University of Mumbai

2020-2021

ANJUMAN-I-ISLAM

M.H. SABOO SIDDIK COLLEGE OF ENGINEERING

BYCULLA - 400 008

Department of Computer Engineering

Project Report

On

"HEART DISEASE PREDICTION"

By

Saiqa Khan	3118029
Mujtaba Shaikh	3118050
Sanurhanaan Shaikh	3118053

Guided By

Prof. Lutful Islam

ANJUMAN-I-ISLAM

M.H. SABOO SIDDIK COLLEGE OF ENGINEERING

BYCULLA - 400 008

This is to certify that,

Roll No	Name
3118029	Saiqa Khan
3118050	Mujtaba Shaikh
3118053	Sanurhanaan Shaikh

Of Third year (T.E Semester VI) degree course in Computer Engineering, have completed the specified project report on,

“Heart Disease Prediction”

As a partial fulfilment of the project work in a satisfactory manner as per the rules of the curriculum laid by the University of Mumbai, during the Academic Year July 2020 — June 2021.

_____

Internal Guide

_____ _____

Internal Examiner External Examiner

_____ _____

Head of Department Principal

This project report entitled “Heart Diseases Prediction by Saiqa Khan(3118029), Mujtaba Shaikh(3118050) and Sanurhanaan Shaikh(3118053) is approved for the degree of Computer Engineering.

EXAMINERS

1.

2.

SUPERVISORS

1.

2.

Date: 15/5/2020

Place: Mumbai

ACKNOWLEDGEMENT

We would like to express our gratitude and appreciation to our parents for motivating and encouraging us throughout the career.

We wish to express our sincere thanks to our principal Dr. Ganesh Kame, M.H. Saboo Siddik College of Engineering for providing us all the facilities, support and wonderful environment to meet our project requirements.

We would also take the opportunity to express our humble gratitude to our Head of Department of Computer Engineering Dr. Zainab Pirani for supporting us in all aspects and for encouraging us with her valuable suggestions to make our project success.

We our highly thankful to our internal project guide Prof. Lutful Islam whose valuable guidance helped us understand the project better, their constant guidance and willingness to share their vast knowledge made us understand this project and its manifestations in great depths and helped us to complete the project successfully.

We would also like to acknowledge with much appreciation the role of the staff of the Computer Department, especially the Laboratory staff, who gave the permission to use the labs when needed and the necessary material to complete the project.

We would like to express our gratitude and appreciate the guidance given by other supervisors and project guides, their comments and tips helped us in improving our presentation skills. Although there may be many who remain unacknowledged in this humble note of appreciation but there are none who remain unappreciated.

Saiqa Khan
Mujtaba Shaikh
Sanurhanaan Shaikh

TABLE OF CONTENT

LIST OF FIGURES

[Indicate the figures wherever used]

ABSTRACT

In recent times, Heart Disease prediction is one of the most complicated tasks in medical field. In the modern era, approximately one person dies per minute due to heart disease. Data science plays a crucial role in processing huge amount of data in the field of healthcare. As heart disease prediction is a complex task, there is a need to automate the prediction process to avoid risks associated with it and alert the patient well in advance. This paper makes use of heart disease dataset available in UCI machine learning repository. The proposed work predicts the chances of Heart Disease and classifies patient's risk level by implementing different data mining techniques such as K – Nearest Neighbor, Random Forest, XGBoost, Adaboost and Gradient Boosting. Thus, this paper presents a comparative study by analysing the performance of different machine learning algorithms. The trial results verify that K – Nearest Neighbour algorithm has achieved the highest accuracy of 97.86% compared to other ML algorithms implemented.

INTRODUCTION AND MOTIVATION

1.1. INTRODUCTION

The work proposed in this paper focus mainly on various data mining practices that are employed in heart disease prediction. Human heart is the principal part of the human body. Basically, it regulates blood flow throughout our body.

Any irregularity to heart can cause distress in other parts of body. Any sort of disturbance to normal functioning of the heart can be classified as a Heart disease. In today's contemporary world, heart disease is one of the primary reasons for occurrence of most deaths. Heart disease may occur due to unhealthy lifestyle, smoking, alcohol and high intake of fat which may cause hypertension. According to the World Health Organization more than 10 million die due to Heart diseases every single year around the world. A healthy lifestyle and earliest detection are only ways to prevent the heart related diseases. The main challenge in today's healthcare is provision of best quality services and effective accurate diagnosis. Even if heart diseases are found as the prime source of death in the world in recent years, they are also the ones that can be controlled and managed effectively. The whole accuracy in management of a disease lies on the proper time of detection of that disease. The proposed work makes an attempt to detect these heart diseases at early stage to avoid disastrous consequences.

Records of large set of medical data created by medical experts are available for analysing and extracting valuable knowledge from it. Data mining techniques are the means of extracting valuable and hidden information from the large amount of data available. Mostly the medical database consists of discrete information. Hence, decision making using discrete data becomes complex and tough task.

Machine Learning (ML) which is subfield of data mining handles large scale well-formatted dataset efficiently. In the medical field, machine learning can be used for diagnosis, detection and prediction of various diseases.

1.2. AIM

The main goal of the paper is to provide a Heart Disease Prediction System for doctors to detect heart diseases at early stage. This in turn will help to provide effective treatment to patients and avoid severe consequences.

1.3. OBJECTIVE

The objective of this article is to develop a Heart Disease Prediction System. The system can discover and extract hidden knowledge associated with diseases from a historical heart data set.

Heart disease prediction system aims to exploit data techniques on medical data and predicting heart disease using different classification algorithms such as K – Nearest Neighbor, Random Forest, XGBoost, Adaboost and Gradient Boosting. The main contributions of this paper are:

- To detect the hidden discrete patterns and thereby analyse the given data.
- Extraction of classified accuracy useful for heart disease prediction
- Remove redundant and irrelevant features with Fast Correlation-Based Feature (FCBF) method.
- Comparison of different data mining algorithms on the heart disease dataset.
- Identification of the best performance-based algorithm for heart disease prediction.

1.4. MOTIVATION

The main motivation of doing this research is to present a heart disease prediction model for the prediction of occurrence of heart disease. Further, this research work is aimed towards identifying the best classification algorithm for identifying the possibility of heart disease in a patient. This work is justified by performing a comparative study and analysis using 5 classification algorithms namely K – Nearest Neighbor, Random Forest, XGBoost, ADAboost and Gradient Boosting are used at different levels of evaluations. Although these are commonly used machine learning algorithms, the heart disease prediction is a vital task involving highest possible accuracy. Hence, the three algorithms are evaluated at numerous levels and types of evaluation strategies.

1.5. SCOPE

Here the scope of the project is that integration of clinical health decision support with computer-based patient records could reduce medical errors, enhance patient safety, decrease unwanted practice variation, and improve patient outcome. This suggestion is promising as data modelling and analysis tools, e.g., data mining, have the potential to generate a knowledge-rich environment which can help to significantly improve the quality of clinical decisions.

1.5.1. LOCAL SCOPE

Machine Learning is a very vast and diverse field and its scope and implementation is increasing day by day. Here the scope of the project is that integration of clinical decision support with computer-based patient records could reduce medical errors, enhance patient safety, decrease unwanted practice variation, and improve patient outcome. This suggestion is promising as data modelling and analysis tools, e.g., data mining, have the potential to generate a knowledge-rich environment which can help to significantly improve the quality of clinical decisions.

1.5.2. GLOBAL SCOPE

The scope of this system aims at giving more sophisticated prediction models, risk calculation tools and feature extraction tools for other clinical risks. The machine learning algorithm with optimal accuracy will yield a good prediction. There is a scope to improvise this system by integrating these approaches and forming a hybrid model that can deliver better outcomes than individual methods.

PROBLEM STATEMENT

Heart disease can be managed effectively with a combination of lifestyle changes, medicine and, in some cases, surgery. With the right treatment, the symptoms of heart disease can be reduced and the functioning of the heart improved. The predicted results can be used to prevent and thus reduce cost for surgical treatment and other expensive.

The overall objective of my work will be to predict accurately with few tests and attributes the presence of heart disease. Attributes considered form the primary basis for tests and give accurate results more or less. Many more input attributes can be taken but our goal is to predict with few attributes and faster efficiency the risk of having heart disease. Decisions are often made based on doctors' intuition and experience rather than on the knowledge rich data hidden in the data set and databases. This practice leads to unwanted biases, errors and excessive medical costs which affects the quality of service provided to patients.

Data mining holds great potential for the healthcare industry to enable health systems to systematically use data and analytics to identify inefficiencies and best practices that improve care and reduce costs. According to (Wurz & Takala, 2006) the opportunities to improve care and reduce costs concurrently could apply to as much as 30% of overall healthcare spending. The successful application of data mining in highly visible fields like e-business, marketing and retail has led to its application in other industries and sectors. Among these sectors just discovering is healthcare. The healthcare environment is still "information rich" but "knowledge poor". There is a wealth of data available within the healthcare systems. However, there is a lack of effective analysis tools to discover hidden relationships and trends in the data for African genres.

REQUIREMENT ANALYSIS

3.1. REVIEW OF LITERATURE

According to Ordonez, the heart disease can be predicted with some basic attributes taken from the patient and in their work have introduced a system that includes the characteristics of an individual human being based on totally 13 basic attributes like sex, blood pressure, cholesterol and others to predict the likelihood of a patient getting affected by heart disease. They have added two more attributes i.e. fat and smoking behaviour and extended the research dataset. The data mining classification algorithms such as Decision Tree, Naive Bayes, and Neural Network are utilized to make predictions and the results are analysed on Heart disease database.

Yılmaz, have proposed a method that uses least squares support vector machine (LS-SVM) utilizing a binary decision tree for classification of cardiogram to find out the patient condition.

Duff, et al. have done a research work involving five hundred and thirty-three patients who had suffered from cardiac arrest and they were integrated in the analysis of heart disease probabilities. They performed classical statistical analysis and data mining analysis using mostly Bayesian networks.

Frawley, et al. have performed a work on prediction of survival of Coronary heart disease (CHD) which is a challenging research problem for medical society. They also used 10-fold cross-validation methods to determine the impartial estimate of the three prediction models for performance comparison purposes.

Parthiban, et al. have proposed a new work in which the heart disease is identified and predicted using the proposed Coactive Neuro-Fuzzy Inference System (CANFIS). Their model works based on the collective nature of neural network adaptive capabilities and based on the genetic algorithm along with fuzzy logic in order to diagnose the occurrence of the disease. The performance of the proposed CANFIS model was evaluated in terms of training performances and classification accuracies. Finally, their results show that the proposed CANFIS model has great prospective in predicting the heart disease.

3.2. EXISTING SYSTEM

The before all existing system works on sets of both Deep learning and Data Mining. The existing system modules generates comprehensive report by implementing the strong prediction algorithm The main aims of the existing system to compare and check the before patient whose having disease outputs and new patient disease and determine future possibilities of the heart disease to a particular patient By Implementing the above mentioned model we will get the goal of developing a system with increased rate of accuracy of estimating the new patient getting heart attack percentage. The model which is proposed for Heart Disease Prediction System is invented for using Deep learning algorithms and approach. But by using all the existing systems the accuracy is very less.

3.3. DRAWBACKS OF EXISTING SYSTEM

1. Prediction of cardiovascular disease results is not accurate.

2. Data mining techniques does not help to provide effective decision making.

3. Cannot handle enormous datasets for patient records.

DESIGN DETAILS

4.1. DESIGN OF THE SYSTEM

4.1.1. DATA COLLECTION

The data is collected from the UCI machine learning repository. The data set is named Heart Disease Dataset and can be found in the UCI machine learning repository. The UCI machine learning repository contains a vast and varied amount of datasets which include datasets from various domains. These data are widely used by machine learning community from novices to experts to understand data empirically. Various academic papers and researches have been conducted using this repository. Heart disease dataset contains data from four institutions-

- Cleveland Clinic Foundation.
- Hungarian Institute of Cardiology, Budapest.
- V.A. Medical Centre, Long Beach, CA.
- University Hospital, Zurich, Switzerland.

