CHAPTER 1

INTRODUCTION

Nowadays, more people die every year from non-communicable diseases compared to infectious diseases. The heart related diseases consume around a million lives of people every year, creating this as the primary reason.

One death among three is due to heart disease in the United States (US). In the year 2016, around 9,20,000 people had heart attacks, and nearly half of them occurred suddenly without prior symptoms.

Sudden death is the only symptom of heart disease. Miserably, most of them belong to young age, particularly in India. In India, heart disease happens 1to1.5 decades in advance when compared to the western countries.

An estimation reports that there are around 45,000,000 will be affected by heart problems.

There seems to be a stable rise in hypertension pervasiveness for the previous 5 decades, which is extra in urban areas than in pastoral zones. It is 25 to 30 percent in urban and 10 to 50 percent in rural zones. An inactive lifestyle is the main reason for death, disease, and disability which multiplies the danger of heart disease.

In the current days, the heart specialized hospitals executed around 2 Lakhs of surgeries, particularly open-hearted every year which is the topmost figure worldwide. It increases by 25 to 30 percent every year constantly.

Heart disease describes a range of conditions that affect your heart. Today, cardiovascular diseases are the leading cause of death worldwide with 17.9 million deaths annually, as per the World Health Organization reports. Various unhealthy activities are the reason for the increase in the risk of heart disease like high cholesterol, obesity, increase in triglycerides levels, hypertension, etc. There are certain signs which the American Heart Association lists like the persons having sleep issues, a certain increase and decrease in heart rate (irregular heartbeat), swollen legs, and in some cases weight gain occurring quite fast; it can be 1-2 kg daily. All these symptoms resemble different diseases also like it occurs in the aging persons, so it becomes a difficult task to get a correct diagnosis, which results in fatality in near future.

But as time is passing, a lot of research data and patients records of hospitals are available

There are many open sources for accessing the patient’s records and researches can be conducted so that various computer technologies could be used for doing the correct diagnosis of the patients and detect this disease to stop it from becoming fatal. Nowadays it is well known that machine learning and artificial intelligence are playing a huge role in the medical industry. We can use different machine learning and deep learning models to diagnose the disease and classify or predict the results. A complete genomic data analysis can easily be done using machine learning models. Models can be trained for knowledge pandemic predictions and also medical records can be transformed and analyzed more deeply for better predictions .

Many studies have been performed and various machine learning models are used for doing the classification and prediction for the diagnosis of heart disease. An automatic classifier for detecting congestive heart failure shows the patients at high risk and the patients at low risk by Melillo et al.; they used machine learning algorithm as CART which stands for Classification and Regression in which sensitivity is achieved as 93.3 percent and specificity is achieved as 63.5 percent.

Then for improving the performance electrocardiogram (ECG) approach is suggested by Rahhal et al. in which deep neural networks are used for choosing the best features and then using them. Then, for detecting heart failures, a clinical decision support system is contributed by Guidi et al for preventing it at an early stage. They tried to compare different machine learning models and deep learning models especially neural networks, as support vector machine, random forest, and CART algorithms. An 87.6 percent accuracy was achieved by random forest and CART, which outperformed everyone used in the classification. Combining the natural language processing with the rule-based approach, Zhang et al achieved 93.37 percent accuracy when the NYHA HF class was found from the unstructured clinical notes.

SVM techniques used for detecting patients who already have diabetes and then predicting heart disease by Parthiban and Srivatsa achieved a 94.60 percent accuracy rate, and the features taken were common like blood sugar level, age of the patient, and their blood pressure data.

Heart disease is very fatal and it should not be taken lightly. Heart disease happens more in males than females, which can be read further from Harvard Health Publishing. Researchers found that, throughout life, men were about twice as likely as women to have a heart attack. That higher risk persisted even after they accounted for traditional risk factors of heart disease, including high cholesterol, high blood pressure, diabetes, body mass index, and physical activity.

The researchers are working on this dataset as it contains certain important parameters like dates from 1998, and it is considered as one of the benchmark datasets when someone is working on heart disease prediction. This dataset dates from 1988 and consists of four databases: Cleveland, Hungary, Switzerland, and Long Beach V, and the results achieved are quite promising.

The main challenge in today's healthcare is provision of best quality services and effective accurate diagnosis . Even if heart diseases are found as the prime source of death in the world in recent years, they are also the ones that can be controlled and managed effectively. The whole accuracy in management of a disease lies on the proper time of detection

The proposed work tries to detect these heart diseases at early stage to avoid disastrous consequences.

Records of large set of medical data created by medical experts are available for analysing and extracting valuable knowledge from it. Data mining techniques are the means of extracting valuable and hidden information from the large amount of data available. Mostly the medical database consists of discrete information. Hence, decision making using discrete data becomes complex and tough task. Machine Learning (ML) which is subfield of data mining handles large scale well-formatted dataset efficiently. In the medical field, machine learning can be used for diagnosis, detection and prediction of various diseases. The main goal of this paper is to provide a tool for doctors to detect heart disease as early stage . This in turn will help to provide effective treatment to patients and avoid severe consequences. ML plays a very important role to detect the hidden discrete patterns and thereby analyse the given data. After analysis of data ML techniques help in heart disease prediction and early diagnosis. This paper presents performance analysis of various ML techniques such as Naive Bayes, Decision Tree, Logistic Regression and Random Forest for predicting heart disease at an early stage.

