



# **Hybrid Heart Care-Advancing Heart Disease Prediction With A Machine Learning Approach**

*A Major Project Report submitted*

*in partial fulfillment for the award of the Degree of*

**Bachelor of Technology  
in  
Computer Science and Engineering  
by**

**DEVABRATH SAHU (U20CS232)**

**E. ABHINASH GOUD (U20CS260)**

**E. LAKSHMA REDDY (U20CS257)**

**G.V.RAVINDRA REDDY (U20CS273)**

*Under the guidance of*  
**Dr. D. THIRIPURASUNDARI**



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING  
SCHOOL OF COMPUTING  
BHARATH INSTITUTE OF HIGHER EDUCATION AND RESEARCH  
(Deemed to be University Estd u/s 3 of UGC Act, 1956)  
CHENNAI 600 073, TAMILNADU, INDIA**

April, 2024



**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**

**BONAFIDE CERTIFICATE**

**This is to Certify that this Innovation Project Report Titled “HYBRID HEART CARE-ADVANCING HEART DISEASE PREDICTION WITH A MACHINE LEARNING APPROACH” is the Bonafide Work of DEVABRATH SAHU(U20CS232),E. ABHINASH GOUD(U20CS260), E.LAKSHMA REDDY(U20CS257) and G.V.RAVINDRA REDDY (U20CN299)**

of Final Year B.Tech. (CSE) who carried out the Majorproject work under my supervision Certified further, that to the best of my knowledge the work reportedhere in does not form part of any other projectreport or dissertation on basis of which a degree or award conferred on an earlieroccasion by any other candidate.

**PROJECT GUIDE**

**HEAD OF THE DEPARTMENT**

Dr. D. THIRIPURASUNDARI

Dr. S. Maruthuperumal

Professor

Professor

Department of CSE

Department of CSE

BIHER

BIHER

Submitted for the Project Viva -Voice held on .....

## DECLARATION

We declare that this Majorproject report titled “**HYBRID HEART CARE-ADVANCING HEART DISEASE PREDICTION WITH A MACHINE LEARNING APPROACH**” submitted in partial fulfillment of the degree of **B. Tech in (Computer Science and Engineering)** is a record of original work carried out by us under the supervision of **Dr.D.THIRIPURASUNDARI**, and has not formed the basis for the award of any other degree or diploma, in this or any other Institution or University. In keeping with the ethical practice in reporting scientific information, due acknowledgements have been made wherever the findings of others have been cited.

DEVABRATH SAHU

U20CS232

E. ABHINASH GOUD

U20CS260

E. LAKSHMA REDDY

U20CS257

G. V. RAVINDRA REDDY

U20CS273

Chennai

Date:-

## ACKNOWLEDGMENTS

We express our sincere thanks to our beloved Honorable Chairman **Dr S.Jagathrakshakan, M.P.**, for continuous and constant encouragement in all academic activities.

We express our deepest gratitude to our beloved President **Dr. J. Sundeep Aanand** President , and Managing Director **Dr.E.Swetha Sundeep Aanand** Managing Director for providing us the necessary facilities to complete our project.

We take great pleasure in expressing sincere thanks to **Dr. K. VijayaBaskar Raju** Pro Chancellor, **M. Sundararajan** Vice Chancellor (vc),**Dr. S. Bhuminathan** Registrar and **Dr. R. Hariprakash** Additional Registrar, **Dr. M. Sunderaraj** Dean Academics for moldings our thoughts to complete our project.

We thank our **Dr. S. Neduncheliyan** Dean, School of Computing for his encouragement and the valuable guidance.

We record indebtedness to our Head, **Dr. S. Maruthuperumal**, Department of Computer Science and Engineering for his immense care and encouragement towards us throughout the course of this project.

We also take this opportunity to express a deep sense of gratitude to our Supervisor **Dr. D. THIRIPURASUNDARI** and our Project Co-Ordinator **Dr. K. BAALAJI** for their cordial support, valuable information and guidance, they helped us in completing this project through various stages.

We thank our department faculty, supporting staff and friends for their help and guidance to complete this project.

**DEVABRATH SAHU (U20CS232)**

**E. ABHINASH GOUD (U20CS260)**

**E, LAKSHMA REDDY (U20CS257)**

**G. V. RAVINDRA REDDY (U20CS273)**

# ABSTRACT

Heart disease (HD) is the critical health issue and numerous people have been suffered by this disease around the world .The HD occurs with common symptoms of breath shortness, physical body weakness and, feet are swollen. Researchers try to come across an efficient technique for the detection of heart disease, as the current diagnosis techniques of heart disease are not much effective in early time identification due to several reasons, such as accuracy and execution time. In this project, we proposed an efficient and accurate system to diagnosis heart disease and the system is based on machine learning techniques. The system is developed based on classification algorithms includes support vector machine, decision tree and Random forest. Then to improve the accuracy we implemented hybrid algorithm by combining all the above algorithm. The performance measuring metrics are used for assessment of the performances of the classifiers. The performances of the classifiers have been checked on the selected features as selected by features selection algorithms. The final prediction is given in flask framework webpage. Additionally, the proposed system can suggest food for positive patients.

## TABLE OF CONTENTS

S.NO	CONTENT	PGNO
	ACKNOWLEDGEMENTS TABLE OF CONTENTS LIST OF TABLES LIST OF FIGURES LIST OF ABBREVIATIONS ABSTRACT	
1	Introduction 1.1 Overview 1.2 Objective 1.3 Challenges in Domain 1.4 Motivation For the Work 1.5 Problem Statement	
2	Literature survey	
3	System analysis 3.1 Existing system 3.1.1 Objective 3.1.2 Scope and Application 3.2 proposed system 3.2.1 Advantages of Proposed System 3.3 Modules	
4	System design 4.1 System architecture 4.1.1 Data Collection 4.1.2 Data Preprocessing 4.1.3 Model Training 4.1.4 Random Forest 4.1.5 SVM	

	<ul style="list-style-type: none"> <li>4.1.6 Decision Tree</li> <li>4.1.7 Hybrid Algorithm</li> <li>4.1.8 Model Testing</li> <li>4.1.9 Install Task</li> </ul>	
5	<ul style="list-style-type: none"> <li>UML Diagrams <ul style="list-style-type: none"> <li>5.1 UML Diagrams <ul style="list-style-type: none"> <li>5.1.1 Use case Diagram</li> <li>5.1.2 Class Diagram</li> <li>5.1.3 Sequence Diagram</li> <li>5.1.4 Dataflow Diagram</li> <li>5.1.5 ER Diagram</li> </ul> </li> </ul> </li> </ul>	
6	Software Environment	
7	<ul style="list-style-type: none"> <li>Implementation <ul style="list-style-type: none"> <li>7.1 Coding</li> <li>7.2 coding Standards <ul style="list-style-type: none"> <li>7.2.1 Naming Conventions</li> <li>7.2.2 Value Conventions</li> <li>7.2.3 Script Writing and Commenting Standards</li> <li>7.2.4 Message Box Formatting</li> </ul> </li> <li>7.3 Code</li> </ul> </li> </ul>	
8	<ul style="list-style-type: none"> <li>System Testing <ul style="list-style-type: none"> <li>8.1 Test Data and Output <ul style="list-style-type: none"> <li>8.1.1 Unit Testing</li> <li>8.1.2 Functional Testing</li> <li>8.1.3 Integration Testing</li> </ul> </li> <li>8.2 Testing Techniques <ul style="list-style-type: none"> <li>8.2.1 Testing <ul style="list-style-type: none"> <li>8.2.1.1 White Box Testing</li> <li>8.2.1.2 Black Box Testing</li> <li>8.2.2.2 Program Testing</li> </ul> </li> </ul> </li> </ul> </li> </ul>	

	8.2.2.3 Security Testing 8.2.2.4 Validation Testing	
9	Screenshots 9.1 Home Screen 9.2 Healthy Case 9.3 Risk Case 9.4 Diet Chart	
10	Conclusion	
11	References	

# 1.INTRODUCTION

## 1.1 OVERVIEW

According to the World Health Organization, every year 12 million deaths occur worldwide due to Heart Disease. Heart disease is one of the biggest causes of morbidity and mortality among the population of the world. Prediction of cardiovascular disease is regarded as one of the most important subjects in the section of data analysis. The load of cardiovascular disease is rapidly increasing all over the world from the past few years. Many researchers have been conducted in attempt to pinpoint the most influential factors of heart disease as well as accurately predict the overall risk. Heart Disease is even highlighted as a silent killer which leads to the death of the person without obvious symptoms. The early diagnosis of heart disease plays a vital role in making decisions on lifestyle changes in high-risk patients and in turn reduces the complications.

Machine learning proves to be effective in assisting in making decisions and predictions from the large quantity of data produced by the health care industry. This project aims to predict future Heart Disease by analyzing data of patients which classifies whether they have heart disease or not using machine-learning algorithm. Machine Learning techniques can be a boon in this regard. Even though heart disease can occur in different forms, there is a



common set of core risk factors that influence whether someone will ultimately be at risk for heart disease or not. By collecting the data from various sources, classifying them under suitable headings & finally analysing to extract the desired data we can say that this technique can be very well adapted to do the prediction of heart disease.

Machine learning is important because it gives enterprises a view of trends in customer behavior and business operational patterns, as well as supports the development of new products. Many of today's leading companies, such as Facebook, Google and Uber, make machine learning a central part of their operations. Machine learning has become a significant competitive differentiator for many companies.

- **Supervised learning:** In this type of machine learning, data scientists supply algorithms with labeled training data and define the variables they want the algorithm to assess for correlations. Both the input and the output of the algorithm is specified.
- **Unsupervised learning:** This type of machine learning involves algorithms that train on unlabeled data. The algorithm scans through data sets looking for any meaningful connection. The data that algorithms train on as well as the predictions or recommendations they output are predetermined.
- **Semi-supervised learning:** This approach to machine learning involves a mix of the two preceding types. Data scientists may feed an algorithm mostly labeled training data, but the model is free to explore the data on its own and develop its own understanding of the data set.
- **Reinforcement learning:** Data scientists typically use reinforcement learning to teach a machine to complete a multi-step process for which there are clearly defined rules. Data scientists program an algorithm to complete a task and give it positive or negative cues as it works out how to complete a task. But for the most part, the algorithm decides on its own what steps to take along the way.

Supervised machine learning requires the data scientist to train the algorithm with both labeled inputs and desired outputs. Supervised learning algorithms are good for the following tasks:

- Binary classification: Dividing data into two categories.
- Multi-class classification: Choosing between more than two types of answers.
- Regression modeling: Predicting continuous values.
- Ensembling: Combining the predictions of multiple machine learning models to produce an accurate prediction.

## 1.2 OBJECTIVE

- To measured with the help of graphical notations such as pie chart, bar chart and line chart. The data can be given in a dynamical data.
- The main objective of this research is to develop an Intelligent System using machine learning technique, namely logistic regression, KNN and Random forest.
- It is implemented as web based application and the result can be notified to the doctors mail id.

## 1.3 CHALLENGES IN THE DOMAIN

- Data Quality: The quality and quantity of data available for model training can greatly affect the accuracy and performance of machine learning models. Data may be missing, noisy, or biased, which can lead to incorrect model predictions.

- **Overfitting:** Overfitting occurs when a model is too complex and learns the training data too well, resulting in poor generalization to new, unseen data.
- **Model Selection:** There are many different types of machine learning models to choose from, and the choice of model can greatly affect the results. The right model needs to be selected based on the specific problem being solved and the available data.
- **Hyperparameter tuning:** Many machine learning models have hyperparameters that need to be tuned to achieve optimal results. This can be a time-consuming and iterative process that requires a good understanding of the underlying model and data.
- **Explainability:** Many machine learning models, especially deep learning models, are considered black boxes, meaning it is difficult to understand how they arrive at their predictions. This lack of transparency can be a challenge for organizations looking to deploy these models in decision-making processes.
- **Computational Requirements:** Training complex machine learning models can require a significant amount of computational resources, which can be a challenge for organizations with limited hardware capabilities.
- **Data Privacy and Security:** With the increasing use of machine learning in sensitive applications, such as healthcare and finance, protecting sensitive data during the model training process is becoming increasingly important.

## 1.4 MOTIVATION FOR THE WORK

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 three classification algorithms namely Naïve logistic regression, KNN and Random forest 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. This will provide researchers and medical practitioners to establish a better.

## 1.5 PROBLEM STATEMENT

The major challenge in heart disease is its detection. There are instruments available

which can predict heart disease but either it are expensive or are not efficient to calculate chance of heart disease in human. 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 sapience, 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.

## 2.LITERATURE SURVEY

With growing development in the field of medical science alongside machine learning various experiments and researches has been carried out in these recent years releasing the relevant significant papers.

Purushottam ,et ,al [1] proposed a paper “Efficient Heart Disease Prediction System” using hill climbing and decision tree algorithms .They used Cleveland dataset and preprocessing of data is performed before using classification algorithms. The Knowledge Extraction is done based on Evolutionary Learning (KEEL), an open- source data mining tool that fills the missing values in the data set.A decision tree follows top-down order. For each actual node selected by hill-climbing algorithm a node is selected by a test at each level. The parameters and their values used are confidence. Its minimum confidence value is 0.25. The accuracy of the system is about 86.7%.

Santhana Krishnan. J ,et ,al [2] proposed a paper “Prediction of Heart Disease Using Machine Learning Algorithms” using decision tree and Naive Bayes algorithm for prediction of heart disease. In decision tree algorithm the tree is built using certain conditions which gives True or False decisions. The algorithms like SVM, KNN are results based on vertical or horizontal split conditions depends on dependent variables. But decision tree for a tree like structure having root node, leaves and branches base on the decision made in each of tree Decision tree also help in the understating the importance of the attributes in the dataset. They have also used Cleveland data set. Dataset splits in 70% training and 30% testing by using some methods. This algorithm gives 91% accuracy. The second algorithm is Naive Bayes, which is used for classification. It can handle complicated, nonlinear, dependent data so it is found suitable for heart disease dataset as this dataset is also complicated, dependent and nonlinear in nature. This algorithm gives an 87% accuracy.