For the purpose of this study, the data set provided by the Cleveland Clinic Foundation is used. This dataset was provided by Robert Detrano, M.D, Ph.D. Reason to choose this dataset is, it has less missing values and is also widely used by the research community.

4.1.2. DATA PREPARATION AND TRAINING

The attributes mentioned in methodology are provided as input to the different ML algorithms such as K - Nearest Neighbor Classifier, Random Forest Classifier, XGBoost Classifier, AdaBoost with Random Forest and Gradient Boosting techniques. The input dataset is split into 80% of the training dataset and the remaining 20% into the test dataset. Training dataset is the dataset which is used to train a model. Testing dataset is used to check the performance of the trained model. For each of the algorithms the performance is computed and analysed based on different metrics used such as accuracy, precision, re-call and F-measure scores as described further.

4.1.3. GUI FOR USER

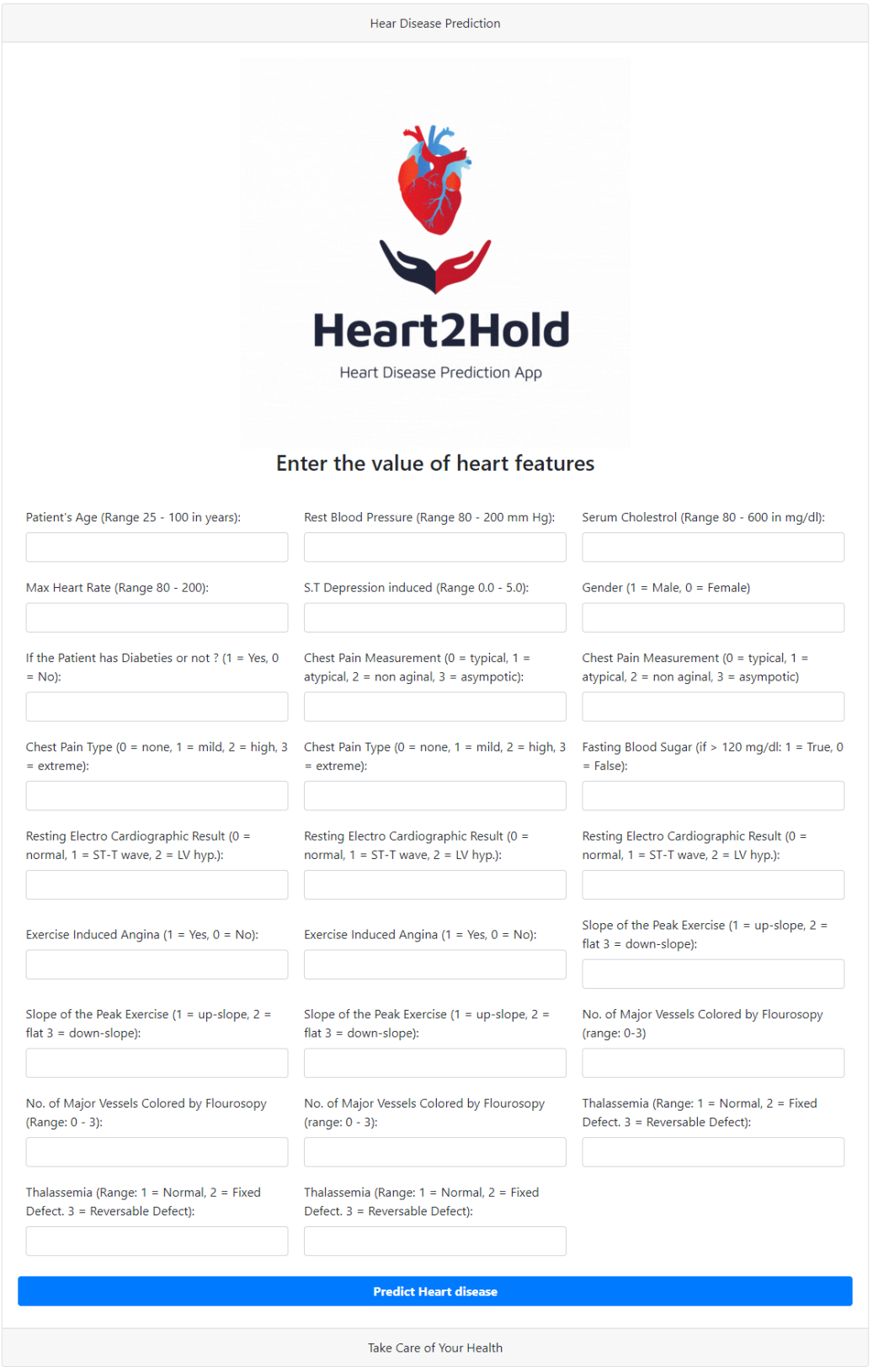


Fig 4.1.3. GUI For User

4.1.4. MODEL SELECTION

The different algorithms explored in this paper are listed as below-

- **K - Nearest Neighbor Classifier (KNN)**

K nearest neighbour is a simple algorithm that stores all available cases and classifies new cases based on a similarity measure (e.g., distance functions). KNN has been used in statistical estimation and pattern recognition already in the beginning of 1970's as a non-parametric technique.

- **Random Forest (RF)**

Random Forest algorithms are used for classification as well as regression. It creates a tree for the data and makes prediction based on that. Random Forest algorithm can be used on large datasets and can produce the same result even when large sets record values are missing.

- **XGBoost Classifier (XGB)**

XGBoost is optimized distributed gradient boosting library designed to highly efficient, flexible and portable. It implements machine learning algorithms under the Gradient Boosting framework. XGBoost provides a parallel tree boosting (also known as GBDT, GBM) that solve many data science problems in a fast and accurate way.

- **AdaBoost with Random Forest (ARF)**

The capability of this hybrid method is evaluated using basic performance measurements (e.g., accuracy, sensitivity, and specificity), Receiver Operating Characteristic (ROC) curve and Area Under the receiver operating characteristic Curve (AUC). Experimental results indicate that the proposed method outperforms a single classifier and other combined classifiers.

- **Gradient Boosting (GB)**

Gradient boosting classifiers are a group of machine learning algorithms that combine many weak learning models together to create a strong predictive model. Decision trees are usually used when doing gradient boosting. Gradient boosting models are becoming popular because of their effectiveness at classifying complex datasets, and have recently been used to win many Kaggle data science competitions.