1.1 OBJECTIVE

The main objective of this research is to develop a heart prediction system. The system can discover and extract hidden knowledge associated with diseases from a historical heart data set.

The heart disease prediction system aims to exploit data mining techniques on medical data sets to assist in the prediction of heart disease.

- Provides a new approach to concealed patterns in the data.
- Helps avoid human biases.
- To implement a Naïve byes classifier that classifies the disease as per the input of the user.
- Reduce the cost of medical tests.

1.2 PROBLEM DEFINITION

A dataset is formed by taking into consideration some of the information of 920 individuals.

The problem is: based on the given information about each individual we have to calculate whether that individual will suffer from heart disease.

1.3 EXISTING SYSTEM

Very few systems use the available clinical data for prediction purposes and even if they do, they are restricted by the large number of association rules that apply.

Diagnosis of the condition solely depends upon the Doctors intuition and patient’s records.

The Disadvantages are:

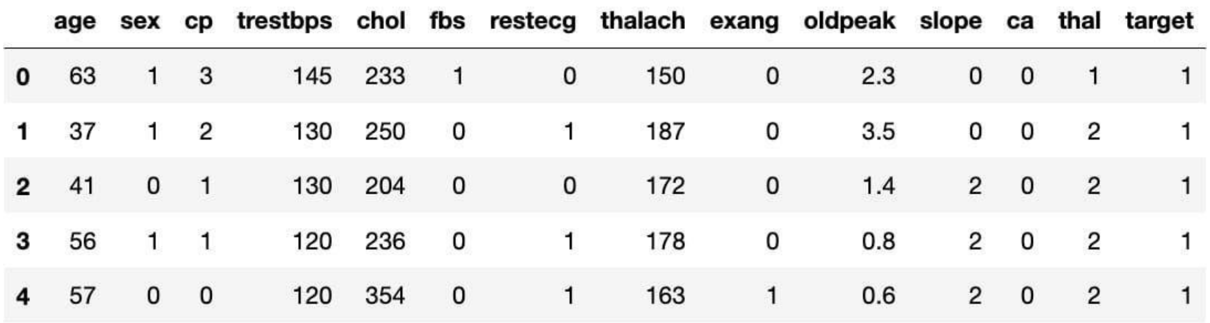
- Detection is not possible at an early stage.
- In the existing system, practical use of various collected data is time- consuming.

1.4 PROPOSED SYSTEM

- The proposed system acts as a decision support system and will prove to be an aid for the physicians with the diagnosis.
- The algorithm Fuzzy c means uses clustering and makes use of clusters and data points the relativity of an attribute.
- Each data point is associated with multiple clusters depending upon the membership degrees.

Advantages:

- High performance and accuracy rate.
- PCM is very flexible and is widely in various domains with high rates of success.



	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

Fig 1 – Dataset

CHAPTER 2

LITERATURE SURVEY

PAPER 1

L Sathish Kumar and A Padma Priya have given a paper named Prediction for similarities of disease by using ID3 algorithm in television and mobile phone.

This paper gives a programmed and concealed way to deal with recognizing designs that are covered up of coronary illness.

Approach

A. Television

Today television has become an integral part of our lives. We have any one information and any one something we got from television. Now a days, people are spending more time on televisions, and no people are there without mobile phone so, through there devices the information can be easily delivered to the people. Now the disease diagnosis method can be easily prediction through television.

B. Mobile Phone

Its component is cheap though mobile phone has limited capacity and speed. It is handy in dealing, talking, making video film, e-mailing, sending pictures etc. Mobile phones are being extensively used by students at all levels, doctors, engineers, service man, jobbers and common man and woman in their day-to-day activities. Today business is next to impossible without a mobile phone. Starting from aviation industry to service sector, the mobile phones are playing a vital role. Here the mobile phones have lot of application so now we can implement disease prediction method to mobile phone. So we can easily predict our disease very simple at the movement.

Advantages:

The given framework utilizes information mining methods

This proposed method helps the people not only to know about the diseases but it can also help to reduce the death rate and count of disease-affected people.

PAPER 2

Mohammed Abdul Khaleel has given a paper in the Survey of Techniques for mining of data on Medical Data for Finding Frequent Diseases locally.

This paper focuses on dissecting information mining procedures that are required for medicinal information mining particularly to find locally visit illnesses.

Data mining is the system of mining data form responsible hidden patterns which can be converted into understandable format. Data mining essential is one of a number of analytical tools for applied to data. It allows users to analyse data from many different databases and categorize it, and summarize the relationships identified. Generally, data mining is the process of searching patterns or correlations among relational databases repository of fields in large databases where data sources can include database, data warehouse, web, information repositories etc. Data mining turns a huge collection of data into knowledge for global challenges.

It is nothing but ‘Knowledge mining data” which is somewhat long and similar to knowledge extraction, data/pattern analysis, data dreading. In India, further societies are listening about the medical fitness and health opinion problems.

Data mining techniques found applications in many areas since their development. New techniques or combination of techniques are created continuously. Techniques like supervised or unsupervised learning are used for prediction different diseases and have the aim to identify (diagnosis) the disease and predict the incidence or make predictions about treatment and survival rate.

