**Heart Disease Data Analysis**

**HETAL SANGANI**

Data Analyst

**Table of Contents:**

* Understanding Business Problem
* Variable Description
* Data Understanding
* Data Cleaning
* Data Visualization
* Build the model

**Understanding Business Problem:**

Cardiovascular diseases are one of the leading causes of deaths globally. To identify the causes and develop a system to predict potential heart attacks in an effective manner is necessary. The data presented has all the information about relevant factors that might have an impact on cardiovascular health. The data needs to be studied in detail for further analysis.

**Objective:** To analyze the dataset that will help to create a model that will predict the heart disease based on various input features

**Domain:** Healthcare

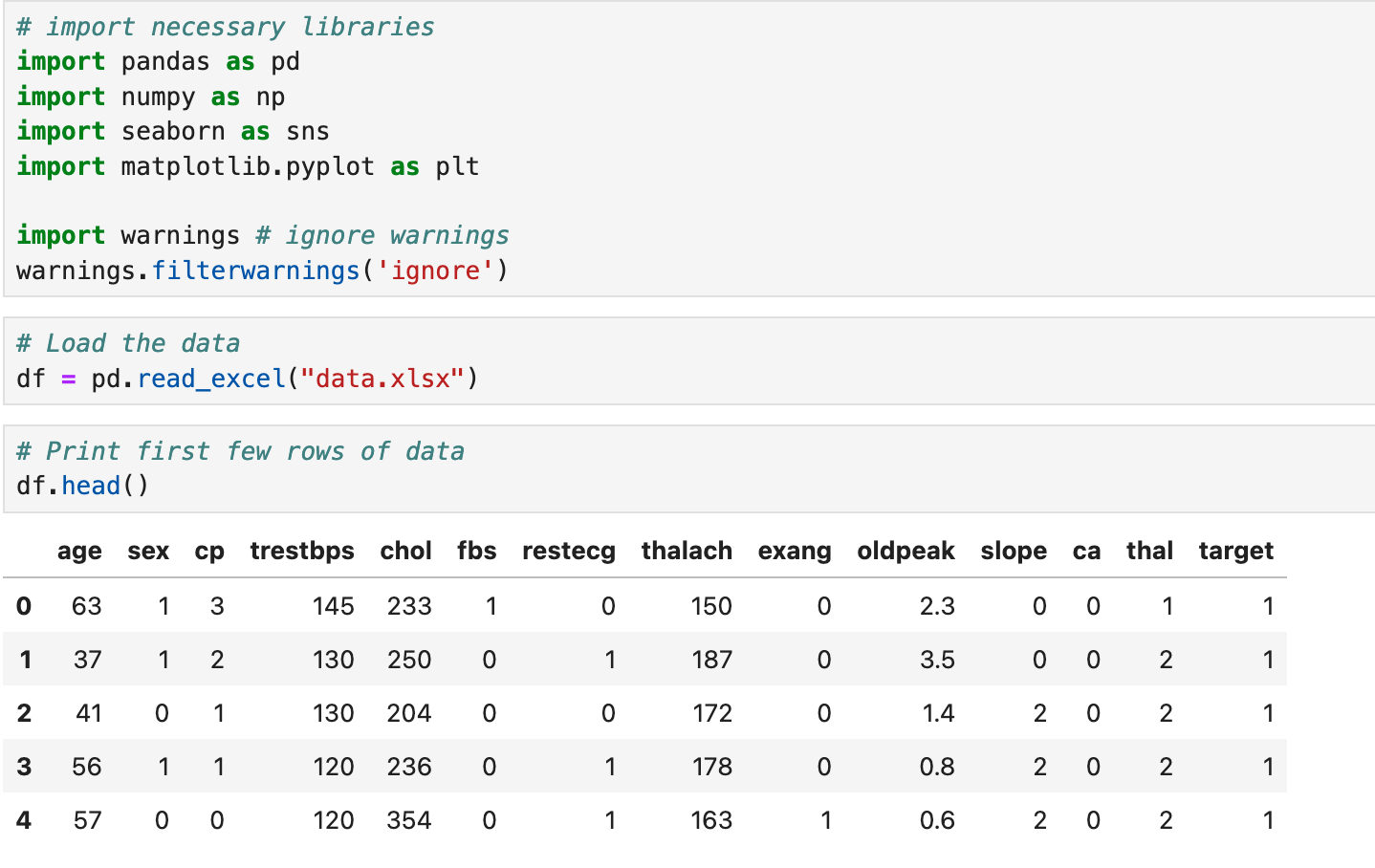
**Dataset:** Heart dataset (data.xlsx)

