha=0.05
from scipy import stats

a = data[data['restecg']==0]['chol']
b = data[data['restecg']==1]['chol']
c = data[data['restecg']==2]['chol']
f_statistic, p_value = stats.f_oneway(a,b,c)

print("F-statistic:", f_statistic)
print("P-value:", p_value)
```

```
F-statistic: 8.354875953313625
P-value: 0.0002526855849998887
```

Since the p-value is less than the alpha value of 0.05 we have to reject the null hypothesis and accept the alternative hypothesis. This means that there is a significant difference between the chol and restecg variables.

```
# H0- There is no significanct difference between the exang and ca variables
# H1- There is a significant difference between the exang and ca variables

#alpha=0.05
from scipy.stats import chi2_contingency

contingency_table = pd.crosstab(data['exang'], data['ca'])

chi2_stat, p_value, dof, expected = chi2_contingency(contingency_table)

print("Chi-squared statistic:", chi2_stat)
print("p-value:", p_value)
```

```
Chi-squared statistic: 50.929762300055955
p-value: 2.3089396527740636e-10
```

Since the p-value is less than the alpha value of 0.05 we have to reject the null hypothesis and accept the alternative hypothesis. This means that there is a significant difference between the exang and ca variables.

```
# H0- There is no significanct difference between the sex and target variables
# H1- There is a significant difference between the sex and target variables

#alpha=0.05
from scipy.stats import chi2_contingency

contingency_table = pd.crosstab(data['sex'], data['target'])

chi2_stat, p_value, dof, expected = chi2_contingency(contingency_table)

print("Chi-squared statistic:", chi2_stat)
print("p-value:", p_value)
```

```
Chi-squared statistic: 97.48157619428007
p-value: 5.435971783244131e-23
```

Since the p-value is less than the alpha value of 0.05 we have to reject the null hypothesis and accept the alternative hypothesis. This means that there is a significant difference between the sex and target variables.

```
# H0- There is no significanct difference between the cp and target variables
# H1- There is a significant difference between the cp and target variables

#alpha=0.05
from scipy.stats import chi2_contingency

contingency_table = pd.crosstab(data['cp'], data['target'])

chi2_stat, p_value, dof, expected = chi2_contingency(contingency_table)

print("Chi-squared statistic:", chi2_stat)
print("p-value:", p_value)
```

```
Chi-squared statistic: 262.8068594193315
p-value: 1.1107521868613481e-56
```

Since the p-value is less than the alpha value of 0.05 we have to reject the null hypothesis and accept the alternative hypothesis. This means that there is a significant difference between the cp and target variables.

```
# H0- There is no significanct difference between the fbs and target variables
# H1- There is a significant difference between the fbs and target variables

#alpha=0.05
from scipy.stats import chi2_contingency

contingency_table = pd.crosstab(data['fbs'], data['target'])

chi2_stat, p_value, dof, expected = chi2_contingency(contingency_table)

print("Chi-squared statistic:", chi2_stat)
print("p-value:", p_value)
```

```
Chi-squared statistic: 1.1827684114951635
p-value: 0.27679312182966676
```

Since the p-value is higher than the alpha value of 0.05 we have to accept the null hypothesis. This means that there is no significant difference between the fbs and target variables.

```
# H0- There is no significanct difference between the restecg and target variables
# H1- There is a significant difference between the restecg and target variables

#alpha=0.05
from scipy.stats import chi2_contingency

contingency_table = pd.crosstab(data['restecg'], data['target'])

chi2_stat, p_value, dof, expected = chi2_contingency(contingency_table)

print("Chi-squared statistic:", chi2_stat)
print("p-value:", p_value)
```

```
Chi-squared statistic: 34.39391609435747
p-value: 3.3998207062427004e-08
```

Since the p-value is less than the alpha value of 0.05 we have to reject the null hypothesis and accept the alternative hypothesis. This means that there is a significant difference between the restecg and target variables.

```
# H0- There is no significanct difference between the exang and target variables
# H1- There is a significant difference between the exang and target variables

#alpha=0.05
from scipy.stats import chi2_contingency

contingency_table = pd.crosstab(data['exang'], data['target'])

chi2_stat, p_value, dof, expected = chi2_contingency(contingency_table)

print("Chi-squared statistic:", chi2_stat)
print("p-value:", p_value)
```

```
Chi-squared statistic: 180.12346747257908
p-value: 4.554727604113628e-41
```

Since the p-value is less than the alpha value of 0.05 we have to reject the null hypothesis and accept the alternative hypothesis. This means that there is a significant difference between the exang and target variables.

```
# H0- There is no significanct difference between the slope and target variables
# H1- There is a significant difference between the slope and target variables

#alpha=0.05
from scipy.stats import chi2_contingency

contingency_table = pd.crosstab(data['slope'], data['target'])

chi2_stat, p_value, dof, expected = chi2_contingency(contingency_table)

print("Chi-squared statistic:", chi2_stat)
print("p-value:", p_value)
```

```
Chi-squared statistic: 139.50146434183944
p-value: 5.100842447616183e-31
```

Since the p-value is less than the alpha value of 0.05 we have to reject the null hypothesis and accept the alternative hypothesis. This means that there is a significant difference between the slope and target variables.

