# PROFESSIONAL TRAINING REPORT

**at**

**Sathyabama Institute of Science and Technology (Deemed to be University)**

Submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering Degree in Computer Science and Engineering

By

## GAJJALA AMRUTH REDDY

**REG. NO: 39110307**

**A picture containing text

Description automatically generated**

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**SCHOOL OF COMPUTING**

**SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY**

**JEPPIAAR NAGAR, RAJIV GANDHI SALAI,**

**CHENNAI – 600119, TAMILNADU**

**APRIL 2022**

|  |  |  |
| --- | --- | --- |
|  | **SATHYABAMA**  **INSTITUTE OF SCIENCE AND TECHNOLOGY** (DEEMED TO BE UNIVERSITY) **Accredited with Grade “A” by NAAC**  (Established under Section 3 of UGC Act, 1956)  JEPPIAAR NAGAR, RAJIV GANDHI SALAI  CHENNAI– 600119  [**www.sathyabama.ac.in**](http://www.sathyabama.ac.in) |  |

# DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

**BONAFIDE CERTIFICATE**

This is to certify that this Project Report is the Bonafide work of **GAJJALA AMRUTH REDDY(Reg. No: 39110307)** who carried out the project entitled “ **HEART DIESASE PREDICTION ”** under my supervision from June 2021 to November 2021.

## Internal Guide

**Dr .PRAVEN**

**Head of the Department**

**Dr. S. VIGNESHWARI, M.E., Ph.D.,**

**Dr. LAKSHMANAN L, M.E., Ph.D.,**



## Submitted for Viva voce Examination held on

**Internal Examiner External Examiner**

**DECLARATION**

I, **GAJJALA AMRUTH REDDY**hereby declare that the project report entitled “**HEART DIESASE PREDICTION”** done by me under the guidance of **Dr. PRAVEN** is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering Degree in Computer Science and Engineering.

## DATE: AMRUTH REDDY

**PLACE: SIGNATURE OF THE CANDIDATE**

**ACKNOWLEDGEMENT**

I am pleased to acknowledge my sincere thanks to **Board of Management** of **SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T. Sasikala M.E., Ph.D**, **Dean**, School of Computing, **Dr. S. Vigneshwari, M.E., Ph.D. and Dr. L. Lakshmanan, M.E., Ph.D., Heads of the Department** of **Computer Science and Engineering** for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr. PRAVEN** for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Computer Science and Engineering** who were helpful in many ways for the completion of the project.

**TRAINING CERTIFICATE**

## TABLE OF CONTENTS

|  |  |  |
| --- | --- | --- |
| **CHAPTER No.** | **TITLE** | **PAGE No** |

ABSTRACT i

LIST OF FIGURES ii

LIST OF TABLES iii

LIST OF ABBREVATIONS iv

1. **INTRODUTION 1**

1.1 MOTIVATION FOR THE WORK 2

1.2 PROBLEM STATEMENT 2

**2 METHODOLOGY 3**

2.1 EXISTING SYSTEM 3

2.2 PROPOSED SYSTEM 3

2.2.1 COLLECTION OF DATASET 4

2.2.2 SELECTION OF ATTRIBUTES 4

2.2.3 PRE-PROCESSING OF DATA 5

2.2.4 BALANCING OF DATA 6

2.2.5 PREDICTION OF DISEASE 7

**3 WORKING OF SYSTEM 8**

3.1 SYSTEM ARCHITECTURE 8

3.2 EXPLAINATION ML ALGORITHMS 9

**4 EXPERIMENTAL ANALYSIS 13**

4.1 SYSTEM CONFIGURATION 13

4.1.1 HARDWARE REQUIREMENTS: 13

4.1.2 SOFTWARE REQUIREMENTS: 13

4.2 DATASET DETAILS 14

**5 PERFORMANCE ANALYSIS 17**

**6 RESULT 19**

**7 CONCLUSION 20**

**8 REFERENCES 21**

**9 APPENDIX 22**

9.1 OUTPUT 23

9.2 SOURCE CODE 25

# 

# ABSTRACT

Machine Learning is used across many ranges around the world. The healthcare industry is no exclusion. Machine Learning can play an essential role in predicting presence/absence of locomotors disorders, Heart diseases and more. Such information, if predicted well in advance, can provide important intuitions to doctors who can then adapt their diagnosis and dealing per patient basis. We work on predicting possible Heart diseases in people using Machine Learning algorithms. In this project we perform the comparative analysis of classifiers like Simple Linear Regression, Polynomial Regression, KNN ,Logistic Regression and we propose an ensemble classifier which perform hybrid classification by taking strong and weak classifiers since it can have multiple number of samples for training and validating the data so we perform the analysis of existing classifier and proposed which can give the better accuracy and predictive analysis.

(i)

# LIST OF FIGURES

## FIGURE NO. FIGURE NAME PAGE NO.

* + - 1. 2.2.1 Collection of data 4
      2. 2.2.2 Correlation matrix 5
      3. 2.2.3 preprocessing of data 5
      4. 2.2.4 Data Balancing 6
      5. 2.2.5 Prediction of disease 7
      6. 4.1 System Architecture 8
      7. 4.2.1 SL and PR models 10
      8. 4.2.2 Logistic Regression 11
      9. 4.2.3 KNN algorithm 12
      10. 5.1 Dataset Attributes 14
      11. 6.1 Confusion matrix 17
      12. 6.2 Correlation Matrix 18
      13. 9.1 Output 23

## 

## (ii)

# LIST OF TABLES

## TABLE NO. TABLE NAME PAGE NO.

## 1 5.1 Attributes of the Dataset 16

## 

## (iii)

**LIST OF ABBREVATIONS**

ML Machine Learning

AI Artificial Intelligence

NN Neural Networks

KNN K-Nearest Neighbor

|  |
| --- |
|  |
|  |
|  |
|  |

# (iv)

# INTRODUCTION

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 researches 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.