**Variable Description:**

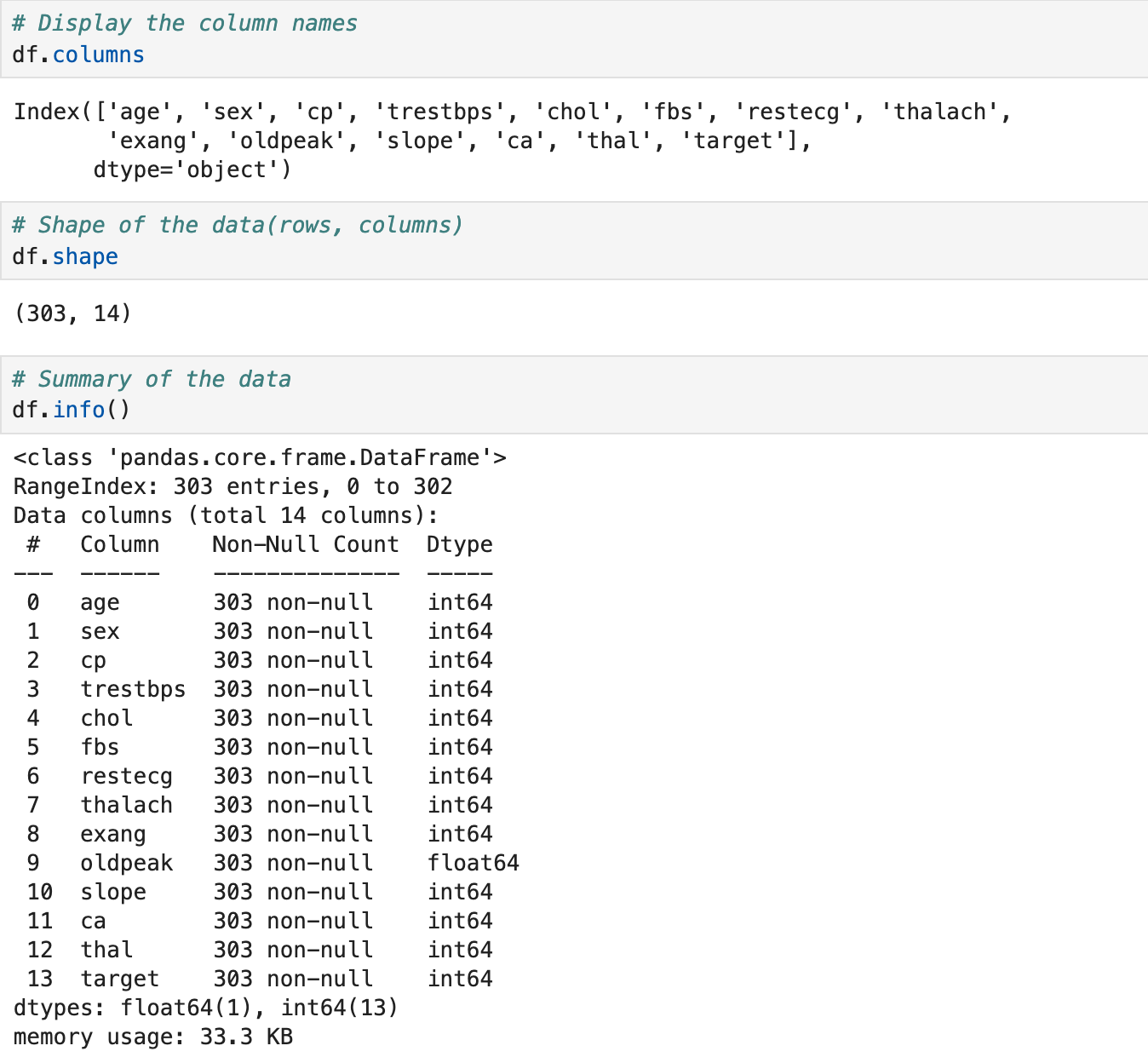
|  |  |
| --- | --- |
| **age** | age in years |
| **sex** | (1 = male, 0 = female) |
| **cp** | Chest pain type |
| **trestbps** | Resting blood pressure (in mm Hg on admission to the hospital) |
| **chol** | serum cholesterol in mg/dl |
| **fbs** | (fasting blood sugar > 120 mg/dl) (1 = true, 0 =false) |
| **restecg** | Resting electrocardiographic results |
| **thalach** | Maximum heart rate achieved |
| **exang** | Exercise induced angina (1 = yes, 0 =no) |
| **oldpeak** | ST depression induced by exercise relative to rest |
| **slope** | The slope of the peak exercise ST segment |
| **ca** | Number of major vessels (0-3) colored by flourosopy |
| **thal** | 3 = normal, 6 = fixed defect, 7 = reversable defect |
| **target** | 1 or 0 |

