Disease Prediction And Analysis

A PROJECT REPORT

Submitted By

Shreyansh Tyagi
(University Roll No:2000290140116)
Sneha Singhal
(University Roll No:2000290140122)
Shweta Rani
(University Roll No:2000290140120)

Submitted in partial fulfillment of the Requirements for the Degree of

MASTER OF COMPUTER APPLICATIONS

Under the Supervision of
Prof:Ankit Verma
Associate professor

Submitted to

DEPARTMENT OF COMPUTER APPLICATIONS
KIET Group of Institutions, Ghaziabad
Uttar Pradesh-201206

(MAY 2022)

CERTIFICATE

Certified that Shreyansh Tyagi (2000290140116), Sneha Singhal(2000290140122) ,Shweta Rani (2000290140120) have carried out the project work having " **Disease Prediction And Analysis**" for Master of Computer Applications from Dr. A.P.J. Abdul Kalam Technical University (AKTU) (formerly UPTU), Technical University, Lucknow under my supervision. The project report embodies original work, and studies are carried out by the student himself / herself and the contents of the project report do not form the basis for the award of any other degree to the candidate or to anybody else from this or any other University/Institution.

Date:26/05/2022

Shreyansh Tyagi (2000290140116)
Sneha Singhal (2000290140122)
Shweta Rani (2000290140120)

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

Date:26/05/2022

Mr. Ankit Verma
Associate Professor
Department of Computer Applications
KIET Group of Institutions, Ghaziabad

Signature of Internal Examiner **Signature of External Examiner**

Dr. Ajay Shrivastava
Head, Department of Computer Applications
KIET Group of Institutions, Ghaziabad

ABSTRACT

Nowadays , people are suffering by various diseases namely chest pain, blood pressure , blood sugar, heart failures, etc. As, elderly people stay alone at their homes when their children are out for work, so in case of emergency immediate medication is not provided. The early prediction of the disease leads to reduce in the chances of mishappening and increases the chances of survival of patients by taking appropriate actions. Human health is wealth, there is nothing more valuable than the good health. So, proper healthcare monitoring is must. In Rural areas, health monitoring is very challenging. Most of the people dies due to lack of medical facilities or because doctor is not available instantly. As a result, it's become more important than ever to predict and prevent heart disease . Data-driven solutions for predicting heart diseases can improve the overall research and prevention process, allowing more people to live a healthy lifestyle. Machine learning comes into play at this point. Heart disease prognosis is aided by machine learning, and the forecasts are fairly accurate.

This project includes data processing and analysis of a heart disease patient's dataset. Different algorithms, such as KNN, Decision Tree, Logistic Regression, Random Forest, and others, were used to train the models and provide predictions. After disease prediction, the alert signal is provided so that patient can be attained. The proposed research was implemented on PYTHON applying different machine learning (ML) algorithms and showed the accuracy of different algorithms used. Random Forest provides the result with highest i.e. 95% accuracy.

Keywords : Machine Learning(ML), Disease Prediction, Random Forest, KNN, Decision Tree, Logistic regression

ACKNOWLEDGEMENT

Success in life is never attained single handedly. My deepest gratitude goes to my thesis supervisor, **Mr. Ankit Verma** for his guidance, help and encouragement throughout my research work. Their enlightening ideas, comments, and suggestions.

Words are not enough to express my gratitude to Dr. Ajay Kumar Shrivastava, Professor and Head, Department of Computer Applications, for his insightful comments and administrative help at various occasions.

Fortunately, I have many understanding friends, who have helped me a lot on many critical conditions.

Finally, my sincere thanks go to my family members and all those who have directly and indirectly provided me moral support and other kind of help. Without their support, completion of this work would not have been possible in time. They keep my life filled with enjoyment and happiness.

Shreyansh Tyagi

Sneha Singhal

Shweta Rani

TABLE OF CONTENT

LIST OF FIGURES

CHAPETR 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 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 analyzing to extract the desired data we can say that this technique can be very well adapted to do the prediction of heart disease.

1.2 OBJECTIVE

The main objective of developing this project are:

1.To develop machine learning model to predict future possibility of heart disease by implementing Machine Learning Algorithm.

2.To determine significant risk factors based on medical dataset which may lead to heart disease.

3.To analyse feature selection methods and understand their working principle.

1.3 PROJECT SCOPE

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 eight classification algorithms namely Logistic Regression, Support Vector Machine (SVM),K-Nearest Neighbor(KNN),XGBoost(XGB),Neural Network(NN),Naïve Bayes, Decision Tree, 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, these algorithms are evaluated at numerous levels and types of evaluation strategies.

1.4 PROBLEM STATEMENT

The major challenge in heart disease is its detection. There are instruments available which can predict heart disease but either it is expensive or 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.

1.5 SYSTEM REQUIREMENTS

1.5.1 Hardware Requirements

Processer	:	Any Update Processer
Ram	:	Min 4GB
Hard Disk	:	Min 100GB

1.5.2 Software requirements

Operating System	:	Windows family
Technology	:	Python3.7
IDE	:	Jupiter notebook

CHAPTER 2 FEASIBILITY STUDY

A feasibility analysis is used to determine the viability of an idea, such as ensuring a project is legally and technically feasible as well as economically justifiable. It tells us whether a project is worth the investment in some cases, a project may not be doable. There can be many reasons for this, including requiring too many resources, which not only prevents those resources from performing other tasks but also may cost more than an organization would earn back by taking on a project that isn't profitable.

A well-designed study should offer a historical background of the business or project, such as a description of the product or service, accounting statements, details of operations and management, marketing research and policies, financial data, legal requirements, and tax obligations. Generally, such studies precede technical development and project implementation.

2.1 Technical Feasibility

This assessment focuses on the technical resources available to the organization. It helps organizations determine whether the technical resources meet capacity and whether the technical team can convert the ideas into working systems. Technical feasibility also involves the evaluation of the hardware, software, and other technical requirements of the proposed system. As an exaggerated example, an organization wouldn't want to try to put Star Trek's transporters in their building currently, this project is not technically feasible. During the technical feasibility step the following must be completed.

a. Identification of hardware requirements for developments.
b. Identification of software requirements for developments.
c. Conduct a preliminary production feasibility assessment.
d. Conduct a preliminary manufacturing assessment.
e. Security assessment for project.

Our project is technically feasible, since all the required technology i.e., hardware and software are available in the market. Since it is a small project, not required must security and storage issue.

2.2 Operational Feasibility

This assessment involves undertaking a study to analyze and determine whether and how well the organization’s needs can be met by completing the project. Operational feasibility studies also examine how a project plan satisfies the requirements identified in the requirements analysis phase of system development.
Our project is Operational feasible since it is very small. A proper training will be schedule to the end users if required.

2.3 Economical Feasibility

In Economic Feasibility study cost and benefit of the project is analyzed. Means under this feasibility study a detail analysis is carried out what will be cost of the project for development which includes all required cost for final development like hardware and software resource required, design and development cost and operational cost and so on. After that it is analyzed whether project will be beneficial in terms of finance for organization or not.
Our project is also Economical feasible since all required hardware and software are available for designing and implementation.

2.4 Behavioral Feasibility

It evaluates and estimates the user attitude or behaviour towards the development of new system. It helps in determining if the system requires special effort to educate, retrain, transfer, and changes in employee's job status on new ways of conducting business.

Our project is Behavioral feasible because it is new interactive system which helps early detection of disease by going through the overall data provided by the user over the period of time.

CHAPTER 3 METHODOLOGY

3.1 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 analyzing 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.

3.2 PROPOSED SYSTEM

The working of the system starts with the collection of data and selecting the important attributes. Then the required data is preprocessed 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.) Model Building
5.) Disease Prediction

3.2.1 Collection of dataset

Initially, we collect a dataset for our heart disease prediction system.We will be using a Dataset from Kaggle for this problem. 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. For this project, 70% of training data is used and 30% of data is used for testing. 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.

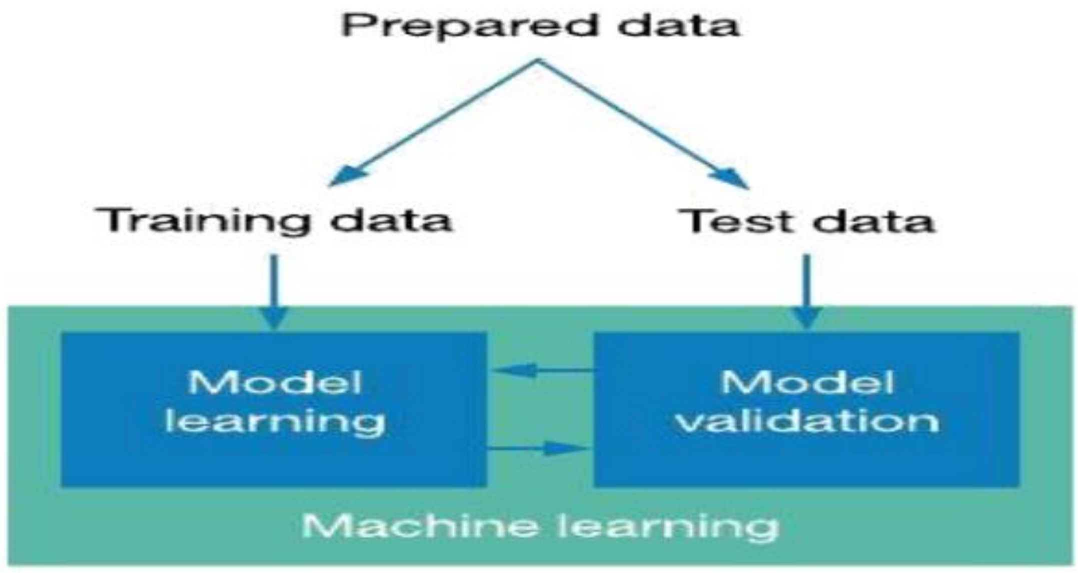


Fig 1 : Collection of data

3.2.2 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.

ATTRIBUTES INFORMATION

Age	:	Age
Sex	:	0:female, 1:male
CP	:	Chest Pain 0:typical angina,1:atypical angina:,2:non
		anginal pain,3:Asmptomatic
Chol	:	Serum Cholesterol in mg/dl
Trestbps	:	Testing Blood Pressure
Restecg	:	Resting Electrocardiographic results(values 0,1,2)
Thalach	:	Maximum heart rate achieved
Fbs	:	Fasting Blood Sugar>120 mg/dl
Oldpeak rest	:	oldpeak=St depression induced by exercise relative to
Slope	:	the slope of the peak exercise ST segment
Exang	:	exercise induced angina
Ca	:	number of major vessels (0-3) colored by flourosopy
Thal	:	thal:3=normal; 6=fixed defect; 7=reversable defect

3.2.3 Pre-processing of Data

Data pre-processing is an important step for the creation of a machine learning model. Initially, data may not be clean or in the required format for the model which can cause misleading outcomes. In pre-processing of data, we transform data into our required format. It is used to deal with noises, duplicates, and missing values of the dataset. Data pre-processing has the activities like importing datasets, splitting datasets, attribute scaling, etc. Preprocessing of data is required for improving the accuracy of the model.

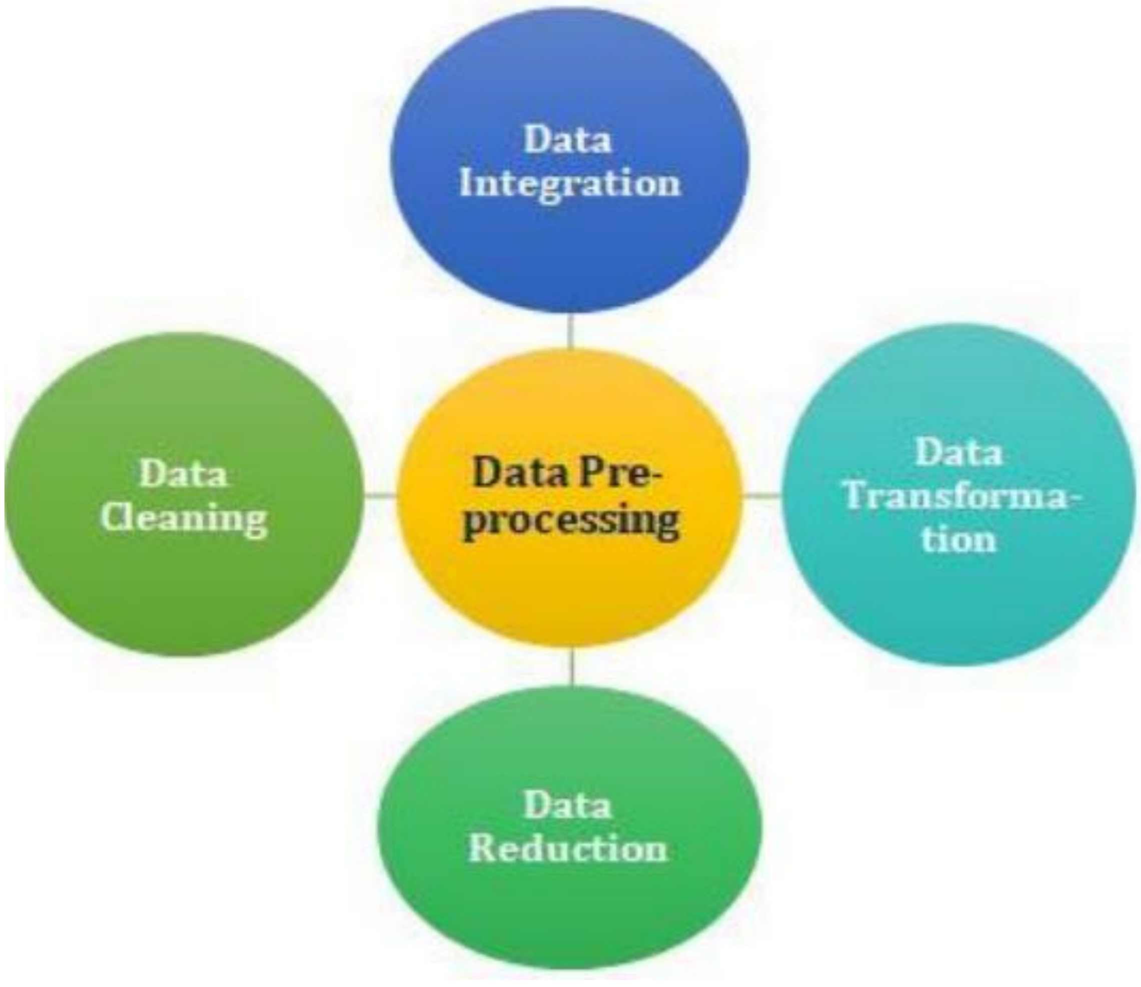


Fig 2 : Data Pre-processing

3.2.4 Model building

After gathering the dataset, the data is ready and can be used to train a machine learning model. We will be using this clean data to train the machine learning eight algorithm .after training the eight models we will be predicting the disease for the input symptoms by combining the predicting of all eight models. This make our overall prediction more robust and accurate.