**1**

* 1. **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 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. **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**

**METHODOLOGY**

**2.1 EXISTING SYSTEM**

Heart disease is even being highlighted as a silent killer which leads to the death of a person without obvious symptoms. The nature of the disease is the cause of growing anxiety about the disease & its consequences. Hence continued efforts are being done to predict the possibility of this deadly disease in prior. So that various tools & techniques are regularly being experimented with to suit the present-day health needs. 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 conclude. This technique can be very well adapted to the do the prediction of heart disease. As the well-known quote says “Prevention is better than cure”, early prediction & its control can be helpful to prevent & decrease the death rates due to heart disease.

**2.2 PROPOSED SYSTEM**

The working of the system starts with the collection of data and selecting the important attributes. Then the required data is pre-processed into the required format. The data is then divided into two parts training and testing data. The algorithms are applied and the model is trained using the training data. The accuracy of the system is obtained by testing the system using the testing data. This system is implemented using the following modules.

1.) Collection of Dataset

2.) Selection of attributes

3.) Data Pre-Processing

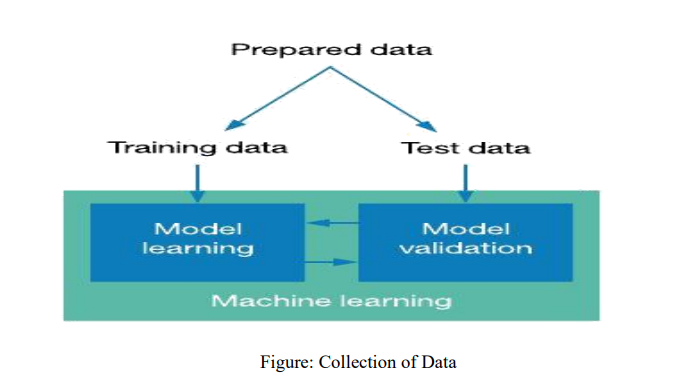
4.) Balancing of Data

5.) Disease Prediction

**3**

**2.2.1 Collection of dataset**

Initially, we collect a dataset for our heart disease prediction system. After the collection of the dataset, we split the dataset into training data and testing data. The training dataset is used for prediction model learning and testing data is used for evaluating the prediction model. The dataset used for this project is Heart Disease UCI. The dataset consists of 76 attributes; out of which, 14 attributes are used for the system.

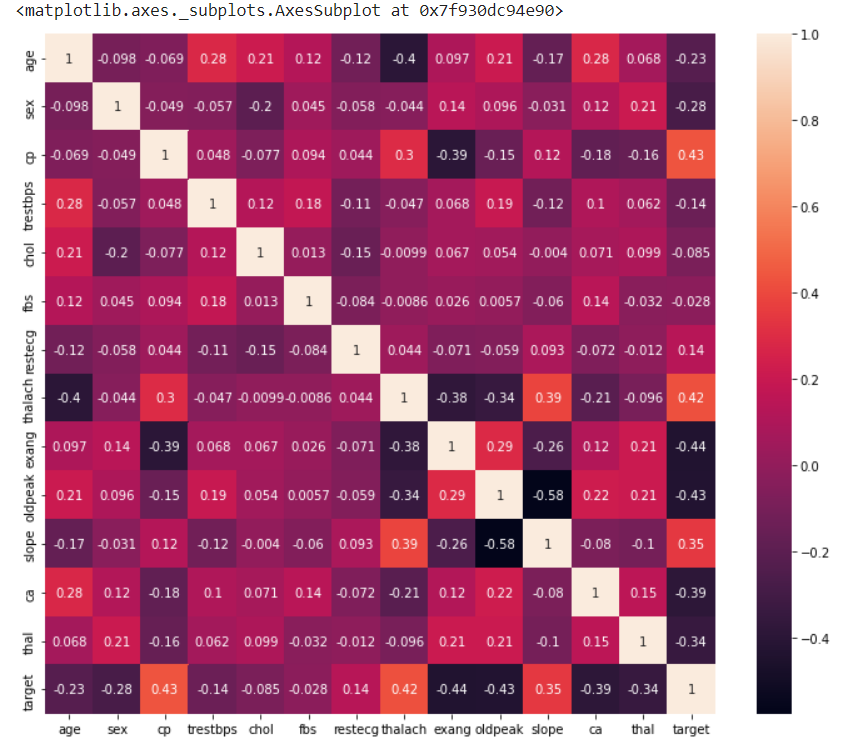


**figure 2.2.1**

**2.2.2 Selection of attributes**

Attribute or Feature selection includes the selection of appropriate attributes for the prediction system. This is used to increase the efficiency of the system. Various attributes of the patient like gender, chest pain type, fasting blood pressure, serum cholesterol, exang ,etc are selected for the prediction. The Correlation matrix is used for attribute selection for this model