**Data Understanding:**

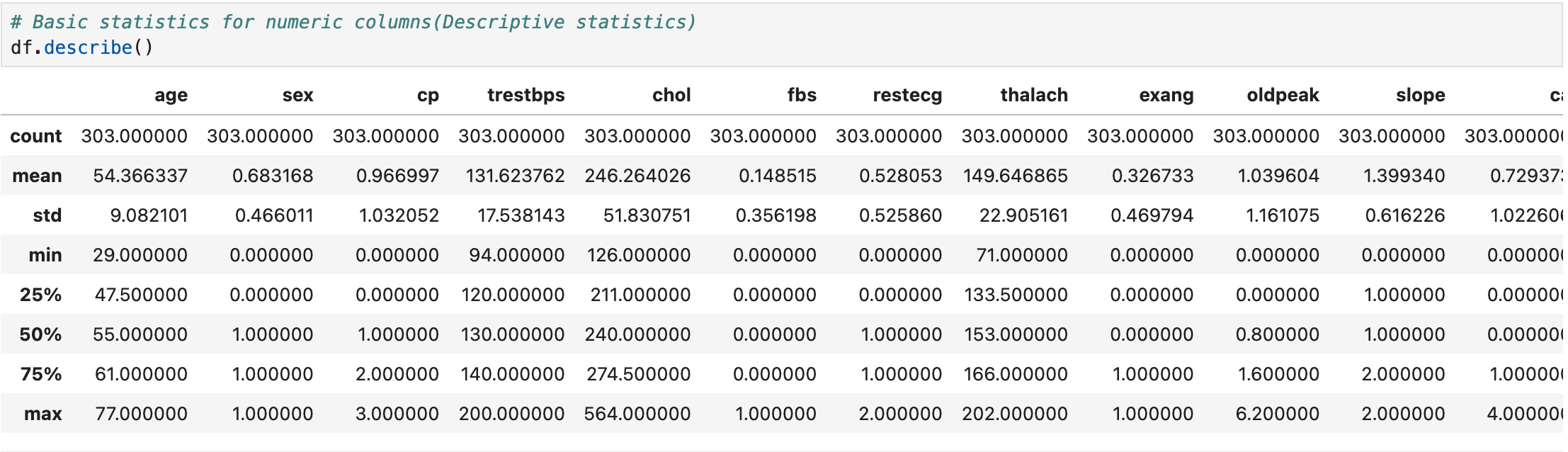
importing necessary library required and understanding the dataset



df.head() returns the first 5 rows of the dataset and df.tail() to get last 5 rows.



df.info() method returns information about the whole DataFrame including the index data type and columns, non-null values, and memory usage.



df.describe() method generates descriptive statistics for us. For numeric data, the result’s index will include count, mean, std, min, max as well as lower, 50(median) and upper percentiles.

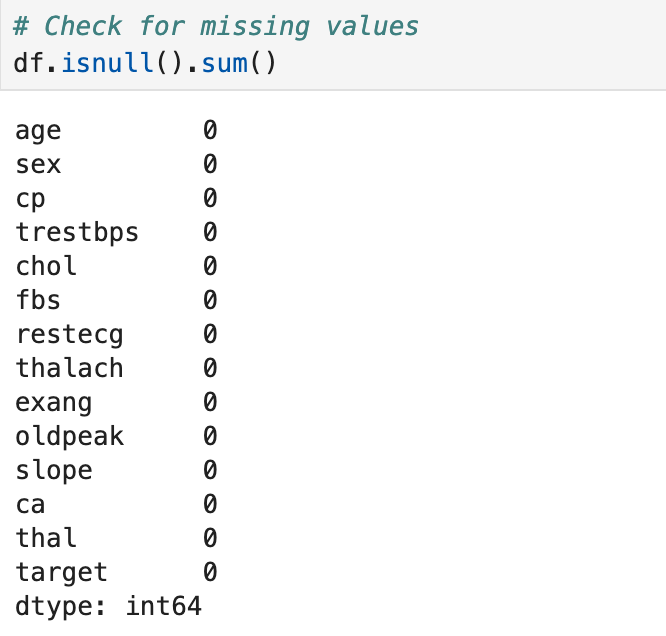
We can easily notice that the minimum age is 29 and the maximum age is 77. We can also see mean and median values of age are almost the same.

**Data Cleaning:**

In this part of the EDA. We will check :

* Missing Values
* Duplicated Values

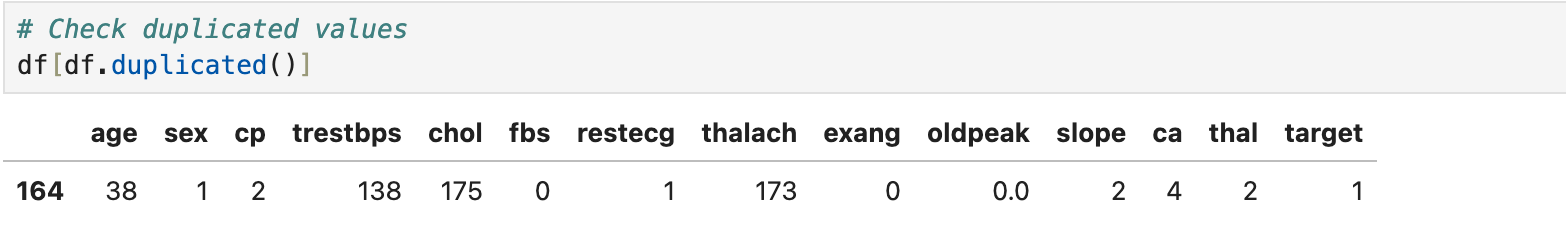
The purpose of data cleaning is to get our data ready to analyze and visualize.