Sonam Nikhar et al [3] proposed paper “ Prediction of Heart Disease Using Machine Learning Algorithms” their research gives point to point explanation of Naïve Bayes and

decision tree classifier that are used especially in the prediction of Heart Disease. Some analysis has been led to think about the execution of prescient data mining strategy on the same dataset, and the result decided that Decision Tree has highest accuracy than Bayesian classifier.

Aditi Gavhane et al [4] proposed a paper “Prediction of Heart Disease Using Machine Learning”, in which training and testing of dataset is performed by using neural network algorithm multi-layer perceptron. In this algorithm there will be one input layer and one output layer and one or more layers are hidden layers between these two input and output layers. Through hidden layers each input node is connected to output layer. This connection is assigned with some random weights. The other input is called bias which is assigned with weight based on requirement the connection between the nodes can be feedforwarded or feedback.

Avinash Golande et al, [5] proposed “Heart Disease Prediction Using Effective Machine Learning Techniques” in which few data mining techniques are used that support the doctors to differentiate the heart disease. Usually utilized methodologies are k-nearest neighbour, Decision tree and Naïve Bayes. Other unique characterization-based strategies utilized are packing calculation, Part thickness, consecutive negligible streamlining and neural systems, straight Kernel selfarranging guide and SVM (Bolster Vector Machine).

Lakshmana Rao et al, [6] proposed “Machine Learning Techniques for Heart Disease Prediction” in which the contributing elements for heart disease are more. So, it is difficult to distinguish heart disease. To find the seriousness of the heart disease among people different neural systems and data mining techniques are used.

Abhay Kishore et al [7] proposed “Heart Attack Prediction Using Deep Learning” in which heart attack prediction system by using Deep learning techniques and to predict the probable aspects of heart related infections of the patient Recurrent Neural System is used. This model uses deep learning and data mining to give the best precise model and least blunders. This paper acts as strong reference model for another type of heart attack prediction models

Senthil Kumar Mohan et al, [8] proposed “Effective Heart Disease Prediction Using Hybrid Machine Learning Techniques” in which their main objective is to improve exactness in cardiovascular problems. The algorithms used are KNN, LR, SVM, NN to produce an improved exhibition level with a precision level of 88.7% through the prediction model for heart disease with hybrid random forest with linear model(HRFLM).

Anjan N. Repaka et al, [9] proposed a model stated the performance of prediction for two classification models, which is analyzed and compared to previous work. The experimental results show that accuracy is improved in finding the percentage of risk prediction of our proposed method in comparison with other models.

Aakash Chauhan et al,[10] proposed “Heart Disease Prediction using Evolutionary Rule Learning”. Data is directly retrieved from electronic records that reduce the manual tasks. The amount of services are decreased and shown major number of rules helps within the best prediction of heart disease. Frequent pattern growth association mining is performed on patient’s dataset to generate strong association.

S:no	AUTHOR	year of publish	DISCRIPTION	ADVANTAGE	DISADVANTAGE
1	Ceerthi Bala U.K and Sarath TV	2023	.This diagnosis is a difficult task i.e. it should be performed precisely and efficiently.	Once it is detected with the particular sensor, the system will alert the driver	Low accuracy
2	Rateb Jabbar et al.	2022	We prepared a heart disease prediction system to predict whether the patient is likely to be diagnosed with a heart disease or not using the medical history of the patient.	When the developed model applies with the real time camera, the drowsiness can be detected considerably	As a result, they found 88% accuracy without sensor, and a sensor system detects 85% accuracy
3	Dr.K.S.Tiwari et al	2021	The correct prediction of heart disease can prevent life threats, and incorrect prediction can prove to be fatal at the same time	A buzzer and camera placed with a microcontroller continuously monitor the pulses with the necessary parameters.	It not detect with diagnosis system.



s:no	author	year of publish	description	advantage	disadvantage
4	Acharya et al.	2021	Combined CNNs, SVMs, and decision trees to predict heart disease risk based on ECG signals	Improved accuracy due to combination of multiple models	Requires large amount of data to train multiple models
5	Mani et al.	2021	Combined k-NN, decision trees, and SVMs to predict heart disease risk based on clinical data	Improved accuracy and interpretability compared to individual models	May be limited by quality of available clinical data

s:no	author	year of publish	description	advantage	disadvantage
------	--------	-----------------	-------------	-----------	--------------

s:no	author	year of publish	description	advantage	disadvantage
	Naksuk et al.	2020	Combined logistic regression, random forests, and neural networks to predict risk of major adverse cardiac events	Achieved higher accuracy than individual models	Limited to predicting specific outcome (major adverse cardiac events)
7	Wei et al.	2020	Combined CNNs and LSTM models to predict heart disease risk based on ECG signals	Achieved higher accuracy than individual models	Limited to predicting risk of coronary artery di

8	Zaman et al.	2019	Combined logistic regression, SVMs, decision trees, and random forests to predict risk of heart disease based on clinical data	Improved accuracy and interpretability compared to individual models	May be limited by quality of available clinical
9	Vigneshwaran et al.	2019	Combined CNNs and SVMs to predict heart disease risk based on ECG signals	Achieved higher accuracy than individual models	Limited to predicting risk of coronary artery disease

## 3. SYSTEM ANALYSIS

### 3.1 EXISTING SYSTEM

- Accurately predicting heart disease risk requires access to a wide range of patient data, including medical history, lifestyle factors, and family history. In many cases, this data may be incomplete or difficult to obtain, making it challenging to predict heart disease risk.
- Some traditional methods for predicting heart disease risk, such as the Framingham risk score, may have limited predictive power.
- Limited ability to adapt: Traditional methods for predicting heart disease risk may not be able to adapt to new data or changing conditions.

#### 3.1.1 OBJECTIVE

- To measured with the help of graphical notations such as pie chart, bar chart and line chart. The data can be given in a dynamical data.
- The main objective of this research is to develop an Intelligent System using machine learning hybrid model.
- It is implemented as web based application.

#### 3.1.2 SCOPE AND APPLICATION

- ML algorithms can be used in primary care settings to predict the risk of heart disease in patients based on a variety of different factors, such as medical history, vital signs, and lab results. This can help healthcare providers identify patients who are at high risk of heart disease and develop targeted prevention and intervention strategies.
- ML algorithms can be used in remote monitoring.
- Electronic health records and public health data.

### 3.2 PROPSOED SYTEM:

- Proposed method is a binary classification problem, where the aim is to predict the presence of heart disease in a patient based on various attributes.
- The first step is to visualize the distribution of the variables by plotting the density maps using matplotlib. Then, the correlation between the variables is visualized using a heatmap from the seaborn library.
- The next step is to prepare the data for the model building. One-hot encoding is used to encode the categorical variables 'cp', 'thal', 'slope', and 'gender' into binary variables.
- The new data is concatenated to the original data and the original categorical variables are dropped. The data is then scaled using the StandardScaler function from the sklearn library.
- Next, the data is split into training and testing sets using the train\_test\_split function from the sklearn library.
- Three different machine learning algorithms support vector machine, decision tree and Random forest are trained on the training data and tested on the test data. Then to improve the accuracy we implemented hybrid algorithm by combining all these algorithm.
- For each of the models, the accuracy, precision, recall, and confusion matrix are printed. The precision, recall, and confusion matrix are used to evaluate the performance of the model and to understand the true positive, false positive, true negative, and false negative rates.
- Final webpage prediction/food suggestion for heart disease positive patients.

### **3.2.1 ADVANTAGE OF PROPSOED SYTEM:**

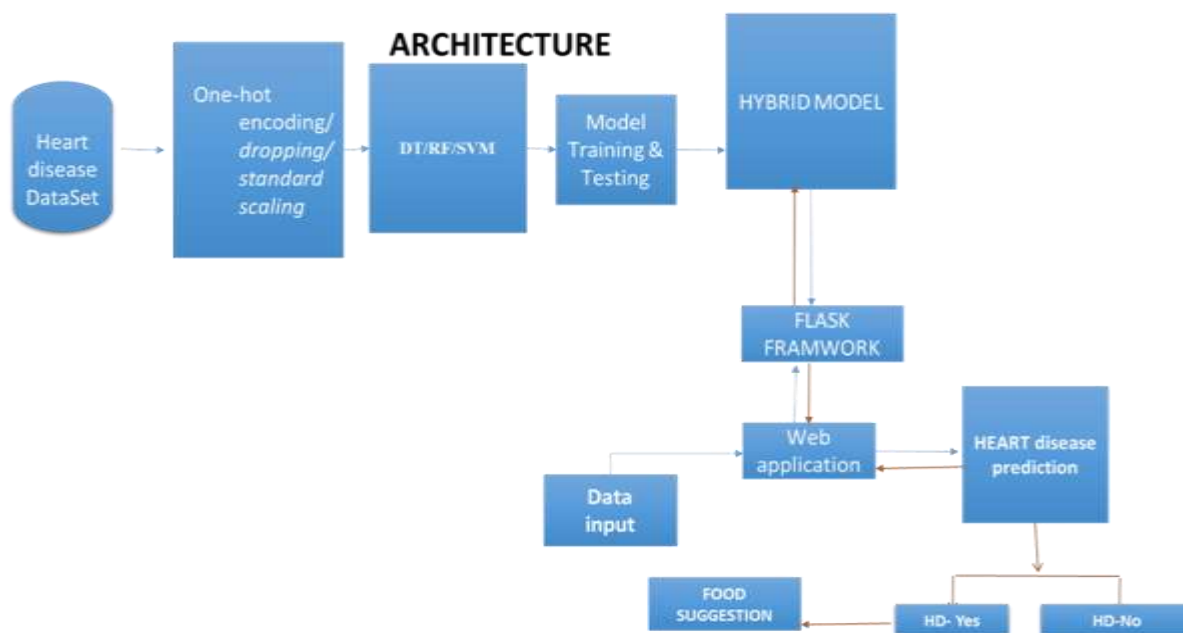
- To get heart disease patient data
- Suggestion food for patient who suffer from heart disease.
- Can be used in diagnostic centres.
- Predict disease in correct time and give knowledge about the disease.

### **3.3 MODULES**

- Data Collection
- Data Preprocessing
- Model Training
- Model Testing
- Flask Framework
- Suggestion & Notification

## 4. SYSTEM DESIGN

### 4.1 SYSTEM ARCHITECTURE:



**Fig.5.1.1 System architecture****4.1.1 DATA COLLECTION**

*Age*: displays the age of the individual.

*Sex*: displays the gender of the individual using the following format :

1 = male

0 = female

*Chest-pain type*: displays the type of chest-pain experienced by the individual using the following format :

1 = typical angina

2 = atypical angina

3 = non — anginal pain

4 = asymptotic

*Resting Blood Pressure*: displays the resting blood pressure value of an individual in mmHg (unit)

*Serum Cholestrol*: displays the serum cholesterol in mg/dl (unit)

*Fasting Blood Sugar*: compares the fasting blood sugar value of an individual with 120mg/dl.

If fasting blood sugar > 120mg/dl then : 1 (true)

else : 0 (false)

*Resting ECG* : displays resting electrocardiographic results

0 = normal

1 = having ST-T wave abnormality

2 = left ventricular hypertrophy

*Max heart rate achieved* : displays the max heart rate achieved by an individual.

*Exercise induced angina :*

1 = yes

0 = no

*ST depression induced by exercise relative to rest:* displays the value which is an integer or float.

*Peak exercise ST segment :*

1 = upsloping

2 = flat

3 = downsloping

*Number of major vessels (0–3) colored by flourosopy :* displays the value as integer or float.

*Thal :* displays the thalassemia :

3 = normal

6 = fixed defect

7 = reversible defect

*Diagnosis of heart disease :* Displays whether the individual is suffering from heart disease or not :

0 = absence

1, 2, 3, 4 = present.

The dataset used in this article is the Cleveland Heart Disease dataset taken from the UCI repository.

#### **4.1.2 DATA PRE-PROCESSING**

- Categorical Variables Into Binary Variables Using One-Hot Encoding.
- StandardScaler()
- Test Train Split
- Data Analysis



- Drop Id

### 4.1.3 MODEL TRAINING

This involves providing the model with input data (also known as features) and the corresponding output data (also known as labels) and allowing the model to learn from the data by adjusting its parameters.

There are several machine learning algorithms to choose. The choice of algorithm depends on the problem being solved and the nature of the data.

### 4.1.4 RANDOM FOREST

Random Forest (RF) is a powerful machine learning algorithm that has been applied to heart disease prediction with great success. In this context, RF is used to model the relationship between a set of predictor variables (such as age, gender, blood pressure, cholesterol levels, and family history of heart disease) and the binary outcome of whether or not a patient has heart disease.

RF is a type of ensemble learning algorithm that builds multiple decision trees and aggregates their results to make predictions. Each tree is trained on a random subset of the data and a random subset of the predictor variables, which helps to reduce overfitting and increase the generalizability of the model.

There are several advantages of using RF in heart disease prediction:

**High accuracy:** RF models have been shown to have high accuracy in predicting heart disease, particularly when compared to other modeling techniques such as logistic regression.

**Can handle complex interactions:** RF can capture complex interactions between predictor variables, which may be difficult to model using traditional statistical techniques.

**Robust to noise and missing data:** RF is robust to noisy and missing data, which is often a problem in medical datasets.

Can handle both continuous and categorical predictor variables: RF can handle both continuous and categorical predictor variables, making it a flexible modeling technique

#### **4.1.5 SVM**

Support Vector Machines (SVM) is a popular machine learning algorithm that can be used for heart disease prediction due to its several advantages.

SVM has been found to have high accuracy in predicting heart disease compared to other machine learning algorithms. This is because SVM is good at handling non-linear relationships in the data, which is often present in medical datasets.