3.2.5 Prediction of Disease

Various machine learning algorithms like SVM, Naive Bayes, Decision Tree, Random Tree, Logistic Regression, Xg-boost are used for classification. Comparative analysis is performed among algorithms and the algorithm that gives the highest accuracy is used for heart disease prediction.

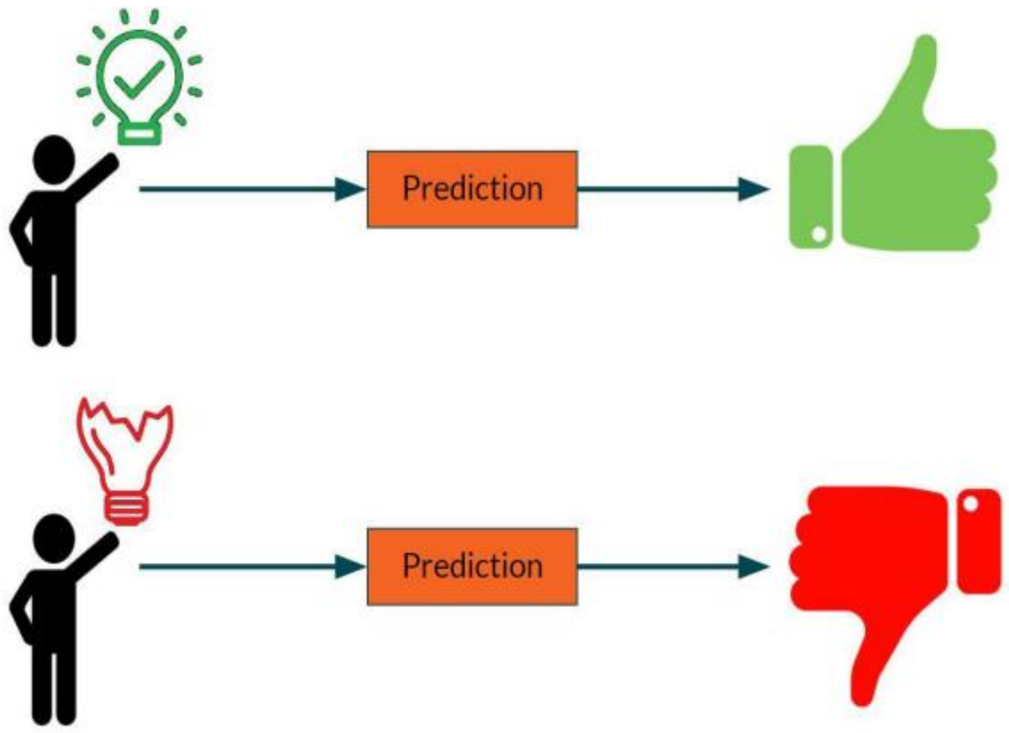
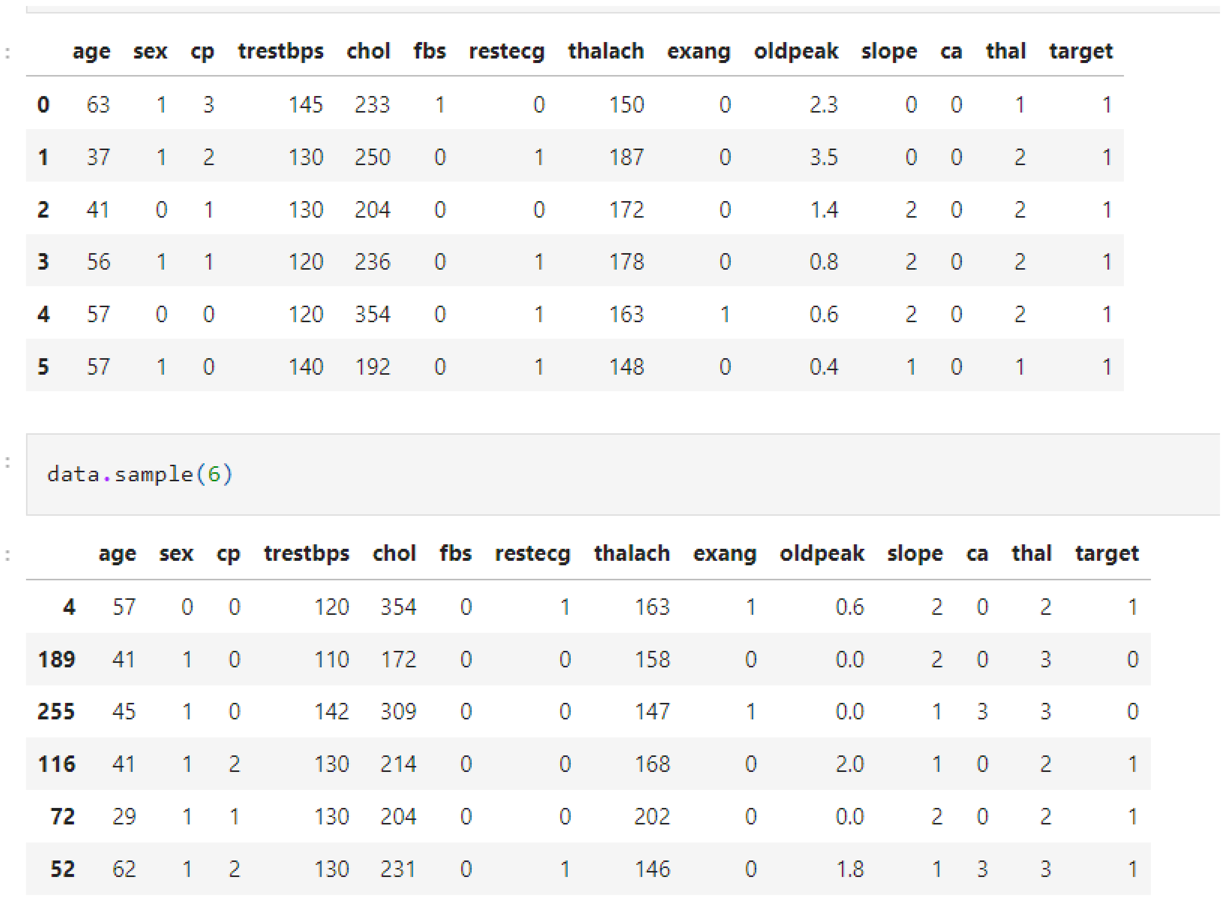


Fig. 3: Prediction of Disease

3.3 DATASET INFORMATION

3.3.1 Few columns of dataset



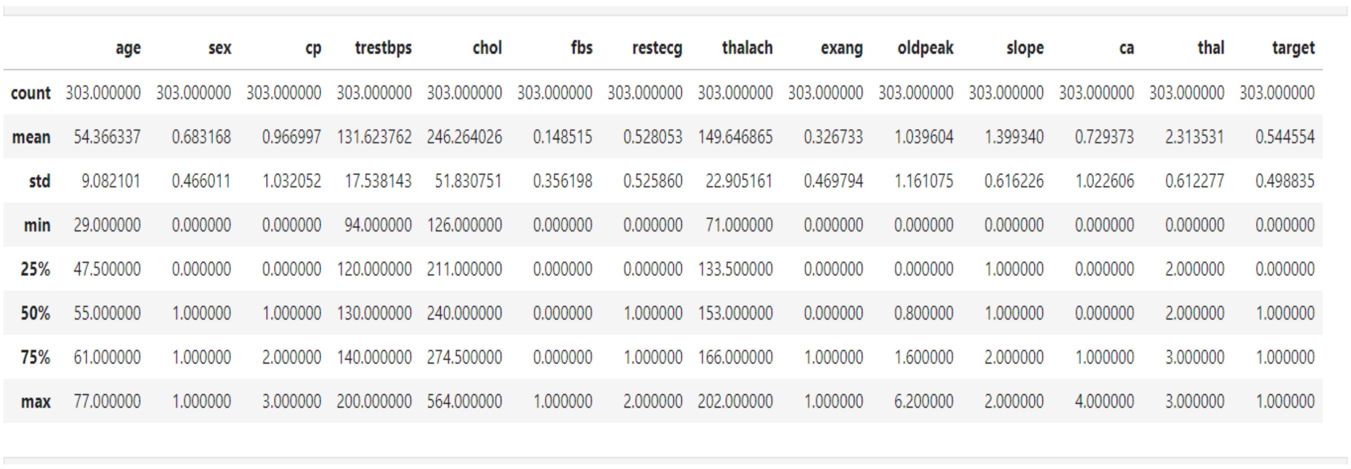
	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
0	63	1	3	145	233	1	0	150	0	2.3	0	0	1	1
1	37	1	2	130	250	0	1	187	0	3.5	0	0	2	1
2	41	0	1	130	204	0	0	172	0	1.4	2	0	2	1
3	56	1	1	120	236	0	1	178	0	0.8	2	0	2	1
4	57	0	0	120	354	0	1	163	1	0.6	2	0	2	1
5	57	1	0	140	192	0	1	148	0	0.4	1	0	1	1

```
data.sample(6)
```

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
4	57	0	0	120	354	0	1	163	1	0.6	2	0	2	1
189	41	1	0	110	172	0	0	158	0	0.0	2	0	3	0
255	45	1	0	142	309	0	0	147	1	0.0	1	3	3	0
116	41	1	2	130	214	0	0	168	0	2.0	1	0	2	1
72	29	1	1	130	204	0	0	202	0	0.0	2	0	2	1
52	62	1	2	130	231	0	1	146	0	1.8	1	3	3	1

Fig. 4 : Dataset Columns

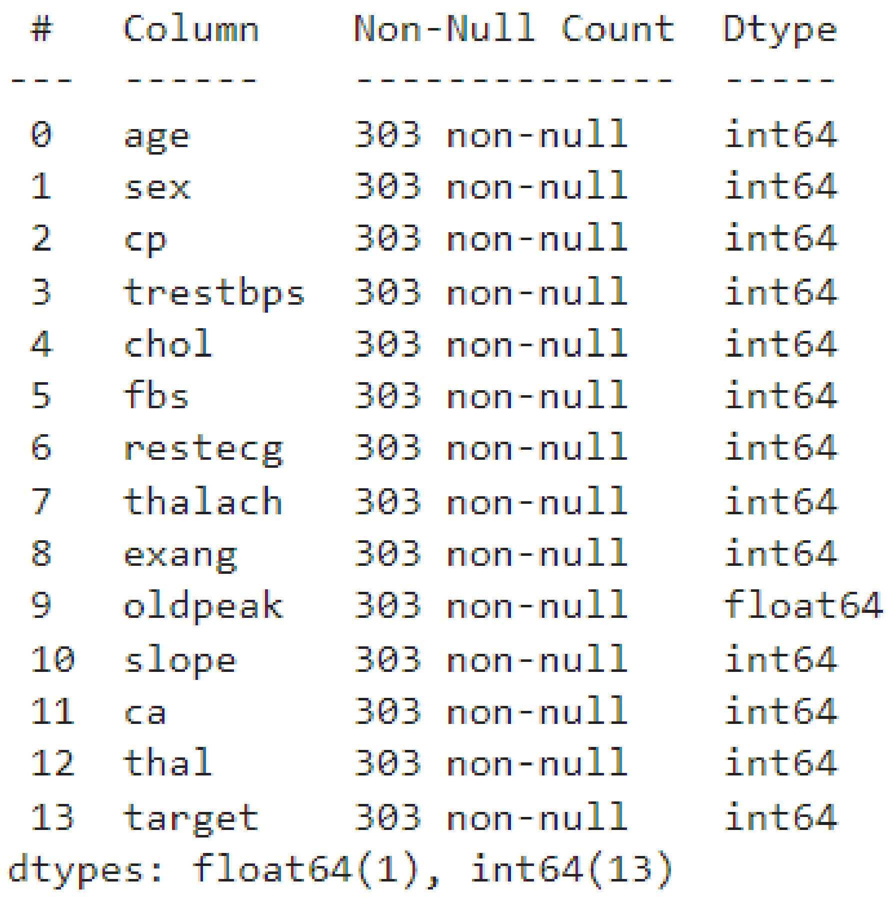
3.3.2 Description of data



	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
count	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000
mean	54.366337	0.683168	0.966997	131.623762	246.264026	0.148515	0.528053	149.646865	0.326733	1.039604	1.399340	0.729373	2.313531	0.544554
std	9.082101	0.466011	1.032052	17.538143	51.830751	0.356198	0.525860	22.905161	0.469794	1.161075	0.616226	1.022606	0.612277	0.498835
min	29.000000	0.000000	0.000000	94.000000	126.000000	0.000000	0.000000	71.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	47.500000	0.000000	0.000000	120.000000	211.000000	0.000000	0.000000	133.500000	0.000000	0.000000	1.000000	0.000000	2.000000	0.000000
50%	55.000000	1.000000	1.000000	130.000000	240.000000	0.000000	1.000000	153.000000	0.000000	0.800000	1.000000	0.000000	2.000000	1.000000
75%	61.000000	1.000000	2.000000	140.000000	274.500000	0.000000	1.000000	166.000000	1.000000	1.600000	2.000000	1.000000	3.000000	1.000000
max	77.000000	1.000000	3.000000	200.000000	564.000000	1.000000	2.000000	202.000000	1.000000	6.200000	2.000000	4.000000	3.000000	1.000000