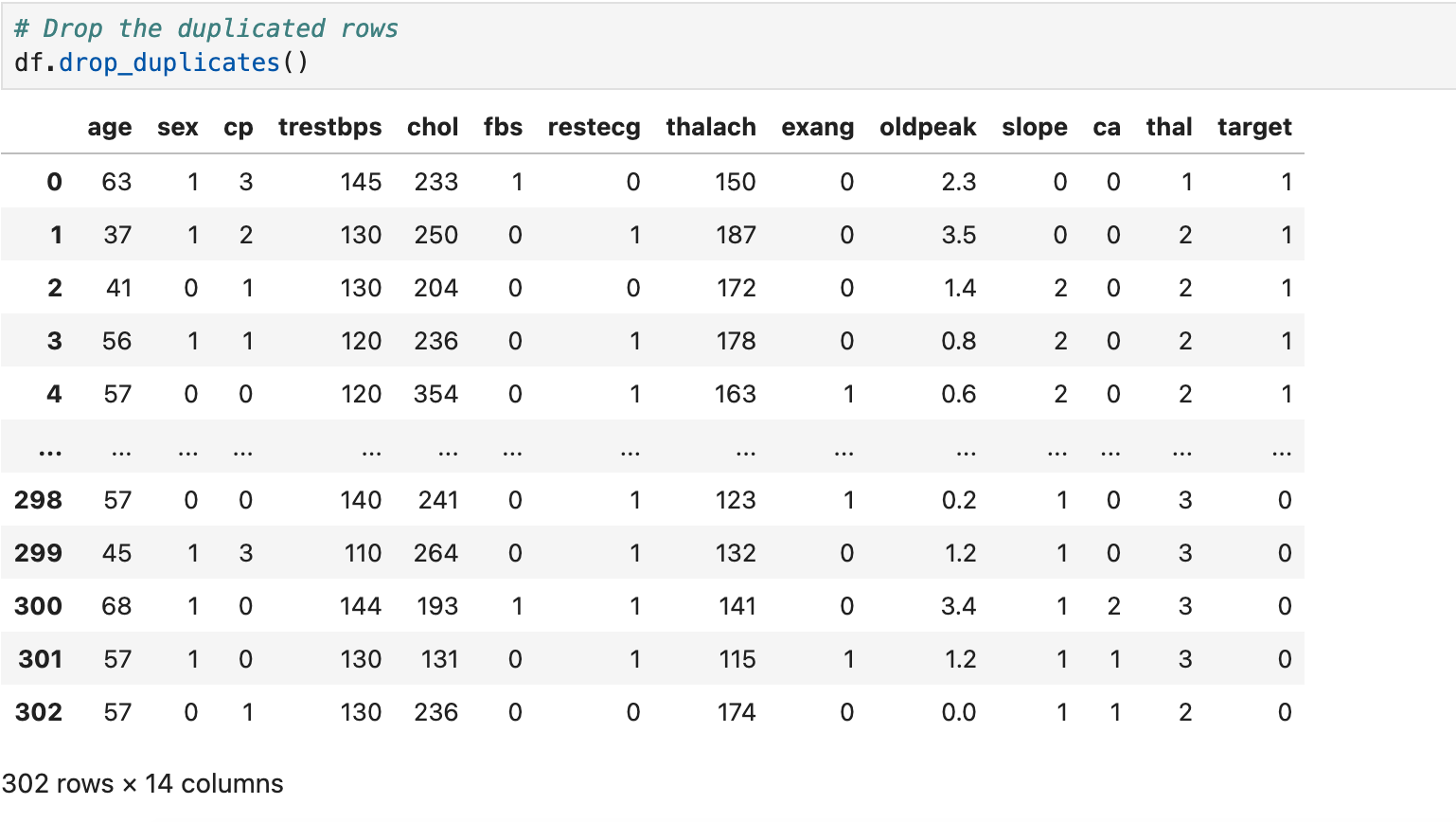
When combining .isnull() method with .sum() we can sum up all the missing values for each column.

There are no missing values in this dataset. We will now proceed to analyze the data, observe patterns, and identify outliers with the help of visualization methods.

Now we will check for duplicated values.



Dataset has only one duplicated row. We can simply drop this row using the drop\_duplicates() method.



**Data Visualization:**

Now, We understood our dataset in general and checked the missing values. We also deleted duplicated values from the data frame.

The next part is data visualization! We have to perform univariate, bivariate and multivariate analysis to see the distribution and relationship between variables.

We will use the seaborn library for statistical data visualization. Seaborn is a data visualization library based on matplotlib.

**Univariate Analysis**

The purpose of the univariate analysis is to understand the distribution of values for a single variable.

We will perform univariate analysis by using visualization techniques.