SVM is relatively robust to overfitting, which is a common problem in machine learning algorithms. SVM can handle high dimensional data and can generalize well to new data, making it a suitable algorithm for heart disease prediction.

SVM provides a clear boundary between classes, making it easy to understand how the algorithm is making its predictions.

#### **4.1.6 DECISION TREE**

Decision Tree (DT) is another popular machine learning algorithm that can be used for heart disease prediction.

Scalability: DT can handle large datasets with thousands of observations and hundreds of features, making it suitable for large medical datasets.

Interactivity: DT can be used interactively with the user, where the user can modify the decision rules to explore different scenarios and understand the impact of different features on the prediction.

Ensemble methods: DT can be combined with other algorithms using ensemble methods, such as Random Forests, to improve the accuracy of the prediction

#### **4.1.7 HYBRID ALGORITHM**

HYBRID ALGORITHM takes three models, a Decision Tree (DT) model, a Random Forest (RF) model, and a Support Vector Machine (SVM) model, and a set of input features ( $X$ ) as input. The function predicts the outcome variable using all three models and returns a hybrid prediction that combines the individual predictions.

The hybrid prediction is made by considering the predictions of all three models.

If all three models agree on the prediction, the hybrid prediction is set to the prediction.

If only two of the models agree, the hybrid prediction is set to the prediction made by the two models that agree.

If no two models agree, the function uses the probability estimates from all three models to make the final prediction.

#### 4.1.8 MODEL TESTING

True Positive: You predicted positive, and it's true.

True Negative: You predicted negative, and it's true.

False Positive: (Type 1 Error): You predicted positive, and it's false.

False Negative: (Type 2 Error): You predicted negative, and it's false.

In this module we test the trained machine learning model using the test dataset.

Accuracy

$$\frac{TP + TN}{TP + FP + TN + FN}$$

The most commonly used metric to judge a model and is actually not a clear indicator of the performance. The worse happens when classes are imbalanced.

Precision

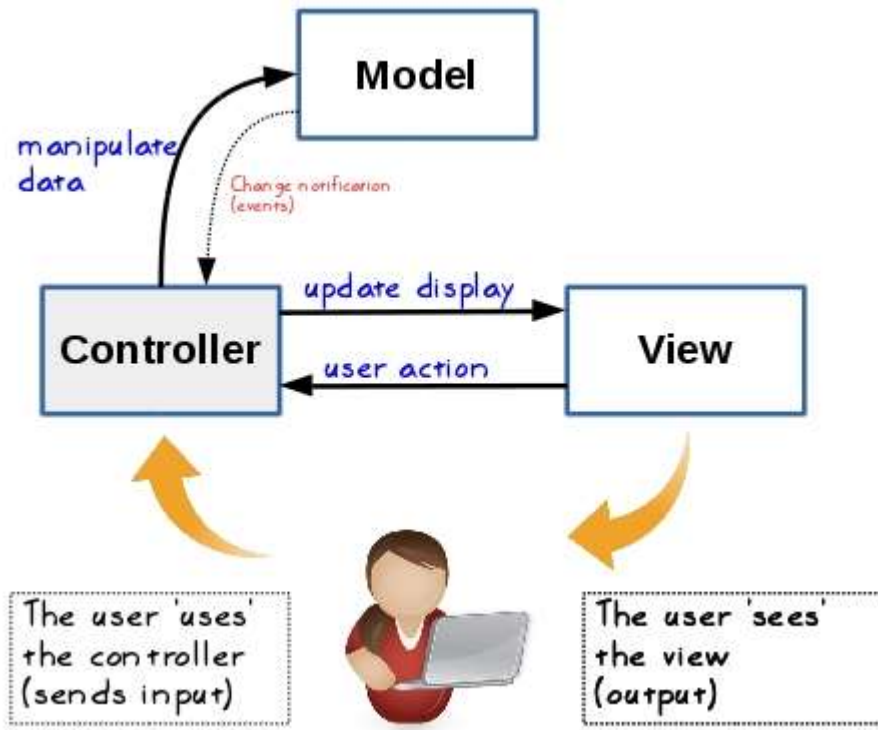
$$\frac{TP}{TP + FP}$$

Percentage of positive instances out of the *total predicted positive* instances.

Confusion Matrix		Target			
		Positive	Negative		
Model	Positive	a	b	Positive Predictive Value	a/(a+b)
	Negative	c	d	Negative Predictive Value	d/(c+d)
		Sensitivity	Specificity	Accuracy = (a+d)/(a+b+c+d)	
		a/(a+c)	d/(b+d)		

### Flask framework Prediction

- The front end of a website is the area with which the user immediately interacts. It contains everything that users see and interact with: text colours and styles, images and videos, graphs and tables, the navigation menu, buttons, and colours.
- HTML, CSS, and JavaScript are used in developing the front end. Flask is used for developing web applications using python.
- We started by importing the Flask class. We then make an instance of this class.
- The ‘\_\_name\_\_’ argument is passed which is the name of the application’s module or package. Flask needs this to know where to look for resources like templates and static files.
- The route() decorator is then used to inform Flask which URL should activate our method. This method returns the message that should be shown in the user’s browser.
- Flask is a lightweight web framework for Python that allows you to build web applications quickly and easily. Flask works by Model-View-Controller (MVC) architecture pattern, where the model represents the data, the view represents the user interface, and the controller acts as an intermediary between the model and the view.



### 4.1.9 Install Flask

Define routes: Routes are the URLs that the user can visit in your web application. You can define routes in Flask by using the `@app.route()` decorator and specifying the URL pattern as a parameter.

## 5.UML DIAGRAMS

### 5.1UML DIAGRAM

## **ARTIFICIAL INTELLIGENCE:**

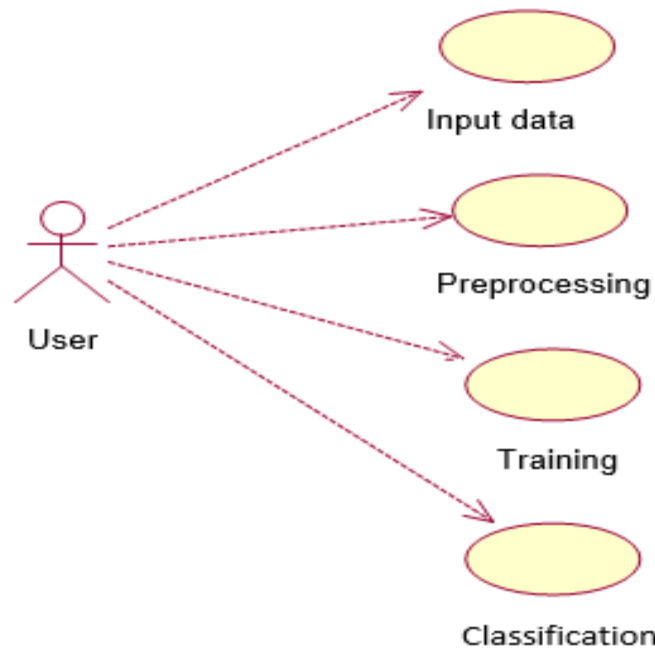
Artificial intelligence (AI) is the simulation of human intelligence processes by machines, especially computer systems enabling it to even mimic human behaviour. Its applications lie in fields of Computer Vision, Natural Language Processing, Robotics, Speech Recognition, etc. Advantages of using AI are improved customer experience, accelerate speed to market, develop sophisticated products, enable cost optimisation, enhance employee productivity and improve operational efficiency. Machine Learning (ML) is a subset of AI which is programmed to think on its own, perform social interaction, learn new information from the provided data and adapt as well as improve with experience. Although training time via Deep Learning (DL) methods is more than Machine Learning methods, it is compensated by higher accuracy in the former case. Also, DL being automatic, large domain knowledge is not required for obtaining desired results unlike in ML.

## **UML DIAGRAMS:**

Unified Modelling Language (UML) is simply another graphical representation of a common semantic model. The proposed system has been designed by using use case diagram, class diagram, sequence diagram, collaboration diagram, state chart diagram and component diagram.

### **5.1.1 USE CASE DIAGRAM**

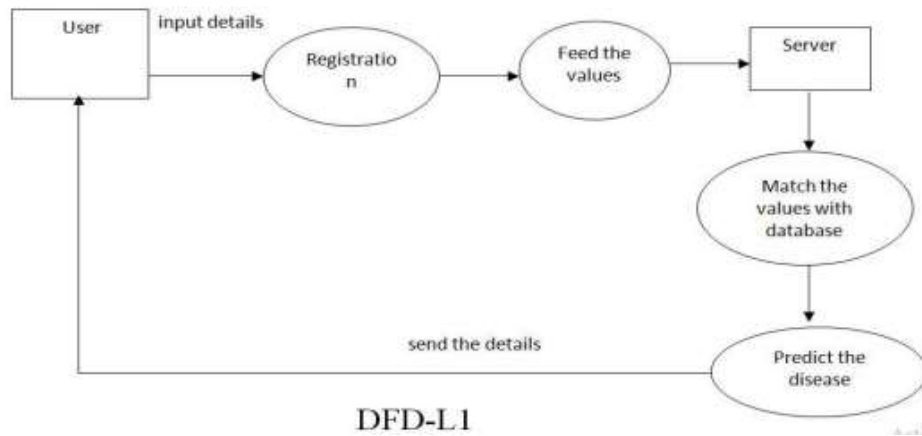
The use case diagram consists of the actors and the use cases. The actors of the system are user, system holder, device controller and the use cases are authentication, checking credentials, basic ON/OFF, allow/deny user, storing NLP commands, Input through voice commands, Deriving Data, Intrusion Detection, Service Maintenance. describes Use Case diagram for Adaptive Automation System (AAS). User and Non-User can login to the device by Admin. Then device controller decides the entry of the user. The user can control the devices through voice commands and these can be monitored by Device Controller. Intrusion detection and maintenance also controlled by the Device Controller (Admin). The user can check for the intruder in the environment and based on the intruder the door can be open/closed. The light can be turned on/off by the system holder and they can alert the emergency system if any poisonous gas or temperature exceeds the normal level. This can be maintained by system maintenance.



### 5.1.2 CLASS DIAGRAM

Class diagram is to model the static view of an application. Class diagrams are the only diagrams which can be directly mapped with object-oriented languages and thus widely used at the time of construction and it is used for general conceptual modelling of the structure of the application, and for detailed modelling translating the models into programming code.

Class diagrams can also be used for data modelling. The classes in a class diagram represent both the main elements, interactions in the application, and the classes to be programmed.



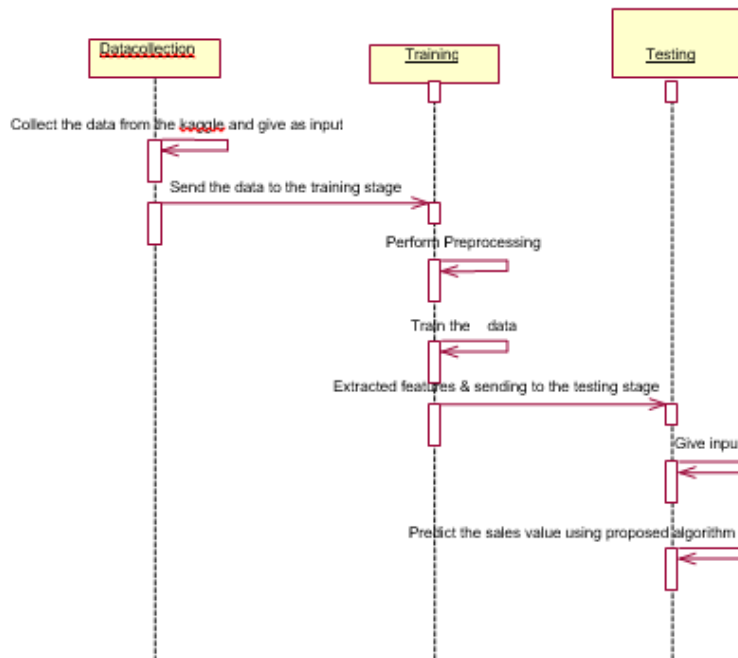
### 5.1.3 Sequence diagram

The control flow between various participants or entity roles of the corresponding system in the form of messages is represented in the Sequence Diagram. The participants are represented within the rectangular object. The swim line or the lifeline that is dragged below every participant represents the lifetime of the corresponding participant.

The UML representation of a class is rectangle containing three

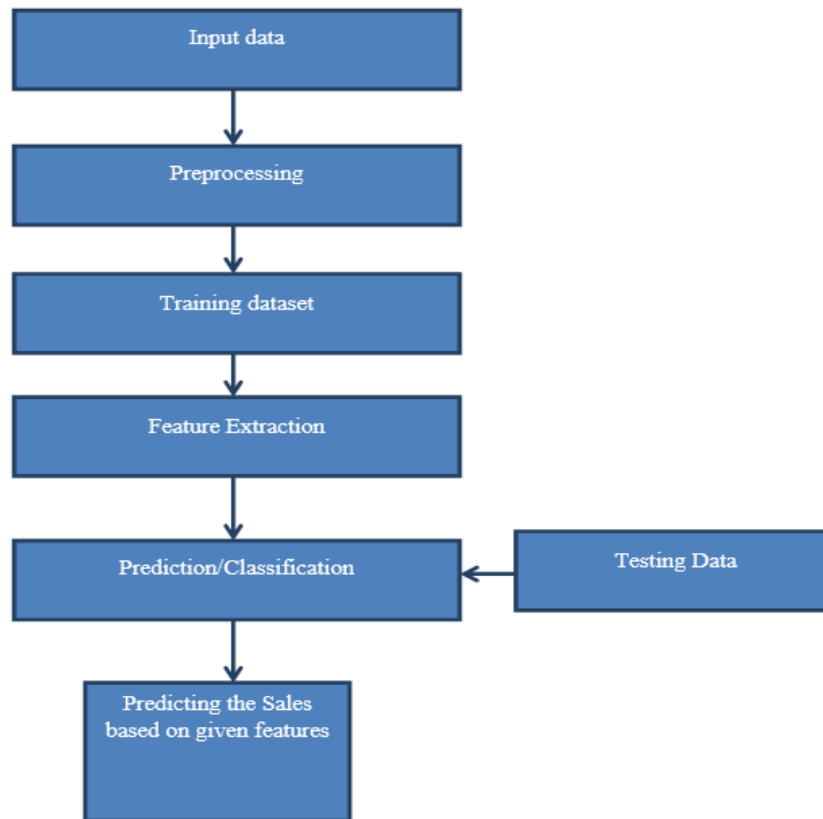
compartments stacked vertically. The top compartments shows the class's name. the middle compartments list the class's attributes. The bottom compartment lists the class operations known as the methods of the class. A class diagram consists of any number of classes which will be connected by the lines, which may have arrows at one or both ends, connecting the boxes. These lines define the relationships, also called associations, between the classes. These lines will have multiplicity to represent the number of instances of the classes.





