VISVESVARAYA TECHNOLOGICAL UNIVERSITY
BELAGAVI-590018

An Internship/Professional work Report

on

HEART DISEASE DATA ANALYSIS

Submitted in partial fulfillment of the requirements for the final year degree in

Bachelor of Engineering in Computer Science and Engineering

of Visvesvaraya Technological University, Belagavi

Submitted by

NITISH K 1RN19CS092

KIRAN YADAV S 1RN19CS067

Carried out at

Inflow Technologies

Under the Guidance of:

Internal Guide:
Mrs. Chethana H R
Assistant Professor
Dept. of CSE

External Guide:
Mr. Arib Nawal
Inflow Technologies
Bengaluru

Department of Computer Science and Engineering

RNS Institute of Technology

(Accredited by NBA upto 30-06-2025)

Channasandra, Dr.Vishnuvardhan Road, Bengaluru-560098

2022-2023

RNS INSTITUTE OF TECHNOLOGY

Channasandra, Dr.Vishnuvardhan Road, Bengaluru-560098

DEPARTMENT OF COMPUTER SCIENCE ENGINEERING

(Accredited by NBA upto 30-06-2025)

CERTIFICATE

Certified that the Internship/Professional Practice work entitled **Heart Disease Data Analysis** has been successfully carried out at **Inflow Technologies** by **Nitish K** bearing USN **1RN19CS092** and **Kiran Yadav S** bearing USN **1RN19CS067** bonafide students of **RNS Institute of Technology** in partial fulfillment of the requirements of final year degree in **Bachelor of Engineering in Computer Science and Engineering of Visvesvaraya Technological University, Belagavi** during academic year **2022-2023**. The internship report has been approved as it satisfies the academic requirements in respect of internship work for the said degree.

Signature of the Guide	Signature of the HoD	Signature of the Principal
Mrs. Chethana H R	**Dr. Kiran P**	**Dr. M K Venkatesha**
Assistant Professor	Professor	Principal
Dept. of CSE	Dept. of CSE	RNSIT

External Viva

Name of the Examiners	**Signature with Date**
1.	
2.	

Acknowledgement

At the very onset, I would like to place on record my gratitude to all those people who have helped me in making this Internship work a reality. Our Institution has played a paramount role in guiding in the right direction. I would like to profoundly thank **Sri. Satish R Shetty**, Managing Director, RNS Group of Companies, Bengaluru for providing such a healthy environment for the successful completion of this Internship Project work.

I would like to thank our beloved Principal, **Dr. M K Venkatesha**, for providing the necessary facilities to carry out this work. I am extremely grateful to **Dr. Kiran P**, Professor and Head, Department of Computer Science and Engineering for having accepted to patronize me in the right direction with all his wisdom.

I would like to express my sincere thanks to our Coordinator and guide, **Mrs. Chethana H R**, Assistant Professor, for her constant encouragement that motivated me for the successful completion of this work. Last but not the least, I am thankful to all the teaching and non-teaching staff members of the Computer Science and Engineering Department for their encouragement and support throughout this work.

Nitish K 1RN19CS092

Kiran Yadav S 1RN19CS092

Abstract

Heart is one of the most vital organs in the human body. The term “heart disease” refers to several types of heart conditions. The diagnosis of heart disease in most cases depends on a complex combination of clinical and pathological data. Because of this complexity, there exists a significant amount of interest among clinical professionals and researchers regarding the efficient and accurate prediction of heart disease. In this paper, we develop a heart disease predict system that can assist medical professionals in predicting heart disease status based on the clinical data of patients. Our approaches include three steps.

We select 13 important clinical features, i.e., age, sex, chest pain type, trestbps, cholesterol, fasting blood sugar, resting ecg, max heart rate, exercise induced angina, old peak, slope, number of vessels colored, and thal. Then we develop a classification algorithm for classifying heart disease based on these clinical features.The accuracy of prediction is near 84 percent .At last we have deployed our app on streamlit cloud.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Organization

1.1.1 Company Profile

Inflow Technologies is a niche player in the Distribution Services industry providing value added distribution in Cyber Security, Networking, Unified Communications and Collaboration, AIDC & POS, Infrastructure & Application Software, Storage Management and Electronic Security products related Services in South Asia. Inflow Technologies enables system integrators & resellers to design, deploy and adopt IT technologies to facilitate their customer needs

1.1.2 Domain / Technology

Data analytics is the process of examining data sets in order to find trends and draw conclusions about the information they contain. This technology is widely used in commercial industries to enable organizations to make more-informed business decisions. They help businesses increase revenue, improve operational efficiency, optimize marketing campaigns and bolster customer service efforts. This also enables organizations to respond quickly to emerging market trends and gain a competitive edge over business rivals. Scientists and researchers makes use of analytics tools to verify or disprove scientific models, theories and hypotheses. Depending on the particular application, the data that's analyzed can consist of either historical records or new information that has been processed for the real-time analytics.

In our increasingly data-driven world, it's more important than ever to have accessible ways to view and understand data. After all, the demand for data skills in employees is steadily increasing each year. Employees and business owners at every level need to have an understanding of data and of its impact. That's where data visualization comes in handy. With the goal of making data more accessible and understandable, data visualization in the form of dashboards is the go-to tool for many businesses to analyze and share information.

1.2 Problem Statement

1.2.1 Problem Definition

The major challenge in heart disease is its detection. Early detection of cardiac diseases can decrease the mortality rate and overall complications. However, it is not possible to monitor patients everyday in all cases accurately and consultation of a patient for 24 hours by a doctor is not available since it requires more patience, time and expertise. Since we have a good amount of data in today's world, we can use various machine learning algorithms to analyze the data for hidden patterns. The hidden patterns can be used for health diagnosis in medicinal data.