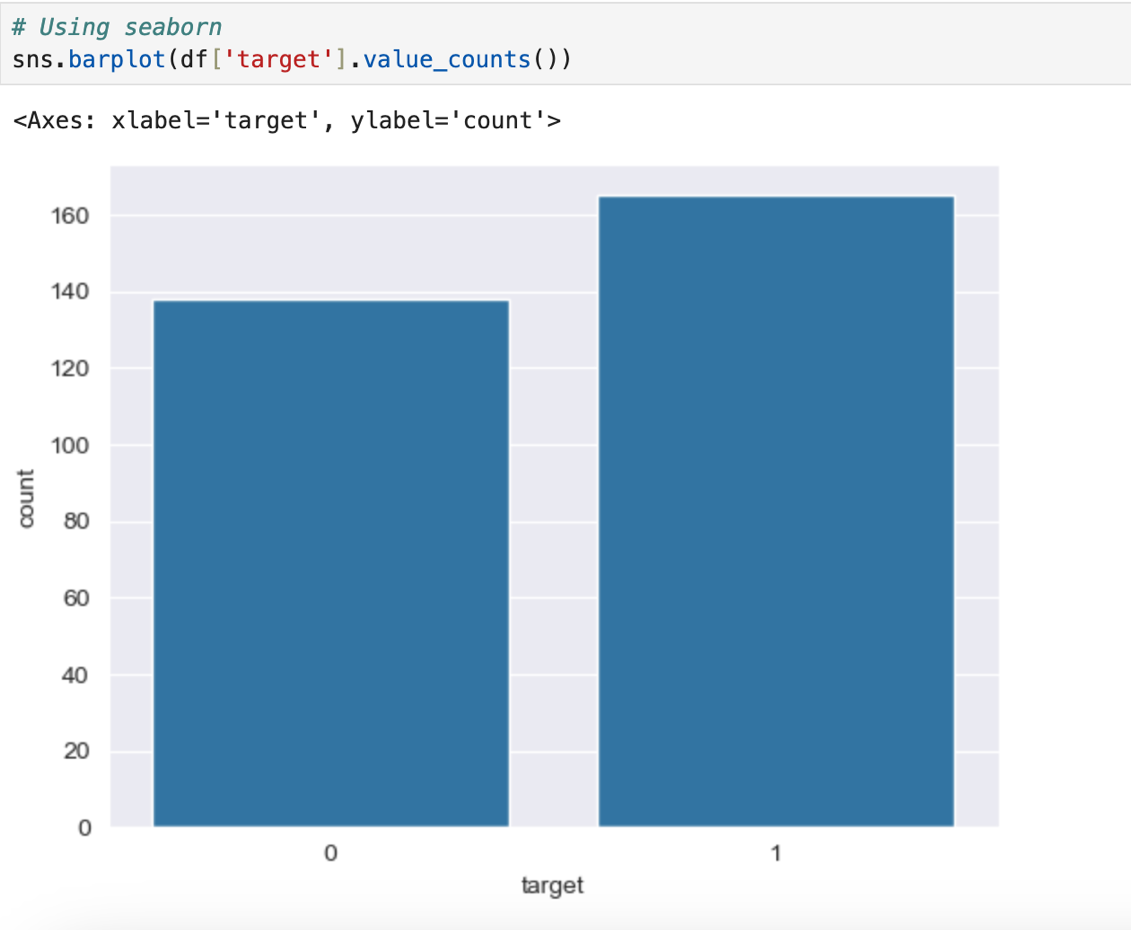
Univariate Analysis for Numerical Features

A screenshot of a computer

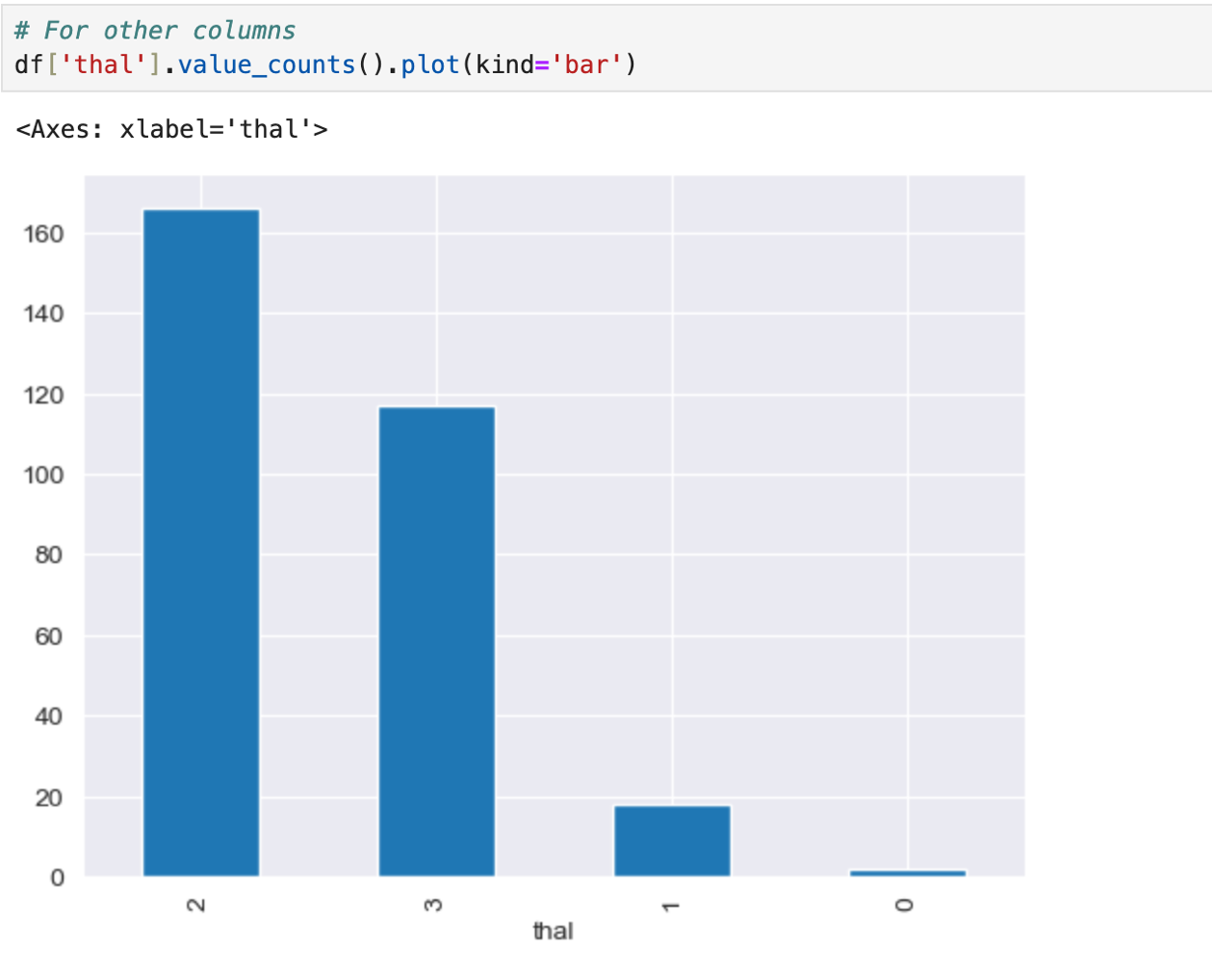
AI-generated content may be incorrect.