4.2. FLOWCHART

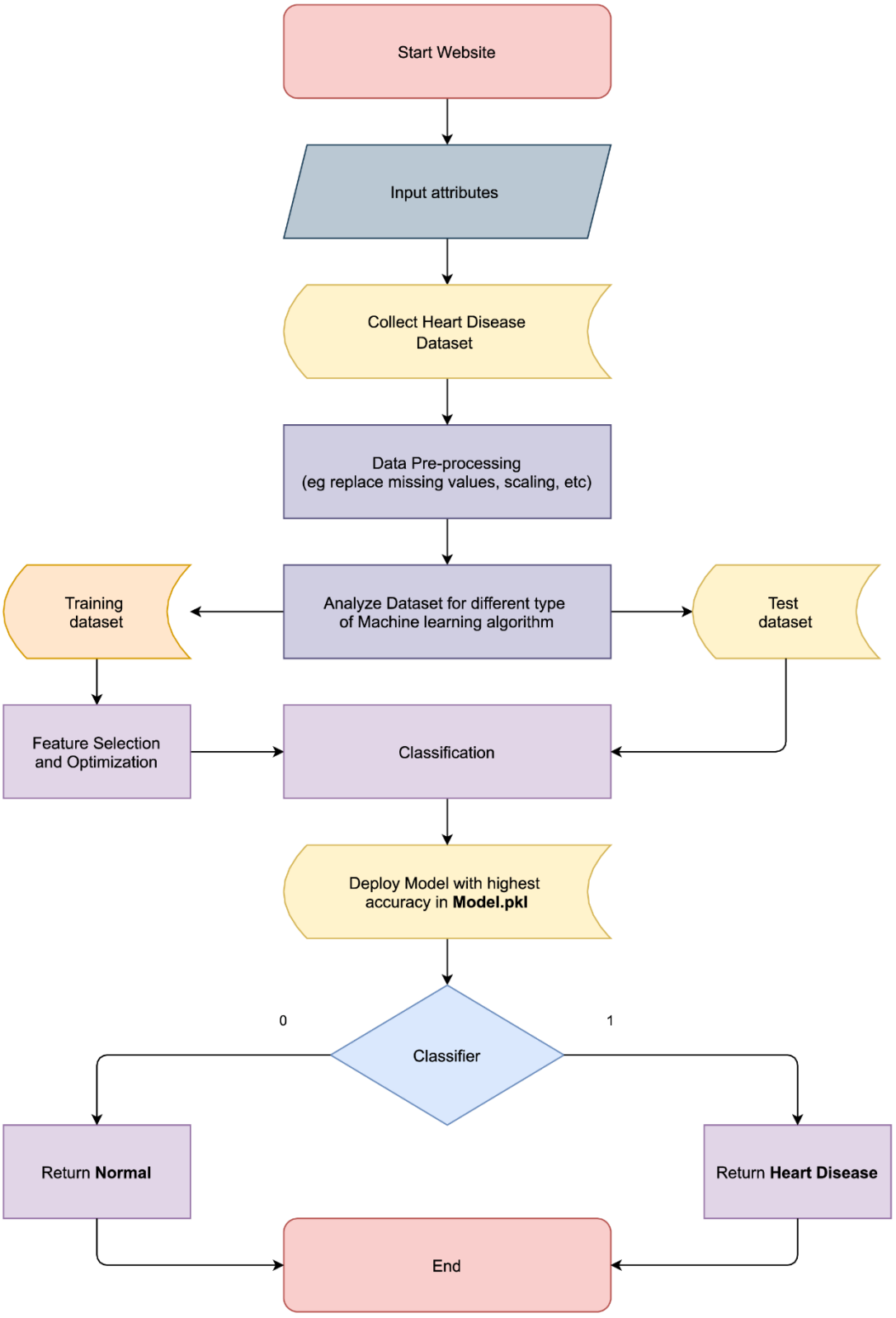


Fig 4.2. Flowchart

4.3. BLOCK DIAGRAM

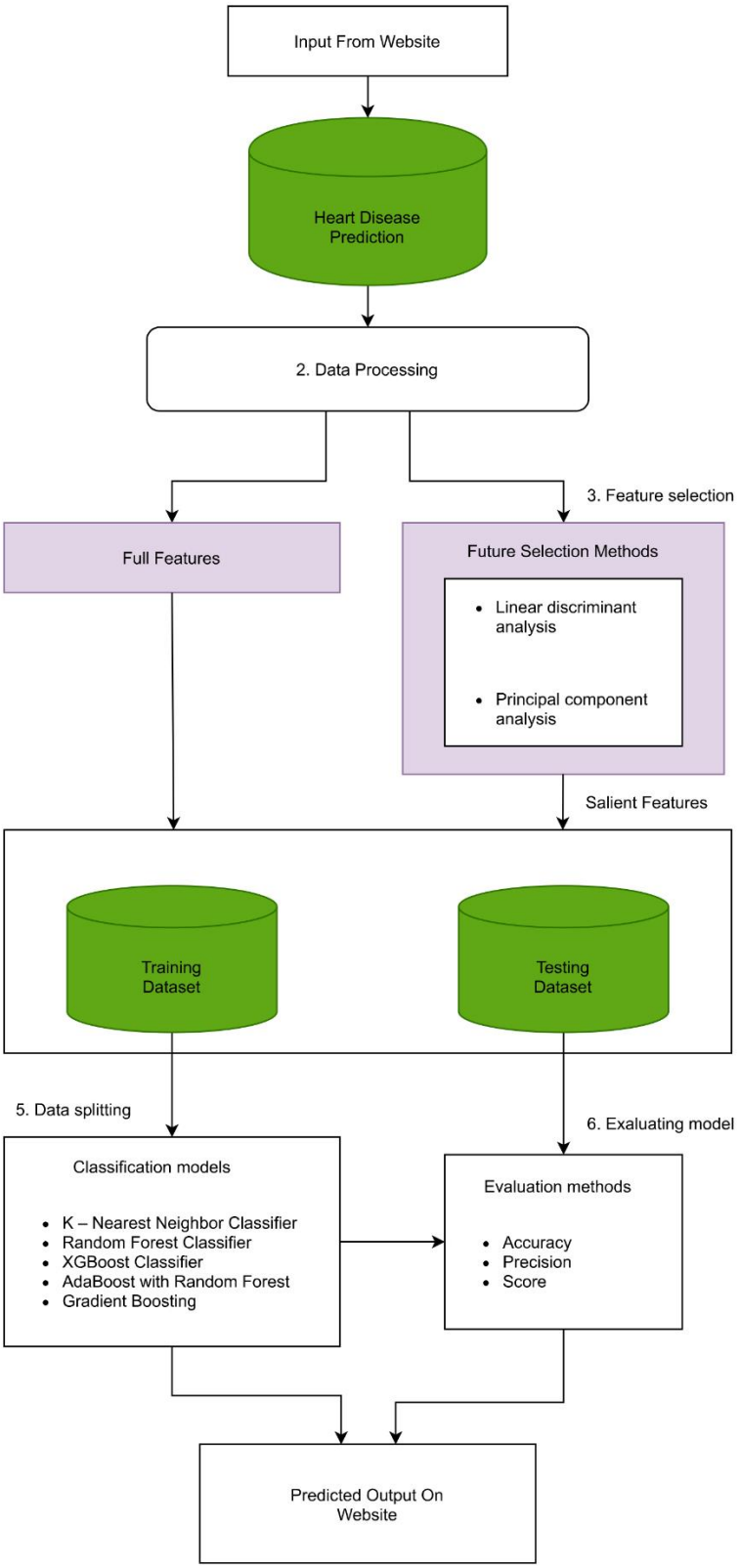


Fig 4.3. Block Diagram

4.4. DATA FLOW DIAGRAM (DFD)

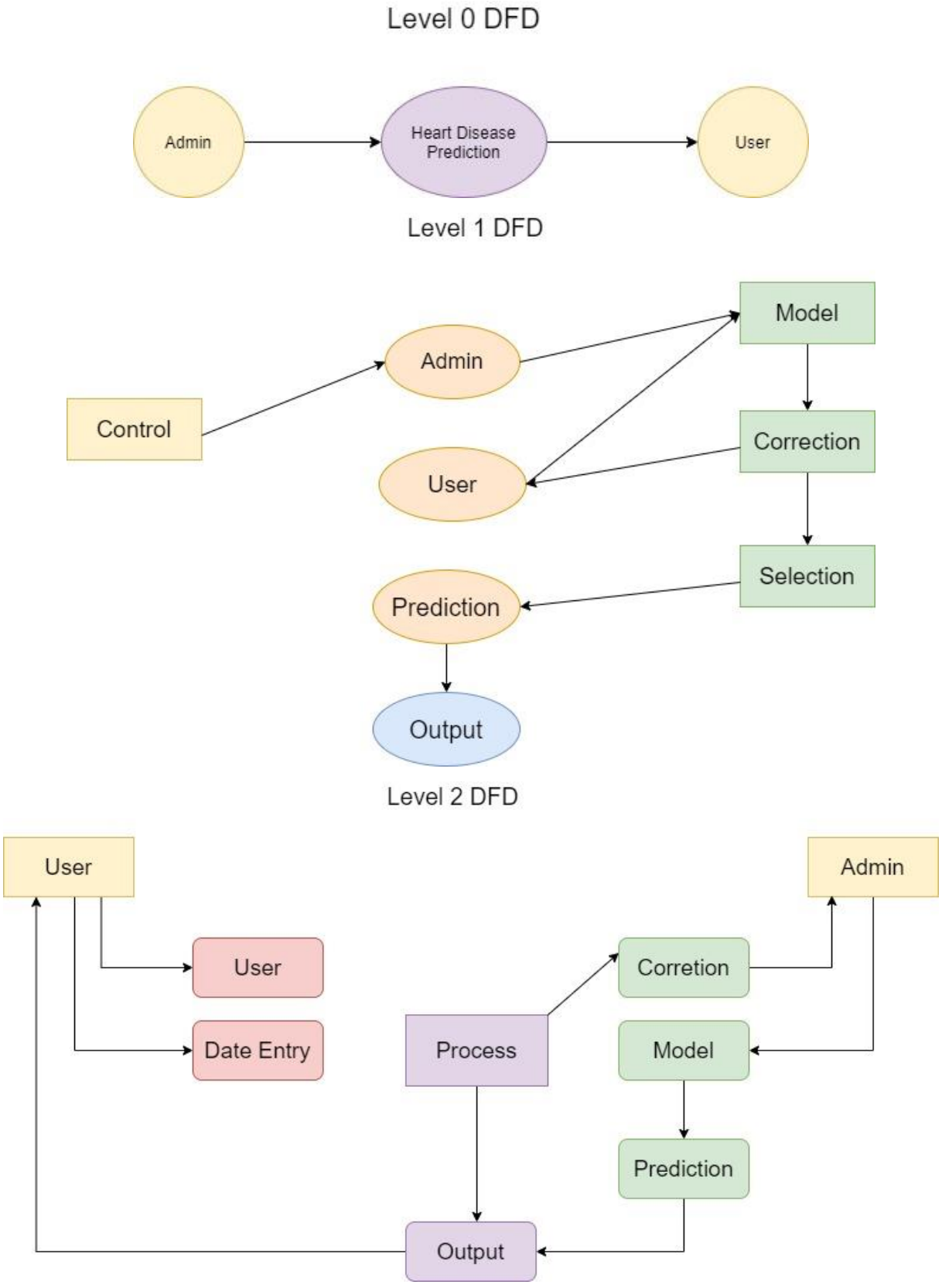


Fig 4.4. Data Flow Diagram (DFD)