**4**



**Figure 2.2.2**

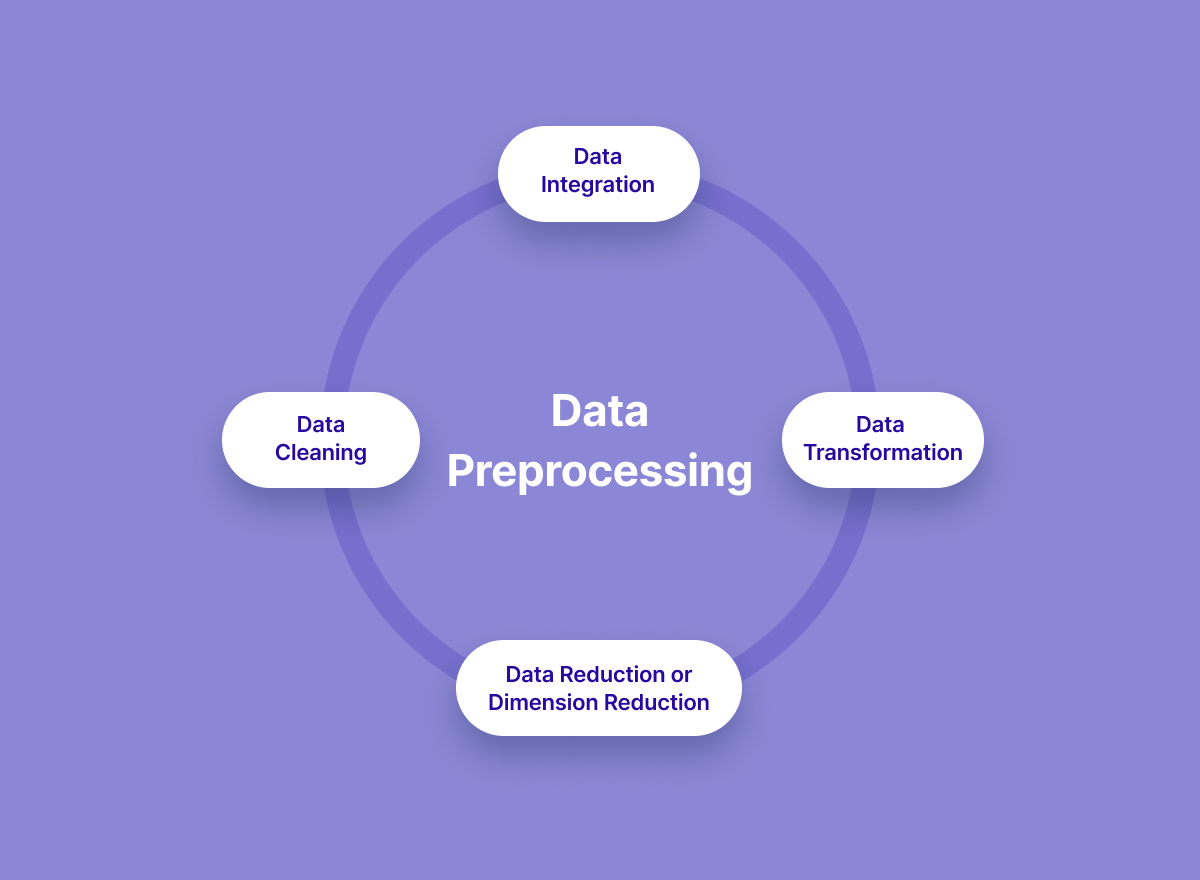
**3.2.3 Pre-processing of Data**

Data pre-processing is an important step for the creation of a machine learning model. Data pre-processing has the activities

* Getting the dataset
* Importing libraries
* Importing datasets
* Finding Missing Data
* Encoding Categorical Data
* Splitting dataset into training and test set
* Feature scaling

Preprocessing of data is required for improving the accuracy of the model.

**5**



**Figure 2.2.3**

**3.2.4 Balancing of Data**

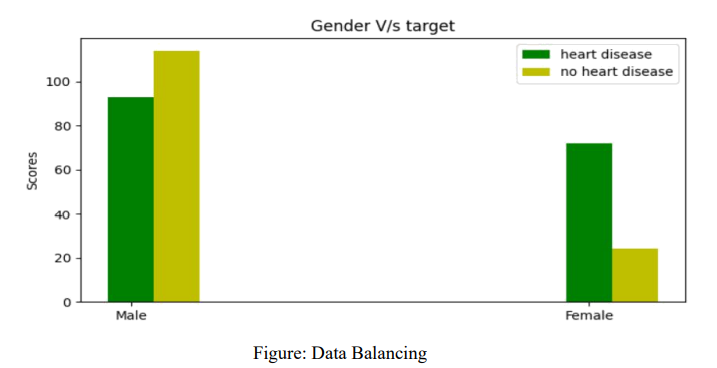
Imbalanced datasets can be balanced in two ways.

They are Under Sampling and Over Sampling

(**a) Under Sampling**: In Under Sampling, dataset balance is done by the reduction of the size of the ample class. This process is considered when the amount of data is adequate.

**(b) Over Sampling**: In Over Sampling, dataset balance is done by increasing the size of the scarce samples. This process is considered when the amount of data is inadequate.

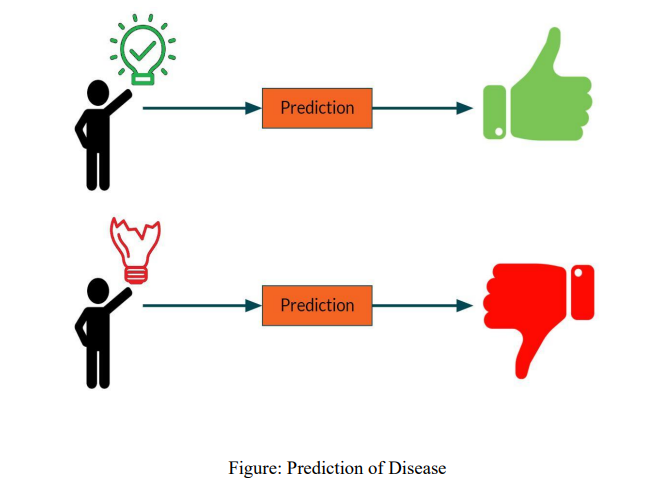
**6**



**figure 2.2.4**

**2.2.5 Prediction of Disease**

Various machine learning algorithms Simple Linear Regression, Polynomial Regression are used for classification. Comparative analysis is performed among algorithms and the algorithm that gives the highest accuracy is used for heart disease prediction.