There are two unique values in target column.

1 = Defective Heart and 0 = Healthy Heart



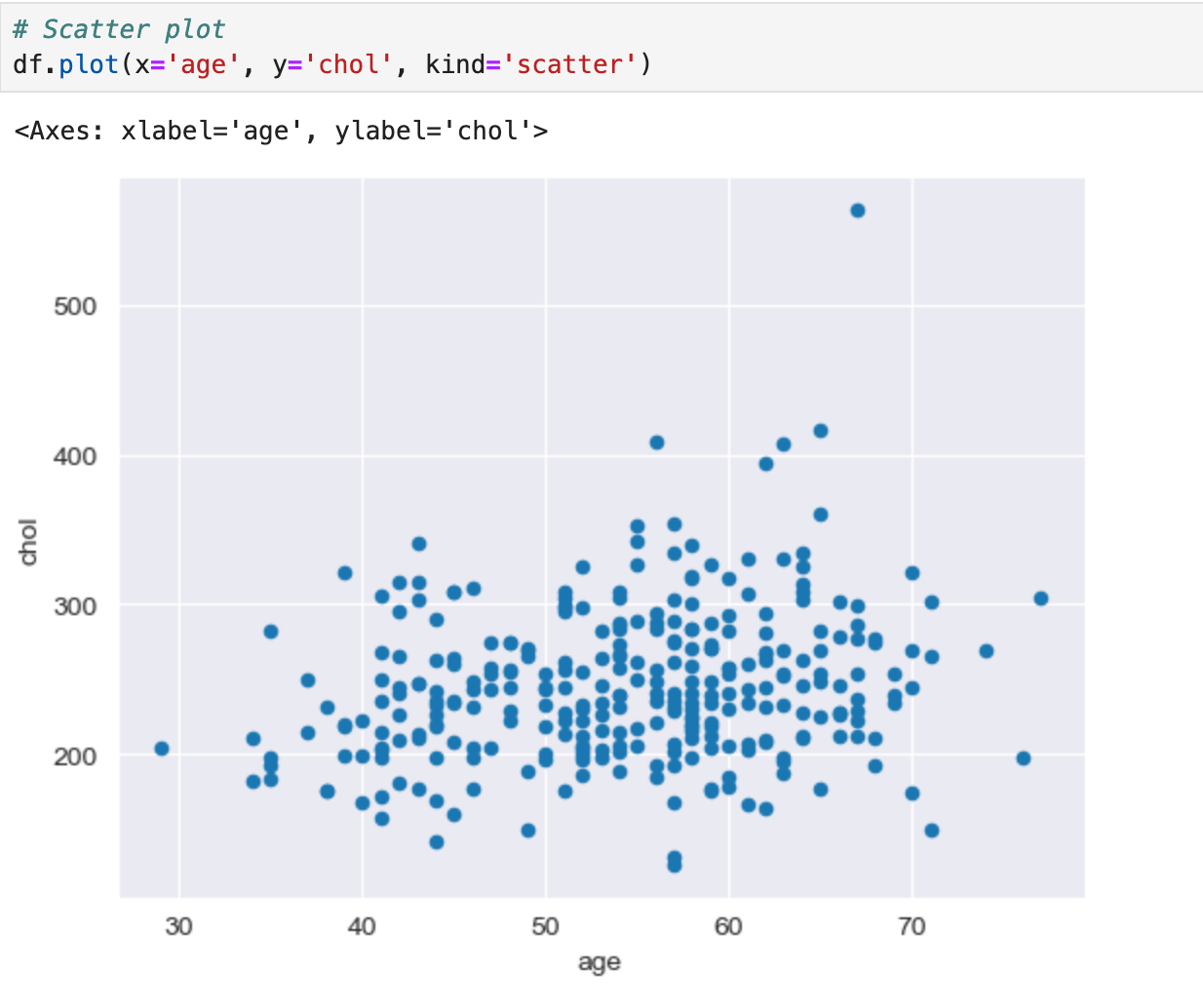
From the above bar graph, we can conclude even when the distribution is not exactly 50:50, but still the data is good enough to use on machine learning algorithms and to predict standard metrics like Accuracy scores. So, we do not need to resample this dataset.