4.5. USE CASE DIAGRAM

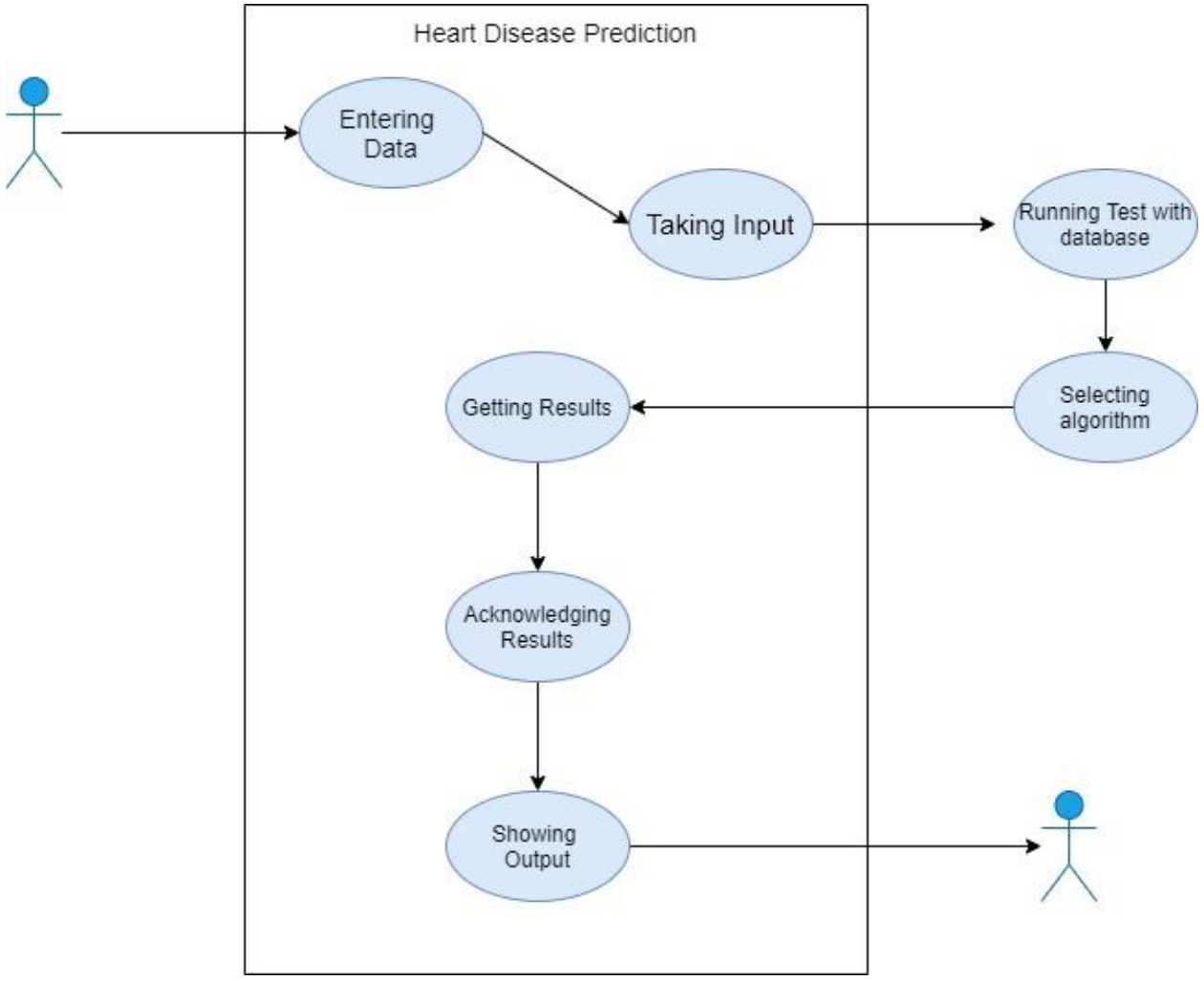


Fig 4.5. Use Case Diagram

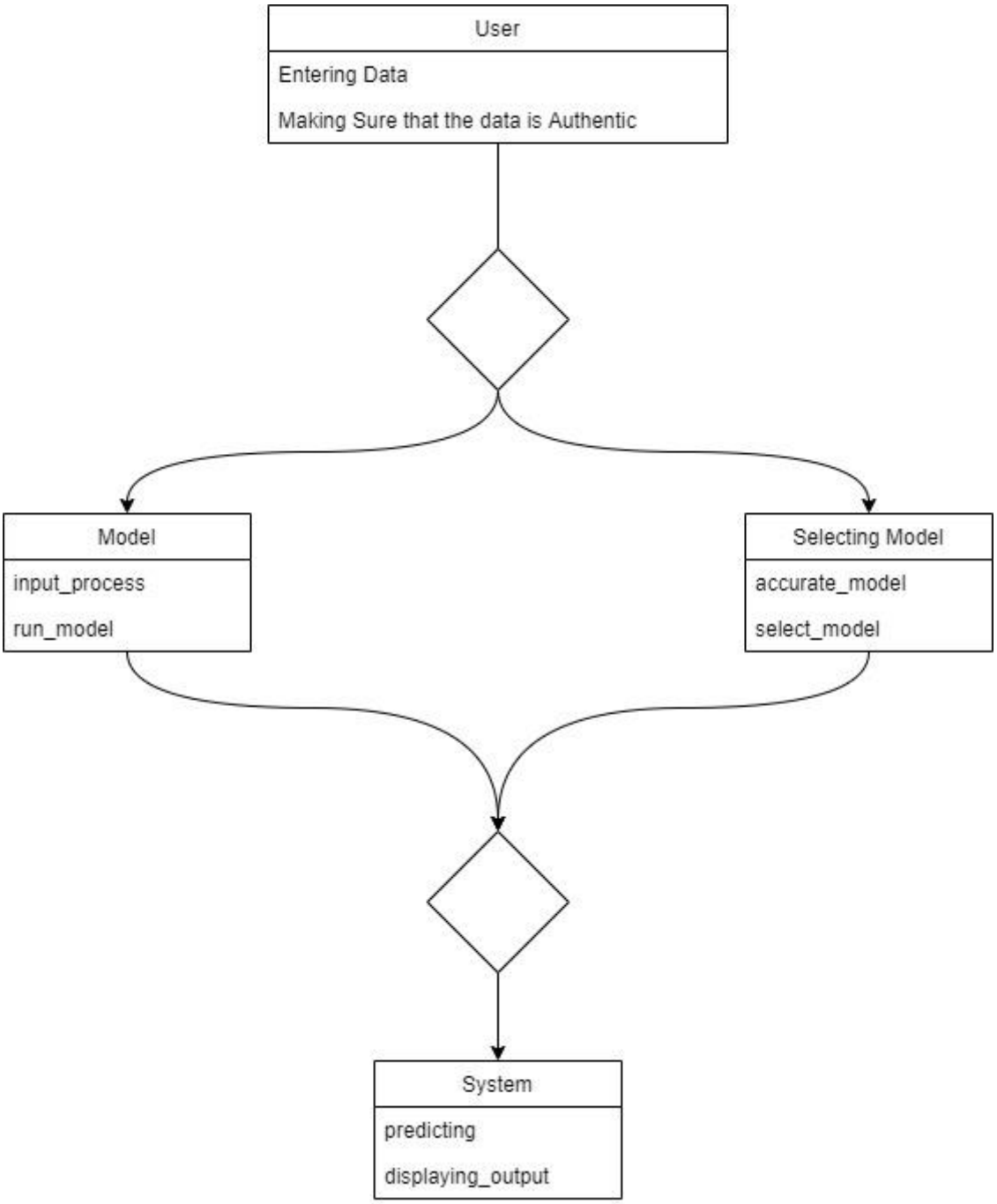


Fig 4.5. Use Case Diagram (2)

IMPLEMENTATION DETAILS

5.1. PROCESS MODEL

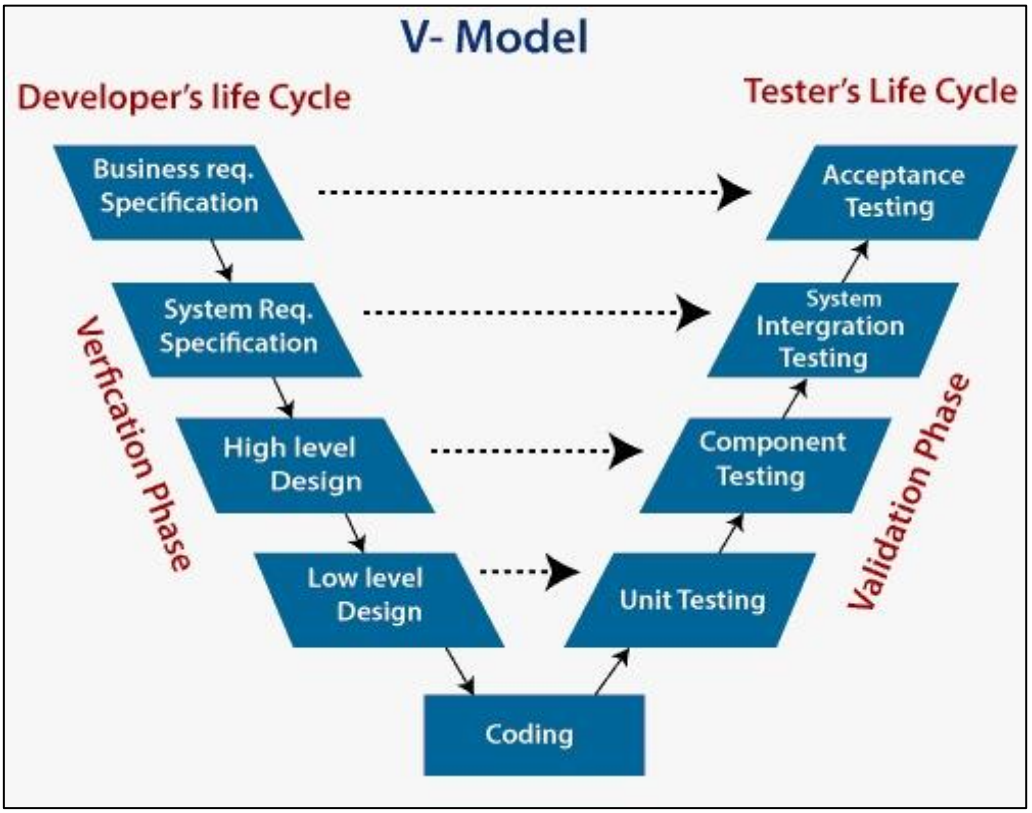


Fig 5.1. Process Model

The V-model is an SDLC model where execution of processes happens in a sequential manner in a V-shape. It is also known as Verification and Validation model.

The V-Model is an extension of the waterfall model and is based on the association of a testing phase for each corresponding development stage. This means that for every single phase in the development cycle, there is a directly associated testing phase. This is a highly-disciplined model and the next phase starts only after completion of the previous phase.

Application: V- Model application is almost the same as the waterfall model, as both the models are of sequential type. Requirements have to be very clear before the project starts, because it is usually expensive to go back and make changes. This model is used in the medical development field, as it is strictly a disciplined domain.

5.2. METHODOLOGY

The proposed work predicts heart disease by exploring the above mentioned four classification algorithms and does performance analysis. The objective of this study is to effectively predict if the patient suffers from heart disease.

The health professional enters the input values from the patient's health report. The data is fed into model which predicts the probability of having heart disease. Figure given below shows the entire process involved.

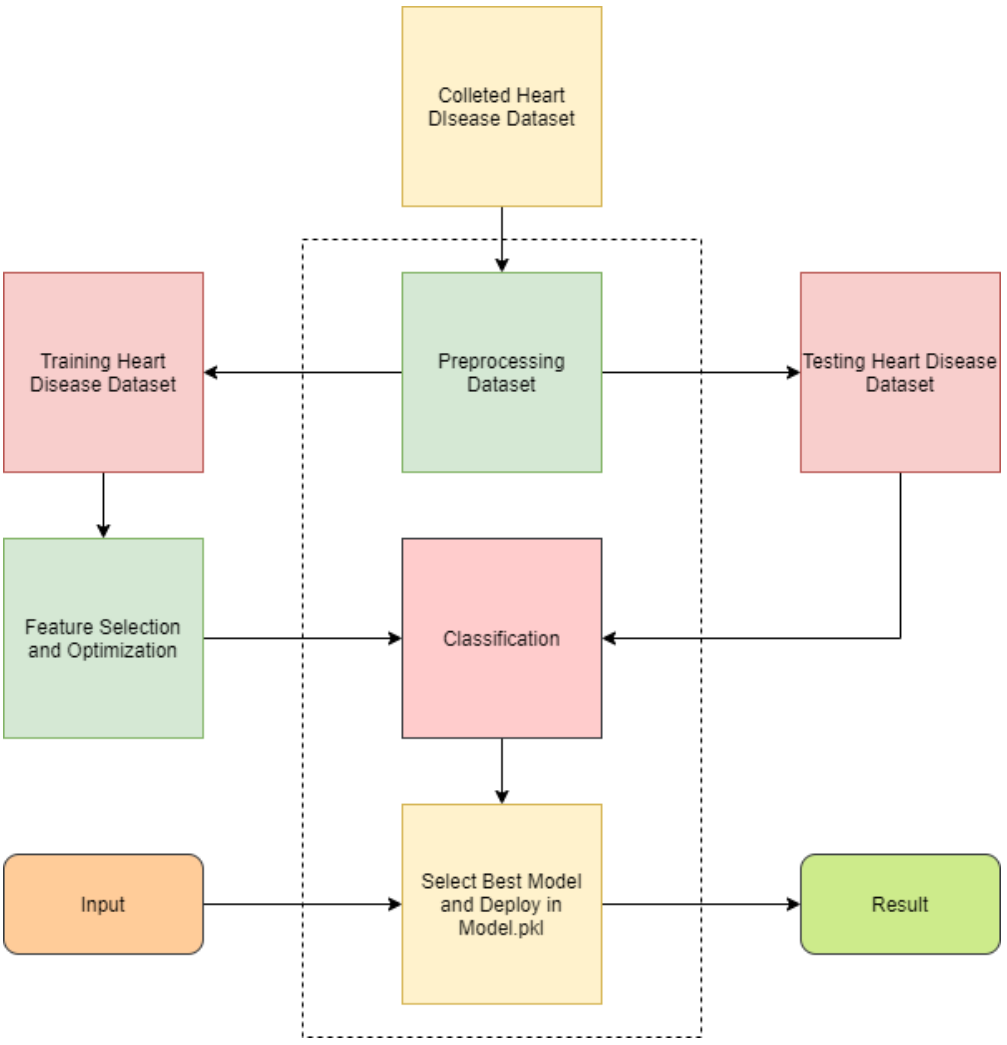


Fig 5.2. System Architecture

DATA COLLECTION AND PREPROCESSING

The dataset used was the Heart disease Dataset which is a combination of 4 different database, but only the UCI Cleveland dataset was used. This database consists of a total of 76 attributes but all published experiments refer to using a subset of only 14 features. Therefore, we have used the already processed UCI Cleveland dataset available in the Kaggle website for our analysis. The complete description of the 14 attributes used in the proposed work is mentioned in Attribute Table shown below.

CLASSIFICATION TASK

From the perspective of Machine learning, heart disease detection can be seen as a classification or clustering problem. On the other hand, we formed a model on the vast set of presence and absence file data; we can reduce this problem to classification. For known families, this problem can be reduced to one classification only - having a limited set of classes, including the heart disease sample, it is easier to identify the right class, and the result would be more accurate than with clustering algorithms. In this section, the theoretical context is given on all the methods used in this research. For the purpose of comparative analysis, five Machine Learning algorithms are discussed. The different Machine Learning (ML) algorithms are K – Nearest Neighbour (KNN), Random Forest (RF), XGBoost (XGB), ADAboost with Random Forest (ADG) and Gradient Boosting (GB). The reason to choose these algorithms is based on their popularity and their performance.

FEATURE SELECTION

In the heart disease datasets, the number of features can reach up to tens of thousands; the heart disease dataset has 14 attributes. Since a large number of irrelevant and redundant attributes are involved in these expression data, the heart disease classification task is made more complex. If complete data are used to perform heart disease classification, accuracy will not be as accurate, and calculation time and costs will be high. Therefore, the feature selection, as a pre-treatment step to machine learning, reduces sizing, eliminates unresolved data, increases learning accuracy, and improves understanding of results. The recent increase in the dimensionality of the data poses a serious problem to the methods of selecting characteristics with regard to efficiency and effectiveness.