#### 5.1.4 DATA FLOW DIAGRAM:

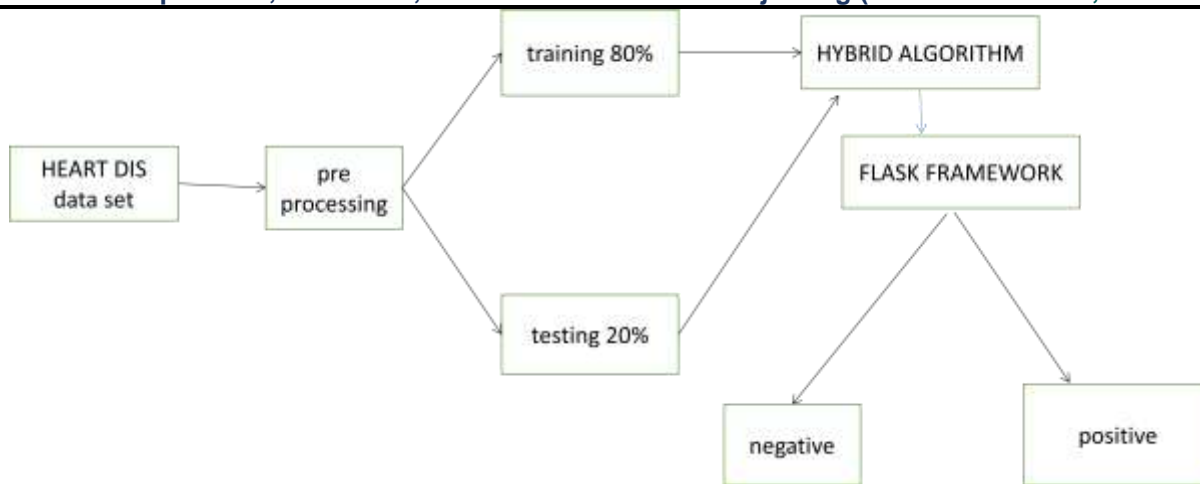
A Data Flow Diagram (DFD) is a traditional way to visualize the information flows within a system. A neat and clear DFD can depict a good amount of the system requirements graphically. It can be manual, automated, or a combination of both. It shows how information enters and leaves the system, what changes the information and where information is stored. The purpose of a DFD is to show the scope and boundaries of a system as a whole. It may be used as a communications tool between a systems analyst and any person who plays a part in the system that acts as the starting point for redesigning a system.



### 5.1.5 ER DIAGRAM:

An Entity–relationship model (ER model) describes the structure of a database with the help of a diagram, which is known as Entity Relationship Diagram (ER Diagram). An ER model is a design or blueprint of a database that can later be implemented as a database. The main components of E-R model are: entity set and relationship set.

An ER diagram shows the relationship among entity sets. An entity set is a group of similar entities and these entities can have attributes. In terms of DBMS, an entity is a table or attribute of a table in database, so by showing relationship among tables and their attributes, ER diagram shows the complete logical structure of a database. Let's have a look at a simple ER diagram to understand this concept.



## 6.SOFTWARE ENVIRONMENT

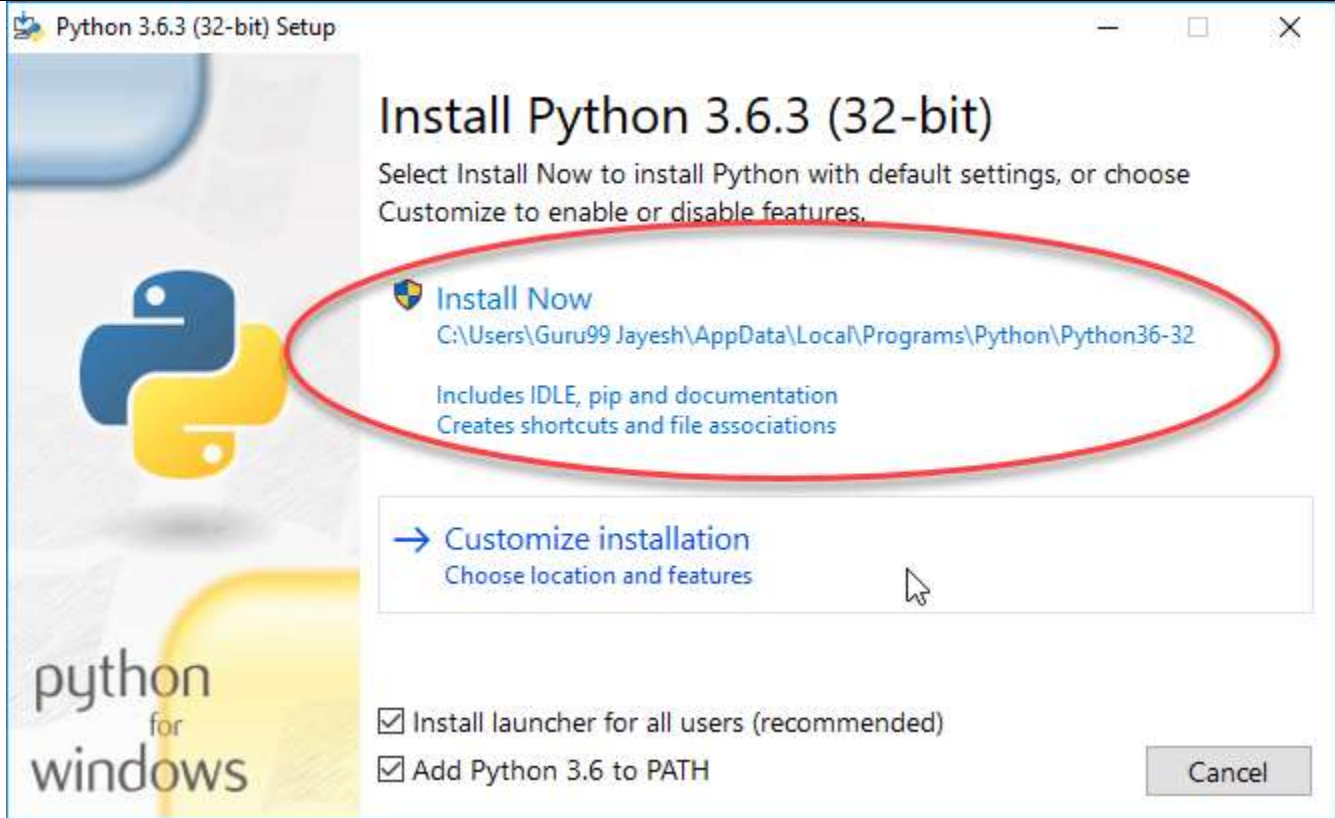
### HOW TO INSTALL PYTHON IDE

Below is a step by step process on how to download and install Python on Windows:

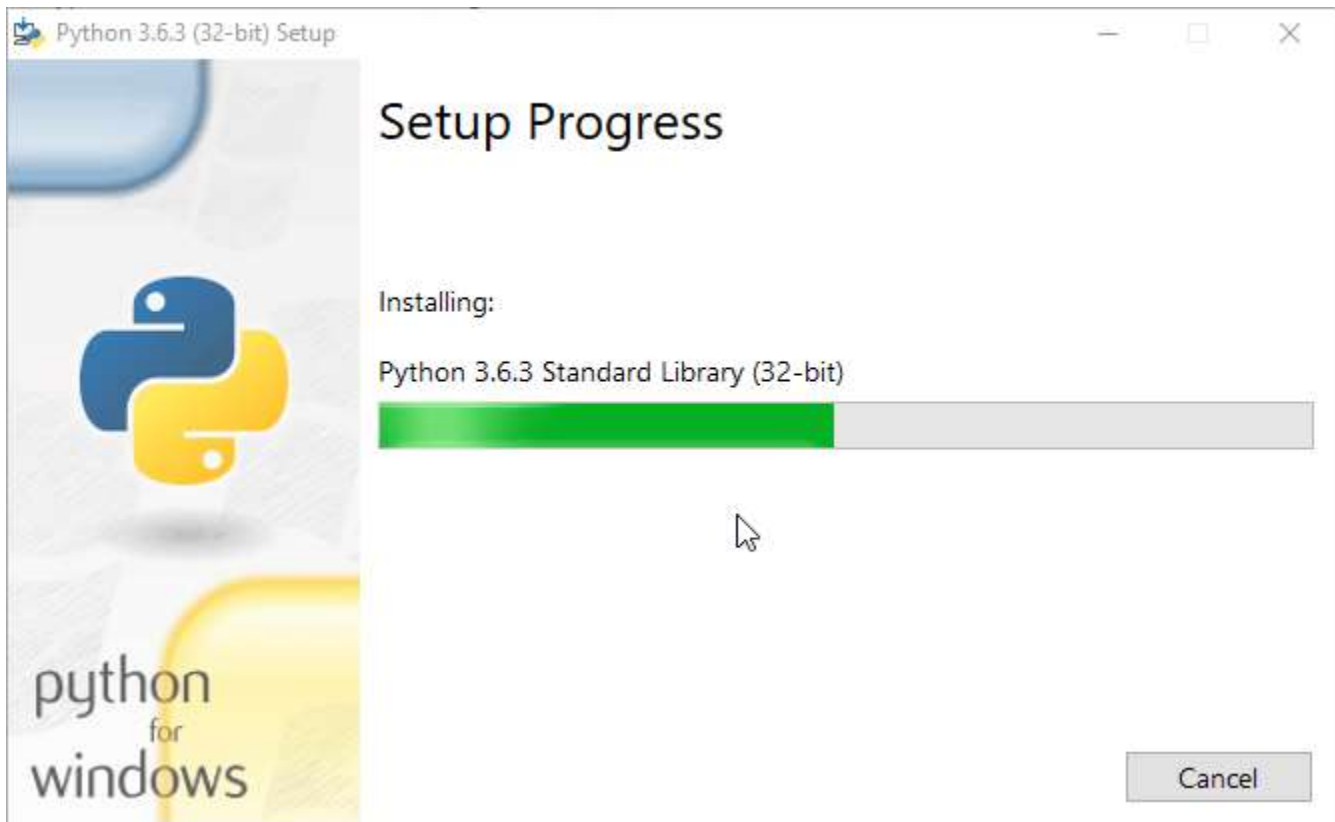
**Step 1)** To download and install Python, visit the official website of Python <https://www.python.org/downloads/> and choose your version. We have chosen Python version 3.6.3



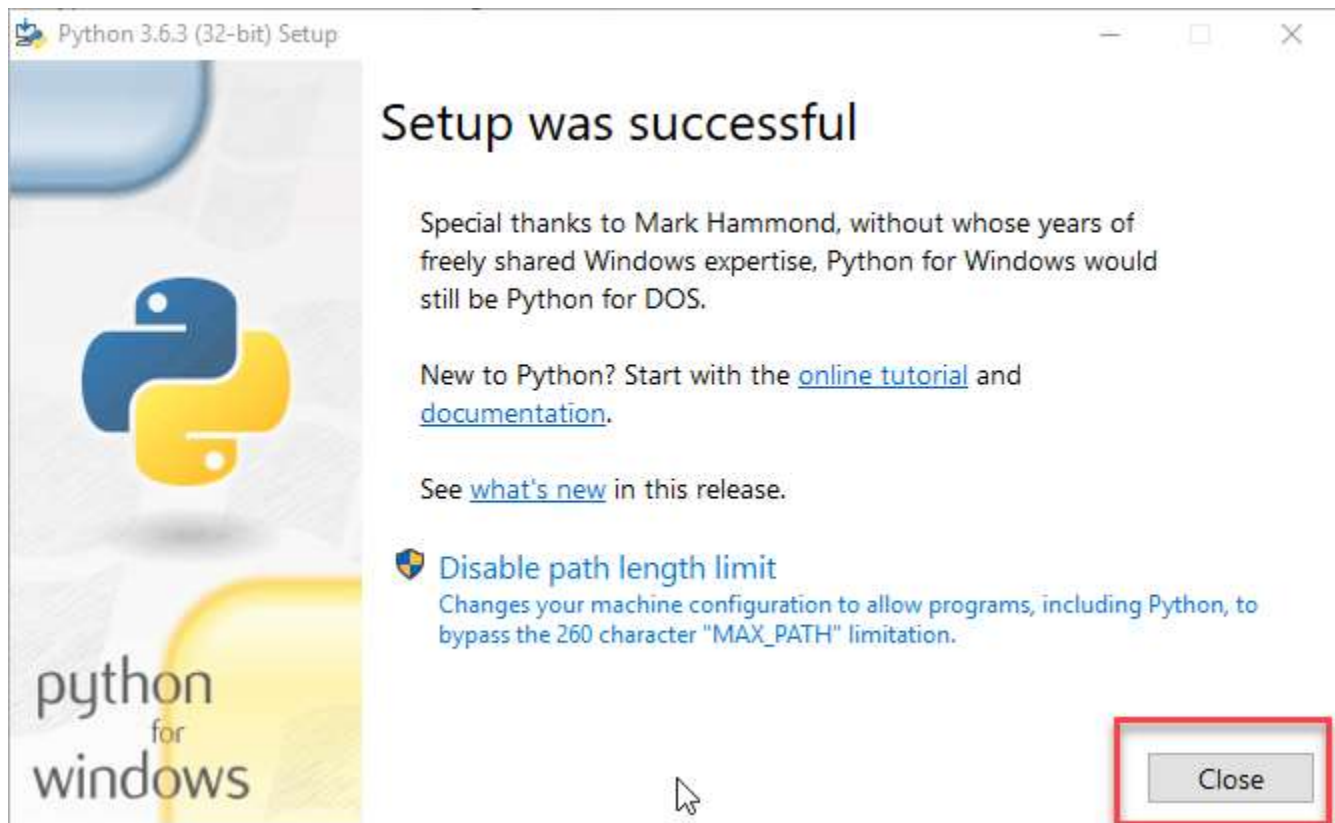
**Step 2)** Once the download is completed, run the .exe file to install Python. Now click on Install Now.