1.2.2 Problem Formulation

Data visualization helps people interact with, and better understand data. Whether simple or complex, the right visualization can bring everyone on the same page, regardless of their level of expertise. The purpose of this data exploration and predictive analysis is to better understand which health factors affect a patient's risk for heart disease. To accomplish this, an introduction to the data will be made, along with a graphical analysis of the health factors in the dataset. The predictive modeling process will be introduced, giving the background for the evaluation of the logistic regression predictive model. This evaluation will consist of reviewing performance metrics from the confusion matrix. Finally, an explanation of the model's calculation will be given for a specific example to demonstrate how the factors drove the prediction

Chapter 2

Requirement Analysis, Tools & Technologies

2.1 Hardware Requirements

The Hardware requirements are very minimal and the program can be run on most of the machines.Table 2.1 gives details of hardware requirements.

Table 2.1: Hardware Requirements

Processor	Intel Core i3 processor
Processor Speed	1.70 GHz
RAM	4 GB
Storage Space	40 GB
Monitor Resolution	1024*768 or 1336*768 or 1280*1024

2.2 Software Requirements

The software requirements are description of features and functionalities of the system.Table 2.2 gives details of software requirements.

Table 2.2: Software Requirements

Operating System	Windows 8.1
IDE	Anaconda
Tools	Tableau
Libraries	Pandas, Numpy, Streamlit, Matplotlib, Seaborn

2.2.1 Anaconda

Anaconda is the birthplace of Python data science. It is a distribution of the Python and R programming languages for scientific computing, that aims to simplify package management and deployment. The distribution includes data-science packages suitable for Windows, Linux, and macOS. It is developed and maintained by Anaconda, Inc., which was founded by Peter Wang and Travis Oliphant in 2012. Anaconda distribution comes with over 250 packages automatically installed, and over 7,500 additional open-source packages can be installed from PyPI as well as the conda package and virtual environment manager. It also includes a GUI, Anaconda Navigator, as a graphical alternative to the command-line interface.

2.2.2 Jupyter Notebook

Jupyter Notebook (formerly IPython Notebook) is a web-based interactive computational environment for creating notebook documents. Jupyter Notebook is built using several open-source libraries, including IPython, ZeroMQ, Tornado, jQuery, Bootstrap, and MathJax. A Jupyter Notebook document is a browser-based REPL containing an ordered list of input/output cells which can contain code, text (using Markdown), mathematics, plots and rich media. Underneath the interface, a notebook is a JSON document, following a versioned schema, usually ending with the ".ipynb" extension. Jupyter Notebook is similar to the notebook interface of other programs such as Maple, Mathematica, and SageMath, a computational interface style that originated with Mathematica in the 1980s. Jupyter interest overtook the popularity of the Mathematica notebook interface in early 2018.

2.2.3 Tableau

There are dozens of tools for data visualization and data analysis. These include Google Charts, Tableau, Grafana, Chartist, FusionCharts, Datawrapper, Infogram. Among all these, tableau is simple and easy to use. Tableau is considered as one of an excellent data visualization and business intelligence tool used for reporting and analyzing vast volumes of data. It is an American company that started in 2003—in June 2019, Salesforce acquired Tableau. It helps users create different charts, graphs, maps, dashboards, and stories for visualizing and analyzing data, which inturn helps in making business decisions.

Tableau Features

- Tableau supports powerful data discovery and exploration that enables users to answer important questions in seconds
- No prior programming knowledge is needed; users without relevant experience can start immediately with creating visualizations using Tableau
- It can connect to several data sources that other BI tools do not support. Tableau enables users to create reports by joining and blending different datasets
- Tableau Server supports a centralized location to manage all published data sources within an organization
- Tableau software's drag-and-drop features, intuitive drill-down capabilities, and natural language querying (that we'll share more about later), new users and data analysts alike can quickly create visualizations and dashboards to gain insight almost instantly
- Tableau software's role-based permissions you can manage who has access to what data down to the row, and you can even define who can make changes to every data source or workbook
- All views and dashboards in Tableau software are mobile and tablet compatible
- Tableau software's Ask Data feature can be a time-saver when it comes to asking simple questions and needing to create quick visualizations
- Additionally, sharing in Tableau software is especially powerful because, instead of sending a static report, you can share a dashboard that is interactive and provides more than just a single view
- Tableau software's expansive and supportive community ignites learning across the organization and equips employees with extensive training and tutorial options, collaborative forums, and support