Fig. 5 : Data Description

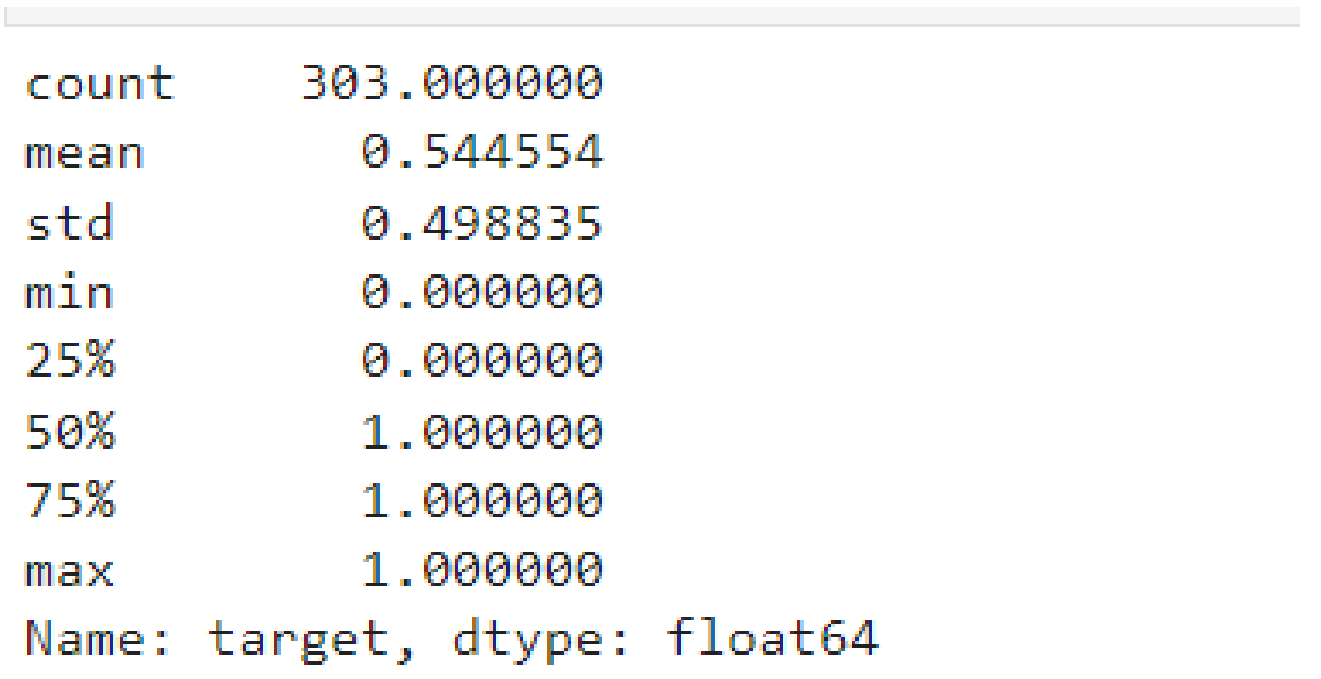
3.3.3. Dataset table structure



```
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   age       303 non-null    int64
 1   sex       303 non-null    int64
 2   cp        303 non-null    int64
 3   trestbps  303 non-null    int64
 4   chol      303 non-null    int64
 5   fbs       303 non-null    int64
 6   restecg   303 non-null    int64
 7   thalach   303 non-null    int64
 8   exang     303 non-null    int64
 9   oldpeak   303 non-null    float64
 10  slope     303 non-null    int64
 11  ca        303 non-null    int64
 12  thal      303 non-null    int64
 13  target    303 non-null    int64
dtypes: float64(1), int64(13)
```

Fig. 6 : Table Structure

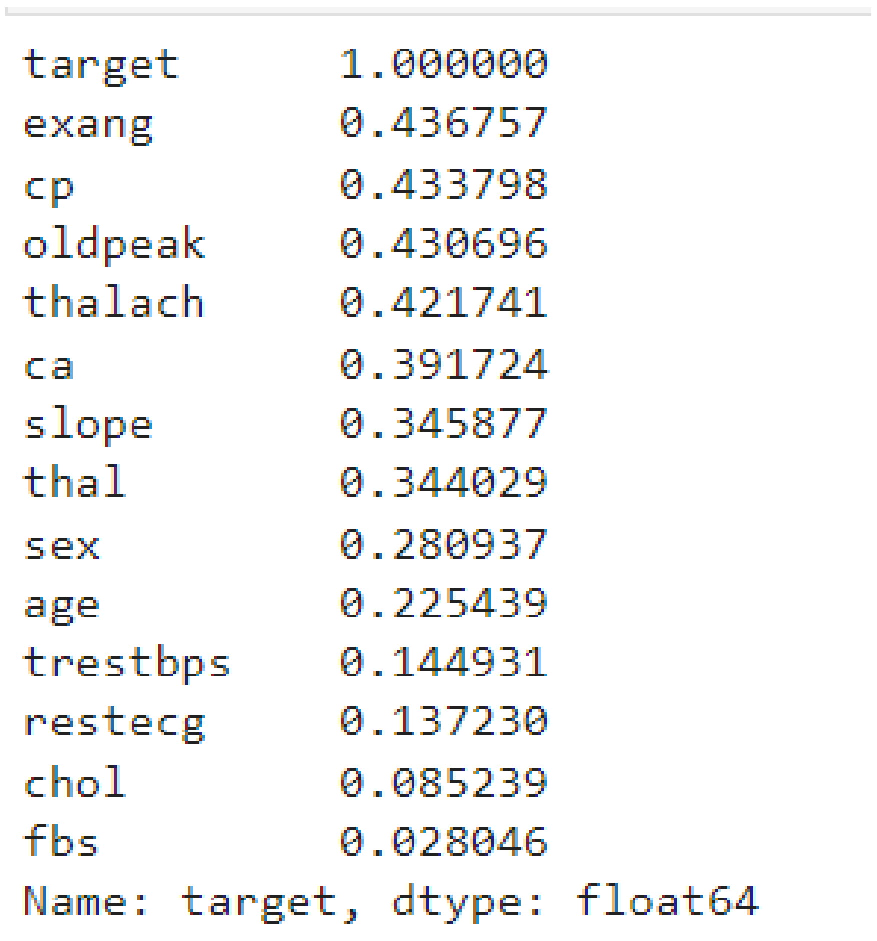
3.3.4 Target description



```
count    303.000000
mean       0.544554
std        0.498835
min        0.000000
25%        0.000000
50%        1.000000
75%        1.000000
max        1.000000
Name: target, dtype: float64
```

Fig. 7 : Target Description

3.3.5 Correlation between columns in the dataset



```
target      1.000000
exang       0.436757
cp          0.433798
oldpeak     0.430696
thalach     0.421741
ca          0.391724
slope       0.345877
thal        0.344029
sex         0.280937
age         0.225439
trestbps    0.144931
restecg     0.137230
chol        0.085239
fbs         0.028046
Name: target, dtype: float64
```

Fig. 8 : Dataset Correlation

This clearly shows that fbs present in the data is weakly correlated while others are closely related.

CHAPTER 4 TECHNOLOGY USED

4.1 PYTHON

Python is a high-level, general-purpose and a very popular programming language.
Python programming language (latest Python 3) is being used in web development, Machine Learning applications, along with all cutting edge technology in Software Indust ry. Python Programming Language is very well suited for Beginners, also for experienced programmers with other programming languages like C++ and Java.
Below are some facts about Python Programming Language:
Python is currently the most widely used multi-purpose, high-level programming language.
Python allows programming in Object-Oriented and Procedural paradigms.Python progra ms generally are smaller than other programming languages like Java.
Programmers have to type relatively less and indentation requirement of the language, makes them readable all the time.Python language is being used by almost all tech-giant c ompanies like – Google, Amazon, Facebook, Instagram, Dropbox, Uber… etc.
The biggest strength of Python is huge collection of standard library which can be used for the following:
Machine Learning
GUI Applications (like Kivy, Tkinter, PyQt etc.)
Web frameworks like Django (used by YouTube, Instagram, Dropbox)
Image processing (like OpenCV, Pillow)
Web scraping (like Scrapy, BeautifulSoup, Selenium)
Test frameworks
Multimedia
Scientific computing
Text processing and many more..

4.2 MACHINE LEARNING

Machine Learning is the field of study that gives computers the capability to learn without being explicitly programmed. ML is one of the most exciting technologies that one would have ever come across. As it is evident from the name, it gives the computer that makes it more similar to humans: *The ability to learn*. Machine learning is actively being used today, perhaps in many more places than one would expect.

With the help of sample historical data, which is known as training data, machine learning algorithms build a mathematical model that helps in making predictions or decisions with out being explicitly programmed. Machine learning brings computer science and statistics together for creating predictive models. Machine learning constructs or uses the algorithm s that learn from historical data. The more we will provide the information, the higher will be the performance.

4.2.1 Supervised Machine Learning

Supervised learning is a type of machine learning method in which we provide sample labeled data to the machine learning system in order to train it, and on that basis, it predicts the output.

The system creates a model using labeled data to understand the datasets and learn about each data, once the training and processing are done then we test the model by providing a sample data to check whether it is predicting the exact output or not.

The goal of supervised learning is to map input data with the output data. The supervised learning is based on supervision, and it is the same as when a student learns things in the supervision of the teacher. The example of supervised learning is **spam filtering**.

4.2.2 Unsupervised Machine Learning

Unsupervised learning is a learning method in which a machine learns without any

supervision.
The training is provided to the machine with the set of data that has not been labeled, class ified, or categorized, and the algorithm needs to act on that data without any supervision. The goal of unsupervised learning is to restructure the input data into new features or a gr oup of objects with similar patterns.

In unsupervised learning, we don't have a predetermined result. The machine tries to find useful insights from the huge amount of data. It can be further classifieds into two categor ies of algorithms:
Clustering
Association

4.2.3 Reinforcement Machine Learning

Reinforcement learning is a feedback-based learning method, in which a learning agent gets a reward for each right action and gets a penalty for each wrong action. The agent learns automatically with these feedbacks and improves its performance. In reinforcement learning, the agent interacts with the environment and explores it. The goal of an agent is to get the most reward points, and hence, it improves its performance.

The robotic dog, which automatically learns the movement of his arms, is an example of Reinforcement learning.