Advantages:

- There are more than 500 patients in the dataset.
- The accuracy offered by naïve bayes is 86.419%. Assess the health of the public and patterns of illness and injury.
- Identify unmet regional health needs.
- Document patterns of health care expenditures on inappropriate, wasteful, potentially harmful services.
- Find cost-effective care providers. Improve the quality of care in hospitals, practitioners' offices, clinics, and various other health care settings.

PAPER 3

Gayathri P and Jaisankar the review of the research associated with heart disease and furthermore.

The overview of numerous classifications of heart disease such as coronary artery disease, coronary heart disease, ischemic heart disease, heart failure, congenital heart disease, cardiovascular disease, hypoplastic left heart syndrome, and valvular heart disease are presented in the research.

Cardiovascular disease develops 7 to 10 years later in women than in men and is still the major cause of death in women over the age of 65 years. The risk of heart disease in women is often underestimated due to the misperception that females are 'protected' against cardiovascular disease. Recent data from the National Health and Nutrition Examination Surveys (NHANES) have shown that over the past two decades the prevalence of myocardial infarctions has increased in midlife (35 to 54 years) women, while declining in similarly aged men. Based on this perspective, several researches have been conducted in the literature recently. So, analysing those diagnosis techniques can lead to new development in this area. Furthermore, self awareness in women and identification of their cardiovascular risk factors needs more attention which should result in a better prevention of cardiovascular events. In this review we summarise the major issues that are important in the diagnosis and treatment of coronary heart disease (CHD) in women.

Accordingly, we present a detailed survey of 47 articles published in the standard journals from the year 2005 to 2013. The survey of the papers related to heart disease and also the survey of many categories of heart disease such as coronary heart disease, coronary artery disease, heart failure, ischemic heart disease, cardiovascular disease, congenital heart disease, valvular heart disease and hypoplastic left heart syndrome are presented in this paper.

Advantages:

- With the prediction of coronary heart disease. Early treatment can be given at the right time which avoids the risk of heart attacks.
- Simple procedure and is easy to obtain the required results.

PAPER 4

M.A. Nishara Banu and B. Gomathy have given a paper named Disease Predicting system using data mining techniques.

In this paper, they talk about MAFIA (Maximal Frequent Itemset algorithm) and K-Means clustering. As classification is important for the prediction of disease. The classification based on MAFIA and K-Means results in inaccuracy.

The healthcare industry collects large amounts of healthcare information which cannot be mined to find unknown information for efficient evaluation. Discovery of buried patterns frequently goes unexploited. Heart disease is a term for defining a huge amount of healthcare conditions that are related to the heart. This medicinal condition defines the unpredicted health conditions that directly control all the parts of the heart. Different data mining techniques such as association rule mining, classification, clustering are used to predict the heart disease in health care industry.

The heart disease database is preprocessed to make the mining process more efficient. The preprocessed data is clustered using clustering algorithms like K-means to cluster relevant data in database. Maximal Frequent Item set Algorithm (MAFIA) is used for mining maximal frequent patterns in heart disease database. The frequent patterns can be classified using C4.5 algorithm as training algorithm using the concept of information entropy.

The results showed that the designed prediction system is capable of predicting the heart attack successfully.

Advantages:

- Data mining helps marketing companies build models based on historical data to predict who will respond to the new marketing campaigns such as direct mail, online marketing campaign.
- Data mining brings a lot of benefits to retail companies in the same way as marketing.
- It also helps the retail companies offer certain discounts for particular products that will attract more customers.