Chapter 3

Design & Implementation

3.1 Data Analysis Life Cycle

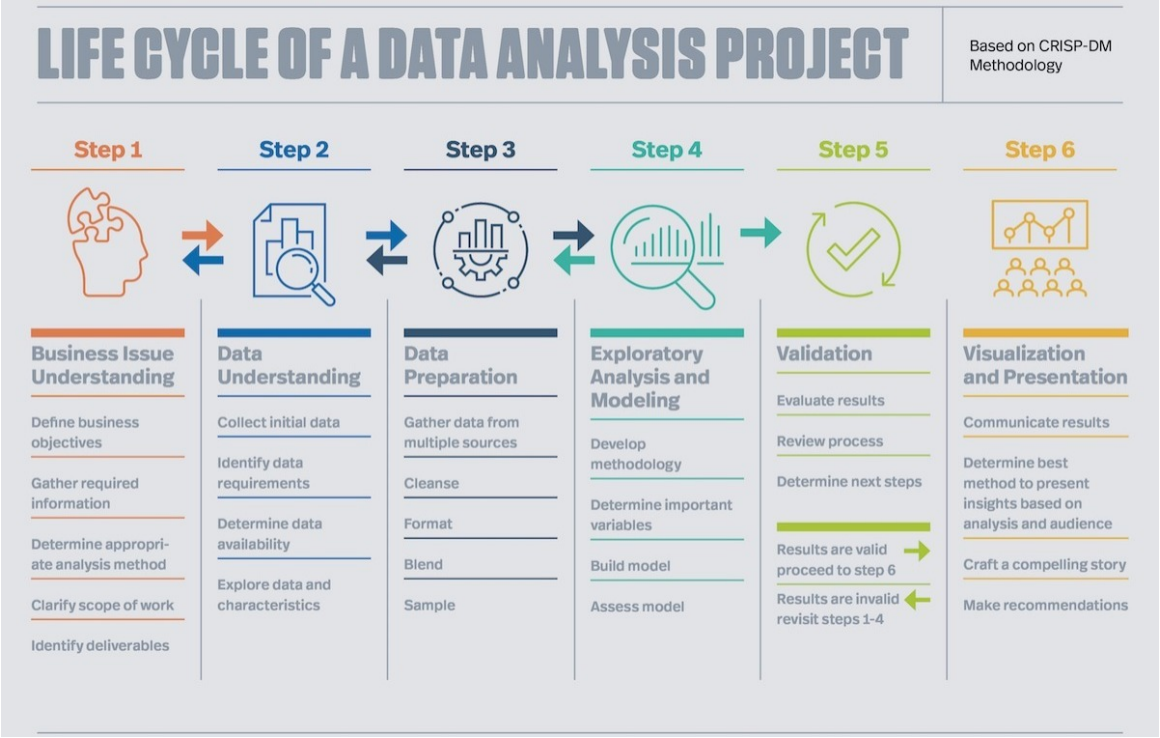

Figure 3.1: Data Analysis Life Cycle

The data analysis lifecycle describes the process of conducting a data analytics project, which consists of six key steps based on the CRISP-DM methodology. Data analysis is the process of examining data sets in order to find trends and draw conclusions about the information they contain. Increasingly, data analytics is done with the aid of specialized systems and software.

When presented with a data project, you will be given a brief outline of the expectations. From that outline, you should identify the key objectives that the business is trying to uncover. When presented with a small dataset, you can use tools like Excel, R, Python, Tableau Prep or Tableau Desktop to help prepare your data for it's cleaning. Once you have organized and identified all the variables in your dataset, you can begin cleaning. Using different statistical modeling methods, you can determine which is the best. Interactive visualization tools like Tableau are tremendously useful in illustrating your conclusions to clients. Being able to tell a story with your data is essential.

3.2 Libraries

3.2.1 NumPy

NumPy is a library for the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays. It also has functions for working in domain of linear algebra, fourier transform, and matrices. NumPy was created in 2005 by Travis Oliphant. It is an open source project and you can use it freely. NumPy stands for Numerical Python.

Main Features

- NumPy aims to provide an array object that is up to 50x faster than traditional Python lists. The array object in NumPy is called ndarray, it provides a lot of supporting functions that make working with ndarray very easy. It is optimized to work with latest CPU architectures.
- We can use the functions in NumPy to work with code written in other languages. We can hence integrate the functionalities available in various programming languages. This helps implement inter-platform functions.
- It has the capability to perform complex operations of the elements like linear algebra, Fourier transform, etc. We have separate modules for each of the complex functions. We have the linalg module for linear algebra functions.

3.2.2 Pandas

pandas is a Python package that provides fast, flexible, and expressive data structures designed to make working with ”relational” or ”labeled” data both easy and intuitive. It aims to be the fundamental high-level building block for doing practical, real world data analysis in Python. Additionally, it has the broader goal of becoming the most powerful and flexible open source data analysis / manipulation tool available in any language. It is already well on its way towards this goal.

Main Features

- Easy handling of missing data (represented as NaN, NA, or NaT) in floating point as well as non-floating point data
- Size mutability: columns can be inserted and deleted from DataFrame and higher dimensional objects.
- Automatic and explicit data alignment: objects can be explicitly aligned to a set of labels, or the user can simply ignore the labels and let Series, DataFrame, etc. automatically align the data for you in computations.
- Powerful, flexible group by functionality to perform split-apply-combine operations on data sets, for both aggregating and transforming data.
- Make it easy to convert ragged, differently-indexed data in other Python and NumPy data structures into DataFrame objects
- Intelligent label-based slicing, fancy indexing, and subsetting of large data sets.
- Intuitive merging and joining data sets.
- Flexible reshaping and pivoting of data sets.
- Hierarchical labeling of axes (possible to have multiple labels per tick).
- Robust IO tools for loading data from flat files (CSV and delimited), Excel files, databases, and saving/loading data from the ultrafast HDF5 format.

3.2.3 Matplotlib

Matplotlib is a low level graph plotting library in python that serves as a visualization utility. It was created by John D. Hunter. It is open source and we can use it freely. Matplotlib is mostly written in python, a few segments are written in C, Objective-C and Javascript for Platform compatibility.

Main Features

- Creates publication quality plots.
- Makes interactive figures that can zoom, pan, update.
- Customizes visual style and layout.
- Export to many file formats.
- Can be embedded in JupyterLab and Graphical User Interfaces.
- Uses a rich array of third-party packages built on Matplotlib.

3.2.4 Seaborn

Seaborn is an amazing visualization library for statistical graphics plotting in Python. It provides beautiful default styles and color palettes to make statistical plots more attractive. It is built on the top of matplotlib library and also closely integrated to the data structures from pandas. It provides dataset-oriented APIs, so that we can switch between different visual representations for same variables.

Main Features

- Built in themes for styling matplotlib graphics.
- Visualizing univariate and bivariate data.
- Fitting in and visualizing linear regression models.
- Seaborn works well with NumPy and Pandas data structures.
- It comes with built in themes for styling Matplotlib graphics.

3.2.5 Streamlit

Streamlit is a free and open-source framework to rapidly build and share beautiful machine learning and data science web apps. It is a Python-based library specifically designed for machine learning engineers. Data scientists or machine learning engineers are not web developers and they're not interested in spending weeks learning to use these frameworks to build web apps. Instead, they want a tool that is easier to learn and to use, as long as it can display data and collect needed parameters for modeling. Streamlit allows you to create a stunning-looking application with only a few lines of code.

The best thing about Streamlit is that you don't even need to know the basics of web development to get started or to create your first web application. So if you're somebody who's into data science and you want to deploy your models easily, quickly, and with only a few lines of code, Streamlit is a good fit. One of the important aspects of making an application successful is to deliver it with an effective and intuitive user interface. Many of the modern data-heavy apps face the challenge of building an effective user interface quickly, without taking complicated steps. Streamlit is a promising open-source Python library, which enables developers to build attractive user interfaces in no time.