```
# H0- There is no significanct difference between the ca and target variables
# H1- There is a significant difference between the ca and target variables

#alpha=0.05
from scipy.stats import chi2_contingency

contingency_table = pd.crosstab(data['ca'], data['target'])

chi2_stat, p_value, dof, expected = chi2_contingency(contingency_table)

print("Chi-squared statistic:", chi2_stat)
print("p-value:", p_value)
```

```
Chi-squared statistic: 250.2780633716478
p-value: 5.670985814186916e-53
```

Since the p-value is less than the alpha value of 0.05 we have to reject the null hypothesis and accept the alternative hypothesis. This means that there is a significant difference between the ca and target variables.

```
# H0- There is no significanct difference between the thal and target variables
# H1- There is a significant difference between the thal and target variables

#alpha=0.05
from scipy.stats import chi2_contingency

contingency_table = pd.crosstab(data['thal'], data['target'])

chi2_stat, p_value, dof, expected = chi2_contingency(contingency_table)

print("Chi-squared statistic:", chi2_stat)
print("p-value:", p_value)
```

```
Chi-squared statistic: 277.1292187088963
p-value: 8.851048373174561e-60
```

Since the p-value is less than the alpha value of 0.05 we have to reject the null hypothesis and accept the alternative hypothesis. This means that there is a significant difference between the thal and target variables.

```
data.drop('fbs',axis=1,inplace=True)
data
```

```
data.to_csv('data.csv', index=False)
```

```
import numpy as np
import pandas as pd

df=pd.read_csv('data.csv')
df
```

```
df=df.values
```

```
df
```

```
data=df[:,0:12]
target=df[:,12]
```

```
from sklearn.preprocessing import MinMaxScaler

target=np.reshape(target, (-1,1))

scaler_data = MinMaxScaler(feature_range=(0,1))
scaler_target = MinMaxScaler()

data_scaled=scaler_data.fit_transform(data)
target_scaled=scaler_target.fit_transform(target)
```

```
from sklearn.model_selection import train_test_split

train_data, test_data, train_target, test_target = train_test_split(data_scaled, target_scaled,test_size=0.1)
```

```
from keras.models import Sequential
from keras.layers import Dense,Dropout
from keras.optimizers import Adam

model = Sequential()
model.add(Dense(64,activation='relu',input_shape=(train_data.shape[1],)))
model.add(Dropout(0.2))
model.add(Dense(32, activation='relu'))
model.add(Dropout(0.2))
model.add(Dense(1, activation='sigmoid'))

optimizer = Adam(learning_rate=0.001)
model.compile(optimizer=optimizer,loss='binary_crossentropy',metrics=['accuracy'])

model.summary()
```

```
from keras.callbacks import ModelCheckpoint,EarlyStopping

checkpoint = ModelCheckpoint('models/model-{epoch:03d}.model',monitor='val_loss',save_best_only=True,mode='auto')
```

```
model.fit(train_data, train_target, epochs=500, batch_size=16, validation_split=0.1,callbacks=[checkpoint])
```

```
from matplotlib import pyplot as plt

plt.plot(model.history.history['loss'], label='Training Loss')
plt.plot(model.history.history['val_loss'], label='Validation Loss')
plt.xlabel('# epochs')
plt.ylabel('loss')
plt.legend()
plt.show()
```

```
from matplotlib import pyplot as plt

plt.plot(model.history.history['accuracy'], label='Training Accuracy')
plt.plot(model.history.history['val_accuracy'], label='Validation Accuracy')
plt.xlabel('# epochs')
plt.ylabel('accuracy')
plt.legend()
plt.show()
```

```
predicted_target = model.predict(test_data)
predicted_target
```

```
predicted_labels=np.argmax(predicted_target,axis=1)
actual_labels=np.argmax(test_target,axis=1)
```

```
from sklearn.metrics import accuracy_score

acc=accuracy_score(actual_labels,predicted_labels)
print('acc:',acc)
```

```
from sklearn.metrics import confusion_matrix
import seaborn as sns

conf_matrix = confusion_matrix(actual_labels, predicted_labels)

plt.figure(figsize=(8, 6))
sns.heatmap(conf_matrix, annot=True, fmt='g', cmap='Blues')
plt.xlabel('Predicted Label')
plt.ylabel('True Label')
plt.title('Confusion Matrix')
plt.show()
```

```
print('actual:',test_target[:10].T)
print('predicted:',predicted_target[:10].T)
```

```
import joblib

joblib.dump(scaler_data,'scaler_data.sav')
joblib.dump(scaler_target,'scaler_target.sav')
```

```
model.save("model.h5")
```

```
from keras.models import load_model
import numpy as np

model = load_model('model.h5')

import joblib

scaler_data = joblib.load('scaler_data.sav')
scaler_target = joblib.load('scaler_target.sav')

input_data = np.array([[60, 0, 0, 125, 258, 0, 141, 1, 2.8, 1, 1, 3],
                       [59, 1, 1, 140, 221, 1, 164, 1, 0.0, 2, 0, 2]])
input_data_scaled = scaler_data.transform(input_data)

predictions = model.predict(input_data_scaled)

print(predictions)
```

List of references

https://www.kaggle.com/datasets/johnsmith88/heart-disease-dataset

https://www.cdc.gov/nchs/data/hus/2017/019.pdf

https://www.ijedr.org/papers/IJEDR1704226.pdf

https://www.sciencedirect.com/science/article/pii/S187705091630638X

http://noiselab.ucsd.edu/ECE228-2020/projects/Report/72Report.pdf

https://www.researchgate.net/publication/364949647_Enhanced_accuracy_for_heart_disease_prediction_using_artificial_neural_network

https://iopscience.iop.org/article/10.1088/1742-6596/1997/1/012022