CHAPTER 3

PROPOSED ARCHITECTURE

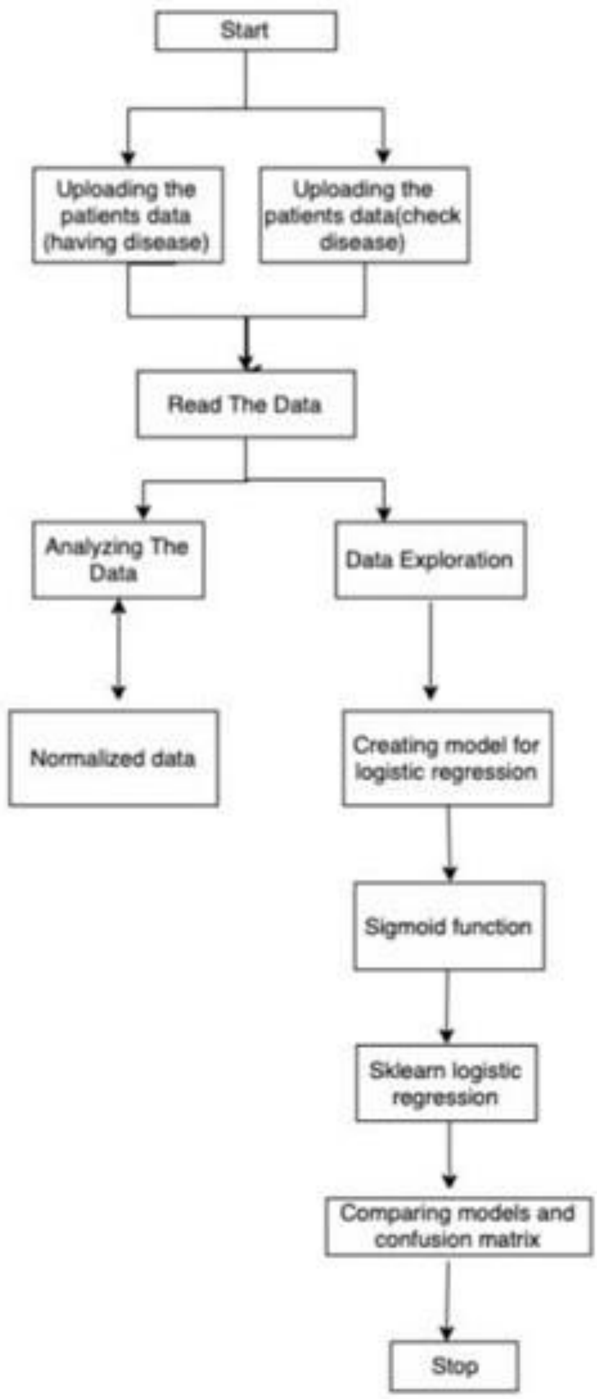

Start
Uploading the patients data (having disease)
Uploading the patients data(check disease)
Read The Data
Analyzing The Data
Data Exploration
Normalized data
Creating model for logistic regression
Sigmoid function
Sklearn logistic regression
Comparing models and confusion matrix
Stop

Fig 2 – Proposed Architecture

3.1 PROCESS

This proposed system has a data which classified if patients have heart disease or not according to features in it. This proposed system can try to use this data to create a model which tries predict (reading data and data Exploration if a patient has this disease or not. In this proposed system, use logistic regression.

By using sk learn library to calculate score. Implements Naive Bayes algorithm to getting accuracy result Finally analysing the results by the help of Comparing Models and Confusion Matrix.

1. From the data we are having, it should be classified into different structured data based on the features of the patients the art.
2. From the availability of the data, we have to create a model which predicts the patient disease using logistic regression algorithm.

- First, we have to import the datasets.
- Read the data sets, the data should contain different variables like age, gender, sex, cp(chest pain), slope, target.
- The data should be explored so that the information is verified.
- Create a temporary variable and also build a model for logistic regression. Here, we use sigmoid function which helps in the graphical representation of the classified data.

By using logistic regression, Naive Bayes the accuracy rate increases.

3.2 DATA PREPROCESSING

Data pre-processing is a step in the data mining and data analysis process that takes raw data and transforms it into a format that can be understood and analysed by computers and machine learning.

Raw, real-world data in the form of text, images, video, etc., is messy. Not only may it contain errors and inconsistencies, but it is often incomplete, and doesn't have a regular, uniform design.

Machines like to process nice and tidy information – they read data as 1s and 0s. So calculating structured data, like whole numbers and percentages is easy. However, unstructured data, in the form of text and images must first be cleaned and formatted before analysis. Depending on your data gathering techniques and sources, you may end up with data that's out of range or includes an incorrect feature, like household income below zero or an image from a set of "zoo animals" that is actually a tree. Your set could have missing values or fields.

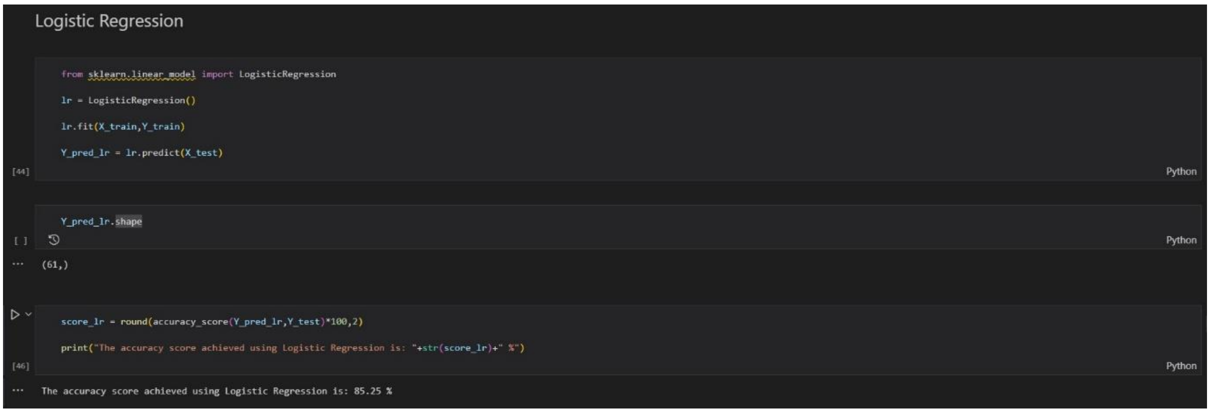
Steps involved:

1. CLEANING: Data that we want to process will not be clean that is it may contain noise or it may contain values missing of we process we cant get good results so to obtain good and perfect results we need to eliminate all this, the process to eliminate all this is data cleaning.
2. TRANSFORMATION: This involves changing data format to one form to other that is making them most understandable by doing normalization, smoothing, and generalization, aggregation techniques on data.
3. INTEGRATION: Data that we need not process may not be from a single source sometimes it can be from different sources we do not integrate them it may be a problem while processing so integration is one of important phase in data pre-processing
4. REDUCTION: When we work on data it may be complex and it may be difficult to understand sometimes so to make them understandable to system we will reduce them to required format so that we can achieve good results.