**Step 3)** You can see Python installing at this point.



**Step 4)** When it finishes, you can see a screen that says the Setup was successful. Now click on “Close”.



## HOW TO INSTALL PYCHARM

Here is a step by step process on how to download and install Pycharm IDE on Windows:

**Step 1)** To download PyCharm visit the website <https://www.jetbrains.com/pycharm/download/> and Click the “DOWNLOAD” link under the Community Section.

# Download PyCharm

Windows macOS Linux

## Professional

Full-featured IDE for Python & Web development

DOWNLOAD

Free trial

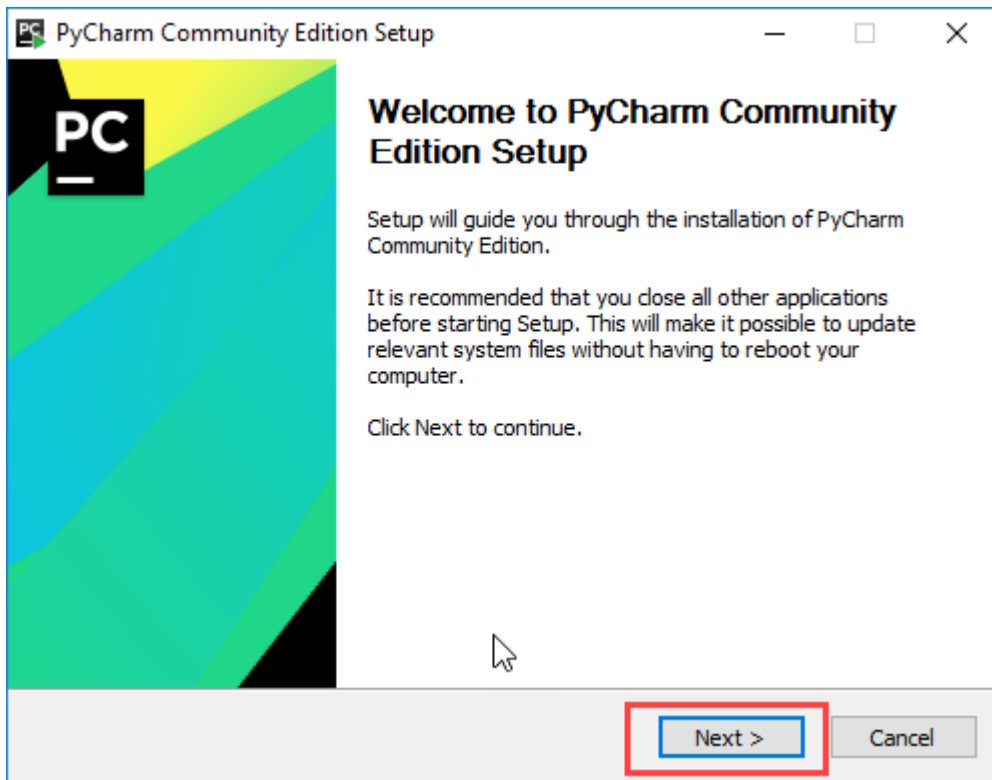
## Community

Lightweight IDE for Python & Scientific development

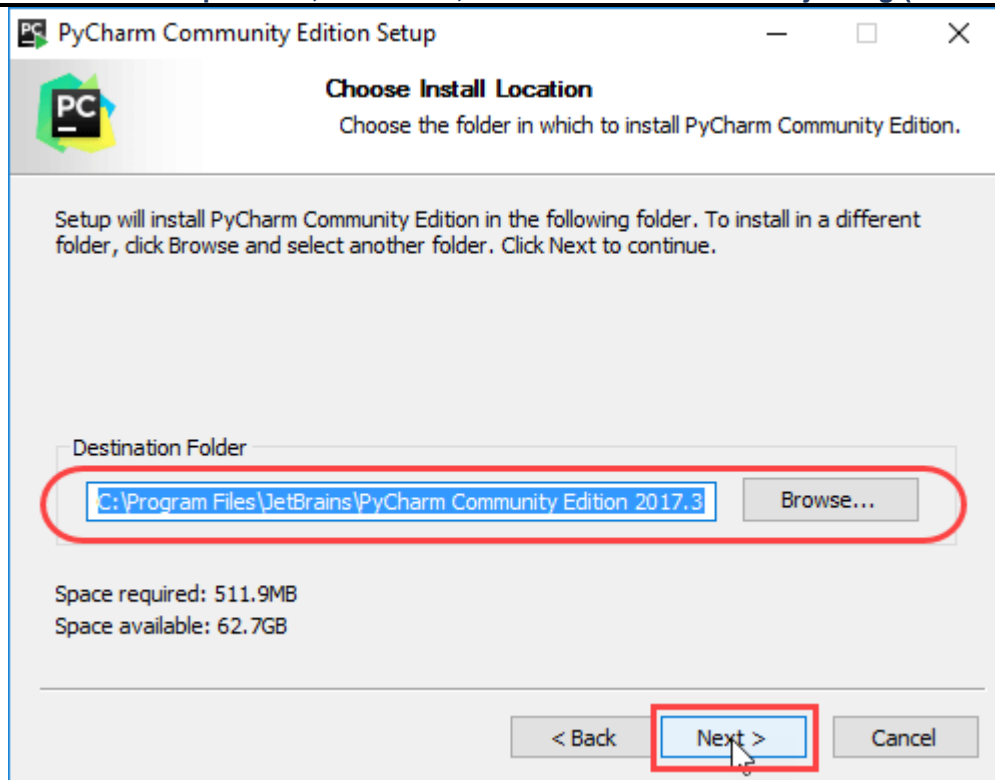
DOWNLOAD

Free, open-source

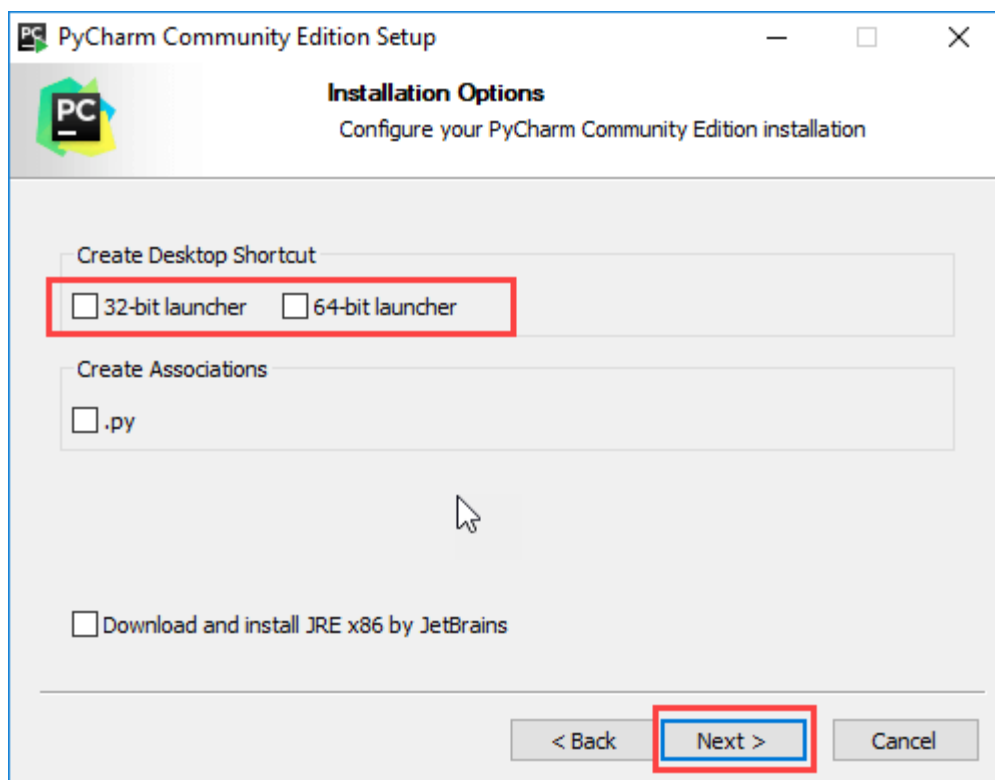
**Step 2)** Once the download is complete, run the exe for install PyCharm. The setup wizard should have started. Click “Next”.