CHAPTER 5 FEATURES ANALYSIS

5.1 "sex" Feature Analysis

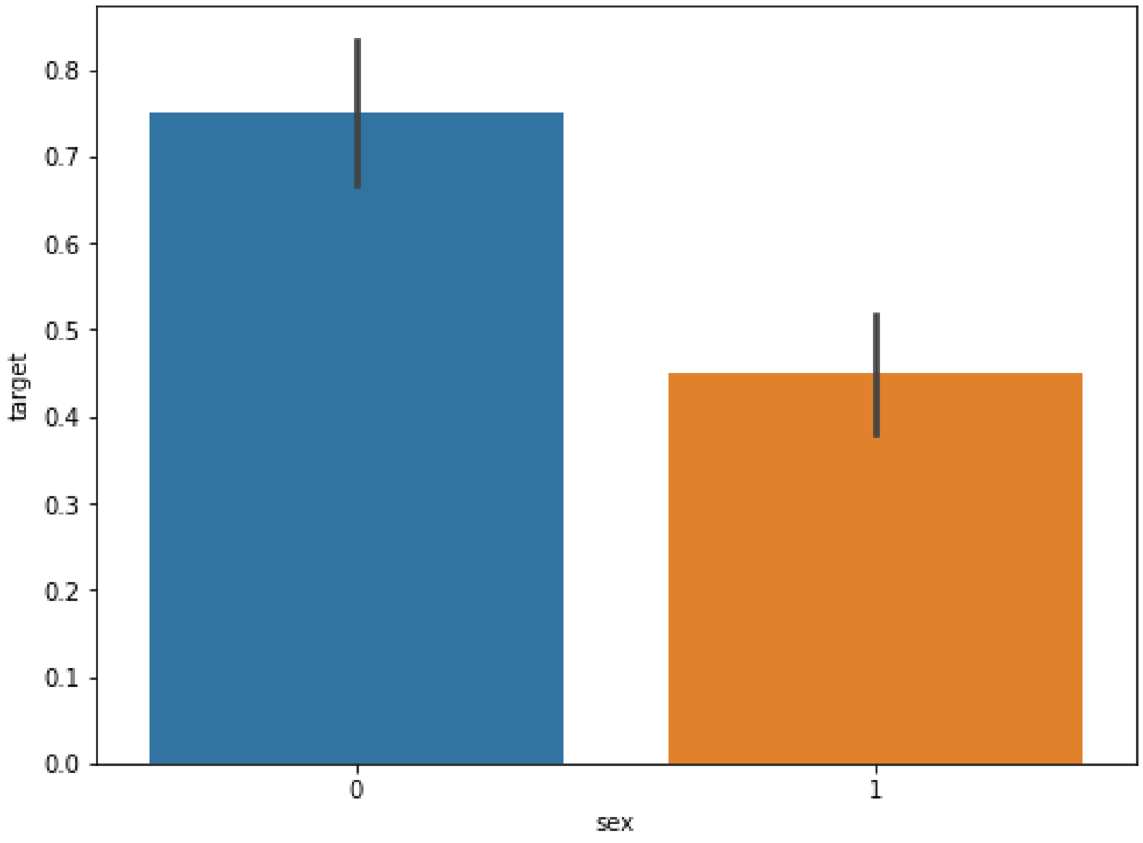


Fig. 9 : "sex" analysis

This features shows that sex of type 0 i.e females are more likely to have problem in comparison to males.

5.2 Chest Pain “cp” Feature Analysis

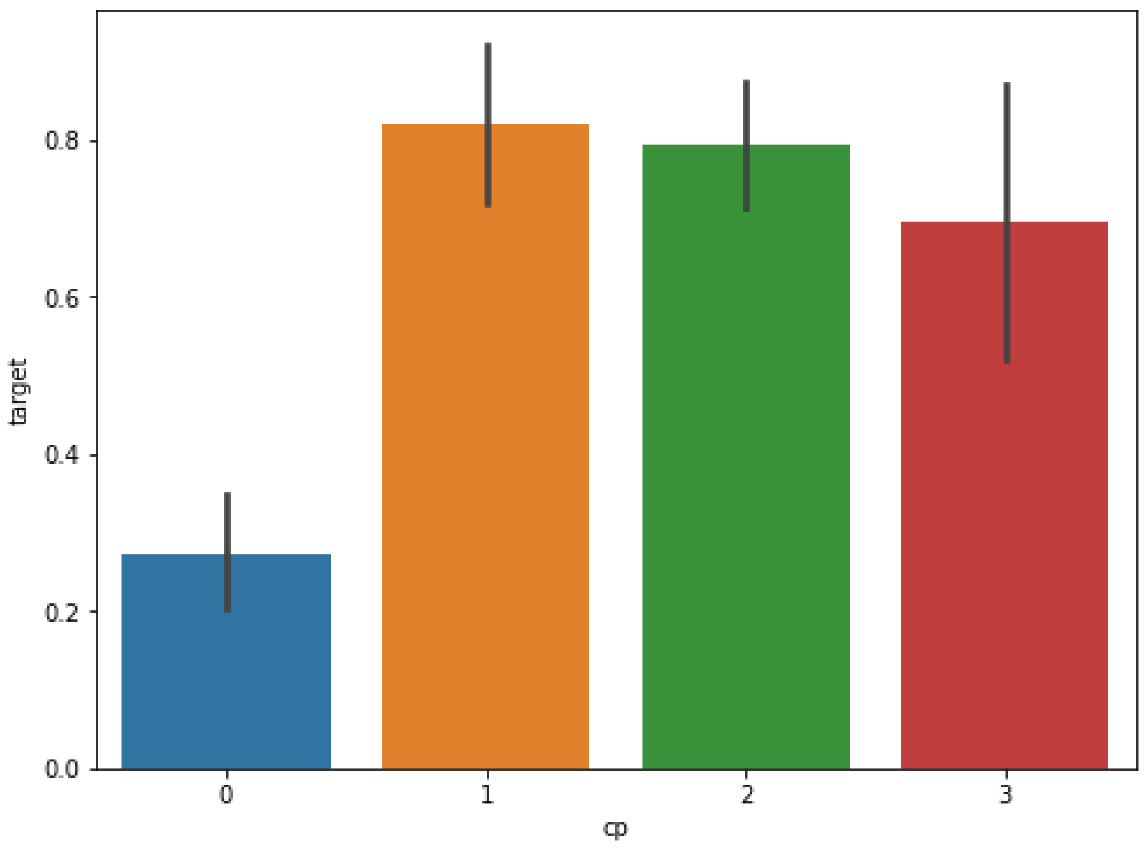


Fig. 10 : “cp” analysis

We notice, that chest pain of '0', i.e. the ones with typical angina are much less likely to have heart problems

5.3 Fasting Blood Sugar “fbs” Feature Analysis

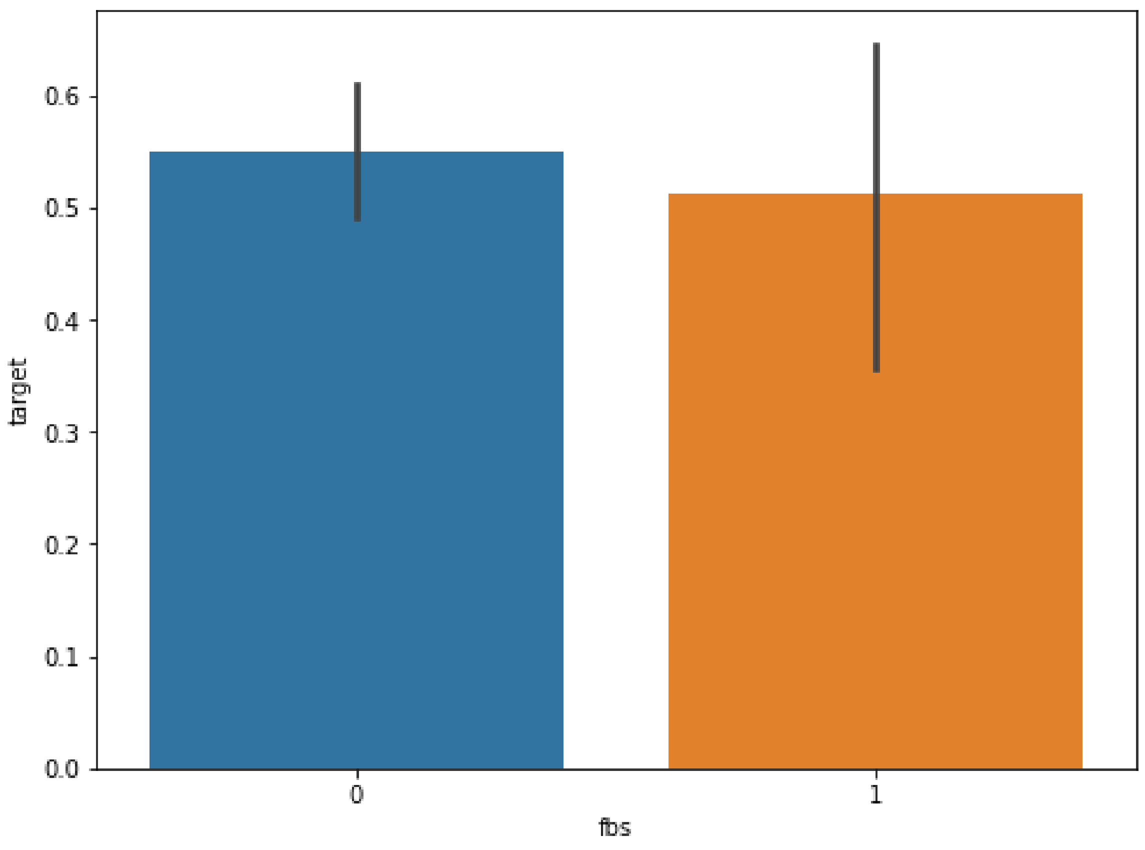


Fig. 11 : “fbs ” analysis

This feature does not contribute much, there is nothing extraordinary here.

5.4 Resting Electrocardiography “restecg” Feature Analysis

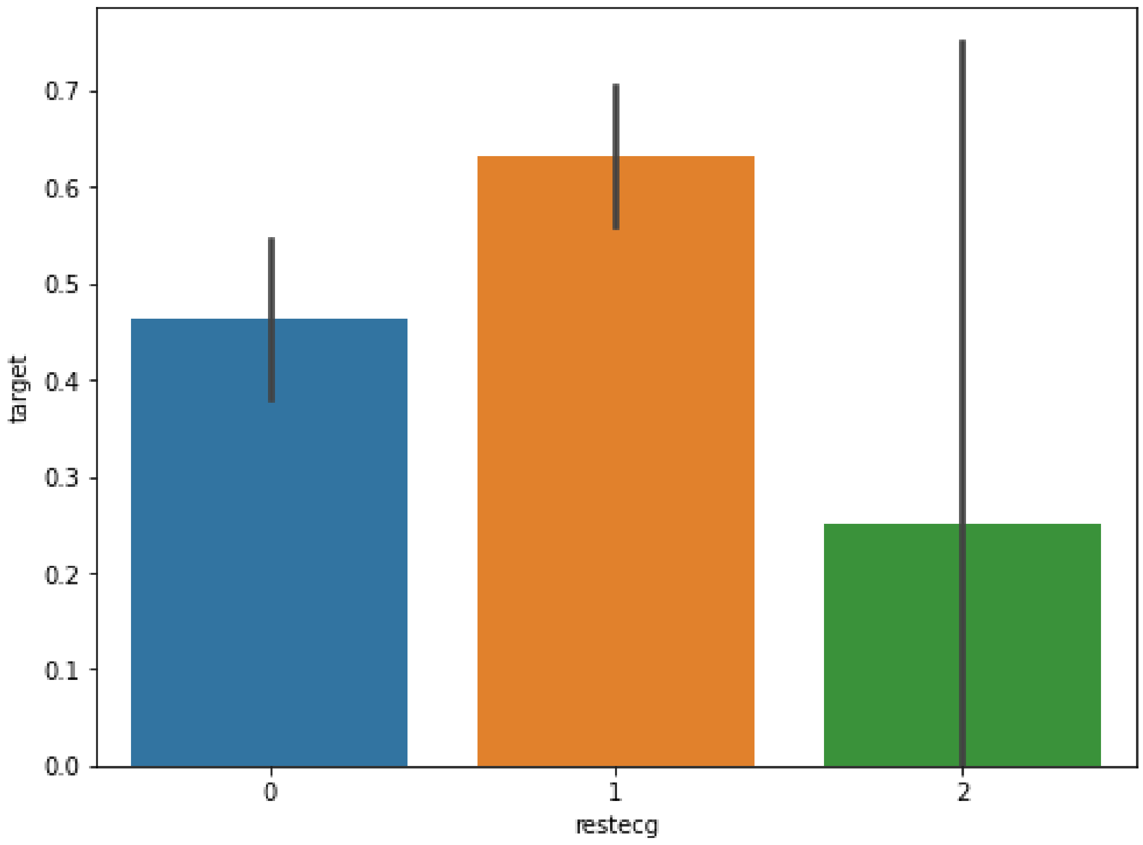


Fig. 12 : “restecg” analysis

We realize that people with restecg '1' and '0' are much more likely to have a heart disease than with restecg '2'

5.5 Exercise Induced Angina “exang” Feature Analysis

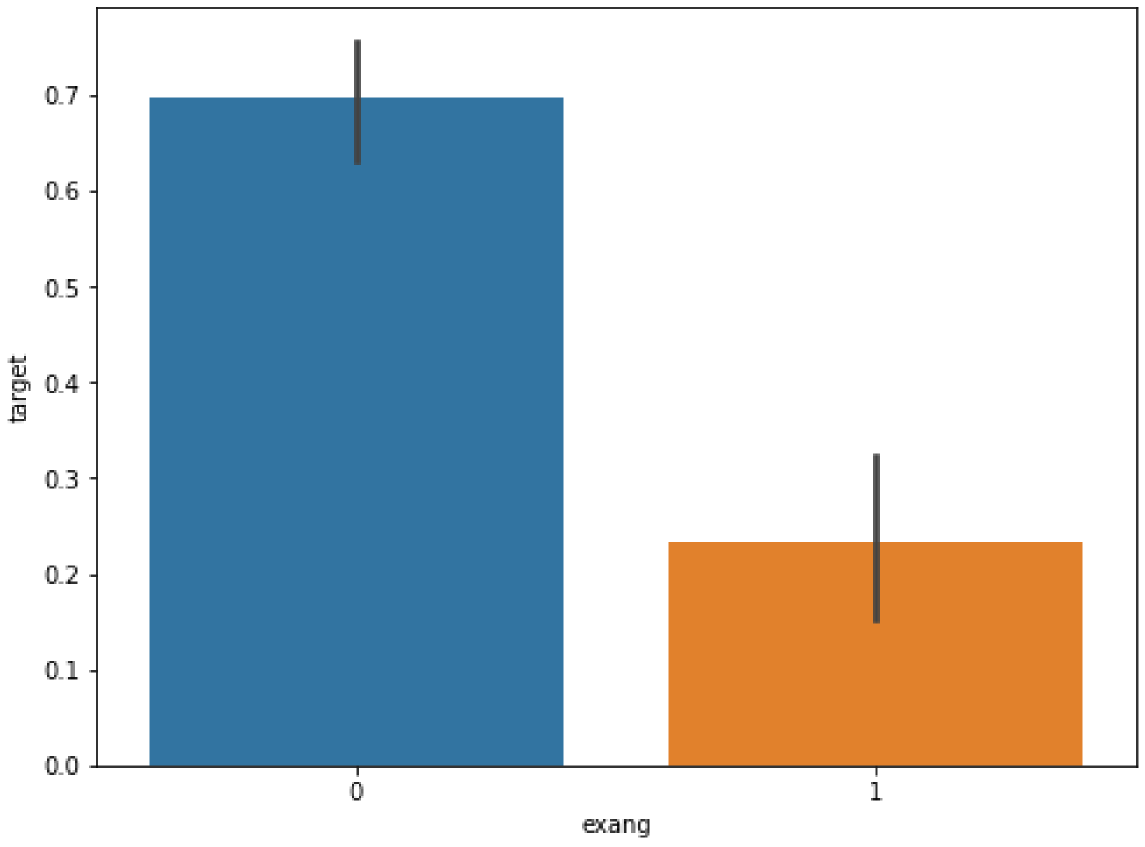


Fig. 13 : “exang” analysis

People with exang=1 i.e. Exercise induced angina are much less likely to have heart problems