CHAPTER 4

ALGORITHMS USED

4.1 Logistic Regression Model



```
from sklearn.linear_model import LogisticRegression

lr = LogisticRegression()

lr.fit(X_train,Y_train)

Y_pred_lr = lr.predict(X_test)
```

```
Y_pred_lr.shape
```

```
(61,)
```

```
score_lr = round(accuracy_score(Y_pred_lr,Y_test)*100,2)

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

```
The accuracy score achieved using Logistic Regression is: 85.25 %
```

Fig 3 – Logistic Regression Algorithm

- We are going to use a logistic regression model.
- Logistic regression is a statistical model that in its basic form uses a logistic function to model a binary dependent variable.
- Logistic regression (or logit regression) is estimating the parameters of a logistic model (a form of binary regression). Mathematically, a binary logistic model has a dependent variable with two possible values, such as pass/fail
- This particular use case is a binary classification, so we are going to classify whether a person has the heart of a disease or not.
- This is a binary classification of either yes or no kinds of questions.
- In the binary classifications logistic regression model is very useful so it’s the best model when it comes to binary classification.

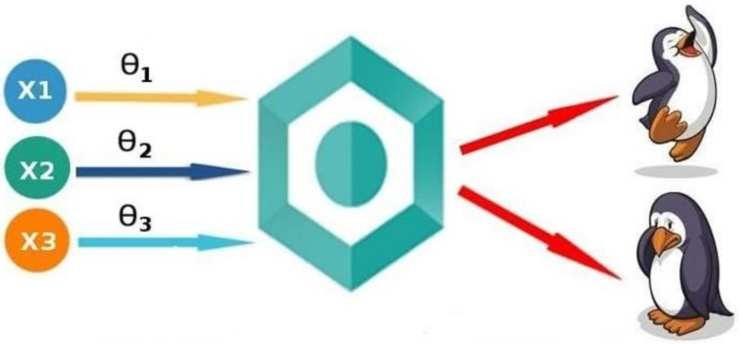
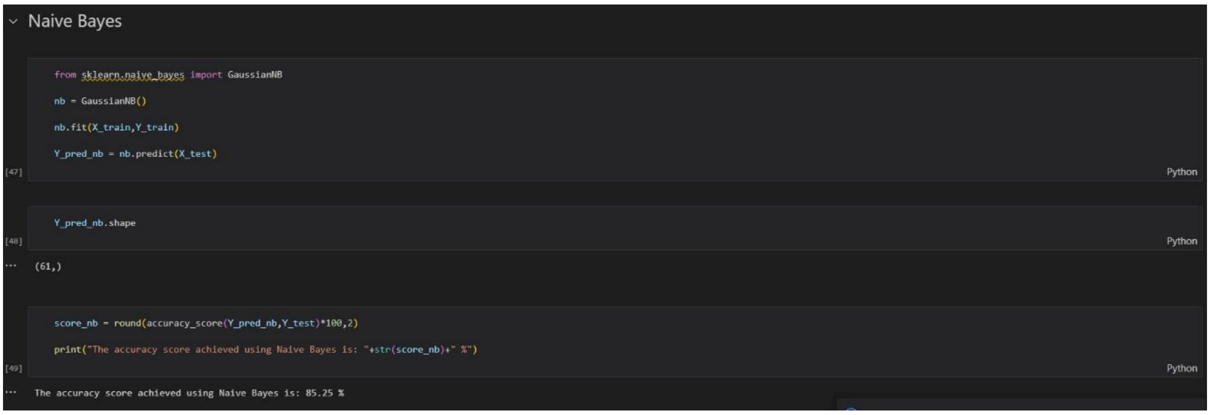


Fig 4 – Logistic Regression Model

4.2 Naive Bayes Model



```
from sklearn.naive_bayes import GaussianNB
nb = GaussianNB()
nb.fit(X_train,Y_train)
Y_pred_nb = nb.predict(X_test)
[47]

Y_pred_nb.shape
[48]
...  (61,)

score_nb = round(accuracy_score(Y_pred_nb,Y_test)*100,2)
print("The accuracy score achieved using Naive Bayes is: "+str(score_nb)+" %")
[49]
...  The accuracy score achieved using Naive Bayes is: 85.25 %
```

Fig 5 – Naïve Bayes Algorithm

The Naive Bayes classification algorithm is a probabilistic classifier. It is based on probability models that incorporate strong independence assumptions. The independence assumptions often do not have an impact on reality. Therefore they are considered as naive.

This algorithm works very fast and can easily predict the class of a test dataset. You can use it to solve multi-class prediction problems as it's quite useful with them. Naive Bayes classifier performs better than other models with less training data if the assumption of independence of features holds.

Step 1: Handling Data.

Step 2: Summarizing the Data.

Step 3: Making a Prediction.

Step 4: Making all the Predictions.

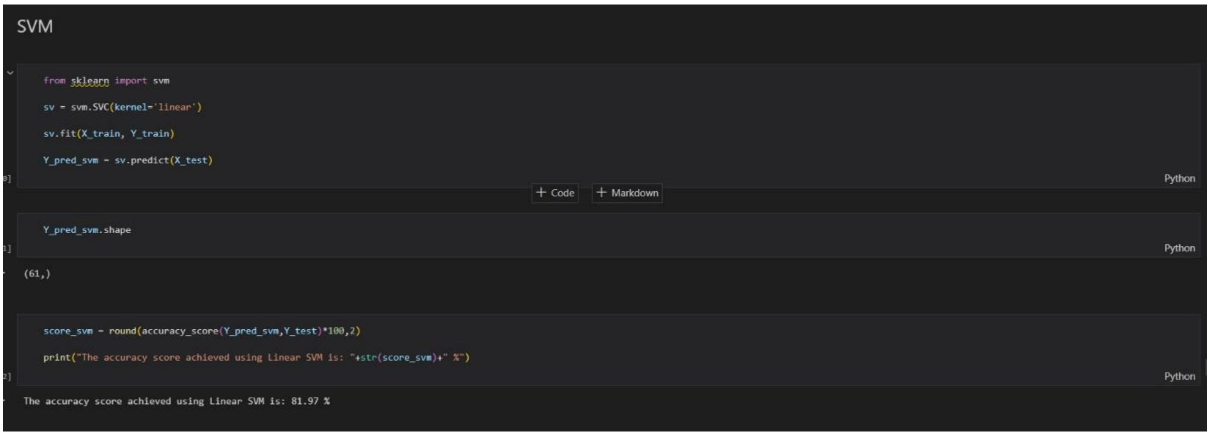
Step 5: Evaluate Accuracy.

Step 6: Tying all Together.

Advantages

- It is easy and fast to predict the class of the test data set.
- It also performs well in multi-class prediction.
- When assumption of independence holds, a Naive Bayes classifier performs better compare to other models like logistic regression and you need less training data.

4.3 Support Vector Machine



SVM