**Step 3)** On the next screen, Change the installation path if required. Click “Next”.

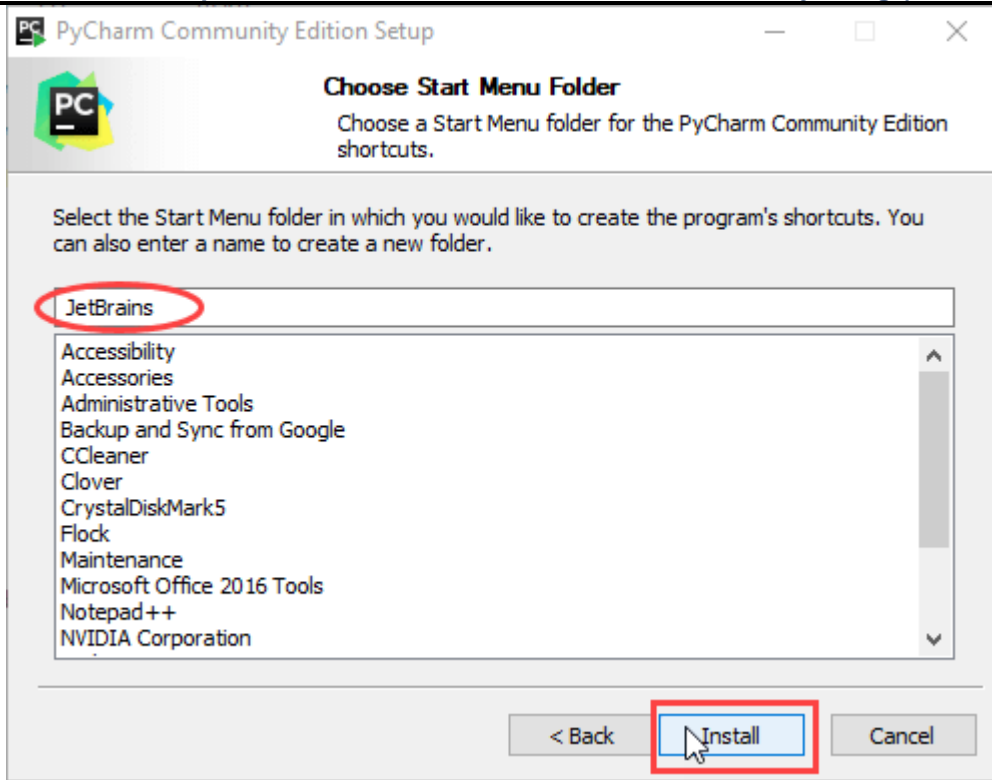


**Step 4)** On the next screen, you can create a desktop shortcut if you want and click on “Next”.

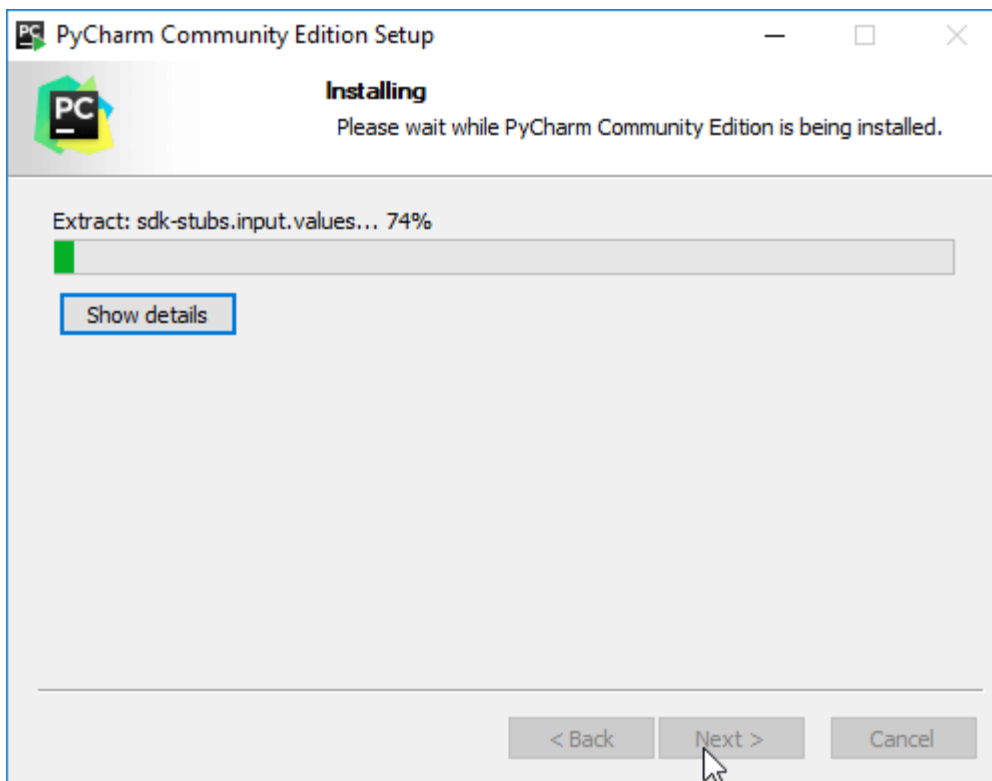


**Step 5)** Choose the start menu folder. Keep selected JetBrains and click on “Install”.



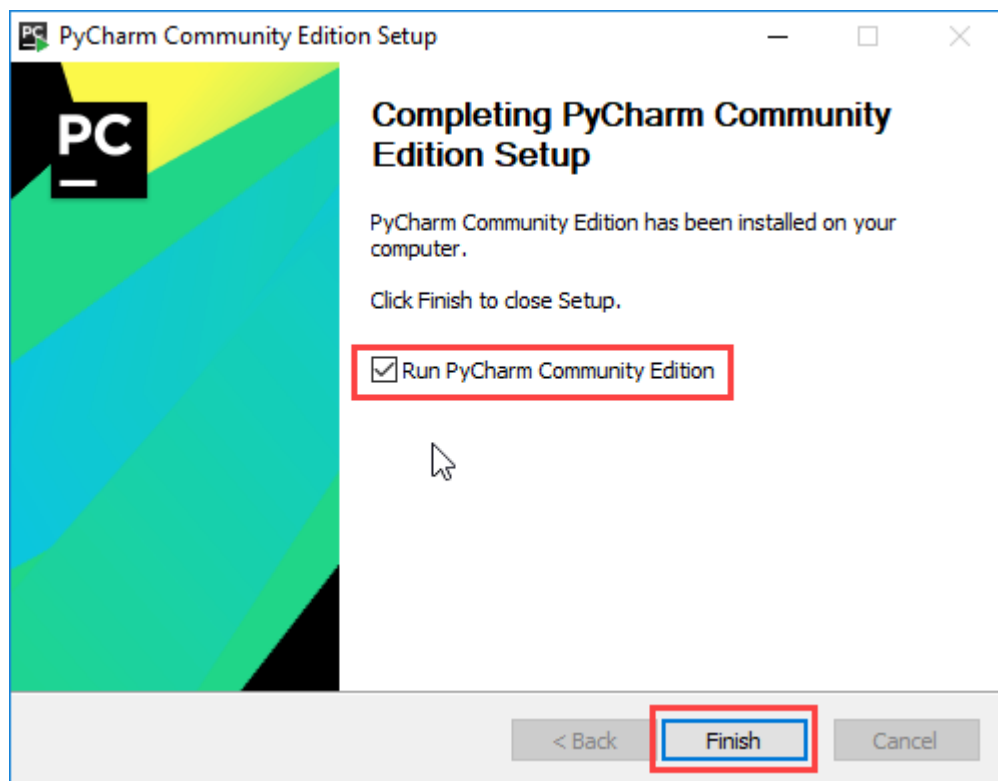


Step 6) Wait for the installation to finish.

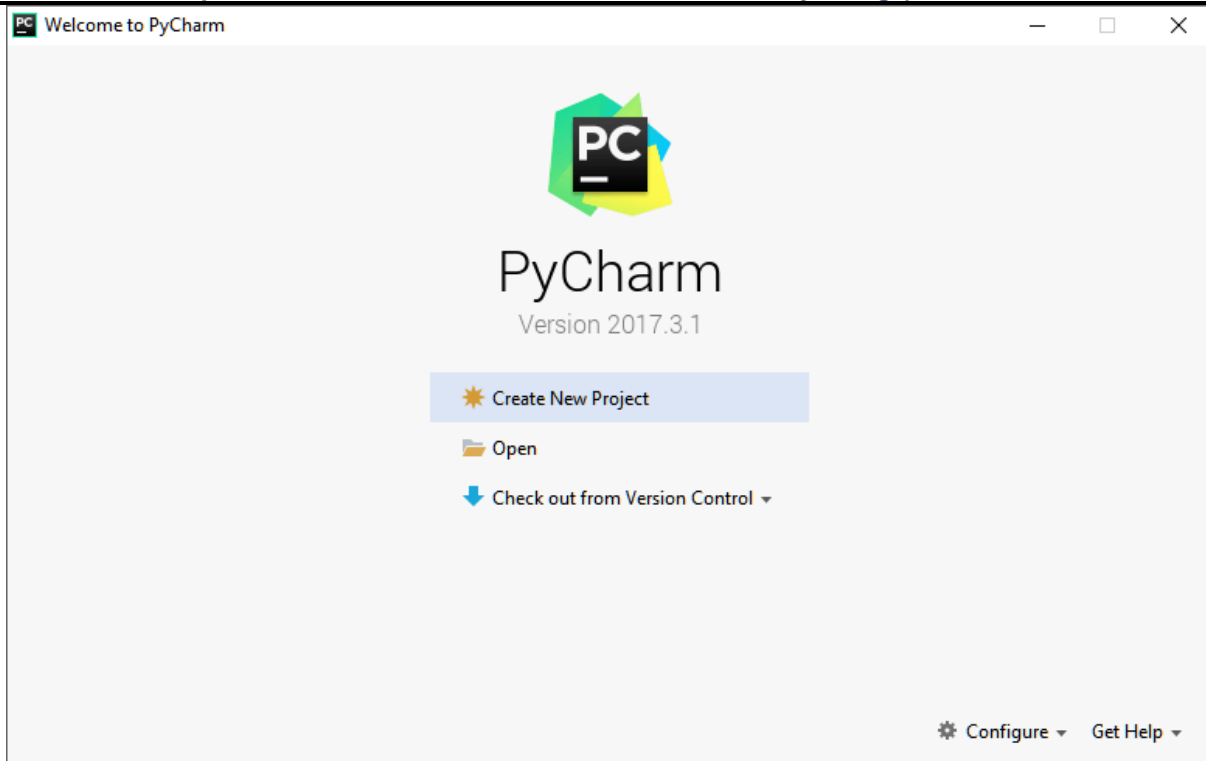


**Step 7)** Once installation finished, you should receive a message screen that PyCharm is installed.

If you want to go ahead and run it, click the “Run PyCharm Community Edition” box first and click “Finish”.

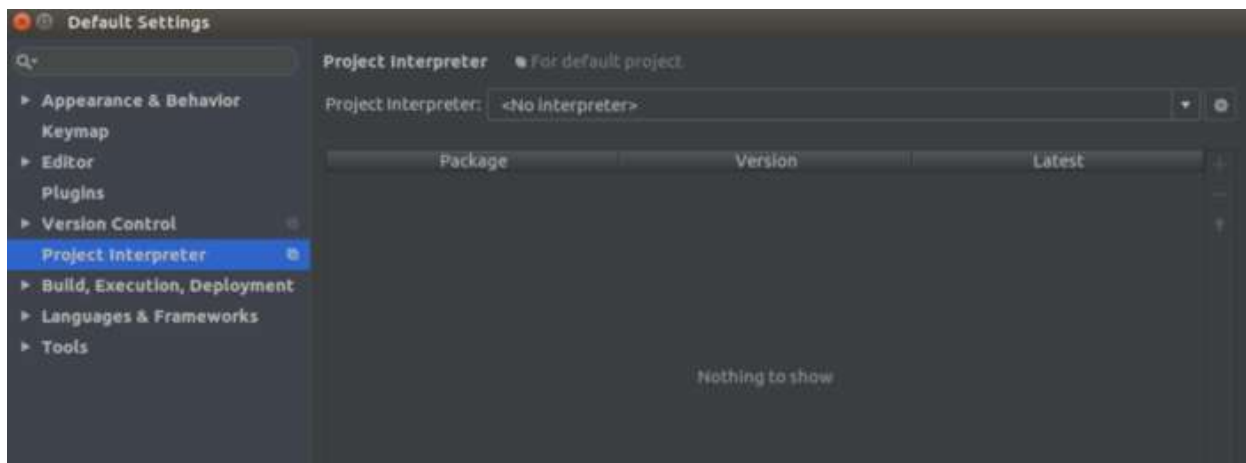


**Step 8)** After you click on “Finish,” the Following screen will appear.

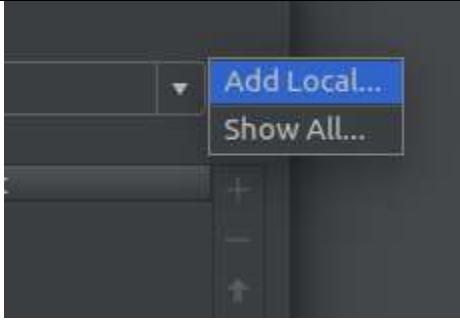


Click on Configure > Settings to open up settings in PyCharm

Search for “Project Interpreter”. My PyCharm looks like this



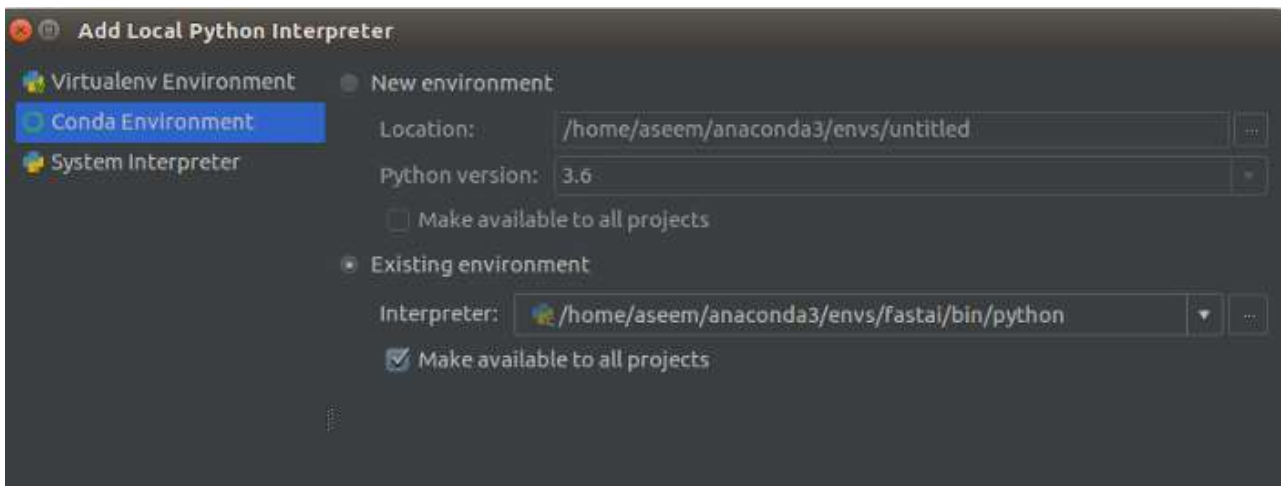
Click on Add local via the settings on the right side



Select “conda environment”

Click on “Existing environment” and navigate to the environment that you want to use. Note that you have to select the bin/python file inside the conda environment for PyCharm to be able to recognise the environment

Make sure to click the “Make available to all projects” if you want the interpreter to be used by multiple projects



Click ok and you are done

# 7. IMPLEMENTATION

## 7.1 CODING

Once the design aspect of the system is finalized the system enters into the coding and testing phase. The coding phase brings the actual system into action by converting the design of the system into the code in a given programming language. Therefore, a good coding style has to be taken whenever changes are required it easily screwed into the system.

## 7.2 CODING STANDARDS

Coding standards are guidelines to programming that focuses on the physical structure and appearance of the program. They make the code easier to read, understand and maintain. This phase of the system actually implements the blueprint developed during the design phase. The coding specification should be in such a way that any programmer must be able to understand the code and can bring about changes whenever felt necessary. Some of the standard needed to achieve the above-mentioned objectives are as follows:

Program should be simple, clear and easy to understand.

Naming conventions

Value conventions

Script and comment procedure

Message box format

### 7.2.1 NAMING CONVENTIONS

Naming conventions of classes, data member, member functions, procedures etc., should be **self-descriptive**. One should even get the meaning and scope of the variable by its name. The conventions are adopted for **easy understanding** of the intended message by the user. So it is customary to follow the conventions. These conventions are as follows:

#### Class names

Class names are problem domain equivalence and begin with capital letter and have mixed cases.

#### Member Function and Data Member name

Member function and data member name begins with a lowercase letter with each subsequent letters of the new words in uppercase and the rest of letters in lowercase.

### 7.2.2 VALUE CONVENTIONS

Value conventions ensure values for variable at any point of time. This involves the following:

- Proper default values for the variables.
- Proper validation of values in the field.
- Proper documentation of flag values.

### 7.2.3 SCRIPT WRITING AND COMMENTING STANDARD

Script writing is an art in which indentation is utmost important. Conditional and looping statements are to be properly aligned to facilitate easy understanding. Comments are included to minimize the number of surprises that could occur when going through the code.

### 7.2.4 MESSAGE BOX FORMAT

When something has to be prompted to the user, he must be able to understand it properly. To achieve this, a specific format has been adopted in displaying messages to the user. They are as follows:

- X – User has performed illegal operation.
- ! – Information to the user.