Main Features

- In addition to the ability to store and persist state, Streamlit also exposes the ability to manipulate state using Callbacks.
- You don't need to spend days or months to create a web app, you can create a really beautiful machine learning or data science app in only a few hours or even minutes.
- It is compatible with the majority of Python libraries (e.g. pandas, matplotlib, seaborn, plotly, Keras, PyTorch, SymPy(latex)).
- No front-end (html, js, css) experience or knowledge is required.
- Less code is needed to create amazing web apps.
- Data caching simplifies and speeds up computation pipelines.

3.3 Dataset

A data set is a collection of related, discrete items of related data that may be accessed individually or in combination or managed as a whole entity. A data set is organized into some type of data structure. For IPL Data Analysis, we have used datasets from kaggle. The two different datasets, from 2008 up to 2021, matches and ball by ball analysis datasets separately.

```
<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 3.2: Description of Match Analysis Dataset

Once data is organized, cleaning need to be done. Data cleaning is the process of fixing or removing incorrect, corrupted, incorrectly formatted, duplicate, or incomplete data within a dataset. When combining multiple data sources, there are many opportunities for data to be duplicated or mislabeled. Unclean data normally comes as a result of human error, scraping data, or combining data from multiple sources. Irrelevant data will slow down and confuse any analysis, hence it is essential to drop out these data. After data cleaning, the matches and ball by ball delivery datasets are merged into one dataset for analysis. Below figure shows the description of merged dataset.

3.4 Code Segment

Code for heartPrediction

```
%%writefile heartPrediction.py
import pickle
import streamlit as st
st.title("Heart Disease Prediction")
st.title("Author Nitish")
st_age          = st.slider('Age',0,100,20)
st_sex          = st.slider('Sex: Male=1, Female=0',0,1,1)
st_cp           = st.slider("Chest pain:",0,3,1)
st_trestbps = st.slider("Resting Bp in mm Hg:",100,400,50)
st_chol         = st.slider("Cholesterol in mg/dl:",100,400,50)
st_fbs          = st.slider("Fasting blood sugar > 120 mg/dl: True
   =1,False=0",0,1,0)
st_restecg  = st.slider("Resting ECG results:",0,2,1)
st_thalach  = st.slider("Maximum heart rate:",60,200,30)
st_exang    = st.slider("Exercize induced angina Yes=1,No=0:"
   ,0,1,0)
st_old      = st.slider("Oldpeak,ST depression induced by exercise
   relative to rest:",0,10,5)
st_slope        = st.slider("Slope of peak exercise ST segment, 1=
   unsloping 2= flat 3= downsloping:",1,3,2)
st_ca       = st.slider("No. of major vessels(0-3):",0,3,2)
st_thal     = st.slider("Thal normal=1,fixed defect=2,reversable
   defect=3",1,3,2)
st_target   = ['Less chance of heart attack','More chance of heart
   attack']

new = pickle.load(open('test','rb'))
```

```
y_pred = new.predict([[st_age,st_sex,st_cp,st_trestbps,st_chol,
    st_fbs,st_restecg,st_thalach,st_exang,st_old,st_slope,st_ca,
    st_thal]])
y_pred = st_target[y_pred[0]]
clicked = st.button("Test")
if(clicked):
        if(y_pred):
                st.success('Prediction Successful')
                y_pred
        else:
                st.error('Error running analysis')
```

3.5 Algorithms

3.5.1 Linear Regression

Regression models a target prediction value based on independent variables. It is mostly used for finding out the relationship between variables and forecasting. Different regression models differ based on – the kind of relationship between dependent and independent variables they are considering, and the number of independent variables getting used. In linear regression analysis is used to predict the value of a variable based on the value of another variable.The variable you want to predict is called the dependent variable. The variable you are using to predict the other variable's value is called the independent variable.

```
from sklearn.linear_model import LogisticRegression
lr = LogisticRegression()
lr.fit(X_train,Y_train)
Y_pred = lr.predict(X_test)
score = round(accuracy_score(Y_pred,Y_test)*100,2)
```

```
print('Accuracy score using Logistic Regression is : '+str(score)+'
    %')
```

3.5.2 Decision Tree

Decision Tree is a Supervised learning technique that can be used for both classification and Regression problems, but mostly it is preferred for solving Classification problems. It is a tree-structured classifier, where internal nodes represent the features of a dataset, branches represent the decision rules and each leaf node represents the outcome. A decision tree simply asks a question, and based on the answer (Yes/No), it further split the tree into subtrees. In a Decision tree, there are two nodes, which are the Decision Node and Leaf Node. Decision nodes are used to make any decision and have multiple branches, whereas Leaf nodes are the output of those decisions and do not contain any further branches.

```
def splitdataset(data):
    X=data.values[:,0:12]
    Y=data.values[:,13]
    X_train, X_test, y_train, y_test=train_test_split(X,Y, test_size
        =0.3, random_state=100)
    return X, Y, X_train, X_test, y_train, y_test

def train_gini(X_train, X_test, y_train):
    clf_gini = DecisionTreeClassifier(criterion = "gini",
            random_state = 100,max_depth=3, min_samples_leaf=5)

    clf_gini.fit(X_train, y_train)
    return clf_gini

def tarin_using_entropy(X_train, X_test, y_train):
```

```
    # Decision tree with entropy
    clf_entropy = DecisionTreeClassifier(
            criterion = "entropy", random_state = 100,
            max_depth = 3, min_samples_leaf = 5)