Attributes of Heart Disease Dataset are:

Attribute	Marked As	Range	Measured in
Patient's Age	Age	25 - 100	in years
Sex	Sex	0 = Female, 1=Male	Instance
Chest Pain Type	cp	0 = typical agina 1 = atypical agina 2 = non-aginal pain 3 = asymptomatic	Instance
Resting Blood Pressure	trestbps	80 - 200	in mm Hg
Serum Cholesterol	chol	80 - 600	in mg/dl
Fasting Blood Sugar	fbs	(If >120 mg/dl ?): 1 = True 0 = False	Instance
Resting Electro cardiogram	restecg	0=normal 1 = ST-T wave abnormality 2 = LV hypertrophy	Instance
Max Heart rate achieved	Thalach	70 - 200	Rate
Exercise Induced angina	exang	(Exercise done?): 0 = No 1 = Yes	Instance
ST depression induced by Exercise relative to rest	oldpeak	0.0 - 5.0	Instance
Slope of the peak exercise ST segment	Slope	0 = up-slope 1 = flat 2 = down-slope	Instance
Number of Major vessel coloured by Fluoroscopy	ca	0 - 3 blood vessel	Instance
Thalassemia (Defect type)	thal	1=normal 2=fixed defect 3=reversible defect	Instance

TECHNOLOGY USED

6.1. HARDWARE SPECIFICATION

- Processor : Intel I3 processor.
- RAM : 4 GB.
- HDD : 1 TB.

6.2. SOFTWARE REQUIREMENTS

Software used in this project are

- Operating System - Windows 2010, Windows 2007/XP
- Jupyter Notebook - for maintaining and executing various machine learning models.
- Github - Versioning Control
- Heroku – Platform used for building and running web applications entirely on cloud.
- Google Chrome Medium - to find reference to do system testing and executing.

6.3. PROGRAMMING LANGUAGE USED

The code has been written in Python programming and all the models building based on the python libraries

Libraries Used:

- NumPy and SciPy
- Matplolib (pyplot, reparams, matshow)
- Gunicorn
- Pandas
- Xgboost
- Scikit-klearn
- Flask

TEST CASE DESIGN

7.1. UNIT TESTING

Test Id	Test Description	Expected Result	Actual Result
A	Filling out the entire form	To fill all columns	To fill all columns (Pass)
B	Entering values that should have two or more categories e.g. sex and chest pain	Entering all categorical features	Entering all categorical features (Pass)
C	Entering values having relative ordering or sorting between the values e.g. fbs, restecg	Entering all Ordinal features	Entering all Ordinal features (Pass)
D	Entering values between any two points or between the min or max values. e.g. age	Entering all continuous features	Entering all continuous features (Pass)
E	To click on predict heart disease button once done filling the entire form	To submit form successfully and give the result	To submit form successfully and give the result (pass)
F	To check result if the patient has a heart disease or not	To show if the patient has a heart disease or not	To show if the patient has a heart disease or not(pass)

7.2. INTEGRATION TESTING

Test Id	Test Description	Expected Result	Actual Result
A	To test the interface link between the form filled and submit button	There should be no errors and the form should submit successfully	There should be no errors and the form should submit successfully(pass)
B	To test the interface link between the submit button and the result of the patient	The result should be quite accurate if the person has the heart disease or not	The result should be quite accurate if the person has the heart disease or not (pass)

7.3. WHITE BOX TESTING

NODE	TASK
1	User visits web Page
2	Input Attributes
3	Attributes filled entirely and properly
4	Attributes not filled entirely and properly
5	Heart Disease Dataset
6	Classification and Testing Model
7	Getting Result
8	Return Heart Disease

DEFINITON

White Box Testing is software testing technique in which internal structure, design and coding of software are tested to verify flow of input-output and to improve design, usability and security. In white box testing, code is visible to testers so it is also called Clear box testing, Open box testing, Transparent box testing, Code-based testing and Glass box testing.

The term "WhiteBox" was used because of the see-through box concept. The clear box or WhiteBox name symbolizes the ability to see through the software's outer shell (or "box") into its inner workings.

The following is our Flow Graph with Edges, Nodes and Regions on the table given above-

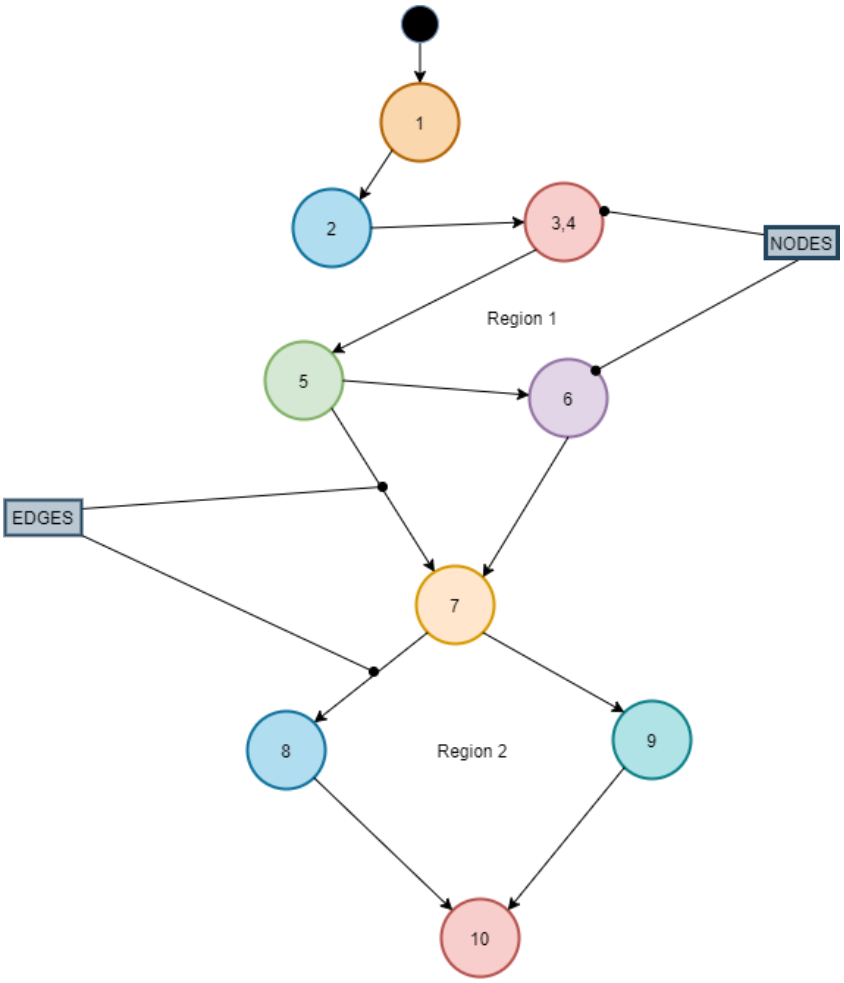


Fig 7.3. White Box Testing

7.4. BLACK BOX TESTING

Black Box Testing is a software testing method in which the functionalities of software applications are tested without having knowledge of internal code structure, implementation details and internal paths. Black Box Testing mainly focuses on input and output of software applications and it is entirely based on software requirements and specifications. It is also known as Behavioural Testing.

Fig 7.4. Black Box Testing

The above Black-Box can be any software system you want to test. For Example, an operating system like Windows, a website like Google, a database like Oracle or even your own custom application. Under Black Box Testing, you can test these applications by just focusing on the inputs and outputs without knowing their internal code implementation.

RESULTS

The results obtained by applying K – Nearest Neighbor, Random Forest, XGBoost, ADAboost with Random Forest and Gradient Boosting are shown in this section. The metrics used to carry our performance analysis of the algorithm are Accuracy score, Precision (P), Recall (R) and F-measure.

Precision metric provides the measure of positive analysis that is correct.

```
Precision (P) = TP / (TP+FP)
```

Recall defines the measure of actual positives that are correct.

```
Recall (R) = TP / (TP+FN)
```

F-measure test accuracy.

```
F-measure = (2*P*R) / (P+R)
```

- True Positive (TP): the patient has the disease and the test is positive.
- False Positive (FP): the patient does not have the disease and the test is positive.
- True Negative (TN): the patient does not have the disease and the test is negative.
- False Negative (FN): the patient has the disease and the test is negative.

In the experiment the pre-processed dataset is used to carry out the experiments and the above mentioned algorithms are explored and applied. The above mentioned performance metrics are obtained using the confusion matrix. Confusion Matrix describes the performance of the model. The confusion matrix obtained by the proposed model for different algorithms is shown below in Confusion Matrix Table. The accuracy score obtained for K-Nearest Neighbor, Random Forest, XGBoost, ADAboost with Random Forest and Gradient Boosting classification techniques is shown below in Table.

Our goal was to compare different classification models and define the most efficient one. From all the tables below, different algorithms performed better depending upon the situation whether cross-validation, calibration and feature selection is used or not. Every algorithm has its intrinsic capacity to outperform other algorithm depending upon the situation. For example, Random Forest performs much better with a large number of datasets than when data is small ADAboost with Random Forest performs better with a smaller number of data sets. Performance of algorithms decreased after boosting in the data, which did not feature, selected while algorithms were performing better without boosting in feature-selected data. This shows the necessity that the data should be feature selected before applying to boost.

For the comparison of the dataset, Confusion matrix after feature selection, parameter tuning and calibration are used because this is a standard process of evaluating algorithms. The precision average value of the best performance without optimization it's for XGB and Gradient with 89.6% than RF with 86.9%. These shows XGB and GB are performing on higher level, after optimized by FCBF we find the best performance of precision it's for K-NN with 97.8% than RF with 86.9% shown

In the last stage, we compared the different algorithms with the proposed optimized model, we find the best one is K-NN with 97.8 % than RF with 86.9 %.

Confusion Matrix Table:

Algorithm	TP	FP	FN	TN
K-Nearest Neighbour	15	0	1	30
Random Forest	14	4	2	26
XGBoost	13	2	3	28
ADAboost with Random Forest	14	6	2	24
Gradient Boosting	15	4	1	26

RESULT TABLE:

Algorithm	Accuracy	Precision	Recall	F-measure
K-Nearest Neighbour	97.826087	1	0.937	0.967
Random Forest	86.956522	0.778	0.875	0.824
XGBoost	89.130435	0.867	0.813	0.839
ADAboost with Random Forest	82.608696	0.7	0.875	0.778
Gradient Boosting	89.130435	0.789	0.938	0.857

Among these 5 models, K – Nearest Neighbour outperforms any other models.