```
from sklearn import svm

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

sv.fit(X_train, Y_train)

Y_pred_svm = sv.predict(X_test)
```

```
Y_pred_svm.shape
```

```
(61,)
```

```
score_svm = round(accuracy_score(Y_pred_svm,Y_test)*100,2)

print("The accuracy score achieved using Linear SVM is: "+str(score_svm)+" %")
```

```
The accuracy score achieved using Linear SVM is: 81.97 %
```

Fig 6 – SVM Algorithm

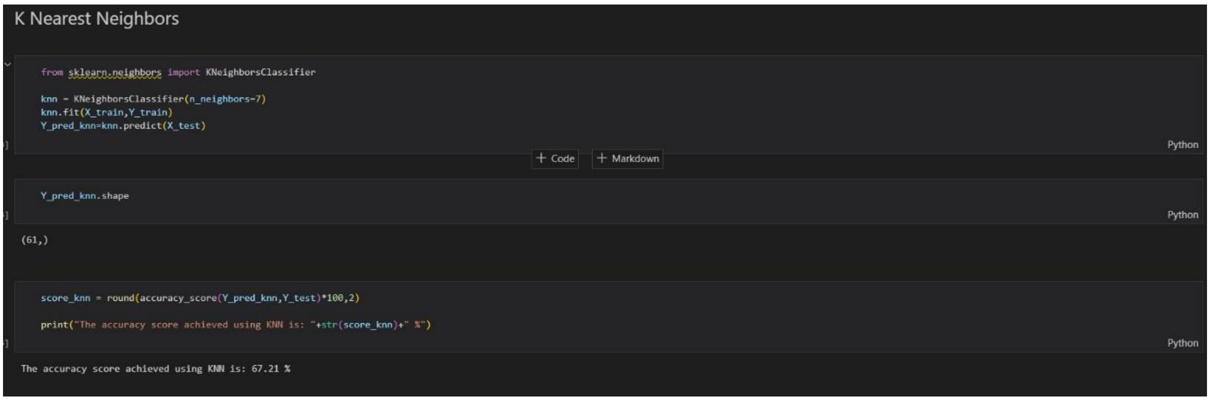
Support Vector Machine (SVM) is a supervised machine learning algorithm capable of performing classification, regression and even outlier detection. The linear SVM classifier works by drawing a straight line between two classes. All the data points that fall on one side of the line will be labeled as one class and all the points that fall on the other side will be labeled as the second. Sounds simple enough, but there’s an infinite amount of lines to choose from. How do we know which line will do the best job of classifying the data? This is where the LSVM algorithm comes in to play. The LSVM algorithm will select a line that not only separates the two classes but stays as far away from the closest samples as possible. In fact, the “support vector” in “support vector machine” refers to two position vectors drawn from the origin to the points which dictate the decision boundary.

4.3.1 How SVM works?

Just for the sake of understanding, we will leave the machines out of the picture for a minute. Now how would a human being like you and me classify a set of objects scattered on the surface of a table? Of course, we will consider all their physical and visual characteristics and then identify based on our prior knowledge. We can easily identify and distinguish apples and oranges based on their colour, texture, shape etc.

Now bringing back the machines, how would a machine identify an apple or an orange. Not surprisingly, it is based on the characteristics that we provide the machine with. It can be size, shape, weight etc. The more features we consider the easier it is to identify and distinguish both.

4.4 K Nearest Neighbors



K Nearest Neighbors

```
from sklearn.neighbors import KNeighborsClassifier

knn = KNeighborsClassifier(n_neighbors=7)
knn.fit(X_train,Y_train)
Y_pred_knn=knn.predict(X_test)
```