    # Performing training
    clf_entropy.fit(X_train, y_train)
    return clf_entropy

def prediction(X_test,clf_object):
    y_pred = clf_object.predict(X_test)
    print("Predicted Values:")
    print(y_pred)
    return y_pred

def cal_accuracy(y_test, y_pred):

    print("Confusion Matrix: ",
        confusion_matrix(y_test, y_pred))

    print ("Accuracy : ",
    accuracy_score(y_test,y_pred)*100)

    print("Report : ",
    classification_report(y_test, y_pred))

def main():
    data = pd.read_csv("heart.csv")
    X, Y, X_train, X_test, y_train, y_test = splitdataset(data)
```

```
    clf_gini = train_gini(X_train, X_test, y_train)
    clf_entropy = tarin_using_entropy(X_train, X_test, y_train)

    # Operational Phase
    print("Results Using Gini Index:")
    print("\n")

    # Prediction using gini
    y_pred_gini = prediction(X_test, clf_gini)
    cal_accuracy(y_test, y_pred_gini)

    print("\n")
    print("-"*120)
    print("\n\n")

    y_pred_entropy = prediction(X_test, clf_entropy)
    cal_accuracy(y_test, y_pred_entropy)

    print("\n")
    print("-"*120)
    print("\n\n")

if __name__ == "__main__":
    main()
```

3.6 Metrics

Mean Absolute Error - MAE