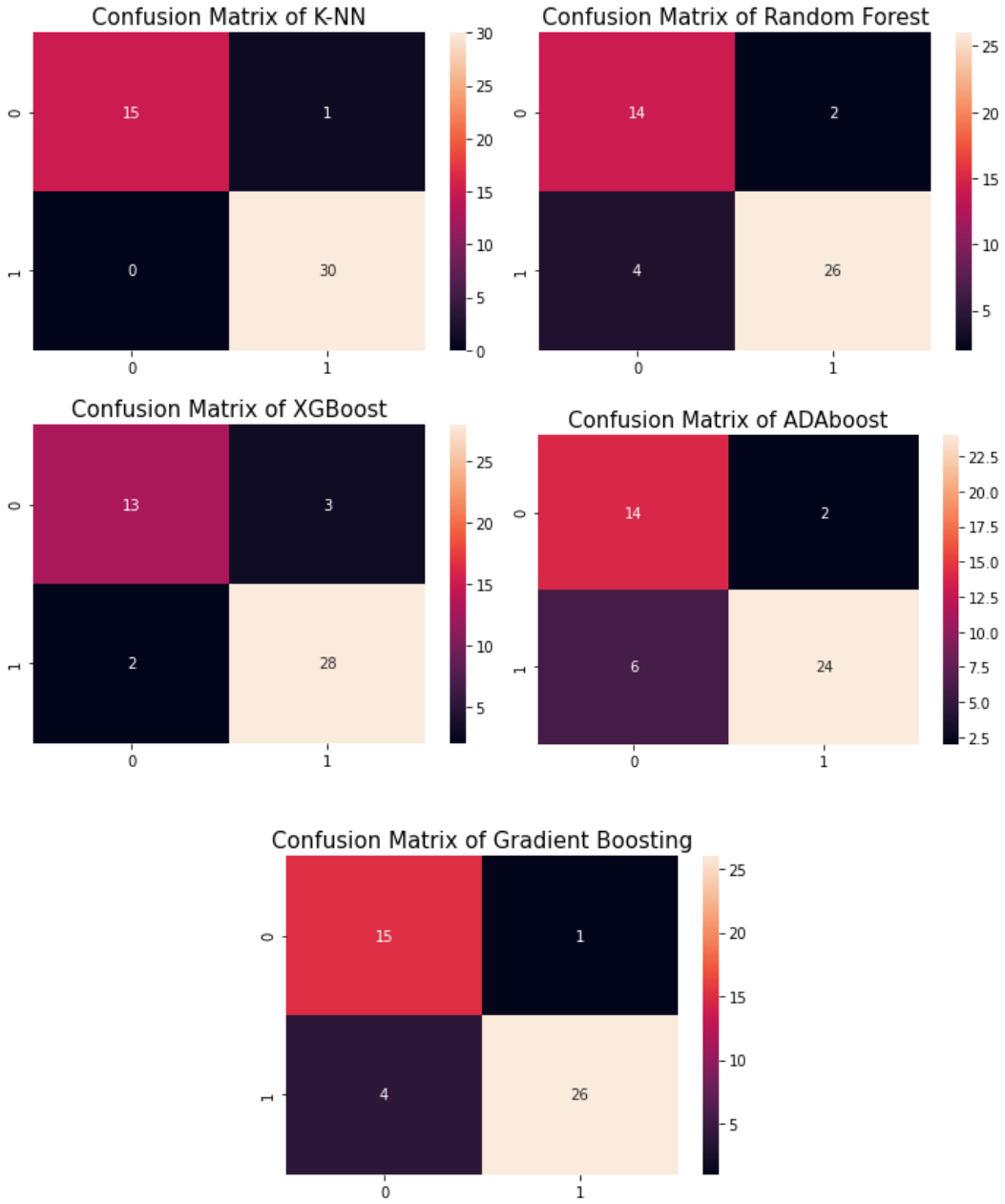
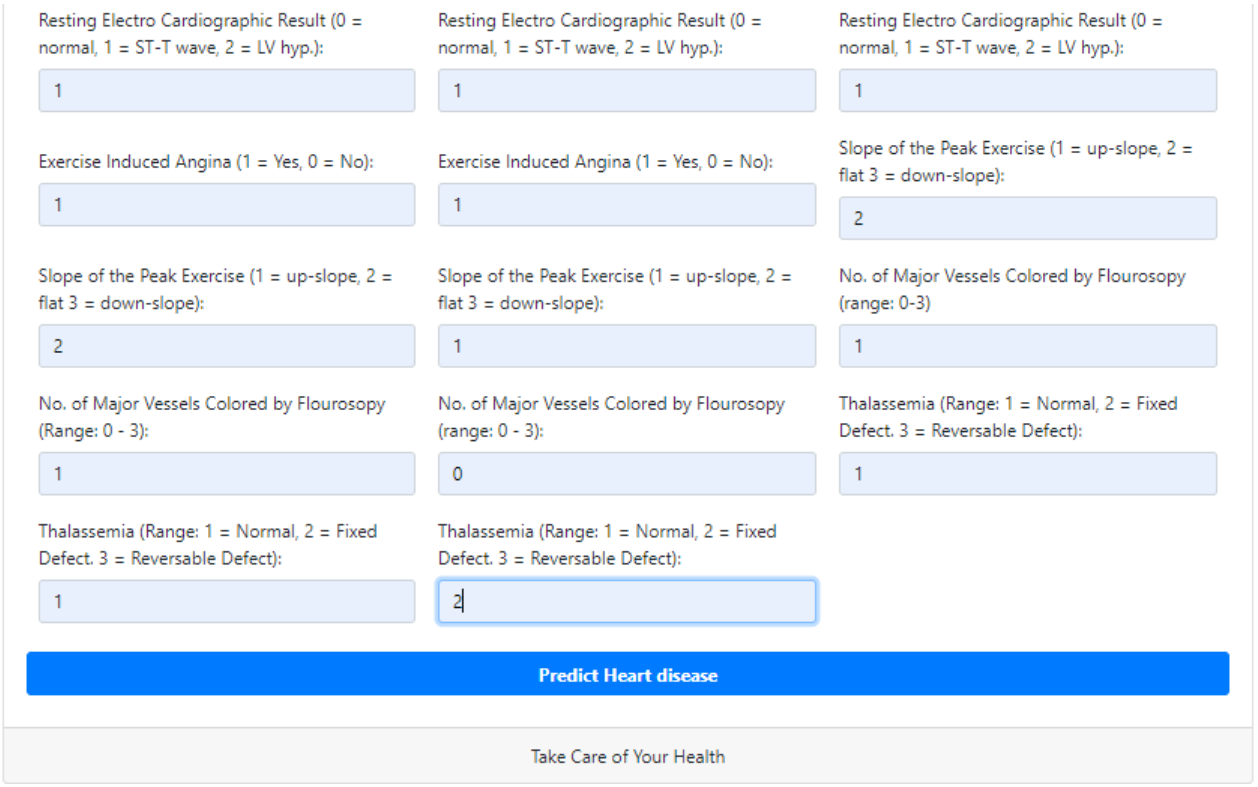


Fig 8.1. Confusion Matrix



Outcome of the given input:

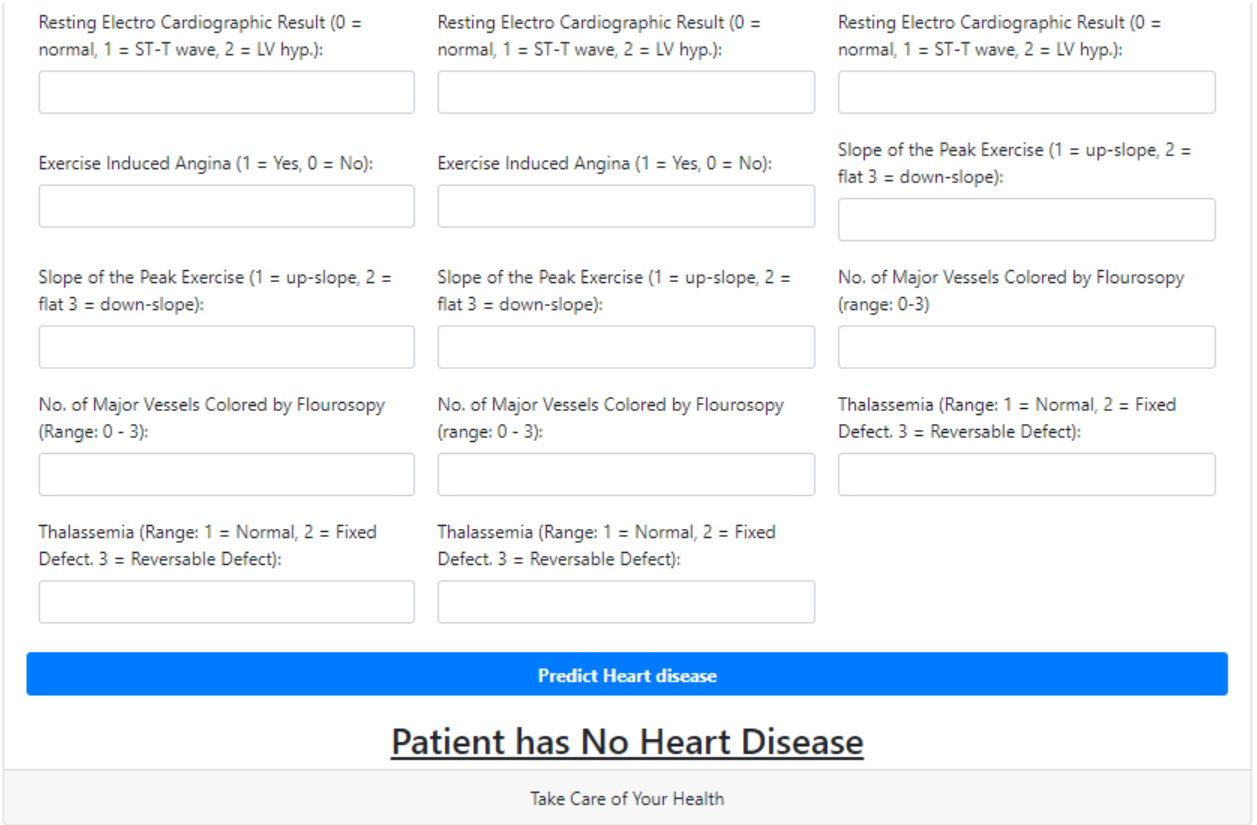


Fig.8.2. Output

PROJECT TIMELINE

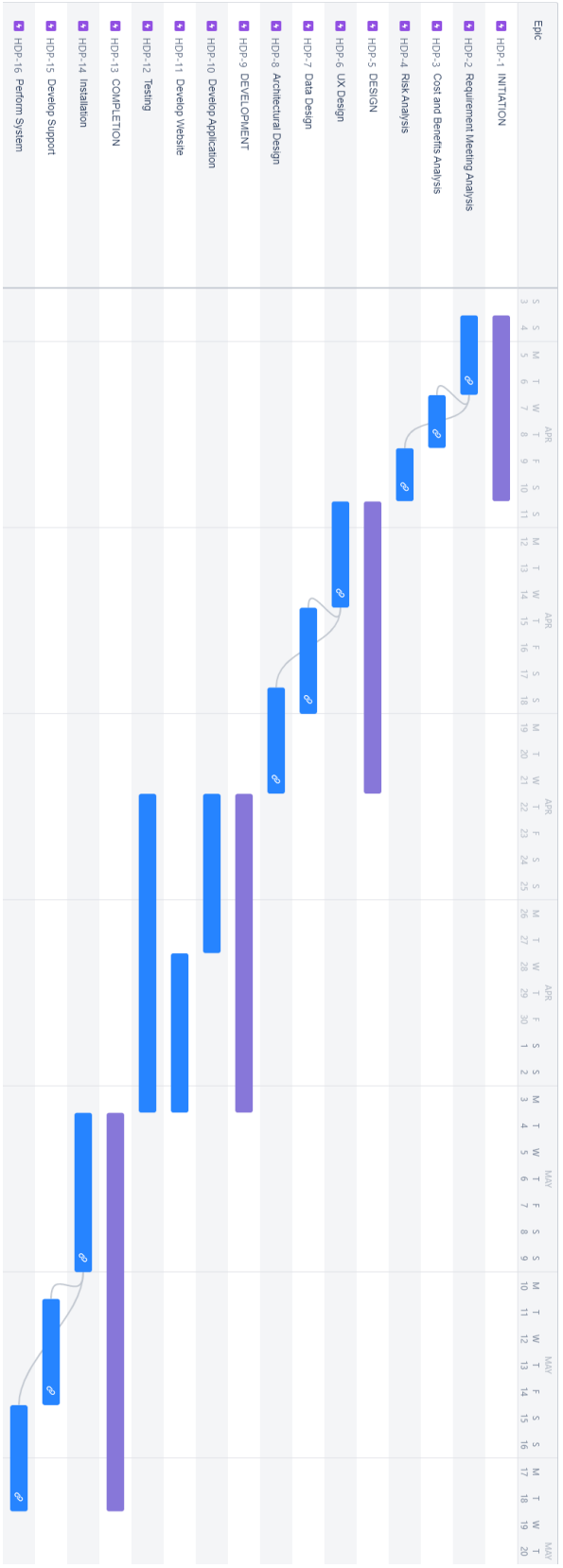


Fig 9.1. Project Timeline

TASK DISTRIBUTION

Mujtaba Shaikh

- Worked on looking for the topics and references before we start the project.
- Helped in selecting suitable features and models.
- Worked on functions to plot the table of results.
- And also Models like Form validation and extracting data from created dataset.
- Created Data Flow Diagram, Use Case Diagram and others for good understanding of various models and features.