***Bivariate Analysis***

Bivariate analysis is the analysis of exactly two variables. We will use bivariate analysis to find relationships between two variables.

For bivariate analysis, we usually use boxplot(categorical vs numerical), scatterplot(numerical vs numerical).

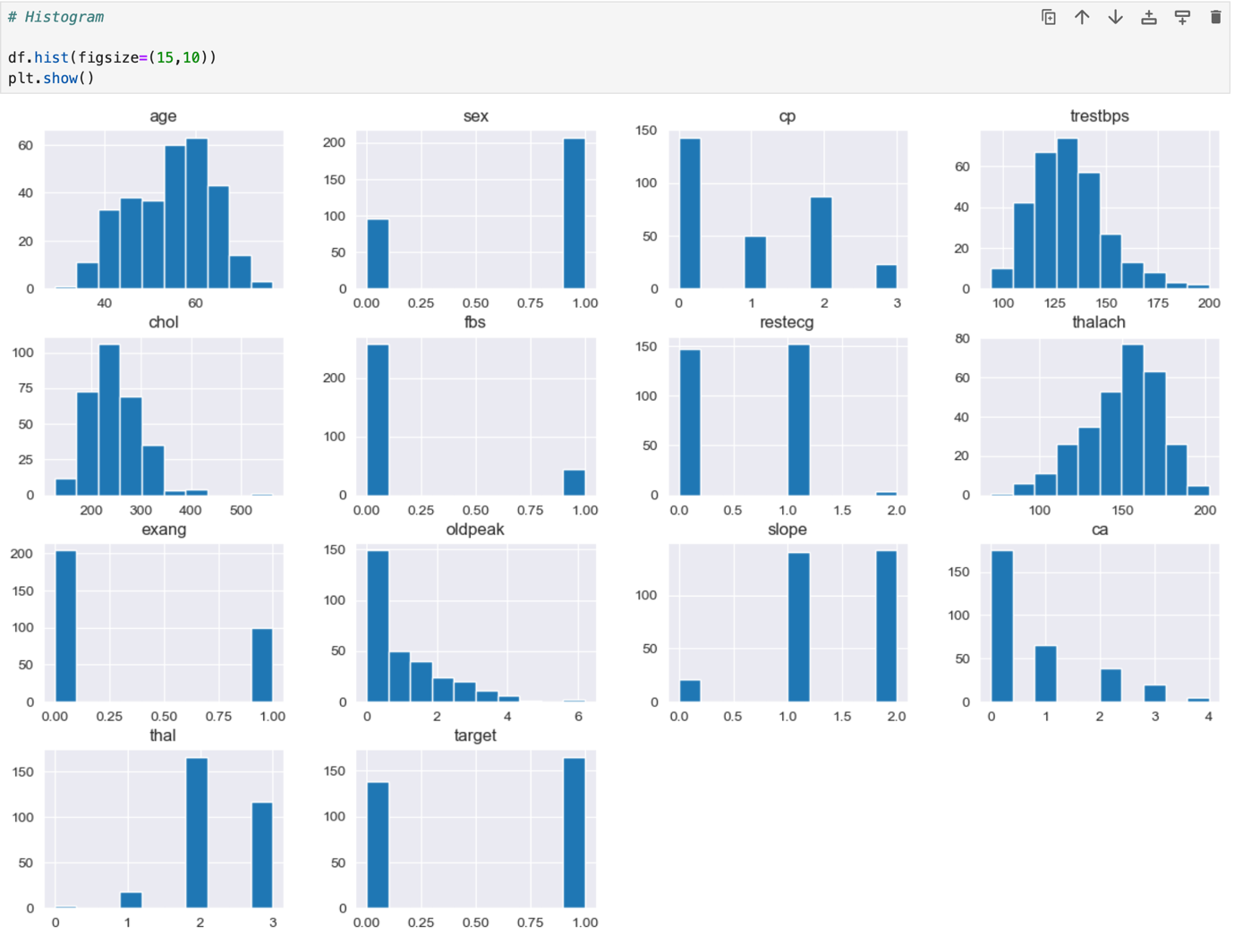


The scatterplot shows us when age is increasing, cholesterol level is also increasing. That means cholesterol levels increase with age, increases the risk of heart attack or stroke.