**figure 2.2.5**

**7**

**WORKING OF SYSTEM**

**4.1 SYSTEM ARCHITECTURE**

The system architecture gives an overview of the working of the system. The working of this system is described as follows: Dataset collection is collecting data which contains patient details. Attributes selection process selects the useful attributes for the prediction of heart disease. After identifying the available data resources, they are further selected, cleaned, made into the desired form. Different classification techniques as stated will be applied on preprocessed data to predict the accuracy of heart disease. Accuracy measure compares the accuracy of different classifiers.

Data pre-processing

Attribute Selection

Classification techniques

Details of Patient

KNN

Simple Linear

Regression

Polynomial Regression

Logistic

Regression

Loan Prediction

Accuracy Score

Fig 4.1

**8**

**figure 4.2.1**

**LOGISTIC REGRESSION ALGORITHM**

Logistic regression is one of the most popular Machine Learning algorithms, which comes under the Supervised Learning technique. It is used for predicting the categorical dependent variable using a given set of independent variables.

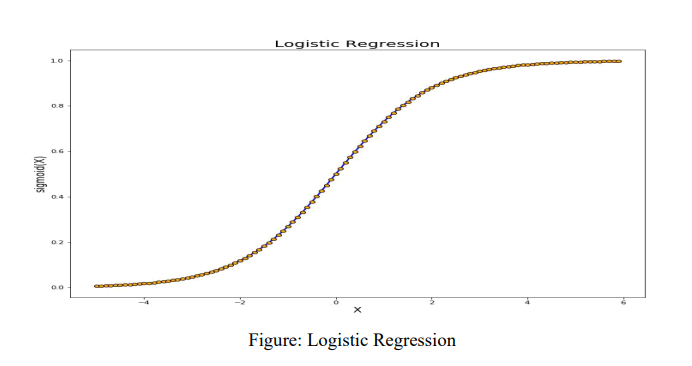
Logistic regression predicts the output of a categorical dependent variable. Therefore, the outcome must be a categorical or discrete value. It can be either Yes or No, 0 or 1, true or False, etc. but instead of giving the exact value as 0 and 1, it gives the probabilistic values which lie between 0 and 1.

Logistic Regression is much similar to the Linear Regression except that how they are used. Linear Regression is used for solving Regression problems, whereas logistic regression is used for solving the classification problems.

In Logistic regression, instead of fitting a regression line, we fit an "S" shaped logistic function, which predicts two maximum values (0 or 1).

The curve from the logistic function indicates the likelihood of something such as whether the cells are cancerous or not, a mouse is obese or not based on its weight, etc.

Logistic Regression is a significant machine learning algorithm because it has the ability to provide probabilities and classify new data using continuous and discrete datasets.

****

**figure 4.2.2**

**11**

# K-Nearest Neighbor (KNN) Algorithm for Machine Learning

* K-Nearest Neighbor is one of the simplest Machine Learning algorithms based on Supervised Learning technique.
* K-NN algorithm assumes the similarity between the new case/data and available cases and put the new case into the category that is most similar to the available categories.
* K-NN algorithm stores all the available data and classifies a new data point based on the similarity.
* K-NN algorithm can be used for Regression as well as for Classification but mostly it is used for the Classification problems.
* K-NN is a **non-parametric algorithm**, which means it does not make any assumption on underlying data.
* It is also called a **lazy learner algorithm** because it does not learn from the training set immediately instead it stores the dataset and at the time of classification, it performs an action on the dataset.
* Suppose there are two categories, i.e., Category A and Category B, and we have a new data point x1, so this data point will lie in which of these categories. To solve this type of problem, we need a K-NN algorithm. With the help of K-NN, we can easily identify the category or class of a particular dataset. Consider the below diagram:



**figure 4.2.3**

**12**

**EXPERIMENTAL ANALYSIS**

**5.1 SYSTEM CONFIGURATION**

**5.1.1 Hardware requirements:**

Processer : Any Update Processer

Ram : Min 4GB

Hard Disk : Min 100GB

**5.1.2 Software requirements:**

Operating System : Windows family

Technology : Python3.7

IDE : Jupiter notebook

**13**

**DATASET DETAILS**

• Of the 304 attributes available in the dataset,14 attributes are considered for the

prediction of the output.

• Heart Disease UCI :

Figure: 5.1

**14**

**Input dataset attributes**

● Gender (value 1: Male; value 0 : Female)

● Chest Pain Type (value 1: typical type 1 angina, value 2: typical type angina, value 3: non-angina pain; value 4: asymptomatic)

● Fasting Blood Sugar (value 1: > 120 mg/dl; value 0:< 120 mg/dl)

● Exang – exercise induced angina (value 1: yes; value 0: no)

● CA – number of major vessels colored by fluoroscopy (value 0 – 3)

● Thal (value 3: normal; value 6: fixed defect; value 7:reversible defect)

● Trest Blood Pressure (mm Hg on admission to the hospital)

● Serum Cholesterol (mg/dl)

● Thalach – maximum heart rate achieved

● Age in Year

● Height in cms

● Weight in Kgs.