Sanurhanaan Shaikh

- Worked on Data Cleaning, Data Imputation and Data Scaling.
- Created Exploratory Analysis on various models.
- Handled result analysis and Accuracy testing for different algorithms.
- Worked on the first three models Logistic KNN, Random Forest and XGBoost.

Saiqa Khan

- Worked on building suitable models and implementing it in the project.
- Worked on web applications for creating User friendly interface.
- Helped in getting the accurate from each model by training and testing dataset.
- Maintained a good flow to project by designing flow charts and block diagram.

All of the members spent effort writing and documentation of the Final Report.

CONCLUSION AND FUTURE SCOPE

11.1. CONCLUSION

The proposed system is GUI-based, user-friendly, scalable, reliable and an expandable system. The proposed working model can also help in reducing treatment costs by providing Initial diagnostics in time. The model can also serve the purpose of training tool for medical students and will be a soft diagnostic tool available for physician and cardiologist. General physicians can utilize this tool for initial diagnosis of cardio-patients. There are many possible improvements that could be explored to improve the scalability and accuracy of this prediction system.

As we have developed a generalized system, in future we can use this system for the analysis of different data sets. The performance of the health's diagnosis can be improved significantly by handling numerous class labels in the prediction process, and it can be another positive direction of research. In DM warehouse, generally, the dimensionality of the heart database is high, so identification and selection of significant attributes for better diagnosis of heart disease are very challenging tasks for future research.

11.2. FUTURE SCOPE

In future, we are planning to introduce an efficient disease prediction system to predict the heart disease with better accuracy utilizing different data mining classification techniques such as Decision Tree, Naïve Bayes, and Support Vector Machine (SVM).

As an extension to this work, and some sort of limitation to the work performed here, different types of classifiers can be included in the analysis and more in depth sensitivity analysis can be performed on these classifiers, also an extension can be made by applying same analysis to other bioinformatics diseases' datasets, and see the performance of these classifiers to classify and predict these diseases.

REFERENCES

[1] J. S. Rumsfeld, K. E. Joynt, and T. M. Maddox, “Big data analytics to improve cardiovascular care: promise and challenges”, Nature Reviews Cardiology, Vol.13, No.6, pp.350, 2016.

[2] W. Dai, T. S. Brisimi, W. G. Adams, T. Mela, V. Saligrama, and I. C. Paschalidis, “Prediction of hospitalization due to heart diseases by supervised learning methods”, International Journal of Medical Informatics, Vol.84, No.3, pp.189–197, 2015.

[3] M. Fatima and M. Pasha, "Survey of machine learning algorithms for disease diagnostic. Journal of Intelligent Learning Systems and Applications", Journal of Intelligent Learning Systems and Applications, Vol.9, No.01, pp.1, 2017.

[4] K. C. Tan, E. J. Teoh, Q. Yu, and K. C. Goh, "A hybrid evolutionary algorithm for attribute selection in data mining", Expert Systems with Applications, Vol.36, No.4, pp.8616-8630, 2009.

[5] A. F. Otoom, E. E. Abdallah, Y. Kilani, A. Kefaye, and M. Ashour, "Effective diagnosis and monitoring of heart disease", International Journal of Software Engineering and Its Applications, Vol.9, No.1, pp. 143-156, 2015.

[6] G. Parthiban and S. K. Srivatsa, "Applying machine learning methods in diagnosing heart disease for diabetic patients", International Journal of Applied Information Systems, Vol.3, No.7, pp.2249-0868, 2012.

[7] V. Chaurasia and S. Pal, "Data mining approach to detect heart diseases", International Journal of Advanced Computer Science and Information Technology, Vol.2, No.4, pp.56-66, 2014.

[8] K. Vembandasamy, R. Sasipriya, and E. Deepa, "Heart Diseases Detection Using Naive Bayes Algorithm", IJISET-International Journal of Innovative Science, Engineering & Technology, Vol.2, pp.441-444, 2015.

[9] X. Liu, X. Wang, Q. Su, M. Zhang, Y.Zhu, Q. Wang, and Q. Wang, "A hybrid classification system for heart disease diagnosis based on the rfrs method", Computational and Mathematical Methods in Medicine, Vol.2017, Article ID 8272091, 11 pages, 2017.

[10] S. R. W. Garner, “The Waikato environment for knowledge analysis”, 2007.

[11] M. Lichman, “UCI Machine Learning Repositry”, [Online]. https://archive.ics.uci.edu/, 2013.

[12] U. H. Dataset, “UCI Machine Learning Repository”, [online]. https//archive.ics.uci.edu/ml/machine-learningdatabases/heartdisease/ Hear.

[13] Çela, E. K., & Frasheri, N. (2012a). A literature review of data mining techniques used in healthcare databases. ICT Innovations 2012 Web Proceedings, 577–582.

[14] David, H. B. F., & Belcy, S. A. (2018). Heart Disease Prediction Using Data Mining Techniques, 6956(October), 1817–1823. https://doi.org/10.21917/ijsc.2018.0253

[15] Desai, S. D., Giraddi, S., Narayankar, P., Pudakalakatti, N. R., & Sulegaon, S. (2019). Back-propagation neural network versus logistic regression in heart disease classification. Advances in Intelligent Systems and Computing, 702, 133–144. https://doi.org/10.1007/978-981-13-0680-8_13

[16] Kiruthika Devi, S., Krishnapriya, S., & Kalita, D. (2016). Prediction of heart disease using data mining techniques. Indian Journal of Science and Technology, 9(39), 1291–1293. https://doi.org/10.17485/ijst/2016/v9i39/102078

[17] L. Van Cauwenberge, “Top 10 Machine Learning Algorithms”, Data Sci. Cent., 2015.

[18] S. Thirumuruganathan, “A Detailed Introduction to K-Nearest Neighbor (KNN) Algorithm”, [Online].

[19] G. Biau, “Analysis of a random forests model,” J. Mach. Learn. Res., Vol.13, pp.1063–1095, 2012.

[20] G. Louppe, “Understanding random forests: From theory to practice”, arXiv Prepr. arXiv1407.7502, 2014.

[21] C. M. Bishop, “Pattern recognition and machine learning”, Inf. Sci. Stat., 2006.

[22] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, “The WEKA Data Mining Software: An Update,” SIGKDD Explor. Newsl., Vol.11, No.1, pp.10–18, 2009.

SOURCE CODE

Heart_disease.ipynb

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

df = pd.read_csv("heart_deasease.csv")
df.head(6)
df.shape

#feature engineering
df.isnull().sum()
plt.figure(figsize=(20,20))
ax = sns.boxplot(data=df)

from scipy import stats
z = np.abs(stats.zscore(df))
print(z)
threshold = 3
print(np.where(z > 3))# The first array contains the list of row numbers and se
cond array respective column numbers

Q1 = df.quantile(0.25)
Q3 = df.quantile(0.75)
IQR = Q3 - Q1
print(IQR)

df = df[(z < 3).all(axis=1)]
df.shape

df = df[~((df < (Q1 - 1.5 * IQR)) |(df > (Q3 + 1.5 * IQR))).any(axis=1)]
df.shape

plt.figure(figsize=(20,20))
ax = sns.boxplot(data=df)

#feature selection
plt.figure(figsize=(20,20))
d = sns.heatmap(df.corr(),cmap="coolwarm",annot= True)
df.describe()
```

```
#feature scalling
from sklearn.preprocessing import StandardScaler
standardScaler = StandardScaler()
dataset = pd.get_dummies(df, columns = ['sex', 'cp', 'fbs', 'restecg', 'exang',
 'slope', 'ca', 'thal'])# creating dummy variable
columns_to_scale = ['age', 'trestbps', 'chol', 'thalach', 'oldpeak'] # we have
taken these columns for scale down
dataset[columns_to_scale] = standardScaler.fit_transform(dataset[columns_to_sca
le])
dataset.head()
dataset.tail()
dataset.describe()

#visualisation
sns.pairplot(df , hue="target", height=3, aspect=1);

#model selection
y = dataset['target']
X = dataset.drop(['target'], axis = 1)

from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2, rand
om_state= 5)

from sklearn.model_selection import cross_val_score
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import confusion_matrix, classification_report, accuracy_s
core

# K - Nearest Neighbor Classifier
knn_classifier = KNeighborsClassifier(n_neighbors = 5)
knn_classifier.fit(X_train, y_train)
score=cross_val_score(knn_classifier,X_train,y_train,cv=10)
y_pred_knn = knn_classifier.predict(X_test)
accuracy_score(y_test, y_pred_knn)
score.mean()

knn_classifier  = KNeighborsClassifier(algorithm='auto', leaf_size=30, metric='
minkowski',
 metric_params=None, n_jobs=1, n_neighbors=5, p=1,
 weights='uniform')
knn_classifier.fit(X_train, y_train)
score=cross_val_score(knn_classifier,X_train,y_train,cv=10)
y_pred_knn = knn_classifier.predict(X_test)
knn_accur = accuracy_score(y_test, y_pred_knn)*100
accuracy_score(y_test, y_pred_knn)
score=cross_val_score(knn_classifier,X_train,y_train,cv=10)
score.mean()
```

```
#confusion matrix
cm = confusion_matrix(y_test, y_pred_knn)
plt.title('Confusion Matrix of K-NN', fontsize = 15)
sns.heatmap(cm, annot = True)
plt.show()

print(classification_report(y_test, y_pred_knn))

# Random Forest Classifier
from sklearn.ensemble import RandomForestClassifier
rf_classifier = RandomForestClassifier(n_estimators = 20, criterion = 'entropy'
, random_state = 51)
rf_classifier.fit(X_train, y_train)
y_pred_rf = rf_classifier.predict(X_test)
rf_accur = accuracy_score(y_test, y_pred_rf)*100
accuracy_score(y_test, y_pred_rf)
score=cross_val_score(rf_classifier,X_train,y_train,cv=10)
score.mean()

#confusion matrix
cm = confusion_matrix(y_test, y_pred_rf)
plt.title('Confusion Matrix of Random Forest', fontsize = 15)
sns.heatmap(cm, annot = True)
plt.show()

# XGBoost Classifier
from xgboost import XGBClassifier
xgb_classifier = XGBClassifier(base_score=0.5,
       colsample_bytree=0.4, gamma=0.2,
       learning_rate=0.1, max_delta_step=0, max_depth=15,
       min_child_weight=1, n_estimators=100, n_jobs=1,
       nthread=None, random_state=23,
       reg_alpha=0, reg_lambda=0.47, scale_pos_weight=1, seed=42,
       silent=None, subsample=1, verbosity=1)
xgb_classifier.fit(X_train, y_train)
y_pred_xgb = xgb_classifier.predict(X_test)
xgb_accur = accuracy_score(y_test, y_pred_xgb)*100
accuracy_score(y_test, y_pred_xgb)
score=cross_val_score(xgb_classifier,X_train,y_train,cv=10)
score.mean()

#confusion matrix
cm = confusion_matrix(y_test, y_pred_xgb)
plt.title('Confusion Matrix of XGBoost', fontsize = 15)
sns.heatmap(cm, annot = True)
plt.show()
```