A screenshot of a computer screen

AI-generated content may be incorrect.

We can see that the overall shape & distribution for negative & positive patients differ vastly. Positive patients exhibit a lower median for ST depression level & thus a great distribution of their data is between 0 & 2, while negative patients are between 1 & 3. In addition, we don’t see many differences between male & female target outcomes.

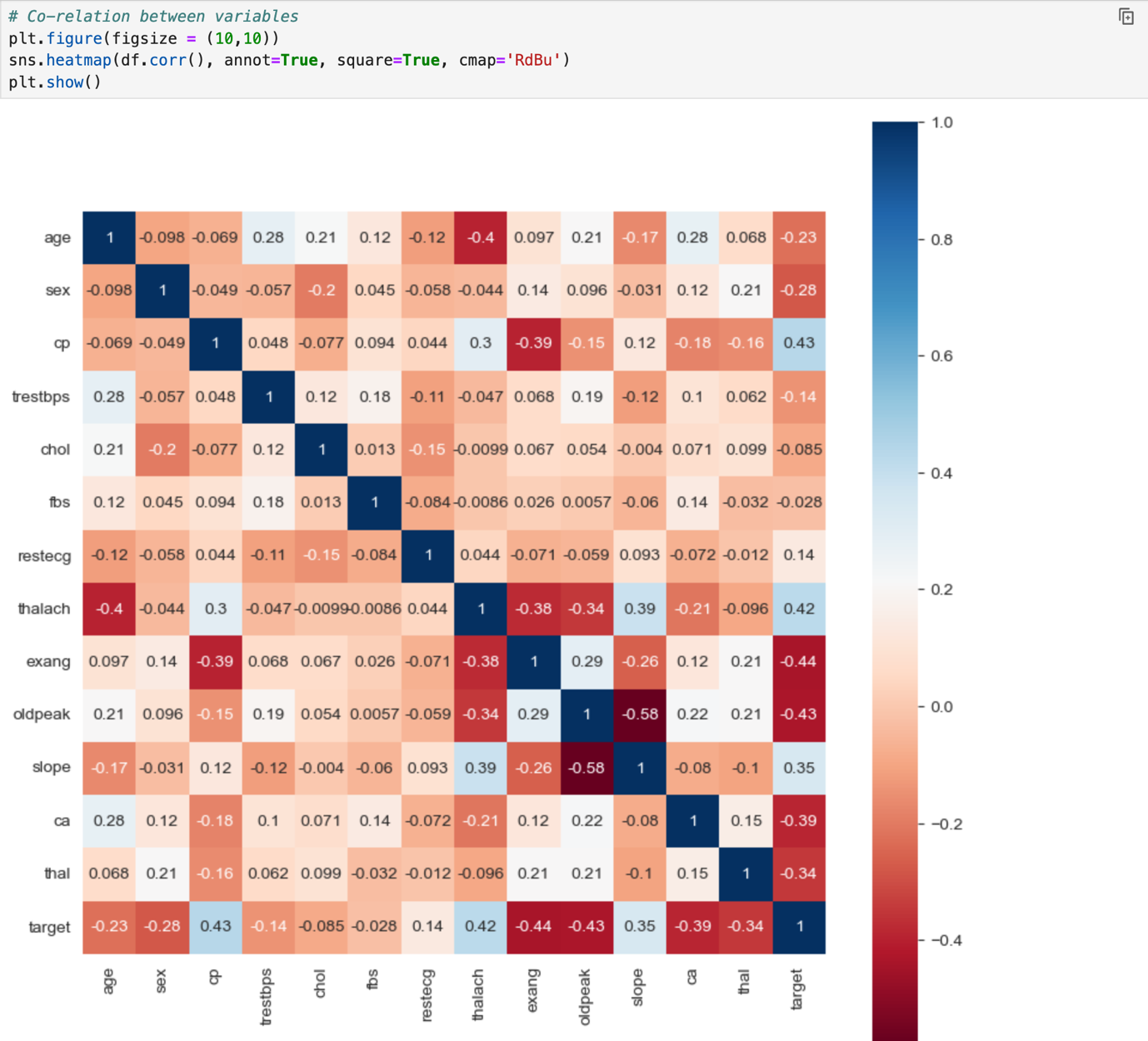


From the above histogram plots, we can see that the features are skewed and not normally distributed. Also, the scales are different between one and another.

**Correlation**

Correlation is used to test relationships between quantitative variables or categorical variables. It’s a measure of how things are related. The heatmap() method shows us the relationship between numeric variables.

We will combine the .corr() method with heatmap so that we will be able to see the relationship in the graph.



From the above Heatmap, we can see that cp and thalach are the features with highest positive correlation whereas exang, oldpeak ,ca and thal are negatively correlated. While other features do not hold much correlation with the response variable "target".

**Outlier Detection**

**A screenshot of a graph

AI-generated content may be incorrect.**

**Train – Test Split**

Now we will distribute the data into training and testing datasets using the train\_test\_split() function.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer program

AI-generated content may be incorrect.

**Conclusion**

Thus, from accuracy score we conclude a good outcome as 81% is the ideal accuracy.