● Cholestrol

● Restecg

**15**

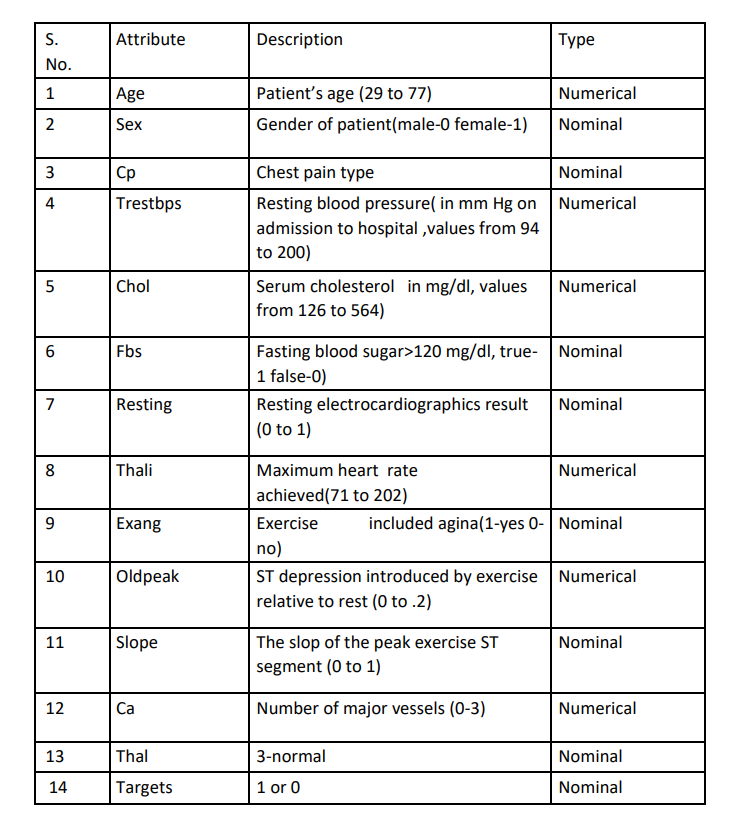
****

TABLE 1: Attributes of the dataset

**16**

**PERFORMANCE ANALYSIS**

In this project, various machine learning algorithms like Logistic Regression, Simple Linear Regression, KNN, Polynomial Regression are used to predict heart disease. Heart Disease UCI dataset, has a total of 76 attributes, out of those only 14 attributes are considered for the prediction of heart disease. Various attributes of the patient like gender, chest pain type, fasting blood pressure, serum cholesterol,

Exang ,etc are considered for this project. The accuracy for individual algorithms has to measure and whichever algorithm is giving the best accuracy, that is considered for the heart disease prediction. For evaluating the experiment, various evaluation metrics like accuracy, confusion matrix, precision, recall, and f1-score are considered. Accuracy- Accuracy is the ratio of the number of correct predictions to the total number of inputs in the dataset.

**Confusion Matrix**- It gives us a matrix as output and gives the total performance of the system.

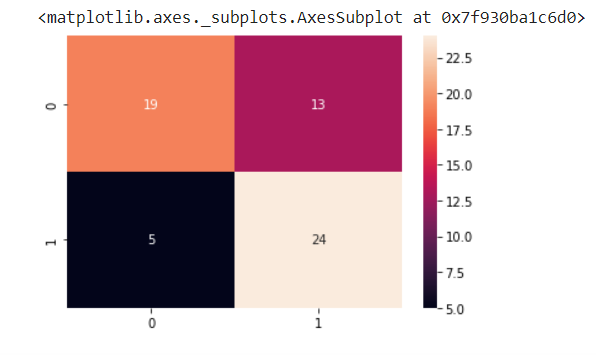


Figure:6.1

**Correlation Matrix:** The correlation matrix in machine learning is used for feature selection. It represents dependency between various attributes.

**17**

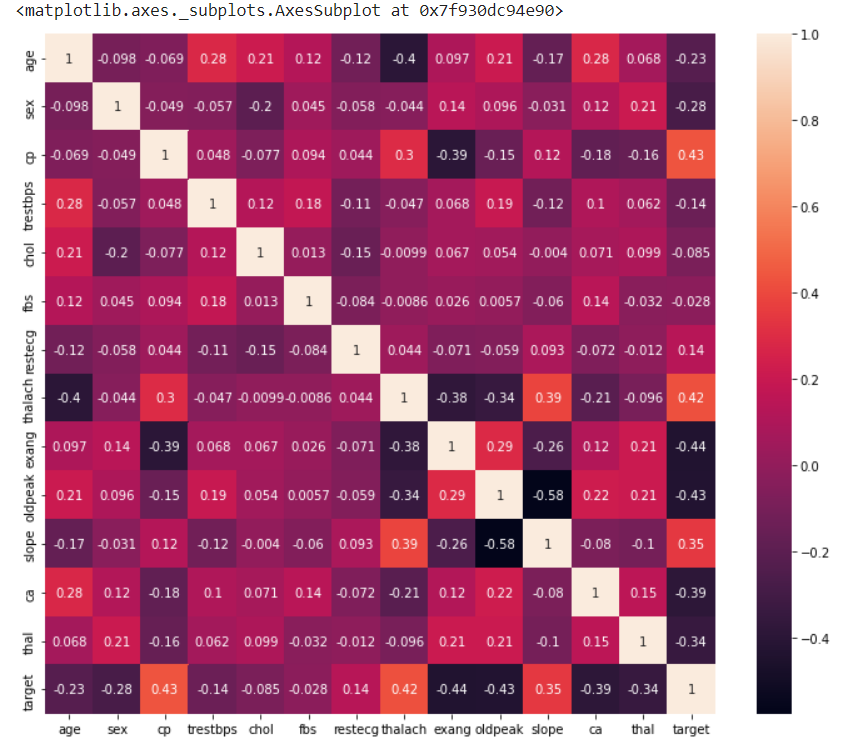


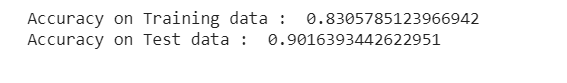
Fig: 6.2

**Precision-** It is the ratio of correct positive results to the total number of positive results predicted by the system.