It is used to measure the accuracy of a given machine learning model by calculating the average difference between the calculated values and actual values.

$$MAE = \frac{\sum |actual_i - predicted_i|}{n}$$

Root Mean Square Error - RMSE

It is the square root of the average of the squared differences between the estimated and the actual value of the variable/feature.

$$RSME = \sqrt[2]{\frac{\sum_{n=1}^{N}(predicted_i - actual_i)^2}{n}}$$

Mean Squared Error - MSE

It measures the average of error squares i.e. the average squared difference between the estimated values and true value.

$$MSE = \frac{\sum_{n=1}^{n}(predicted_i - actual_i)^2}{n}$$

R Square Score - R^2

It is a measure that provides information about the goodness of fit of a model. In the context of regression, it is a statistical measure of how well the regression line approximates the actual data. Here, SSR and SST stands for Sum Squared Regression and Total Sum Of Squares respectively.

$$R^2 = 1 - \frac{\sum(actual_i - predicted_i)^2}{\sum(actual_i - mean)^2}$$

$$R^2 = 1 - \frac{SSR}{SST}$$

Chapter 4

Observations & Results

4.1 Testing

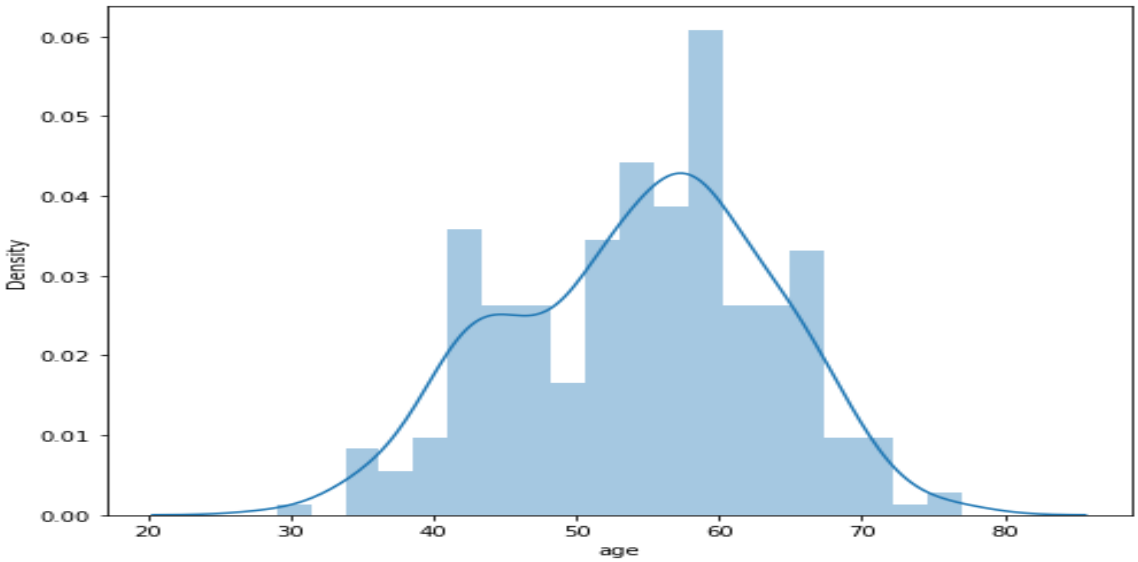

Figure 4.1: Check age distribution in dataset

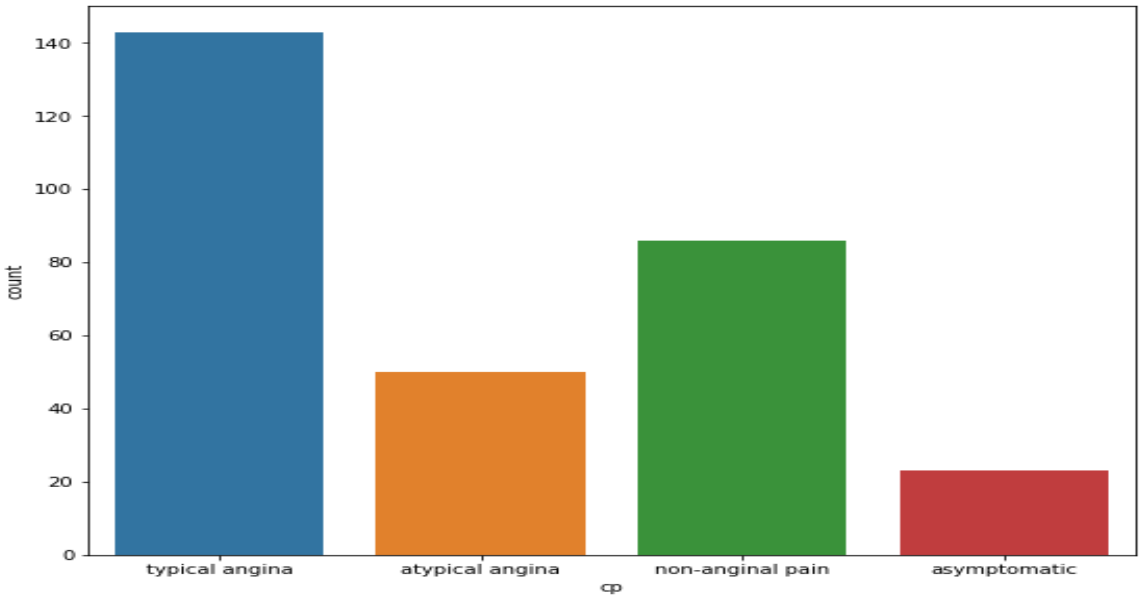

Figure 4.2: check chest pain in dataset