5.6 Slope Feature Analysis

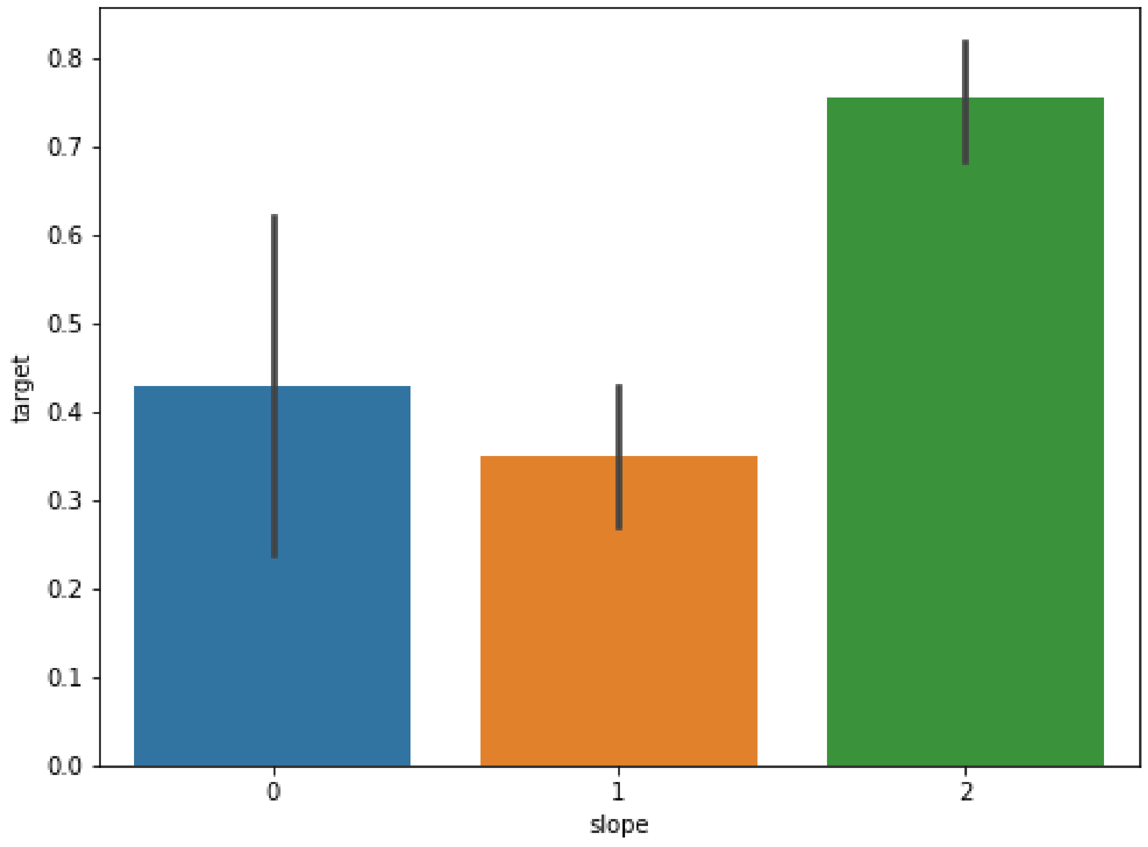


Fig . 14 : slope analysis

We observe, that Slope '2' causes heart pain much more than Slope '0' and '1'

5.7 “ca” Feature Analysis

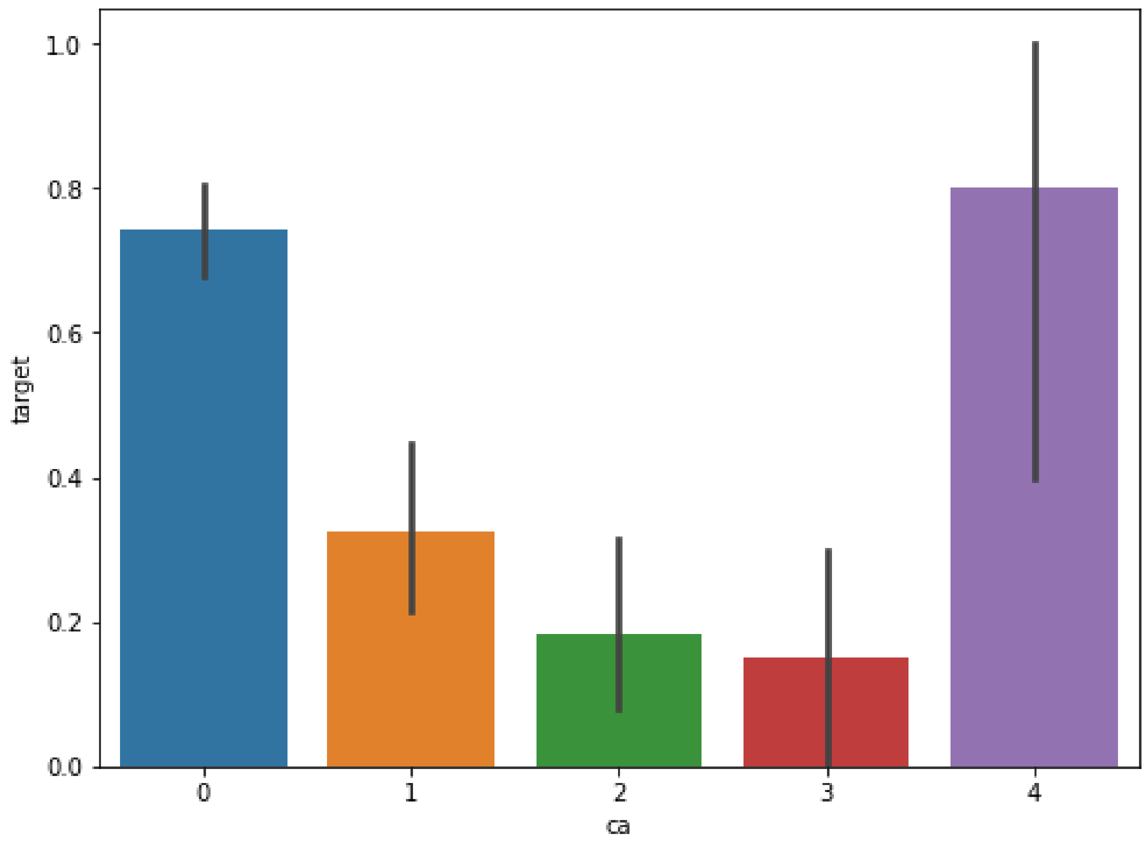


Fig. 15 : “ca” analysis

ca=4 has astonishingly large number of heart patients

5.8 “thal” Feature Analysis

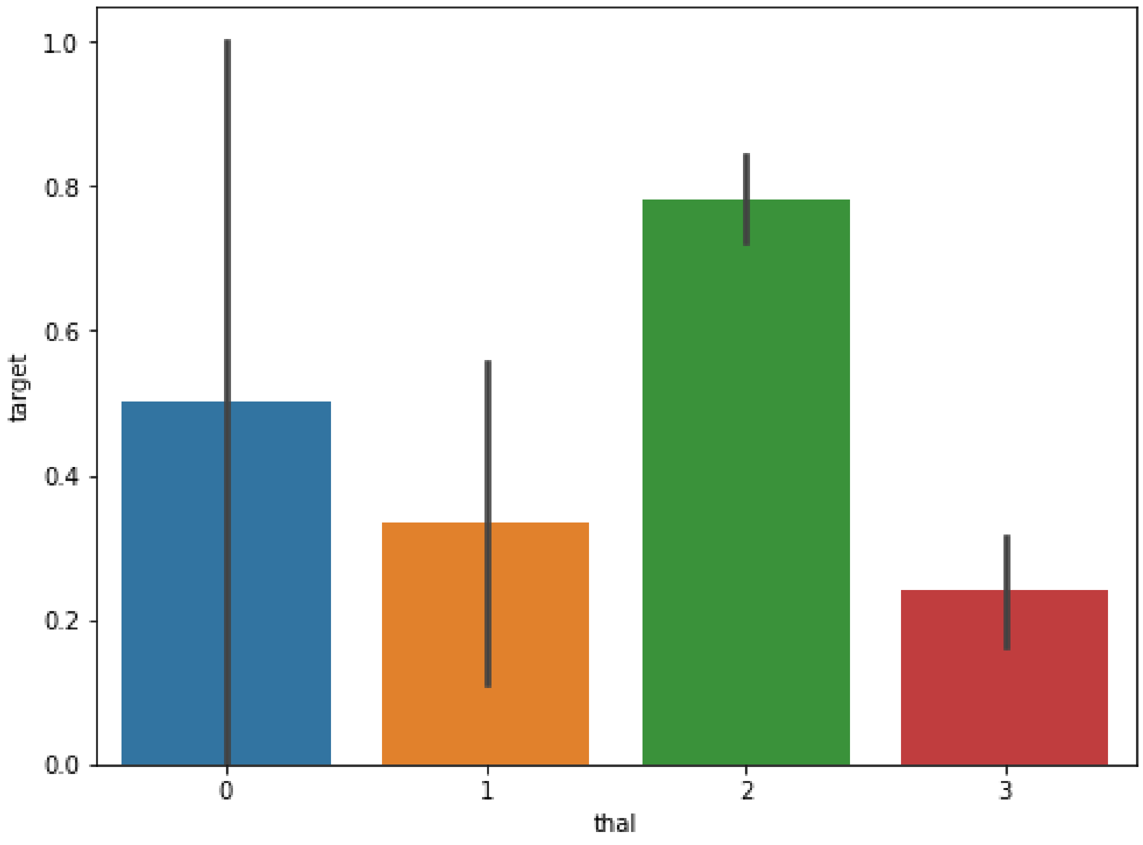


Fig. 16 : “thal” analysis

Thal =3 has less number of heart disease patients.

CHAPTER 6 ALGORITHMS USED

6.1 SUPPORT VECTOR MACHINE (SVM)

Support Vector Machine or SVM is one of the most popular Supervised Learning algorithms, which is used for Classification as well as Regression problems. However, primarily, it is used for Classification problems in Machine Learning.
The goal of the SVM algorithm is to create the best line or decision boundary that can segregate n-dimensional space into classes so that we can easily put the new data point in the correct category in the future. This best decision boundary is called a hyperplane.SVM chooses the extreme points/vectors that help in creating the hyperplane. These extreme cases are called support vectors, and hence the algorithm is termed as Support Vector Machine.

Support vector machines (SVMs) are powerful yet flexible supervised machine learning algorithms which are used both for classification and regression. But generally, they are used in classification problems. In the 1960s, SVMs were first introduced but later they got refined in 1990. SVMs have their unique way of implementation as compared to other machine learning algorithms. Lately, they are extremely popular because of their ability to handle multiple continuous and categorical variables.

The followings are important concepts in SVM –

Support Vectors - Data Points that are closest to the hyperplane are called support vectors. Separating line will be defined with the help of these data points.

Hyperplane - As we can see in the above diagram, it is a decision plane or space which is divided between a set of objects having different classes.

Margin - It may be defined as the gap between two lines on the closest data points of different classes. It can be calculated as the perpendicular distance from the line to the 12 support vectors. Large margin is considered as a good margin and small margin is considered as a bad margin.

Types of SVM:

SVM can be of two types:

● **Linear SVM**: Linear SVM is used for linearly separable data, which means if a dataset can be classified into two classes by using a single straight line, then such data is termed as linearly separable data, and classifier is used called as Linear SVM classifier.

● **Non-linear SVM:** Non-Linear SVM is used for non-linearly separated data, which means if a dataset cannot be classified by using a straight line, then such data is termed as non-linear data and classifier used is called as Non-linear SVM classifier.

The objective of the support vector machine algorithm is to find a hyperplane in an N-Dimensional space (N - the number of features) that distinctly classifies the data points.