It is expressed as: Recall-It is the ratio of correct positive results to the total number of positive results predicted by the system.

It is expressed as: F1 Score-It is the harmonic mean of Precision and Recall. It measures the test accuracy. The range of this metric is 0 to 1

**Accuracy Score**



**18**

**RESULT**

* This project aims to know whether the patient has heart disease or not .The records in the dataset are divided into the training set and test sets. After preprocessing the data.
* The data classification technique namely Simple Linear Regression ,Polynomial Regression were applied. The project involved analysis of the heart disease patient dataset with proper data
* The goal of our heart disease prediction project is to determine if a patient should be diagnosed with heart disease or not, which is a binary outcome, so: Positive result = 1, the patient will be diagnosed with heart disease a processing.

**19**

# CONCLUSION

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

# This project provides the deep insight into machine learning techniques for classification of heart diseases. The role of classifier is crucial in healthcare industry so that the results can be used for predicting the treatment which can be provided to patients. The existing techniques are studied and compared for finding the efficient and accurate systems. Machine learning techniques significantly improves accuracy of cardiovascular risk prediction through which patients can be identified during an early stage of disease and can be benefitted by preventive treatment. It can be concluded that there is a huge scope for machine learning algorithms in predicting cardiovascular diseases or heart related diseases

# With the increasing number of deaths due to heart diseases, it has become mandatory to develop a system to predict heart diseases effectively and accurately. The motivation for the study was to find the most efficient ML algorithm for detection of heart diseases.

# The correct prediction of heart disease can prevent life threats, and incorrect prediction can prove to be fatal at the same time. In this paper different machine learning algorithms and deep learning are applied to compare the results and analysis of the UCI Machine Learning Heart Disease dataset.

**20**

# REFERENCES:

# [1] Soni J, Ansari U, Sharma D & Soni S (2011). Predictive data mining for medical diagnosis: an overview of heart disease prediction. International Journal of Computer Applications, 17(8), 43-8

# [2] Ganna A, Magnusson P K, Pedersen N L, de Faire U, Reilly M, Ärnlöv J & Ingelsson E (2013). Multilocus genetic risk scores for coronary heart disease prediction. Arteriosclerosis, thrombosis, and vascular biology, 33(9), 2267-72.

# [3] <https://www.researchgate.net/publication/349470771_Using_Machine_Learning_for_Heart_Disease_Prediction>

**21**

**APPENDIX**

**Python**

Python is an interpreted, high-level, general purpose programming language created by Guido Van Rossum and first released in 1991, Python's design philosophy emphasizes code Readability with its notable use of significant White space. Its language constructs and object oriented approach aim to help programmers write clear, logical code for small and large-scale projects. Python is dynamically typed and garbage collected. It supports multiple programming paradigms, including procedural, object-oriented, and functional programming.

**Sklearn**

Scikit-learn (Sklearn) is the most useful and robust library for machine learning in Python. It provides a selection of efficient tools for machine learning and statistical modeling including classification, regression, clustering and dimensionality reduction via a consistent interface in Python. This library, which is largely written in Python, is built upon NumPy, SciPy and Matplotlib.

**Numpy**

NumPy is a library for the python programming language, adding support for large, multi- dimensional arrays and matrices, along with a large collection of high level mathematical functions to operate on these arrays. The ancestor of NumPy, Numeric, was originally created by Jim with contributions from several other developers. In 2005, Travis created NumPy by incorporating features of the competing Numarray into Numeric, with extensive modifications. NumPy is open source software and has many contributors.

**Matplotlib**

Matplotlib is a plotting library for the Python programming language and its numerical mathematics extension NumPy. It provides an object-oriented API for embedding plots into applications using general-purpose GUI toolkits like Tkinter, wxPython, Qt, or GTK. There is also a procedural "pylab " interface based on a state machine (like OpenGL), designed to closely resemble that of MATLAB, though its use is discouraged.

**Seaborn:**

Seaborn is a Python data visualization library based on matplotlib. It provides a highlevel interface for drawing attractive and informative statistical graphics. Seaborn is a library in Python predominantly used for making statistical graphics. Seaborn is a data visualization library built on top of matplotlib and closely integrated with pandas data structures in Python. Visualization is the central part of Seaborn which helps in exploration and understanding of da

**22**

**Source Code:**

%matplotlib inline

import matplotlib.pyplot as plt

import numpy as np

import pandas as pd

import seaborn as sns

import warnings

warnings.filterwarnings("ignore")

from sklearn import linear\_model

from sklearn.metrics import mean\_squared\_error, r2\_score

from sklearn.preprocessing import PolynomialFeatures

from sklearn.model\_selection import train\_test\_split

from sklearn.linear\_model import LogisticRegression

from sklearn.metrics import confusion\_matrix, classification\_report # this creates a confusion matrix

#from sklearn.metrics import plot\_confusion\_matrix                  # draws a confusion matrix

from sklearn.metrics import accuracy\_score

[1]

%matplotlib inline

[2]

**import** matplotlib.pyplot **as** plt  
**import** numpy **as** np  
**import** pandas **as** pd  
  
**from** sklearn **import** linear\_model  
**from** sklearn.metrics **import** mean\_squared\_error, r2\_score  
**from** sklearn.preprocessing **import** PolynomialFeatures

## Simple Linear Regression

[3]

df = pd.read\_csv(**'heart.csv'**)

[4]

df.head()

[5]

*# X and Y need to be two-dimensional arrays of shape (n\_samples, n\_features) and (n\_sample, n\_targets)*