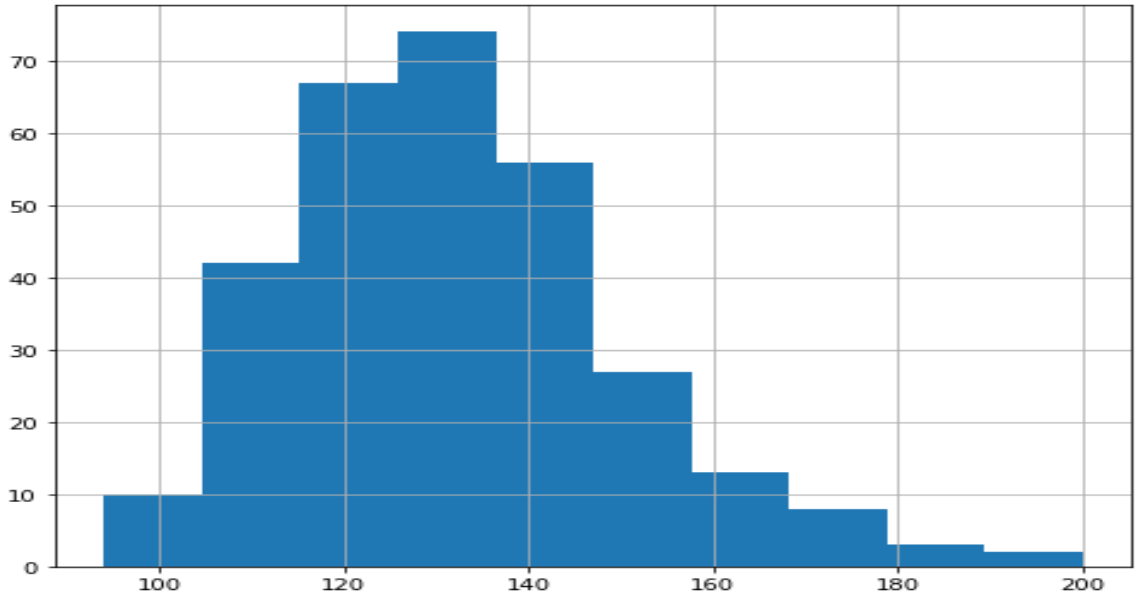

Figure 4.3: check resting blood pressure distribution

Figure 4.4: Heart Disease Analysis

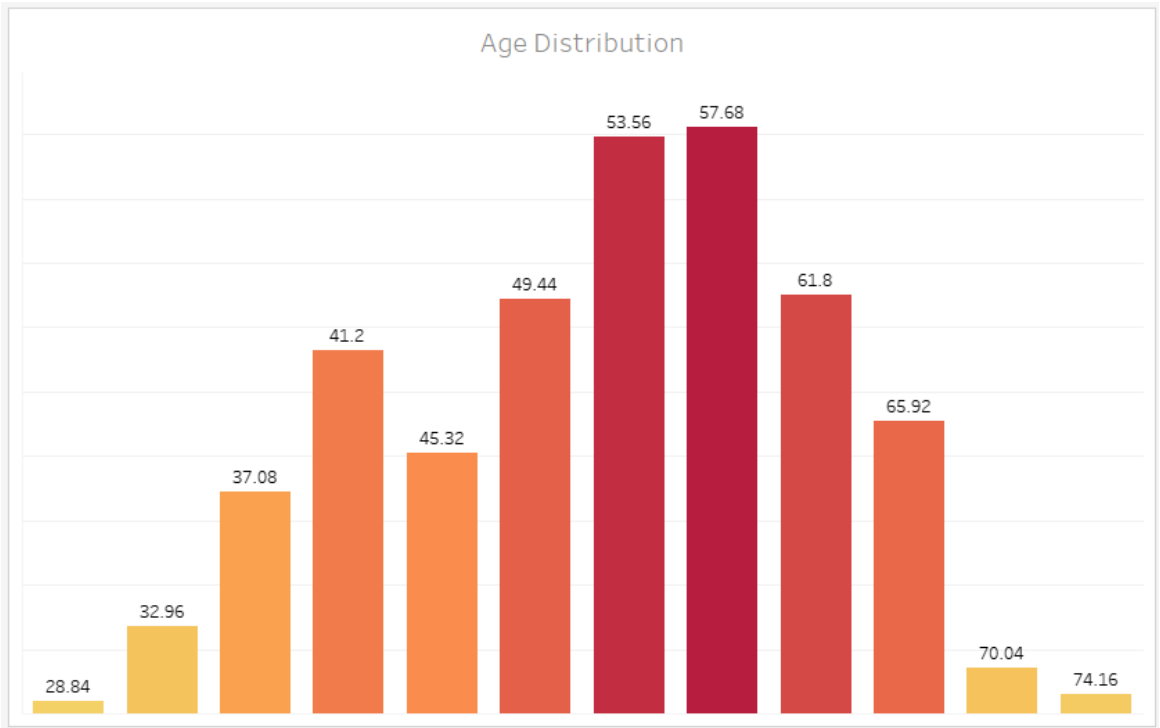

Figure 4.5: Description of Heart Disease Analysis
The age distribution of various patients has been shown in the graph

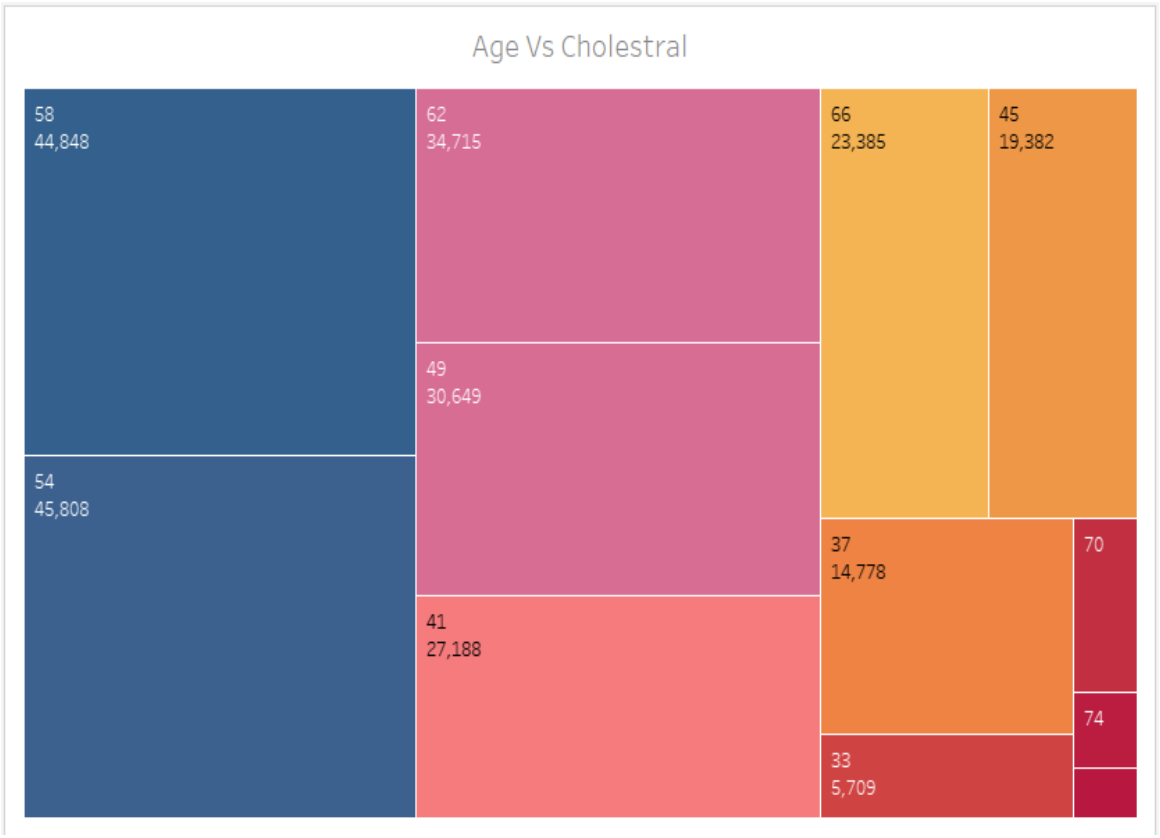

Figure 4.6: Age vs Cholestral

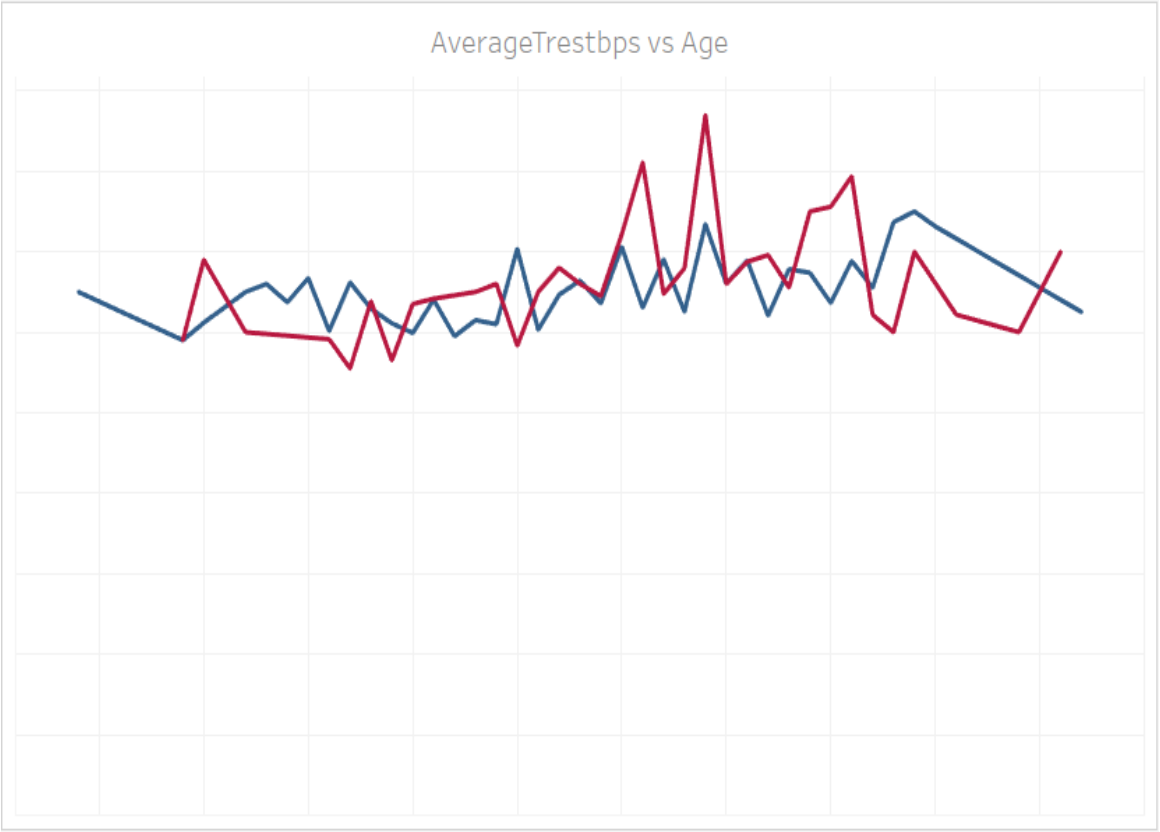

Figure 4.7: AverageTrestbps vs Age

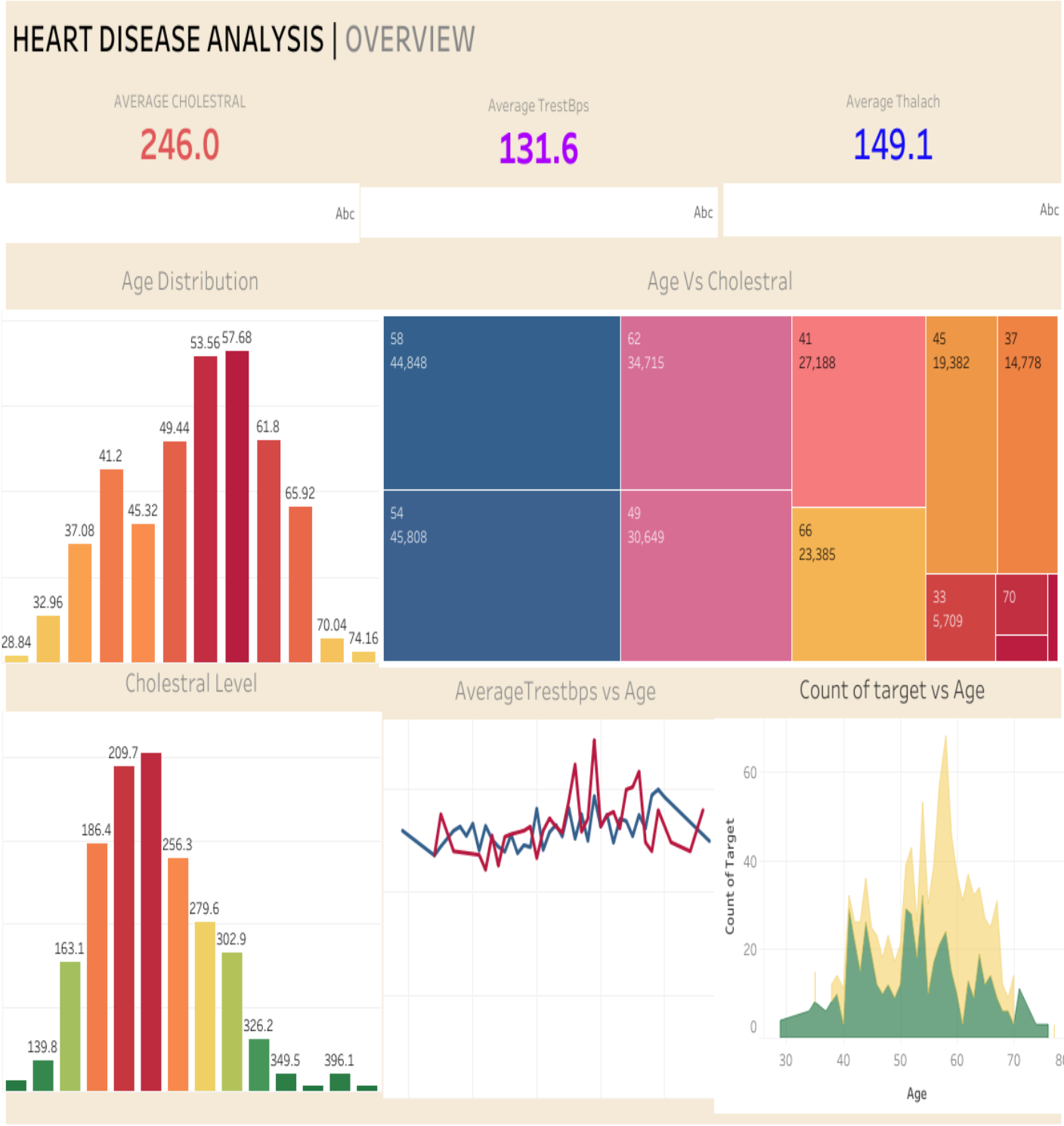

Figure 4.8: Heart Disease analysis — Overview dashboard

This helps us to conclude that with higher cholestrol and with age there are more chances of getting heart disease

Chapter 5

Conclusion & Future Enhancements

5.1 Conclusion

Heart diseases are a major killer in India and throughout the world, application of promising technology like machine learning to the initial prediction of heart diseases will have a profound impact on society. The early prognosis of heart disease can aid in making decisions on lifestyle changes in high-risk patients and in turn reduce the complications, which can be a great milestone in the field of medicine. The number of people facing heart diseases is on a raise each year. This prompts for its early diagnosis and treatment. The utilization of suitable technology support in this regard can prove to be highly beneficial to the medical fraternity and patients.In this report, the two different machine learning algorithms used to measure the performance are Decision Tree and Logistic Regression applied on the dataset. The vizualization of the dataset has been done using tableau.

5.2 Future Enhancements

As a future enhancement, we have decided to work on improving the accuracy using several other of the machine models available so that the predictions become even more clearer. Developing a distributed and real-time healthcare analytics system using traditional analytical tools is extremely complex, while exploiting open source big data technologies can do it in a simpler and more effective way.

References

[1] Wes McKinney, *"Python for Data Analysis: Data Wrangling with pandas, NumPy, and Jupyter"* 3rd Edition, OReilly, 2022

[2] Life Cycle of Data Analysis, `https://www.northeastern.edu/graduate/blog/data-analysis-project-lifecycle/`

[3] Kaggle Datasets `https://www.kaggle.com/datasets/johnsmith88/heart-disease-dataset`

[4] Streamlit, `https://docs.streamlit.io/`

[5] Tableau, `https://www.tableau.com/learn/articles/data-visualization`