### 7.3 CODE:

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="stylesheet" href="/static/style/heart_disease.css">
  <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css">
```

```
<title>Model: Heart Disease Diagnosis</title>
</head>

<body>
  <div class="topnav">
    <a class="active" href="/"> <span> Home </span></a>
    <div class="heading">Heart Disease Diagnosis</div>
  </div>
  <!-- card structure -->
  <div class="container">
    <!-- Result form -->

    <form action="javascript:void(0);" method="post" id="form-id">
      <!-- Tell me your age -->

      <div class="main-frame" style="display:block" id="div1">
        <div class="sub-frame">
          <div class="question-age">
            <div class="question">Can I Know your Age?</div>
            <div class="option">
              <div class="range-wrap">
                <input type="range" name="age" class="range" min="1" max="80">
                <output class="bubble"></output>
              </div>
            </div>
          </div>
          <div class="button" onclick="replace('div1','div2')"> <span> Next </span> </button>
        </div>
      </div>
    </form>
  </div>
</div>
```



```
<div class="main-frame" style="display:none" id="result-box">
```

```
<div class="sub-frame">
```

```
<div class="result">
```

```
<div class="question">How are You</div>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
<!-- Can I know your gender -->
```

```
<div class="main-frame" id="div2" style="display:none">
```

```
<div class="sub-frame">
```

```
<div class="question-gender">
```

```
<div class="question">Can I know your Sex?</div>
```

```
<div class="option">
```

```
<label for="male">
```

```
<input type="radio" name="gender" id="male" value="1" checked> <i
```

```
class="fa fa-male"></i>
```

```
</label>
```

```
<label for="female">
```

```
<input type="radio" name="gender" id="female" value="0"><i class="fa fa-
```

```
female"></i>
```

```
</label>
```

```
</div>
```

```
<button class="button" onclick="replace('div2','div3')"> <span> Next </span> </button>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
<!-- What type of chest pain you have? -->
```

```
<div class="main-frame" id="div3" style="display:none">
```

```
<div class="sub-frame">
```

```
<div class="question-chest-pain-type">
```

```
<div class="question">What type of chest pain you have?</div>
```

```
<div class="option">
```

```
<div class="col">
```

```
<div class="row">
```

```
<label for="typical-agina">
```

```
<input type="radio" name="chest-pain-type" id="typical-agina" value="1" checked>
```

```
<p> Typical Agina </p>
```

```
</label>
```

```
</div>
```

```
<div class="row">
```

```
<label for="atypical-agina">
```

```
<input type="radio" name="chest-pain-type" id="atypical-agina" value="2">
```

```
<p>Atypical Agina</p>
```

```
</label>
```

```
</div>
```

```
</div>
```

```
<div class="col">
```

```
<div class="row">
```

```
<label for="non-aginal-pain">
```

```
<input type="radio" name="chest-pain-type" id="non-aginal-pain" value="3">
```

```
<p>Non-aginal pain</p>
```

```
</label>
```

```
</div>
```

```
<div class="row">
```

```
<label for="Asymptomatic">
```

```
<input type="radio" name="chest-pain-type" id="Asymptomatic" value="4">
```

```
<p> Asymptomatic</p>
```

```
</label>
```

```
</div>
```

```
</div>

</div>

<button class="button" onclick="replace('div3','div4')"> <span> Next </span> </button>

</div>

</div>

</div>

<!-- Is your fasting blood sugar > 120 mg/dl -->

<div class="main-frame" id="div4" style="display:none">

  <div class="sub-frame">

    <div class="question-fasting-blood-sugar">

      <div class="question">Is your fasting blood sugar > 120 mg/dl?</div>

      <div class="option">

        <div class="col">

          <div class="row">

            <label for="fbs-yes">

              <input type="radio" name="fasting-blood-sugar" id="fbs-yes" value="1" checked>

              <p> Yes </p>

            </label>

          </div>

          <div class="row">

            <label for="fbs-no">

              <input type="radio" name="fasting-blood-sugar" id="fbs-no" value="0">

              <p>No</p>

            </label>

          </div>

        </div>

      </div>

    </div>

    <div class="button" onclick="replace('div4','div5')"> <span> Next </span> </button>

  </div>

</div>
```

```
</div>

</div>

<!-- Do your exercise induced angina? -->

<div class="main-frame" id="div5" style="display:none">
  <div class="sub-frame">
    <div class="question-induced-agina">
      <div class="question">Do your exercise induced angina?</div>
      <div class="option">
        <div class="col">
          <div class="row">
            <label for="ag-yes">
              <input type="radio" name="induced-agina" id="ag-yes" value="1" checked>
              <p> Yes </p>
            </label>
          </div>
          <div class="row">
            <label for="ag-no">
              <input type="radio" name="induced-agina" id="ag-no" value="0">
              <p>No</p>
            </label>
          </div>
        </div>
      </div>
      <div class="button" onclick="replace('div5','div6')"> <span> Next </span> </button>
    </div>
  </div>
</div>

<!-- what is your resting systolic blood pressure? -->
```

```
<div class="main-frame" style="display:none" id="div6">
```

```
<div class="sub-frame">
```

```
<div class="question-resting-systolic">
```

```
<div class="question">What is your resting systolic blood pressure?</div>
```

```
<div class="option">
```

```
<div class="range-wrap">
```

```
<input type="range" class="range" name="resting-blood-pressure" min="50"
max="170">
```

```
<output class="bubble"></output>
```

```
</div>
```

```
</div>
```

```
<button class="button" onclick="replace('div6','div7')"> <span> Next </span> </button>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
<!-- what is the slope of your peak exercise ST segment? -->
```

```
<div class="main-frame" style="display:none" id="div7">
```

```
<div class="sub-frame">
```

```
<div class="question-st-segment">
```

```
<div class="question">What is the slope of your peak exercise ST segment?</div>
```

```
<div class="option">
```

```
<div class="col">
```

```
<div class="row">
```

```
<label for="upsloping">
```

```
<input type="radio" name="peak-exercise-st" id="upsloping" value="1" checked>
```

```
<p> Up Sloping </p>
```

```
</label>
```

```
</div>
```

```
<div class="row">
```

```

        <label for="flat">
            <input type="radio" name="peak-exercise-st" id="flat" value="2">
            <p>Flat</p>
        </label>
    </div>
</div>
<div class="col">
    <div class="row">
        <label for="downsloping">
            <input type="radio" name="peak-exercise-st" id="downsloping" value="3">
            <p>Down Sloping </p>
        </label>
    </div>
</div>
</div>
<button class="button" onclick="replace('div7','div8')"> <span> Next </span> </button>
</div>
</div>
</div>

<!-- What is your ST depression value induced by exercise -->
<div class="main-frame" style="display:none" id="div8">
    <div class="sub-frame">
        <div class="question-st-depression">
            <div class="question">What is your ST depression value, induced by exercise, relative to
rest?
            </div>
            <div class="option">
                <div class="range-wrap">
                    <input type="range" class="range" name="st-depressed-value" min="-10" max="10"
                    step="0.01">

```

```
<output class="bubble"></output>

</div>

</div>

<button class="button" onclick="replace('div8','div9')"> <span> Next </span> </button>

</div>

</div>

</div>

<!-- Tell me the maximum heart-rate achieved by you? -->

<div class="main-frame" style="display:none" id="div9">

  <div class="sub-frame">

    <div class="question-heart-rate">

      <div class="question">Tell me the maximum heart-rate achieved by you?</div>

      <div class="option">

        <div class="range-wrap">

          <input type="range" class="range" name="heart-rate-value" min="50" max="250">

          <output class="bubble"></output>

        </div>

      </div>

    </div>

    <button class="button" onclick="replace('div9','div10')"> <span> Next </span> </button>

  </div>

</div>

</div>

<!-- Enter your serum cholestrol in mg/dl -->

<div class="main-frame" style="display:none" id="div10">

  <div class="sub-frame">

    <div class="question-serum-cholestrol">

      <div class="question">Enter your serum cholestrol in mg/dl?</div>

      <div class="option">

        <div class="range-wrap">
```

```
<input type="range" class="range" name="serum-cholesterol-value" min="0"
max="800">

<output class="bubble"></output>

</div>

</div>

<button class="button" onclick="replace('div10','div11')"> <span> Next </span> </button>

</div>

</div>

</div>

<!-- Which of this is your resting ElectroCardioGraph (ECG) -->

<div class="main-frame" id="div11" style="display:none">

<div class="sub-frame">

<div class="question-resting-ecg">

<div class="question"> Which of this is your resting ElectroCardioGraph (ECG)?</div>

<div class="option">

<div class="col">

<label for="ecg-normal">

<input type="radio" name="resting-ecg" id="ecg-normal" value="0" checked>

<p> Normal </p>

</label>

<label for="ecg-abnormal">

<input type="radio" name="resting-ecg" id="ecg-abnormal" value="1">

<p>Having ST - T wave abnormality</p>

</label>

<label for="ecg-hyper">

<input type="radio" name="resting-ecg" id="ecg-hyper" value="2">
```



```
<p>Left ventricular hypertrophy</p>
</label>
</div>
</div>
<button class="button" onclick="submitForm('div11');" <span> Submit </span> </button>
</div>
</div>
</div>
</form>

</div>
<script src="/static/script/heart_disease.js"></script>
</body>

</html>

import pandas as pd
import numpy as np
from matplotlib import pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.svm import SVC

import seaborn as sn

# Load the data
data = pd.read_csv("heart.csv")

# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(data.drop("target", axis=1), data["target"], test_size=0.2,
random_state=42)

# Train three different models on the training set
```

```
dt_model = DecisionTreeClassifier(random_state=42)
```

```
dt_model.fit(X_train, y_train)
```

```
rf_model = RandomForestClassifier(random_state=42)
```

```
rf_model.fit(X_train, y_train)
```

```
svm_model = SVC(probability=True, random_state=42)
```

```
svm_model.fit(X_train, y_train)
```

```
# Define a function to make predictions using the hybrid model
```

```
def make_hybrid_prediction(dt_model, rf_model, svm_model, X):
```

```
    dt_pred = dt_model.predict(X)
```

```
    rf_pred = rf_model.predict(X)
```

```
    svm_pred = svm_model.predict(X)
```

```
    hybrid_pred = np.zeros_like(dt_pred)
```

```
    for i in range(len(dt_pred)):
```

```
        if dt_pred[i] == rf_pred[i] == svm_pred[i]:
```

```
            hybrid_pred[i] = dt_pred[i]
```

```
        else:
```

```
            if dt_pred[i] == rf_pred[i]:
```

```
                hybrid_pred[i] = dt_pred[i]
```

```
            elif dt_pred[i] == svm_pred[i]:
```

```
                hybrid_pred[i] = dt_pred[i]
```

```
            elif rf_pred[i] == svm_pred[i]:
```

```
                hybrid_pred[i] = rf_pred[i]
```

```
        else:
```

```
            prob = np.array([dt_model.predict_proba([X.iloc[i]]),
```

```
                            rf_model.predict_proba([X.iloc[i]]),
```

```
                            svm_model.predict_proba([X.iloc[i]])])
```

```
            hybrid_pred[i] = np.argmax(np.max(prob, axis=0))
```

```
return hybrid_pred

# Test the hybrid model on the testing set
X_test_pred = make_hybrid_prediction(dt_model, rf_model, svm_model, X_test)

# Evaluate the performance of the hybrid model
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score, confusion_matrix

print("Accuracy: ", accuracy_score(y_test, X_test_pred))
print("Precision: ", precision_score(y_test, X_test_pred))
print("Recall: ", recall_score(y_test, X_test_pred))
print("F1 score: ", f1_score(y_test, X_test_pred))
cm = confusion_matrix(y_test, X_test_pred)
print("Confusion Matrix :\n", confusion_matrix(y_test, X_test_pred))
# plot_confusion_matrix of HM
array = [[58,6],
         [0,58]]
df_confusion_matrix = pd.DataFrame(array, index = [i for i in "01"],
                                   columns = [i for i in "01"])
plt.figure(figsize = (2,2))
sn.heatmap(df_confusion_matrix, annot=True)
plt.title('Confusion matrix of HYBRID Model', y=1.1)
plt.ylabel('Actual label')
plt.xlabel('Predicted label')
plt.show()
# Make predictions on new, unknown data
unknown_data = pd.DataFrame({
    'age': [57],
    'sex': [1],
    'cp': [0],
```

```
'trestbps': [110],  
'chol': [201],  
'fbs': [0],  
'restecg': [1],  
'thalach': [126],  
'exang': [1],  
'oldpeak': [1.5],  
'slope': [1],  
'ca': [0],  
'thal': [1]  
})
```

```
unknown_data_pred = make_hybrid_prediction(dt_model, rf_model, svm_model, unknown_data)
```