[6]

heart\_y = df[**"age"**].to\_numpy()  
heart\_y.shape

(303,)

[7]

heart\_y = heart\_y.reshape((-1,1))  
heart\_y.shape

(303, 1)

[8]

heart\_X = df[**'chol'**].to\_numpy()  
heart\_X = heart\_X.reshape((-1, 1))  
heart\_X.shape

(303, 1)

[9]

type(heart\_y)

numpy.ndarray

[11]

*# See slide on train/test dataset*  
*#Train is generally 80 % of your dataset*  
*#Test is generally 20 % of your dataset*

[12]

*# Split the data into training/testing sets*  
  
heart\_X\_train = heart\_X[:242]  
heart\_X\_test = heart\_X[242:]

[13]

*# Split the targets into training/testing sets*  
  
heart\_y\_train = heart\_y[:242]  
heart\_y\_test = heart\_y[242:]

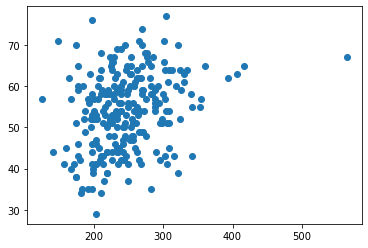
[14]

heart\_X\_train.shape, heart\_y\_train.shape

((242, 1), (242, 1))

[17]

plt.scatter(heart\_X\_train, heart\_y\_train)  
plt.show()

****

[18]

heart\_X\_test.shape, heart\_X\_train.shape

((61, 1), (242, 1))

[19]

*# Create linear regression object*  
  
regr = linear\_model.LinearRegression()

[22]

??regr.fit

[23]

*# Train the model using the training sets*  
  
regr.fit(heart\_X\_train, heart\_y\_train)

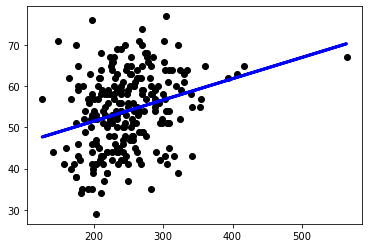
LinearRegression()

[24]

heart\_y\_pred = regr.predict(heart\_X\_train)

[25]

plt.scatter(heart\_X\_train, heart\_y\_train,  color=**'black'**)  
plt.plot(heart\_X\_train, heart\_y\_pred, color=**'blue'**, linewidth=3)  
  
plt.show()

****

[26]

*# Make predictions using the testing set*  
  
heart\_y\_pred = regr.predict(heart\_X\_test)

[27]

*# y = ax + b*

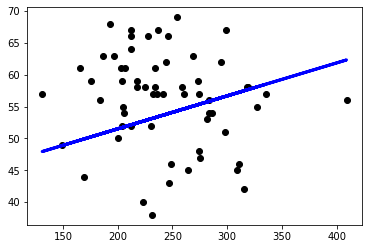
[30]

*# The coefficients*  
  
print(**'Coefficients: \n'**, regr.coef\_)  
  
*# The intercept*  
print(**'Intercept: \n'**, regr.intercept\_)  
  
*# The mean squared error*  
  
print(**'Mean squared error: %.2f'**  
      % mean\_squared\_error(heart\_y\_test, heart\_y\_pred))  
  
*# The coefficient of determination: 1 is perfect prediction*  
  
print(**'Coefficient of determination: %.2f'**  
      % r2\_score(heart\_y\_test, heart\_y\_pred))

Coefficients:   
 [[0.05169721]]  
Intercept:   
 [41.162605]  
Mean squared error: 71.51  
Coefficient of determination: -0.33

[29]

*# Plot outputs*  
  
plt.scatter(heart\_X\_test, heart\_y\_test,  color=**'black'**)  
plt.plot(heart\_X\_test, heart\_y\_pred, color=**'blue'**, linewidth=3)  
  
plt.show()

****

## Polynomial Regression

[34]

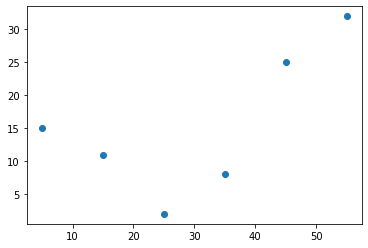
??regr.fit

[35]

x = np.array([5, 15, 25, 35, 45, 55]).reshape((-1, 1))  
y = np.array([15, 11, 2, 8, 25, 32])

[36]

plt.scatter(x, y)  
plt.show()

****

[37]

x.shape

(6, 1)

[51]

*# degree can be greater than 2*  
  
transformer = PolynomialFeatures(degree=2, include\_bias=**False**)  
transformer.fit(x)  
x\_ = transformer.transform(x)

[52]

*# Can replace the above three statements with*  
*# x\_ = PolynomialFeatures(degree=2, include\_bias=False).fit\_transform(x)*

[53]

print(x\_)

[[ 5. 25.]  
 [ 15. 225.]  
 [ 25. 625.]  
 [ 35. 1225.]  
 [ 45. 2025.]  
 [ 55. 3025.]]