The advantages of support vector machines are:

● Effective in high dimensional spaces.

● Still effective in cases where the number of dimensions is greater than the number of samples.

● Uses a subset of training points in the decision function (called support vectors), so it is also memory efficient.

● Versatile: different kernel functions can be specified for the decision function. Common kernels are provided, but it is also possible to specify custom kernels. The disadvantages of support vector machines include:

● If the number of features is much greater than the number of samples, avoid over-fitting in choosing Kernel functions and regularization term is crucial. SVMs do not directly provide probability estimates, these are calculated using an expensive five-fold cross-validation.

The disadvantages of support vector machines include:

● If the number of features is much greater than the number of samples, avoid over-fitting in choosing Kernel functions and regularization term is crucial. SVMs do not directly provide probability estimates, these are calculated using an expensive five-fold cross-validation.

Function used :

```
from sklearn import svm

sv = svm.SVC(kernel='linear')

sv.fit(X_train, Y_train)

Y_prediction_svm = sv.predict(X_test)
Y_prediction_svm.shape
score_of_svm = round(accuracy_score(Y_prediction_svm,Y_test)*100,2)
print("The accuracy score achieved using Linear SVM is: "+str(score_of_svm)+" %")
```

The accuracy score achieved using Linear SVM is: 81.97 %

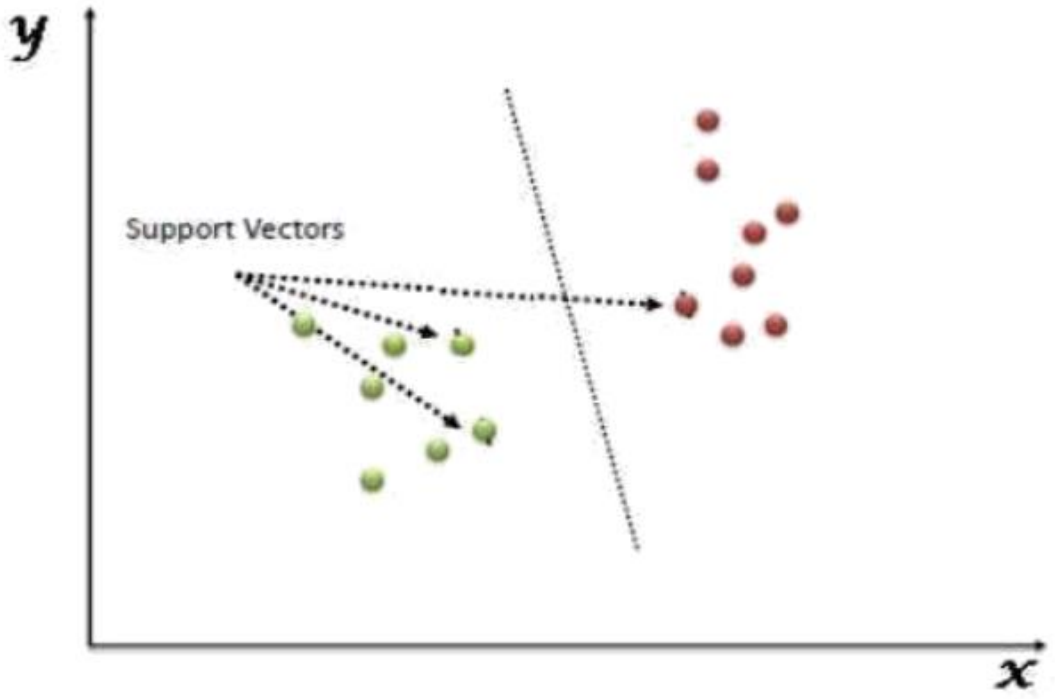


Fig. 17 :Support Vector Machine

6.2 NAIVE BAYES

Naive Bayes algorithm is a supervised learning algorithm, which is based on Bayes theorem and used for solving classification problems. It is mainly used in text classification that includes a high-dimensional training dataset.

Naive Bayes Classifier is one of the simple and most effective Classification algorithms which helps in building the fast machine learning models that can make quick predictions.

It is a probabilistic classifier, which means it predicts on the basis of the probability of an object. Some popular examples of Naïve Bayes Algorithm are spam filtration, Sentimental analysis, and classifying articles.
It is a classification technique based on Bayes' Theorem with an assumption of independence among predictors. In simple terms, a Naive Bayes classifier assumes that the presence of a particular feature in a class is unrelated to the presence of any other feature.
The Naive Bayes model is easy to build and particularly useful for very large data sets. Along with simplicity, Naive Bayes is known to outperform even highly sophisticated classification methods.

The Naive Bayes algorithm is comprised of two words Naive and Bayes, Which can be described as:

● **Naive**: It is called Naive because it assumes that the occurrence of a certain feature is independent of the occurrence of other features. Such as if the fruit is identified on the basis of color, shape, and taste, then red, spherical, and sweet fruit is recognized as an apple. Hence each feature individually contributes to identify that it is an apple without depending on each other.

● **Bayes**: It is called Bayes because it depends on the principle of Bayes' Theorem.

Bayes’s theorem

Bayes' theorem is also known as Bayes' Rule or Bayes' law, which is used to determine the probability of a hypothesis with prior knowledge. It depends on the conditional probability.

The formula for Bayes' theorem is given as:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

Where,
P(A|B) is Posterior probability:Probability of hypothesis A on the observed event B.

P(B|A) is Likelihood probability: Probability of the evidence given that the probability of a hypothesis is true.

P(A) is Prior Probability: Probability of hypothesis before observing the evidence.

P(B) is Marginal Probability: Probability of Evidence.

Types of Naive Bayes model:

There are three types of Naive Bayes Model, which are given below:

• **Gaussian**: The Gaussian model assumes that features follow a normal distribution. This means if predictors take continuous values instead of discrete, then the model assumes that these values are sampled from the Gaussian distribution.

• **Multinomial**: The Multinomial Naïve Bayes classifier is used when the data is multinomial distributed. It is primarily used for document classification problems, it means a particular document belongs to which category such as Sports, Politics, education, etc. The classifier uses the frequency of words for the predictors.

• **Bernoulli:** The Bernoulli classifier works similar to the Multinomial classifier, but the predictor variables are the independent Booleans variables. Such as if a particular word is present or not in a document. This model is also famous for document classification tasks.

Function Used:

```
from sklearn.naive_bayes import GaussianNB

gaussianb = GaussianNB()

gaussianb.fit(X_train,Y_train)

Y_prediction_gaussianb = gaussianb.predict(X_test)
score_of_gaussianb = round(accuracy_score(Y_prediction_gaussianb,Y_test)*100,2)

print("The accuracy score achieved using Naive Bayes is: "+str(score_of_gaussianb)+" %")
```

```
The accuracy score achieved using Naive Bayes is: 85.25 %
```

6. 3 DECISION TREE

Decision Tree is a Supervised learning technique that can be used for both classification and regression problems, but mostly it is preferred for solving classification problems. It is a tree-structured classifier, where internal nodes represent the features of a dataset, branches represent the decision rules and each leaf node represents the outcome. In a Decision Tree, there are two nodes, which are the Decision Node and Leaf Node.

Decision nodes are used to make any decision and have multiple branches, whereas Leaf nodes are the output of those decisions and do not contain any further branches. The decisions or the test are performed on the basis of features of the given dataset. It is a graphical representation for getting all the possible solutions to a problem/decision based on given conditions. It is called a Decision Tree because, similar to a tree, it starts with the root node, which expands on further branches and constructs a tree-like structure. In order to build a tree, we use the CART algorithm, which stands for Classification and Regression Tree algorithm. A Decision Tree simply asks a question, and based on the answer (Yes/No), it further split the tree into subtrees.

The Decision Tree Algorithm belongs to the family of supervised machine learning algorithms. It can be used for both a classification problem as well as for a regression problem.
The goal of this algorithm is to create a model that predicts the value of a target variable, for which the decision tree uses the tree representation to solve the problem in which the leaf node corresponds to a class label and attributes are represented on the internal node of the tree.
There are various algorithms in Machine learning, so choosing the best algorithm for the given dataset and problem is the main point to remember while creating a machine learning model. Below are the two reasons for using the Decision

Tree:

• Decision Trees usually mimic human thinking ability while making a decision, so it is easy to understand.

• The logic behind the decision tree can be easily understood because it shows a tree-like structure.
In Decision Tree the major challenge is to identify the attribute for the root node in each level. This process is known as attribute selection. We have two popular attribute selection measures:

1. Information Gain: When we use a node in a Decision Tree to partition the training instances into smaller subsets, the entropy changes. Information gain is a measure of this change in entropy.
Entropy is the measure of uncertainty of a random variable, it characterizes the impurity of an arbitrary collection of examples. The higher the entropy the more the information content.
2. Gini Index: Gini Index is a metric to measure how often a randomly chosen element would be incorrectly identified. It means an attribute with lower Gini index should be preferred. Sklearn supports “Gini” criteria for Gini Index and by default, it takes “gini” value.
Working:
In a Decision Tree, for predicting the class of the given dataset, the algorithm starts from the root node of the tree. This algorithm compares the values of the root attribute with the record (real dataset) attribute and, based on the comparison, follows the branch and jumps to the next node.

For the next node, the algorithm again compares the attribute value with the other sub-nodes and moves further. It continues the process until it reaches the leaf node of the tree. The complete process can be better understood using the below algorithm:
● Step-1: Begin the tree with the root node, says S, which contains the complete dataset.
● Step-2: Find the best attribute in the dataset using Attribute Selection Measure (ASM).
● Step-3: Divide the S into subsets that contains possible values for the best attributes.
● Step-4: Generate the Decision Tree node, which contains the best attribute.
● Step-5: Recursively make new decision trees using the subsets of the dataset created in step -3. Continue this process until a stage is reached where you cannot further classify the nodes and call the final node as a leaf node.

Function Used:

```
from sklearn.tree import DecisionTreeClassifier

maximum_accuracy = 0

for x in range(200):
    decisiontree = DecisionTreeClassifier(random_state=x)
    decisiontree.fit(X_train,Y_train)
    Y_prediction_decisiontree = decisiontree.predict(X_test)
    current_accuracy = round(accuracy_score(Y_prediction_decisiontree,Y_test)*100,2)
    if(current_accuracy>maximum_accuracy):
        maximum_accuracy = current_accuracy
        best_x = x

#print(max_accuracy)
#print(best_x)

decisiontree = DecisionTreeClassifier(random_state=best_x)
decisiontree.fit(X_train,Y_train)
Y_prediction_decisiontree = decisiontree.predict(X_test)
print(Y_prediction_decisiontree.shape)
score_of_decisiontree = round(accuracy_score(Y_prediction_decisiontree,Y_test)*100,2)

print("The accuracy score achieved using Decision Tree is: 
"+str(score_of_decisiontree)+" %")
```

The accuracy score achieved using Decision Tree is: 81.97 %

6.4 RANDOM FOREST

Random Forest is a supervised learning algorithm. It is an extension of machine learning classifiers which include the bagging to improve the performance of Decision Tree. It combines tree predictors, and trees are dependent on a random vector which is independently sampled. The distribution of all trees are the same. Random Forests splits nodes using the best among of a predictor subset that are randomly chosen from the node itself, instead of splitting nodes based on the variables. The time complexity of the worst case of learning with Random Forests is O(M(dnlogn)) , where M is the number of growing trees, n is the number of instances, and d is the data dimension.

It can be used both for classification and regression. It is also the most flexible and easy to use algorithm. A forest consists of trees. It is said that the more trees it has, the more robust a forest is. Random Forests create Decision Trees on randomly selected data samples, get predictions from each tree and select the best solution by means of voting. It also provides a pretty good indicator of the feature importance.

Random Forests have a variety of applications, such as recommendation engines, image classification and feature selection. It can be used to classify loyal loan applicants, identify fraudulent activity and predict diseases. It lies at the base of the Boruta algorithm, which selects important features in a dataset.
Random Forest is a popular machine learning algorithm that belongs to the supervised learning technique. It can be used for both Classification and Regression problems in ML. It is based on the concept of ensemble learning, which is a process of combining multiple classifiers to solve a complex problem and to improve the performance of the model.

As the name suggests, "Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset." Instead of relying on one decision tree, the random forest takes the prediction from each tree and based on the majority votes of predictions, and it predicts the final output.

The greater number of trees in the forest leads to higher accuracy and prevents the problem of overfitting.

Assumptions:

Since the random forest combines multiple trees to predict the class of the dataset, it is possible that some decision trees may predict the correct output, while others may not. But together, all the trees predict the correct output. Therefore, below are two assumptions for a better Random forest classifier:

● There should be some actual values in the feature variable of the dataset so that the classifier can predict accurate results rather than a guessed result.

● The predictions from each tree must have very low correlations.

Algorithm Steps:

It works in four steps:

● Select random samples from a given dataset.

● Construct a Decision Tree for each sample and get a prediction result from each Decision Tree.

● Perform a vote for each predicted result.

● Select the prediction result with the most votes as the final prediction.

Advantages:

● Random Forest is capable of performing both Classification and Regression tasks.

● It is capable of handling large datasets with high dimensionality.

● It enhances the accuracy of the model and prevents the overfitting issue.

Disadvantages:

Although Random Forest can be used for both classification and regression tasks, it is not more suitable for Regression tasks.