```
# Print the predicted target value for the unknown data
```

```
print(unknown_data_pred)
```

## 8.SYSTEM TESTING

Testing is performed to identify errors. It is used for quality assurance. Testing is an integral part of the entire development and maintenance process. The goal of the testing during phase is to verify that the specification has been accurately and completely incorporated into the design, as well as to ensure the correctness of the design itself. For example the design must not have any logic faults in the design is detected before coding commences, otherwise the cost of fixing the faults will be considerably higher as reflected. Detection of design faults can be achieved by means of inspection as well as walkthrough.

Testing is one of the important steps in the software development phase. Testing checks for the errors, as a whole of the project testing involves the following test cases:

- Static analysis is used to investigate the structural properties of the Source code.
- Dynamic testing is used to investigate the behavior of the source code by executing the program on the test data.

### 8.1 TEST DATA AND OUTPUT

#### 8.1.1 UNIT TESTING

Unit testing is conducted to verify the functional performance of each modular component of the software. Unit testing focuses on the smallest unit of the software design (i.e.), the module. The white-box testing techniques were heavily employed for unit testing.

#### 8.1.2 FUNCTIONAL TESTS

Functional test cases involved exercising the code with nominal input values for which the expected results are known, as well as boundary values and special values, such as logically related inputs, files of identical elements, and empty files.

Three types of tests in Functional test:

- Performance Test
- Stress Test
- Structure Test

### 8.1.3 INTEGRATION TESTING

Integration testing is a systematic technique for construction the program structure while at the same time conducting tests to uncover errors associated with interfacing. i.e., integration testing is the complete testing of the set of modules which makes up the product. The objective is to take untested modules and build a program structure tester should identify critical modules. Critical modules should be tested as early as possible. One approach is to wait until all the units have passed testing, and then combine them and then tested. This approach is evolved from unstructured testing of small programs. Another strategy is to construct the product in increments of tested units. A small set of modules are integrated together and tested, to which another module is added and tested in combination. And so on. The advantages of this approach are that, interface dispenses can be easily found and corrected.

The major error that was faced during the project is linking error. When all the modules are combined the link is not set properly with all support files. Then we checked out for interconnection and the links. Errors are localized to the new module and its intercommunications. The product development can be staged, and modules integrated in as they complete unit testing. Testing is completed when the last module is integrated and tested.

## 8.2 TESTING TECHNIQUES / TESTING STRATERGIES

### 8.2.1 TESTING

Testing is a process of executing a program with the intent of finding an error. A good test case is one that has a high probability of finding an as-yet –undiscovered error. A successful test is one that uncovers an as-yet- undiscovered error. System testing is the stage of implementation, which is aimed at ensuring that the system works accurately and efficiently as expected before live operation commences. It verifies that the whole set of programs hang together. System testing requires a test consists of several key

activities and steps for run program, string, system and is important in adopting a successful new system.

This is the last chance to detect and correct errors before the system is installed for user acceptance testing.

The software testing process commences once the program is created and the documentation and related data structures are designed. Software testing is essential for correcting errors. Otherwise the program or the project is not said to be complete. Software testing is the critical element of software quality assurance and represents the ultimate the review of specification design and coding. Testing is the process of executing the program with the intent of finding the error. A good test case design is one that as a probability of finding an yet undiscovered error. A successful test is one that uncovers an yet undiscovered error. Any engineering product can be tested in one of the two ways:

### 8.2.1.1 WHITE BOX TESTING

This testing is also called as Glass box testing. In this testing, by knowing the specific functions that a product has been design to perform test can be conducted that demonstrate each function is fully operational at the same time searching for errors in each function. It is a test case design method that uses the control structure of the procedural design to derive test cases. Basis path testing is a white box testing.

Basis path testing:

- Flow graph notation
- Cyclometric complexity
- Deriving test cases
- Graph matrices Control

### 8.2.1.2 BLACK BOX TESTING

In this testing by knowing the internal operation of a product, test can be conducted to ensure that “all gears mesh”, that is the internal operation performs according to specification and all internal components have been adequately exercised. It fundamentally focuses on the functional requirements of the software.

The steps involved in black box test case design are:

- Graph based testing methods
- Equivalence partitioning
- Boundary value analysis
- Comparison testing

#### **8.2.2.2 PROGRAM TESTING:**

The logical and syntax errors have been pointed out by program testing. A syntax error is an error in a program statement that in violates one or more rules of the language in which it is written. An improperly defined field dimension or omitted keywords are common syntax error. These errors are shown through error messages generated by the computer. A logic error on the other hand deals with the incorrect data fields, out-off-range items and invalid combinations. Since the compiler s will not deduct logical error, the programmer must examine the output. Condition testing exercises the logical conditions contained in a module. The possible types of elements in a condition include a Boolean operator, Boolean variable, a pair of Boolean parentheses A relational operator or on arithmetic expression. Condition testing method focuses on testing each condition in the program the purpose of condition test is to deduct not only errors in the condition of a program but also other a errors in the program.

#### **8.2.2.3 SECURITY TESTING:**

Security testing attempts to verify the protection mechanisms built in to a system well, in fact, protect it from improper penetration. The system security must be tested for invulnerability from frontal attack must also be tested for invulnerability from rear attack. During security, the tester places the role of individual who desires to penetrate system.

#### **8.2.2.4 VALIDATION TESTING**

At the culmination of integration testing, software is completely assembled as a package. Interfacing errors have been uncovered and corrected and a final series of software



test-validation testing begins. Validation testing can be defined in many ways, but a simple definition is that validation succeeds when the software functions in manner that is reasonably expected by the customer. Software validation is achieved through a series of black box tests that demonstrate conformity with requirement. After validation test has been conducted, one of two conditions exists.

\* The function or performance characteristics confirm to specifications and are accepted.

\* A validation from specification is uncovered and a deficiency created.

Deviation or errors discovered at this step in this project is corrected prior to completion of the project with the help of the user by negotiating to establish a method for resolving deficiencies. Thus the proposed system under consideration has been tested by using validation testing and found to be working satisfactorily. Though there were deficiencies in the system they were not catastrophic

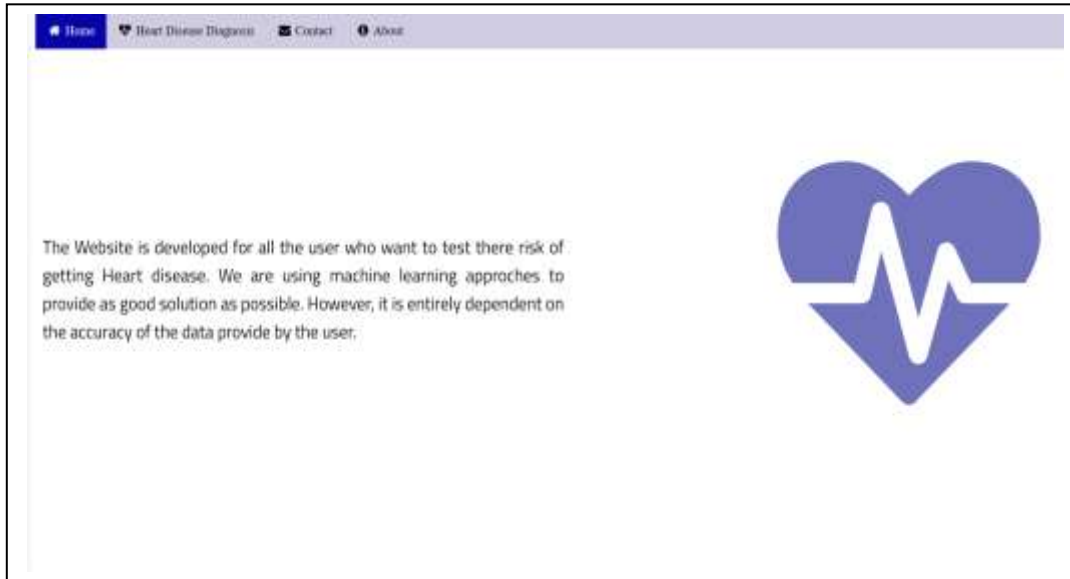
#### **8.2.2.5 USER ACCEPTANCE TESTING**

User acceptance of the system is key factor for the success of any system. The system under consideration is tested for user acceptance by constantly keeping in touch with prospective system and user at the time of developing and making changes whenever required. This is done in regarding to the following points.

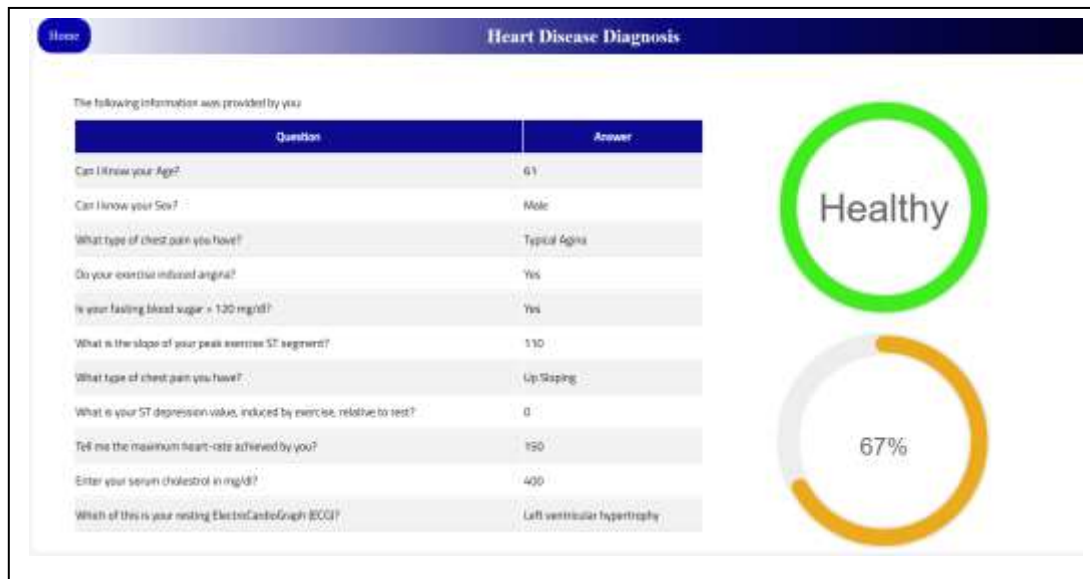
- Input screen design.
- Output screen design.

# 9.SCREENSHOTS

SCREENS:



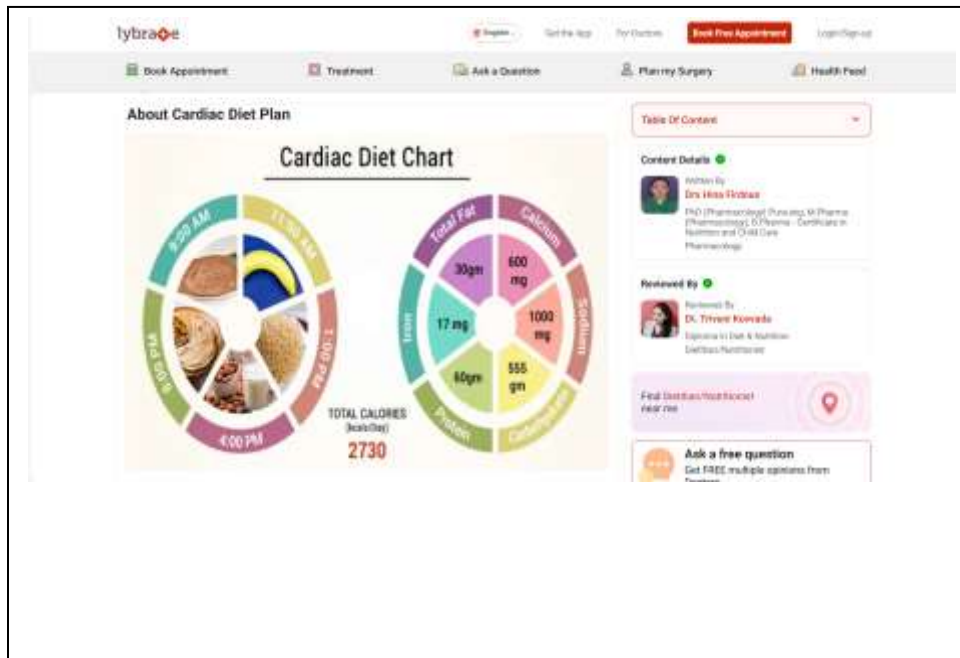
9.1 Home Screen



9.2 Healthy Case



### 9.3 Risk Case



### 9.4 Diet Chart

## 10. CONCLUSION

This project focuses on machine learning model for predicting heart disease. After data collection we perform data pre-processing and analysis, model training, and evaluation. The dataset is visualized using a density map and a correlation map. The density map provides a visual representation of the distribution of each feature in the dataset, and the correlation map provides information about the correlation between the different features. In preprocessing we performs one-hot encoding on the categorical features. After one-hot

encoding, the categorical features are dropped from the data, and the remaining features are normalized using the StandardScaler. Finally Implemented hybrid algorithm and flask framework prediction.

## 11. REFERENCES

- [1] S. I. Ansarullah and P. Kumar, “A systematic literature review on cardiovascular disorder identification using knowledge mining and machine learning method,” *Int. J. Recent Technol. Eng.*, vol. 7, no. 6S, pp. 1009–1015, 2019
- [2] A. U. Haq, J. P. Li, J. Khan, M. H. Memon, S. Nazir, S. Ahmad, G. A. Khan, and A. Ali, “Intelligent machine learning approach for effective recognition of diabetes in E-healthcare using clinical data,” *Sensors*, vol. 20, no. 9, p. 2649, May 2020
- [3] A. U. Haq, J. Li, M. H. Memon, M. H. Memon, J. Khan, and S. M. Marium, “Heart disease prediction system using model of machine learning and sequential backward selection algorithm for features selection,” in *Proc. IEEE 5th Int. Conf. Converg. Technol. (ICT)*, Mar. 2019, pp. 1–4

- [4] U. Haq, J. Li, M. H. Memon, J. Khan, and S. U. Din, "A novel integrated diagnosis method for breast cancer detection," *J. Intell. Fuzzy Syst.*, vol. 38, no. 2, pp. 2383–2398, 2020.
- [5] Attia, Z., Karras, B., & Kheir, M. (2020). Heart disease diagnosis using machine learning techniques: a review. *Journal of Ambient Intelligence and Humanized Computing*, 11(7), 2953-2964.
- [6] Pablico, L. S., Su, H. Y., & Chen, K. Y. (2020). A Review on Machine Learning Techniques for Heart Disease Diagnosis. *Journal of Medical Systems*, 44(9), 176.
- [7] Ribeiro, A. H., Ribeiro, M. H., Paixão, G. M. M., Oliveira, D. M., & Ribeiro, A. F. (2019). Machine learning models for predicting acute myocardial infarction. *Expert Systems with Applications*, 124, 152-159.
- [8] Pathak, Y., Kumar, M., & Bhatia, V. (2021). Comparative Analysis of Machine Learning Techniques for Heart Disease Diagnosis. *International Journal of Intelligent Systems and Applications*, 13(2), 40-48.
- [9] Liu, F., Zhang, Q., & Yang, F. (2019). Application of machine learning algorithms in diagnosis of heart disease. *Journal of Healthcare Engineering*, 2019, 1-11.
- [10] Abdi, A., Ahmadi, M., & Alikhani, M. (2021). Machine learning-based approach for predicting the risk of heart disease. *Journal of Ambient Intelligence and Humanized Computing*, 12(6), 6149-6160.
- [11] Sarwar, M. H., & Javaid, N. (2019). Heart disease prediction using machine learning and data mining techniques: a review. *2019 International Conference on Engineering and Emerging Technologies (ICEET)*, 1-6.
- [12] Liu, X., Xie, J., Yang, X., & Deng, W. (2021). A novel feature selection and classification method for heart disease detection based on machine learning. *Computer Methods and Programs in Biomedicine*, 204, 106078.