```
Y_pred_knn.shape
```

```
(61,)
```

```
score_knn = round(accuracy_score(Y_pred_knn,Y_test)*100,2)

print("The accuracy score achieved using KNN is: "+str(score_knn)+" %")
```

```
The accuracy score achieved using KNN is: 67.21 %
```

Fig 7 – KNN Algorithm

This algorithm is used to solve the classification model problems. K-nearest neighbor or K-NN algorithm basically creates an imaginary boundary to classify the data. When new data points come in, the algorithm will try to predict that to the nearest of the boundary line.

Therefore, larger k value means smother curves of separation resulting in less complex models. Whereas, smaller k value tends to overfit the data and resulting in complex models.

4.4.1 Pros and Cons of KNN

In this section we'll present some of the pros and cons of using the KNN algorithm.

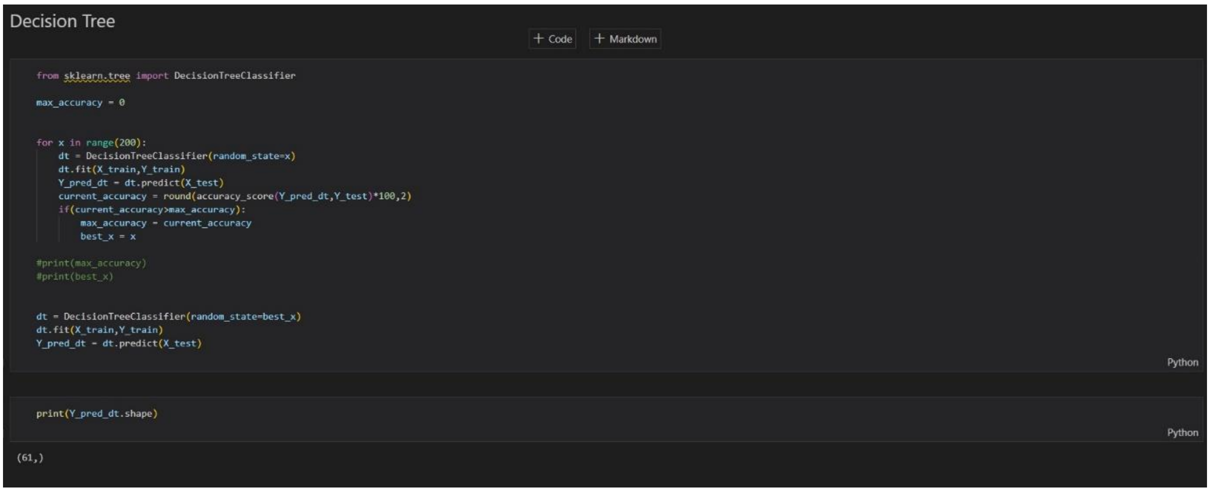
Pros

- It is extremely easy to implement
- Since the algorithm requires no training before making predictions, new data can be added seamlessly.
- There are only two parameters required to implement KNN i.e. the value of K and the distance function (e.g. Euclidean or Manhattan etc.)

Cons

- The KNN algorithm doesn't work well with high dimensional data because with large number of dimensions, it becomes difficult for the algorithm to calculate distance in each dimension.
- The KNN algorithm has a high prediction cost for large datasets. This is because in large datasets the cost of calculating distance between new point and each existing point becomes higher.

4.5 Decision Tree



```
from sklearn.tree import DecisionTreeClassifier

max_accuracy = 0


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

#print(max_accuracy)
#print(best_x)


dt = DecisionTreeClassifier(random_state=best_x)
dt.fit(X_train,Y_train)
Y_pred_dt = dt.predict(X_test)
```