Function Used:

```
from sklearn.ensemble import RandomForestClassifier

maximum_accuracy = 0


for x in range(2000):
    random_forest = RandomForestClassifier(random_state=x)
    random_forest.fit(X_train,Y_train)
    Y_prediction_random_forest = random_forest.predict(X_test)
    current_accuracy = round(accuracy_score(Y_prediction_random_forest,Y_test)*100,2)
    if(current_accuracy>maximum_accuracy):
        maximum_accuracy = current_accuracy
        best_x = x

#print(max_accuracy)
#print(best_x)

random_forest = RandomForestClassifier(random_state=best_x)
random_forest.fit(X_train,Y_train)
Y_prediction_random_forest = random_forest.predict(X_test)
Y_prediction_random_forest.shape
score_of_random_forest =
round(accuracy_score(Y_prediction_random_forest,Y_test)*100,2)

print("The accuracy score achieved using Decision Tree is:
"+str(score_of_random_forest)+" %")
```

The accuracy score achieved using Random Forest is: 95.16 %

6.5 LOGISTIC REGRESSION

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 itsweight, 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.

Advantages:

Logistic Regression is one of the simplest machine learning algorithms and is easy to implement yet provides great training efficiency in some cases. Also due to these reasons, training a model with this algorithm doesn't require high computation power.
The predicted parameters (trained weights) give inference about the importance of each feature. The direction of association i.e. positive or negative is also given. So we can use Logistic Regression to find out the relationship between the features.
This algorithm allows models to be updated easily to reflect new data, unlike Decision Tree or Support Vector Machine. The update can be done using stochastic gradient descent.

Logistic Regression outputs well-calibrated probabilities along with classification results. This is an advantage over models that only give the final classification as results. If a training example has a 95% probability for a class, and another has a 55% probability for the same class, we get an inference about which training examples are more accurate for the formulated problem.

Disadvantages:

Logistic Regression is a statistical analysis model that attempts to predict precise probabilistic outcomes based on independent features. On high dimensional datasets, this may lead to the model being over-fit on the training set, which means overstating the accuracy of predictions on the training set and thus the model may not be able to predict accurate results on the test set. This usually happens in the case when the model is trained on little training data with lots of features. So on high dimensional datasets, Regularization techniques should be considered to avoid overfitting (but this makes the model complex). Very high regularization factors may even lead to the model being under-fit on the training data.
Non linear problems can't be solved with logistic regression since it has a linear decision surface. Linearly separable data is rarely found in real world scenarios. So the transformation of non linear features is required which can be done by increasing the number of features such that the data becomes linearly separable in higher dimensions.

Non-Linearly Separable Data:

It is difficult to capture complex relationships using logistic regression. More powerful and complex algorithms such as Neural Networks can easily outperform this algorithm .

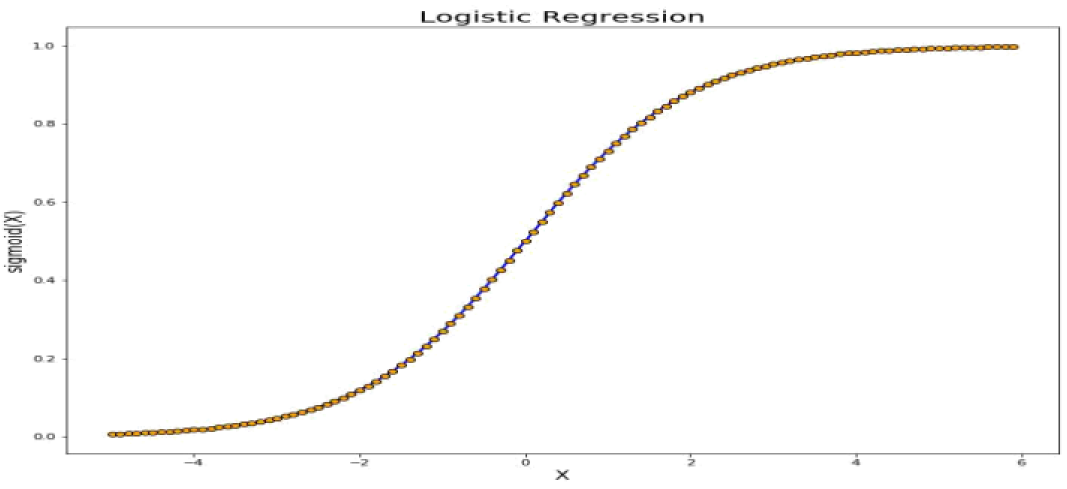


Fig. 18 : Logistic Regression

FUNCTION USED

```
from sklearn.linear_model import LogisticRegression

logis = LogisticRegression()

logis.fit(X_train,Y_train)

Y_prediction_logis = logis.predict(X_test)
Y_prediction_logis.shape
score_of_logis = round(accuracy_score(Y_prediction_logis,Y_test)*100,2)

print("The accuracy score achieved using Logistic Regression is: "+str(score_of_logis)+"
%")
```

The accuracy score achieved using Logistic Regression is: 85.25 %

6.6 XGBOOST

XG-boost is an implementation of Gradient Boosted decision trees. It is a type of Software library that was designed basically to improve speed and model performance. In this algorithm, decision trees are created in sequential form. Weights play an important role in XG-boost. Weights are assigned to all the independent variables which are then fed into the decision tree which predicts results. Weight of variables predicted wrong by the tree is increased and these the variables are then fed to the second decision tree. These individual classifiers/predictors then assemble to give a strong and more precise model. It can work on regression, classification, ranking, and user-defined predict.

1. Regularization: XG-boost has in-built L1 (Lasso Regression) and L2 (Ridge Regression) regularization which prevents the model from overfitting. That is why, XG-boost is also called regularized form of GBM (Gradient Boosting Machine). While using Scikit Learn libarary, we pass two hyper-parameters (alpha and lambda) to XG-boost related to regularization. alpha is used for L1 regularization and lambda is used for L2 regularization.

2. Parallel Processing: XG-boost utilizes the power of parallel processing and that is why it is much faster than GBM. It uses multiple CPU cores to execute the model. While using Scikit Learn libarary, nthread hyper-parameter is used for parallel processing. nthread represents number of CPU cores to be used. If you want to use all the available cores, don't mention any value for nthread and the algorithm will detect automatically.

3. Handling Missing Values: XG-boost has an in-built capability to handle missing values. When XG-boost encounters a missing value at a node, it tries both the left and right hand split and learns the way leading to higher loss for each node. It then does the same when working on the testing data.

4. Cross Validation: XG-boost allows user to run a cross-validation at each iteration of the boosting process and thus it is easy to get the exact optimum number of boosting iterations in a single run. This is unlike GBM where we have to run a grid-search and only a limited values can be tested.

5. Effective Tree Pruning: A GBM would stop splitting a node when it encounters a negative loss in the split. Thus it is more of a greedy algorithm. XG-boost on the other hand make splits upto the max_depth specified and then start pruning the tree backwards and remove splits beyond which there is no positive gain.

Function Used

```
import xgboost as xgbmodel

xgb = xgbmodel.XGBClassifier(objective="binary:logistic", random_state=42)
xgb.fit(X_train, Y_train)

Y_prediction_xgbmodel = xgb.predict(X_test)
Y_prediction_xgbmodel.shape
score_of_xgb = round(accuracy_score(Y_prediction_xgbmodel,Y_test)*100,2)

print("The accuracy score achieved using XGBoost is: "+str(score_of_xgb)+" %")
```

```
The accuracy score achieved using XGBoost is: 85.25 %
```

6.7 Neural Networks

Neural networks reflect the behavior of the human brain, allowing computer programs to recognize patterns and solve common problems in the fields of AI, machine learning, and deep learning.

Neural networks, also known as artificial neural networks (ANNs) or simulated neural networks (SNNs), are a subset of machine_learning and are at the heart of deep learning algorithms. Their name and structure are inspired by the human brain, mimicking the way that biological neurons signal to one another.

Artificial neural networks (ANNs) are comprised of a node layers, containing an input layer, one or more hidden layers, and an output layer. Each node, or artificial neuron, connects to another and has an associated weight and threshold. If the output of any individual node is above the specified threshold value, that node is activated, sending data to the next layer of the network. Otherwise, no data is passed along to the next layer of the network.

Types of neural networks

Neural networks can be classified into different types, which are used for different purposes. While this isn't a comprehensive list of types, the below would be representative of the most common types of neural networks that you'll come across for its common use cases.

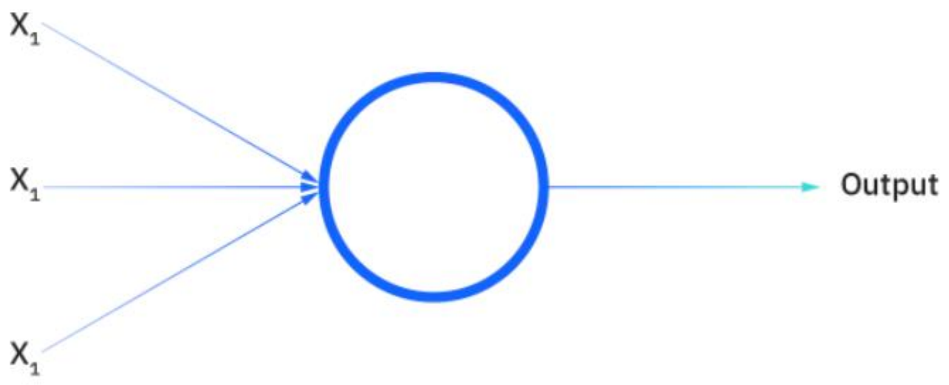


Fig. 19 : Neural Network

Feedforward neural networks or multi-layer perceptrons (MLPs), are what we’ve primarily been focusing on within this article. They are comprised of an input layer, a hidden layer or layers, and an output layer. While these neural networks are also commonly referred to as MLPs, it’s important to note that they are comprised of sigmoid neurons, not perceptrons, as most real-world problems are nonlinear. Data usually is fed into these models to train them, and they are the foundation for computer vision, natural_language processing, and other neural networks.

Convolutional neural networks (CNNs) are similar to feedforward networks, but they’re usually utilized for image recognition, pattern recognition, and/or computer vision. These networks harness principles from linear algebra, particularly matrix multiplication, to identify patterns within an image.

Recurrent neural networks (RNNs) are identified by their feedback loops. These learning algorithms are primarily leveraged when using time-series data to make predictions about future outcomes, such as stock market predictions or sales forecasting.

Function Used:

```
from keras.models import Sequential
from keras.layers import Dense
sequential_model = Sequential()
sequential_model.add(Dense(11,activation='relu',input_dim=13))
sequential_model.add(Dense(1,activation='sigmoid'))
sequential_model.compile(loss='binary_crossentropy',optimizer='adam',metrics=['accurac
y'])
sequential_model.fit(X_train,Y_train,epochs=300)

Y_prediction_neural_network = sequential_model.predict(X_test)
Y_prediction_neural_network.shape
rounded = [round(x[0]) for x in Y_prediction_neural_network]

Y_prediction_neural_network = rounded
score_of_neural_network =
round(accuracy_score(Y_prediction_neural_network,Y_test)*100,2)

print("The accuracy score achieved using Neural Network is:
"+str(score_of_neural_network)+" %")

#Note: Accuracy of 85% can be achieved on the test set, by setting epochs=2000, and
number of nodes = 11.
```

```
The accuracy score achieved using Neural Network is: 80.33 %
```

6.8 K-NEAREST NEIGHBOUR (KNN)

The k-Nearest-Neighbours (KNN) is a non-parametric classification method, which is simple but effective in many cases . For a data record t to be classified, its k nearest neighbours are retrieved, and this forms a neighbourhood of t. Majority voting among the data records in the neighbourhood is usually used to decide the classification for with or without consideration of distance-based weighting. However, to apply KNN we need to choose an appropriate value for k, and the success of classification is very much dependent on this value. In a sense, the KNN method is biased by k. There are many ways of choosing the k value, but a simple one is to run the algorithm many times with different k values and choose the one with the best performance.

In order for KNN to be less dependent on the choice of k, Wang proposed to look at multiple sets of nearest neighbours rather than just one set of k-nearest neighbours. The proposed formalism is based on contextual probability, and the idea is to aggregate the support of multiple sets of nearest neighbours for various classes to give a more reliable support value, which better reveals the true class of t. However, in its basic form the method is relatively slow, which needs O(n2) to classify a new Instance, though it is indeed less dependent on k and is able to achieve classification performance close to that for the best k.

Function used

```
from sklearn.neighbors import KNeighborsClassifier
kneighborsclassifier = KNeighborsClassifier(n_neighbors=7)
kneighborsclassifier.fit(X_train,Y_train)
Y_prediction_kneighborsclassifier=kneighborsclassifier.predict(X_test)
Y_prediction_kneighborsclassifier.shape
score_of_kneighborsclassifier =
round(accuracy_score(Y_prediction_kneighborsclassifier,Y_test)*100,2)
print("The accuracy score achieved using KNN is: "+str(score_of_kneighborsclassifier)+"
%")
```

The accuracy score achieved using KNN is: 67.21 %

CHAPTER 7 RESULT

Models used	Accuracy(in %)
Logistic Regression (LR)	85.25
Naïve Bayes(NB)	85.25
Support Vector Machine (SVM)	81.97
K-Nearest Neighbour (KNN)	67.21
Decision Tree(DT)	81.97
Random Forest(RF)	95.16
XGBoost(XGB)	85.25
Neural Network(NN)	81.97

Table : Models with accuracy

The result of the study shown in the above table clearly shows that Random forest has the highest Accuracy and KNN has the lowest accuracy. Every algorithm has accuracy more than eighty percent only KNN has 67 percent accuracy.