```
# AdaBoost with Random Forest
from sklearn.ensemble import AdaBoostClassifier
from sklearn.ensemble import RandomForestClassifier
ada_clf = AdaBoostClassifier(RandomForestClassifier(n_estimators=100), n_estima
tors=100)
ada_clf.fit(X_train, y_train)
y_pred_adb = ada_clf.predict(X_test)
arf_accur= accuracy_score(y_test, y_pred_adb)*100
accuracy_score(y_test, y_pred_adb)
score=cross_val_score(ada_clf,X_train,y_train,cv=10)
score.mean()

#confusion matrix
cm = confusion_matrix(y_test, y_pred_adb)
plt.title('Confusion Matrix of ADAboost', fontsize = 15)
sns.heatmap(cm, annot = True)
plt.show()

# Gradient Boosting
from sklearn.ensemble import GradientBoostingClassifier
gbc_clf = GradientBoostingClassifier()
gbc_clf.fit(X_train, y_train)
y_pred_adb = gbc_clf.predict(X_test)
gb_accur = accuracy_score(y_test, y_pred_adb)*100
accuracy_score(y_test, y_pred_adb)
score=cross_val_score(gbc_clf,X_train,y_train,cv=10)
score.mean()

#confusion matrix
cm = confusion_matrix(y_test, y_pred_adb)
plt.title('Confusion Matrix of Gradient Boosting', fontsize = 15)
sns.heatmap(cm, annot = True)
plt.show()

# Accuracy of all models
accur = {'Classifier':['K Nearest Neighbour', 'Random Forest', 'XGBoost Classif
ier', 'Adaboost with Ran. F', 'Gradient Boost'],
         'Accuracy':[knn_accur, rf_accur, xgb_accur, arf_accur, gb_accur]}
df = pd.DataFrame(accur)
df

# save model
## Pickle
from xgboost import XGBClassifier
import pickle
```

```
# save model
pickle.dump(knn_classifier, open('model.pkl', 'wb'))

# load model
Heart_disease_detector_model = pickle.load(open('model.pkl', 'rb'))

# predict the output
y_pred = Heart_disease_detector_model.predict(X_test)

# confusion matrix
print('Confusion matrix of K – Nearest Neighbor model: \n',confusion_matrix(y_t
est, y_pred),'\n')

# show the accuracy
print('Accuracy of K – Nearest Neighbor  model = ',accuracy_score(y_test, y_pre
d))
```

Index.html

```
<!-
- This is GUI for Heart disease Detection Application Using Machine Learning Classi
fer -->

<!DOCTYPE html>
<html>
<head>
        <title>
                Heart Disease Prediction
        </title>
        <link rel="shortcut icon" href="static/images/heart.png" type="image/x-
icon">
        <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.
2/css/bootstrap.min.css">
</head>
<body>
<div class="container">
        <div class="card">
                <div class="card-header text-center"> Hear Disease Prediction</div>
                <div class="card-body">

        <form class=" m-auto" action="{{url_for('predict')}} "method="post">
        <center>
        <img class="center" src="static/images/heart.gif" alt="Girl in a jacket">

        <h3>Enter the value of heart features</h3>
        <br>
        </center>

        <div class="form-group">
          <table style="padding: 10px; width: 100%">
            <tr>
                <td style="padding: 10px; width: 33%;"><label>Patient's Age (Range
25 - 100 in years):</label>
                <input class="form-
control" type="text" name="age" placeholder=" " required="required"></td>
                <td style="padding: 10px; width: 33%;" ><label for="trestbps">Rest
Blood Pressure (Range 80 - 200 mm Hg):</label>
                <input class="form-
control" type="text" name="trestbps" placeholder=" " required="required"></td>
                <td style="padding: 10px;width: 33%;"><label for="chol">Serum Chole
strol (Range 80 - 600 in mg/dl):</label>
                <input class="form-
control" type="text" name="chol" placeholder=" " required="required"></td>
            </tr>
```

```
              <td style="padding: 10px; width: 33%;"><label for="thalach">Max Hear
t Rate (Range 80 - 200):</label>
                <input class="form-
control" type="text" name="thalach" placeholder=" " required="required"></td>
                <td style="padding: 10px; width: 33%;"><label for="oldpeak">S.T Dep
ression induced (Range 0.0 - 5.0):</label>
                <input class="form-
control" type="text" name="oldpeak" placeholder=" " required="required"></td>
                <td style="padding: 10px; width: 33%;"><label for="sex_0">Gender (1
 = Male, 0 = Female)</label>
                <input class="form-
control" type="text" name="sex_0" placeholder=" " required="required"></td>
            </tr>

            <tr>
              <td style="padding: 10px; width: 33%;"><label for="sex_1">If the Pati
ent has Diabeties or not ? (1 = Yes, 0 = No):</label>
        <input class="form-
control" type="text" name="sex_1" placeholder=" " required="required"></td>
              <td style="padding: 10px; width: 33%;"><label for="cp_0">Chest Pain M
easurement (0 = typical, 1 = atypical, 2 = non aginal, 3 = asympotic):</label>
  <input class="form-
control" type="text" name="cp_0" placeholder=" " required="required"></td>
  <td style="padding: 10px; width: 33%;"><label for="cp_1">Chest Pain Measurement (
0 = typical, 1 = atypical, 2 = non aginal, 3 = asympotic)</label>
        <input class="form-
control" type="text" name="cp_1" placeholder=" " required="required"></td>
            </tr>
            <tr>
              <td style="padding: 10px; width: 33%;"><label for="cp_2">Chest Pain T
ype (0 = none, 1 = mild, 2 = high, 3 = extreme):</label>
        <input class="form-
control" type="text" name="cp_2" placeholder=" " required="required"></td>
              <td style="padding: 10px; width: 33%;"><label for="cp_3">Chest Pain T
ype (0 = none, 1 = mild, 2 = high, 3 = extreme):</label>
      <input class="form-
control" type="text" name="cp_3" placeholder=" " required="required"></td>
              <td style="padding: 10px; width: 33%;"><label for="fbs_0">Fasting Blo
od Sugar (if > 120 mg/dl: 1 = True, 0 = False):</label>
        <input class="form-
control" type="text" name="fbs_0" placeholder=" " required="required"></td>
            </tr>
            <tr>
              <td style="padding: 10px; width: 33%;"><label for="restecg_0">Resting
 Electro Cardiographic Result (0 = normal, 1 = ST-T wave, 2 = LV hyp.):</label>
  <input class="form-
control" type="text" name="restecg_0" placeholder=" " required="required"></td>
```

```
            <td style="padding: 10px; width: 33%;"><label for="restecg_1">Resting
 Electro Cardiographic Result (0 = normal, 1 = ST-T wave, 2 = LV hyp.):</label>
       <input class="form-
control" type="text" name="restecg_1" placeholder=" " required="required"></td>
            <td style="padding: 10px; width: 33%;"><label for="restecg_2">Resting
 Electro Cardiographic Result (0 = normal, 1 = ST-T wave, 2 = LV hyp.):</label>
       <input class="form-
control" type="text" name="restecg_2" placeholder=" " required="required"></td>
          </tr>
          <tr>
          <td style="padding: 10px; width: 33%;"><label for="exang_0">Exercise In
duced Angina (1 = Yes, 0 = No):</label>
     <input class="form-
control"  type="text" name="exang_0" placeholder=" " required="required"></td>
          <td style="padding: 10px; width: 33%;"><label for="exang_1">Exercise In
duced Angina (1 = Yes, 0 = No):</label>
       <input class="form-
control"  type="text" name="exang_1" placeholder=" " required="required"></td>
          <td style="padding: 10px; width: 33%;"><label for="slope_0">Slope of th
e Peak Exercise (1 = up-slope, 2 = flat 3 = down-slope):</label>
  <input class="form-
control"  type="text" name=" slope_0" placeholder=" " required="required"></td>
          </tr>
          <tr>
            <td style="padding: 10px; width: 33%;"><label for="slope_1">Slope of
the Peak Exercise (1 = up-slope, 2 = flat 3 = down-slope):</label>
       <input class="form-
control"  type="text" name=" slope_1" placeholder=" " required="required"></td>
            <td style="padding: 10px; width: 33%;"><label for="slope_2">Slope of
the Peak Exercise (1 = up-slope, 2 = flat 3 = down-slope):</label>
     <input class="form-
control"  type="text" name="  slope_2" placeholder=" " required="required"></td>
            <td style="padding: 10px; width: 33%;"><label for="ca_0"> No. of Majo
r Vessels Colored by Flourosopy (range: 0-3)</label>
     <input class="form-
control" type="text" name=" ca_0" placeholder=" " required="required"></td>
          </tr>
          <tr>
            <td style="padding: 10px; width: 33%;"><label for="ca_1"> No. of Majo
r Vessels Colored by Flourosopy (Range: 0 - 3):</label>
     <input class="form-
control" type="text" name="ca_1" placeholder="" required="required"></td>
            <td style="padding: 10px; width: 33%;"><label for="ca_2"> No. of Majo
r Vessels Colored by Flourosopy (range: 0 - 3):</label>
     <input class="form-
control" type="text" name="ca_2" placeholder="" required="required"></td>
```

```
              <td style="padding: 10px; width: 33%;"><label for="thal_1">Thalassemi
a (Range: 1 = Normal, 2 = Fixed Defect. 3 = Reversable Defect):</label>
      <input class="form-
control" type="text" name=" thal_1" placeholder="" required="required"></td>
            </tr>
            <tr>
              <td style="padding: 10px; width: 33%;"><label for="thal_2">Thalassemi
a (Range: 1 = Normal, 2 = Fixed Defect. 3 = Reversable Defect):</label>
      <input class="form-
control" type="text" name="thal_2" placeholder=" " required="required"></td>
              <td style="padding: 10px; width: 33%;"><label for="thal_2">Thalassemi
a (Range: 1 = Normal, 2 = Fixed Defect. 3 = Reversable Defect):</label>
      <input class="form-
control" type="text" name="thal_3" placeholder=" " required="required"></td>
            </tr>
          </table>
          </div>
        <!-- Show button -->
        <div class="button_cont" align="center"><a class="button_css" href="https:/
/github.com/hannan13231/Heart2Hold.git" target="_blank" rel="nofollow noopener">
            <button type="submit" class="btn btn-primary btn-block btn-
large"><strong>Predict Heart disease</strong></button></a></div>
        </form>
                </div>
        <center>
        <h2><u>{{prediction_text}}</u></h2>
        </center>
                <div class="card-footer text-
center"> Take Care of Your Health</div>

  </div>
        </div>
</div>

<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></sc
ript>
<script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.16.0/umd/popper.min
.js"></script>
<script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"><
/script>
</body>
</html>
```

App.py

```
import numpy as np
import pandas as pd
from flask import Flask, request, render_template
import pickle

app = Flask(__name__)
model = pickle.load(open('model.pkl', 'rb'))

@app.route('/')
def home():
    return render_template('index.html')

@app.route('/predict',methods=['POST'])
def predict():
    input_features = [float(x) for x in request.form.values()]
    features_value = [np.array(input_features)]

    features_name = [ "age", "trestbps","chol","thalach", "oldpeak", "sex_0",
                      "  sex_1", "cp_0", "cp_1", "cp_2", "cp_3","  fbs_0",
                       "restecg_0","restecg_1","restecg_2","exang_0","exang_1",
                      "slope_0","slope_1","slope_2","ca_0","ca_1","ca_2","thal_1”,
                       "thal_2","thal_3"]

    df = pd.DataFrame(features_value, columns=features_name)
    output = model.predict(df)

    if output == 1:
        res_val = "Heart Disease"
    else:
        res_val = "No Heart Disease "

    return render_template('index.html', prediction_text='Patient has {}'.format(re
s_val))

if __name__ == "__main__":
    app.run()
```