[54]

model = linear\_model.LinearRegression()  
model.fit(x\_, y)

LinearRegression()

[55]

r\_sq = model.score(x\_, y)  
  
print(**'coefficient of determination:'**, r\_sq)  
  
print(**'intercept:'**, model.intercept\_)  
  
print(**'coefficients:'**, model.coef\_)

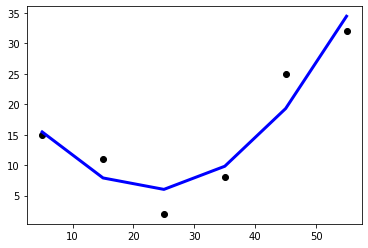
coefficient of determination: 0.8908516262498564  
intercept: 21.372321428571425  
coefficients: [-1.32357143 0.02839286]

[57]

*# y = ax^2 + bx + c*

[56]

plt.scatter(x, y,  color=**'black'**)  
plt.plot(x, model.predict(x\_), color=**'blue'**, linewidth=3)  
  
plt.show()

****

## Underfitting / Overfitting

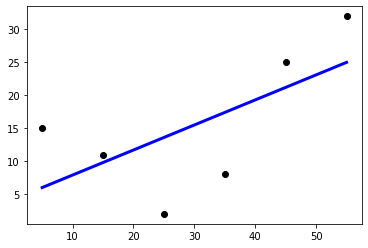
[58]

*# Using linear fit for above example (Under-fitting)*  
  
regr.fit(x, y)  
  
y\_pred = regr.predict(x)  
  
*# The coefficients*  
  
print(**'Coefficients: \n'**, regr.coef\_)  
  
*# The mean squared error*  
  
print(**'Mean squared error: %.2f'**  
      % mean\_squared\_error(y, y\_pred))  
  
*# The coefficient of determination: 1 is perfect prediction*  
  
print(**'Coefficient of determination: %.2f'**  
      % r2\_score(y, y\_pred))

Coefficients:   
 [0.38]  
Mean squared error: 61.47  
Coefficient of determination: 0.41

[59]

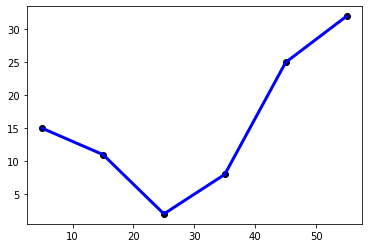
plt.scatter(x, y,  color=**'black'**)  
plt.plot(x, y\_pred, color=**'blue'**, linewidth=3)  
  
plt.show()

****

[60]

*# Over-fitting*  
  
x\_ = PolynomialFeatures(degree=10, include\_bias=**False**).fit\_transform(x)  
  
model = linear\_model.LinearRegression().fit(x\_, y)  
  
r\_sq = model.score(x\_, y)  
  
print(**'coefficient of determination:'**, r\_sq)  
  
print(**'intercept:'**, model.intercept\_)  
  
print(**'coefficients:'**, model.coef\_)  
  
plt.scatter(x, y,  color=**'black'**)  
plt.plot(x, model.predict(x\_), color=**'blue'**, linewidth=3)  
  
plt.show()

coefficient of determination: 1.0  
intercept: 15.020359754678793  
coefficients: [-2.31761841e-11 -2.47328301e-10 -3.40704264e-09 -3.95084163e-08  
 -3.63619813e-07 -2.06682371e-06 2.01144641e-07 -7.26480195e-09  
 1.15642163e-10 -6.83323029e-13]

****

## Train/Test split to avoid under/over fitting

[76]

transformer = PolynomialFeatures(degree=5, include\_bias=**False**)  
  
transformer.fit(heart\_X\_train)

PolynomialFeatures(degree=5, include\_bias=False)

[77]

heart\_X\_train\_trans = transformer.transform(heart\_X\_train)

[78]

heart\_X\_test\_trans = transformer.transform(heart\_X\_test)

[79]

model = linear\_model.LinearRegression().fit(heart\_X\_train\_trans, heart\_y\_train)

[80]

heart\_y\_pred = model.predict(heart\_X\_test\_trans)

[81]

*# The coefficients*  
  
print(**'Coefficients: \n'**, model.coef\_)  
  
*# The mean squared error*  
  
print(**'Mean squared error of test: %.2f'**  
      % mean\_squared\_error(heart\_y\_test, heart\_y\_pred))  
  
*# The coefficient of determination: 1 is perfect prediction*  
  
print(**'Coefficient of determination of test: %.2f'**  
      % r2\_score(heart\_y\_test, heart\_y\_pred))

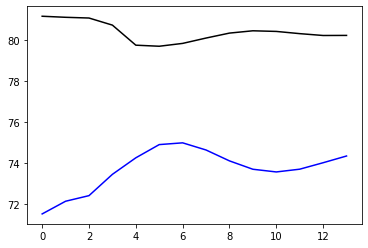
Coefficients:   
 [[-6.99138317e+00 4.90413789e-02 -1.63078074e-04 2.59006206e-07  
 -1.57136132e-10]]  
Mean squared error of test: 74.25  
Coefficient of determination of test: -0.38

[82]

mse\_list\_test = list()  
mse\_list\_train = list()  
  
**for** i **in** range(1, 15):  
    transformer = PolynomialFeatures(degree=i, include\_bias=**False**)  
    transformer.fit(heart\_X\_train)  
    heart\_X\_train\_trans = transformer.transform(heart\_X\_train)  
    heart\_X\_test\_trans = transformer.transform(heart\_X\_test)  
    model = linear\_model.LinearRegression().fit(heart\_X\_train\_trans, heart\_y\_train)  
    heart\_y\_pred = model.predict(heart\_X\_test\_trans)  
    mse\_list\_test.append(mean\_squared\_error(heart\_y\_test, heart\_y\_pred))  
    heart\_y\_pred = model.predict(heart\_X\_train\_trans)  
    mse\_list\_train.append(mean\_squared\_error(heart\_y\_train, heart\_y\_pred))

[86]

plt.plot(mse\_list\_test, **'b'**)  
plt.plot(mse\_list\_train, **'k'**)  
plt.show()

****

[87]

mse\_list\_test = np.array(mse\_list\_test)  
np.argmin(mse\_list\_test)+1

1

[88]

np.argmin([1,2,3])

0

## That's it folks

# 

# 31