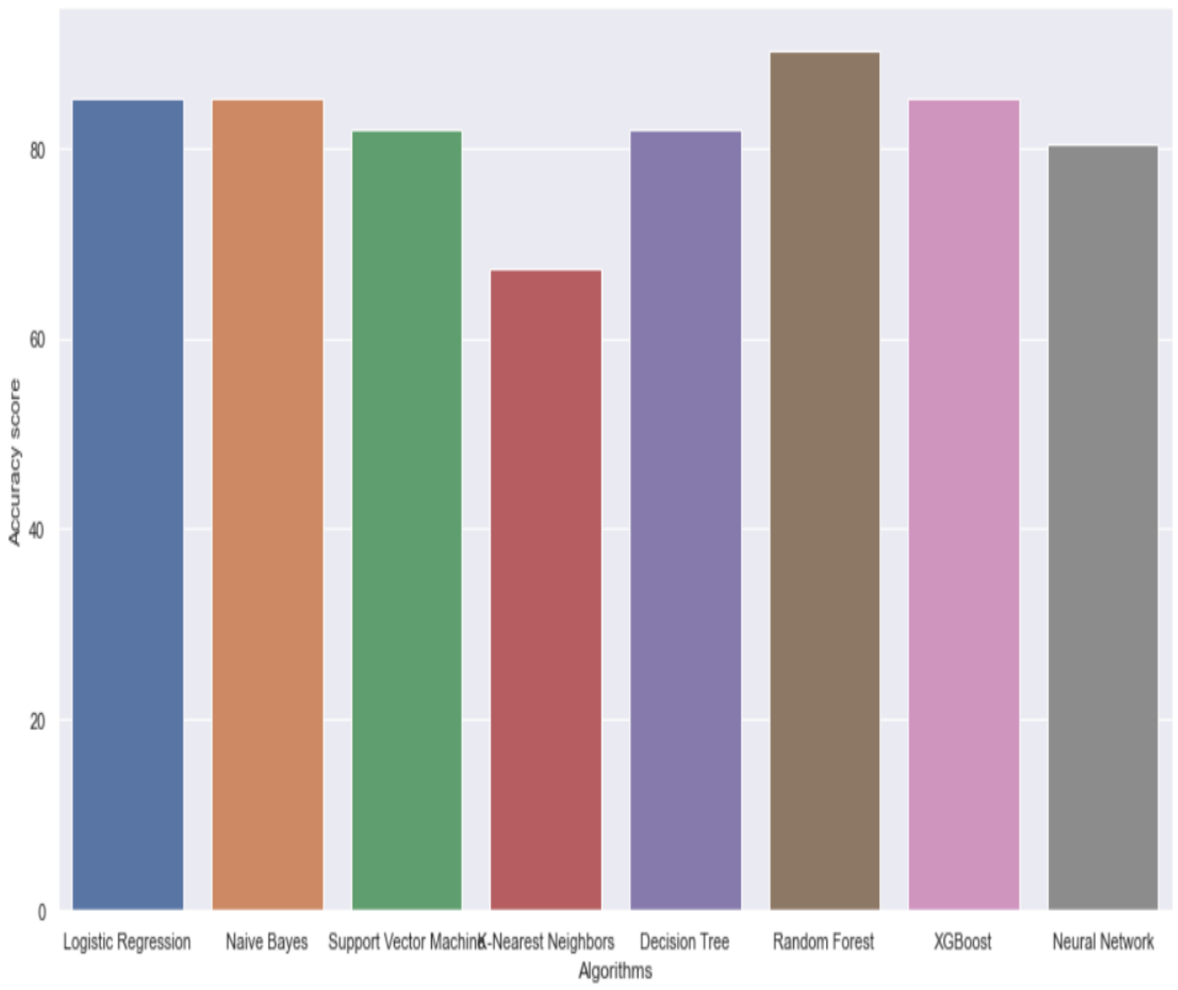


Fig. 20 : Final Result

From above graph, it can be easily predicted that Random Forest provides the best result with higher accuracy.

CHAPTER 8   FUTURE SCOPE

The future scope of the proposed methology is to use the model in any sensor based devices like smart watches etc, which can sense the pulses , etc. and send this data to the cloud with the help of IOT. Data will be stored on cloud and if there is any uneven fluctuation in the readings then the alert message will be send to the concerned people like family members or doctors. This model also helps in COVID cases, as using sensor based devices, doctors can check the patient without touching them. Doctors get the data from the cloud and accordingly can provide provide medication or prescription. This model helps in reduction of data rates as disease will be predict in early stage and medication can be provided before any emergency, so the chances of death of patient will be reduced which leads to the reduction of data rate. This model works best for the people living in nuclear family because in case of any issue alert signals will be sent to the concerned persons in the form of emails, SMS , etc. and the appropriate action can be taken .

So, in future this model will be fit in any sensor based device and then it will be used in the prediction of diseases which will be helpful in COVID cases, rural areas, etc.

CHAPTER 9 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.

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. This study compares the accuracy score of Decision Tree, Logistic Regression, Random Forest and Naive Bayes algorithms for predicting heart disease using UCI machine learning repository dataset. The result of this study indicates that the Random Forest algorithm is the most efficient algorithm with accuracy score of 95.16% for prediction of heart disease. In future the work can be enhanced by developing a web application based on the Random Forest algorithm as well as using a larger dataset as compared to the one used in this analysis which will help to provide better results and help health professionals in predicting the heart disease effectively and efficiently.

CHAPTER 10 REFERENCES

1. Ajay D Wasan , Alex M Dressler, Andrea G Gillman. “A narrative review of data collection and analysis guidelines for comparative effectiveness research in chronic pain using patient-reported outcomes and electronic health records”,2019.

2. Rinkal Keniya , Aman Khakharia ,Vruddhi Shah ,Vrushabh Gada ,Ruchi Manjalkar , Tirth Thaker , Mahesh Warang ,Ninad Mehendale. “Disease Prediction from various symptoms using machine learning”.

3. S. Mohan, C. Thirumalai, G. Srivastava, Effective heart disease prediction using hybrid machine learning techniques,IEEE Access 7, 81542 (2019).

4. Y. Khourdi_, M. Bahaj, Heart disease prediction and classification using machine learning algorithms optimized by particle swarm optimization and ant colony optimization,Int. J. Intell. Eng. Syst. 12(1), 242 (2019).

5. Marouane Fethi Ferjani . “Disease Prediction using Machine Learning”,2020.

6. F. Q. Yuan, “Critical issues of applying machine learning to condition monitoring for failure diagnosis,” in 2016 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM), 2016, pp. 1903–1907.

7. R.D.H.D.P. Sreevalli, K.P.M. Asia, Prediction of diseases using random forest classification algorithm.

8. Iqbal H. Sarker. “Machine Learning: Algorithms, Real-World Applications and Research Directions”, 2021.

9. Batta Mahesh. “Machine Learning Algorithms- A Review”, 2018.

10. W. Richert, L. P. Coelho, “Building Machine Learning Systems with Python”, Packt Publishing Ltd., ISBN 978-1-78216-140-0

11. J. M. Keller, M. R. Gray, J. A. Givens Jr., “A Fuzzy K-Nearest Neighbor Algorithm”, IEEE Transactions on Systems, Man and Cybernetics, Vol. SMC-15, No. 4, August 1985 .

12. https://www.geeksforgeeks.org/machine-learning

13. S. Marsland, Machine learning: an algorithmic perspective. CRC press, 2015.

14. M. Bkassiny, Y. Li, and S. K. Jayaweera, “A survey on machine learning techniques in cognitive radios,” IEEE Communications Surveys & Tutorials, vol. 15, no. 3, pp. 1136–1159, Oct. 2012.

15. https://en.wikipedia.org/wiki/Instance-based_learning

16. R. S. Sutton, “Introduction: The Challenge of Reinforcement Learning”, Machine Learning, 8, Page 225-227, Kluwer Academic Publishers, Boston, 1992

17. P. Harrington, “Machine Learning in action”, Manning Publications Co., Shelter Island, New York, 2012

18. A. Mir, S.N. Dhage, in 2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA) (IEEE, 2018).

19. Y. Khourdi_, M. Bahaj, Heart disease prediction and classification using machine learning algorithms optimized by particle swarm optimization and ant colony optimization, Int. J. Intell. Eng. Syst. 12(1), 242 (2019).

20. S. Vijayarani, S. Dhayanand, Liver disease prediction using svm and naive bayes algorithms, International Journal of Science, Engineering and Technology Research (IJSETR) 4(4), 816 (2015)

21. S. Mohan, C. Thirumalai, G. Srivastava, Effective heart disease prediction using hybrid machine learning techniques, IEEE Access 7, 81542 (2019).

22. T.V. Sriram, M.V. Rao, G.S. Narayana, D. Kaladhar, T.P.R. Vital, Intelligent Parkinson disease prediction using machine learning algorithms, International Journal of Engineering and Innovative Technology (IJEIT) 3(3), 1568 (2013).

23. A.S. Monto, S. Gravenstein, M. Elliott, M. Colopy, J. Schweinle, Clinical signs and symptoms predicting influenza infection, Archives of internal medicine 160(21), 3243 (2000)

24 R.D.H.D.P. Sreevalli, K.P.M. Asia, Prediction of diseases using random forest classification algorithm.

25 . D.R. Langbehn, R.R. Brinkman, D. Falush, J.S. Paulsen,M. Hayden, an International Huntington's Disease Collaborative Group, A new model for prediction of the age of onset and penetrance for huntington's disease based on cag length, Clinical genetics 65(4), 267 (2004).

26. G. Battineni, G. G. Sagaro, N. Chinatalapudi, and F Amenta, “Applications of machine learning predictive models in the chronic disease diagnosis,” Journal of Personalized Medicine, vol. 10, no. 2, p. 21, 2020.

27. B. Manjulatha and P. Suresh, “An ensemble model for predicting chronic diseases using machine learning algorithms,” in Smart Computing Techniques and Applications, pp. 337–345, Springer, New York, NY, USA, 2021.

28. C.-H. Jen, C.-C. Wang, B. C. Jiang, Y.-H. Chu, and M.-S. Chen, “Application of classification techniques on development an early-warning system for chronic illnesses,” Expert Systems with Applications, vol. 39, no. 10, pp. 8852–8858, 2012.

29. D. Gupta, S. Khare, and A. Aggarwal, “A method to predict diagnostic codes for chronic diseases using machine learning techniques,” in Proceedings of the 2016 International

Conference on Computing, Communication and Automation (ICCCA), pp. 281–287, IEEE, Greater Noida, India, April 2016.

30. M. Chen, H. Yixue, K. Hwang, L Wang, and L. Wang, “Disease prediction by machine learning over big data from healthcare communities,” Ieee Access, vol. 5, pp. 8869–8879, 2017.

31. R. Ge, R. Zhang, and P Wang, “Prediction of chronic diseases with multi-label neural network,” IEEE Access, vol. 8, pp. 138210–138216, 2020.

32. H. MacLeod, S. Yang, O. Kim, C. Kay, and S. Natarajan, “Identifying rare diseases from behavioural data: a machine learning approach,” in Proceedings of the 2016 IEEE First International Conference on Connected Health: Applications, Systems and Engineering Technologies (CHASE), pp. 130–139, IEEE, Washington, DC, USA, June 2016.

33. M. A. Myszczynska, P. N. Ojamies, A. M. B. Lacoste et al., “Applications of machine learning to diagnosis and treatment of neurodegenerative diseases,” Nature Reviews Neurology, vol. 16, no. 8, pp. 440–456, 2020.

34. I. Preethi and K. Dharmarajan, “Diagnosis of chronic disease in a predictive model using machine learning algorithm,” in Proceedings of the 2020 International Conference on Smart Technologies in Computing, Electrical and Electronics (ICSTCEE), pp. 191–96, IEEE, Bengaluru, India, October 2020.

35. J. Wiens and E. S Shenoy, “Machine learning for healthcare: on the verge of a major shift in healthcare epidemiology,” Clinical Infectious Diseases, vol. 66, no. 1, pp. 149–153, 2018.

36. S. Swaminathan, K. Qirko, T. Smith et al., “A machine learning approach to triaging patients with chronic obstructive pulmonary disease,” PLoS One, vol. 12, no. 11, Article ID e0188532, 2017.

37. Z. Wang, J. W. Chung, X. Jiang, Y. Cui, M. Wang, and A. Zheng, "Machine learning-based prediction system for chronic kidney disease using associative classification technique," International Journal of Engineering & Technology, vol. 7, pp. 1161–1167, 2018.

38. A. Kumar and A. Pathak, "A machine learning model for early prediction of multiple diseases to cure lives," Turkish Journal of Computer and Mathematics Education (TURCOMAT), vol. 12, no. 6, pp. 4013–4023, 2021.

39. C. Kalaiselvi, "Diagnosing of heart diseases using average K-nearest neighbor algorithm of data mining," in Proceedings of the 2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom), pp. 3099–3103, IEEE, New Delhi, India, March 2016.

40. D. Jain and V. Singh, "Feature selection and classification systems for chronic disease prediction: a review," Egyptian Informatics Journal, vol. 19, no. 3, pp. 179–189, 2018.