```
print(Y_pred_dt.shape)
```

```
(61,)
```

Fig 8 – Decision Tree Method

A decision tree is a type of supervised machine learning used to categorize or make predictions based on how a previous set of questions were answered. The model is a form of supervised learning, meaning that the model is trained and tested on a set of data that contains the desired categorization.

The decision tree may not always provide a clear-cut answer or decision. Instead, it may present options so the data scientist can make an informed decision on their own. Decision trees imitate human thinking, so it’s generally easy for data scientists to understand and interpret the results.

4.5.1 How Does the Decision Tree Work?

Root node: The base of the decision tree.

Splitting: The process of dividing a node into multiple sub-nodes.

Decision node: When a sub-node is further split into additional sub-nodes.

Leaf node: When a sub-node does not further split into additional sub-nodes; represents possible outcomes.

Pruning: The process of removing sub-nodes of a decision tree.

Branch: A subsection of the decision tree consisting of multiple nodes.

A decision tree resembles, well, a tree. The base of the tree is the root node. From the root node flows a series of decision nodes that depict decisions to be made. From the decision nodes are leaf nodes that represent the consequences of those decisions. Each decision node represents a question or split point, and the leaf nodes that stem from a decision node represent the possible answers. Leaf nodes sprout from decision nodes similar to how a leaf sprouts on a tree branch. This is why we call each subsection of a decision tree a "branch." Let's take a look at an example for this. You're a golfer, and a consistent one at that. On any given day you want to predict where your score will be in two buckets: below par or over par.

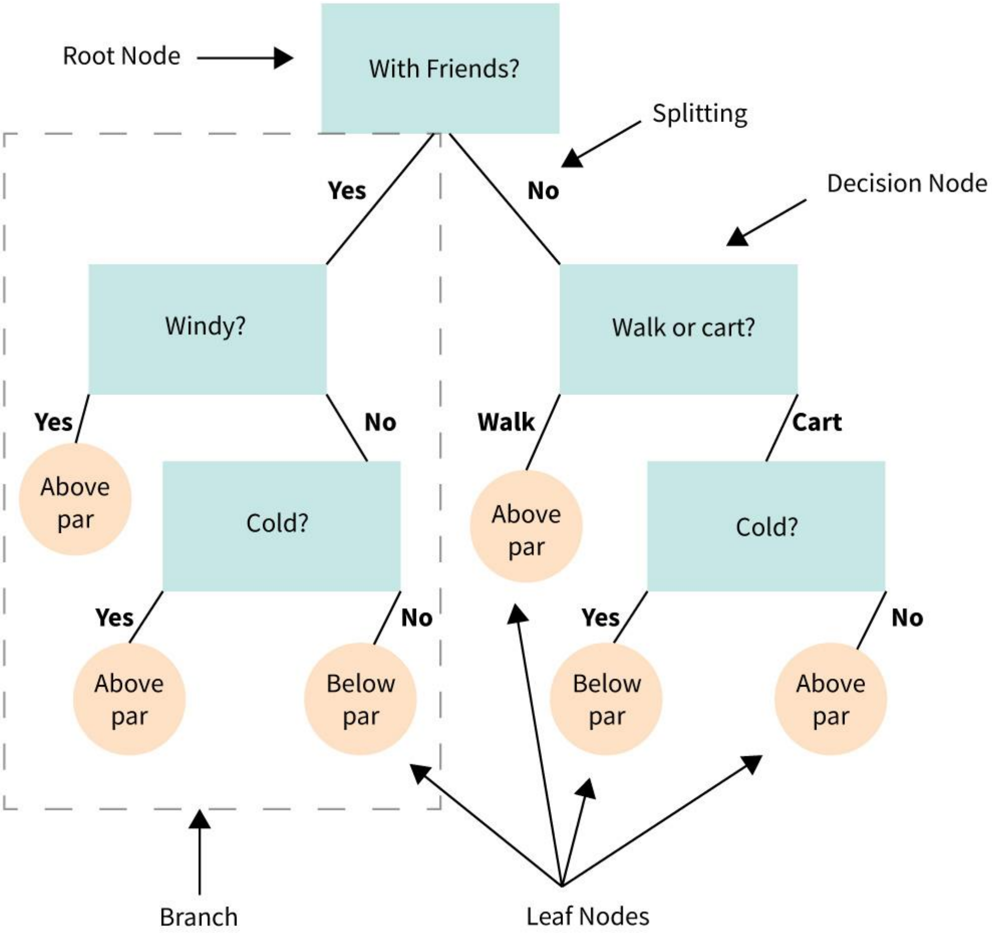
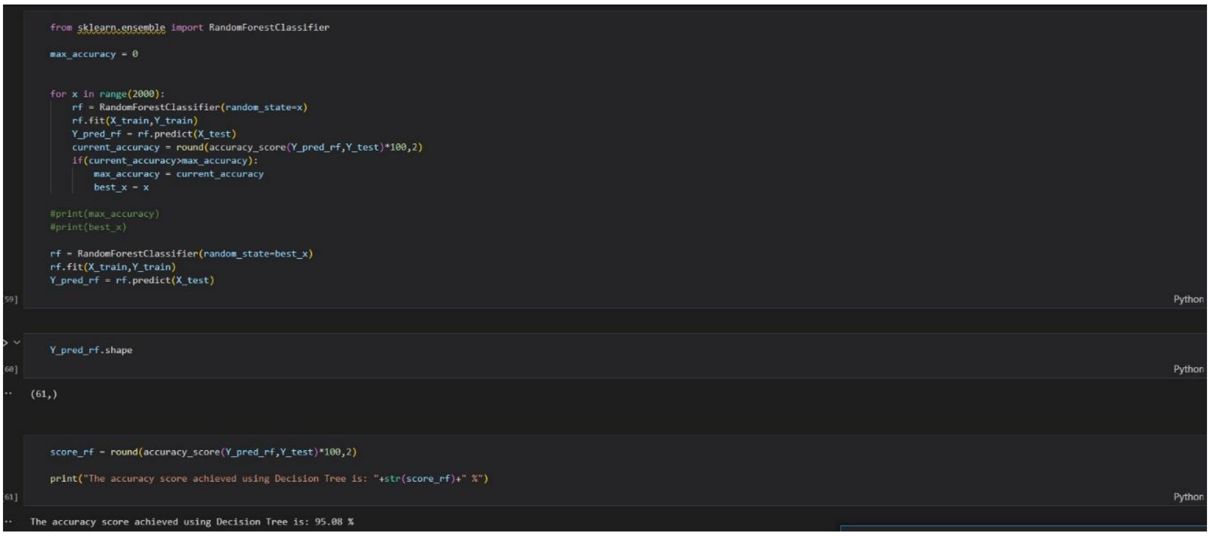


Fig 9 – Workflow Process

In this example, there are two leaf nodes: below par or over par. Each of the input variables will determine decision nodes. Was it windy? Cold? Did you golf with friends? Did you walk or take a cart? With enough data on your golfing habits (and assuming you are a consistent golfer), a decision tree could help predict how you will do on the course on any given day.

4.6 Random Forest Model



```
from sklearn.ensemble import RandomForestClassifier

max_accuracy = 0


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

#print(max_accuracy)
#print(best_x)

rf = RandomForestClassifier(random_state=best_x)
rf.fit(X_train,Y_train)
Y_pred_rf = rf.predict(X_test)
```

```
Y_pred_rf.shape
```

```
(61,)
```

```
score_rf = round(accuracy_score(Y_pred_rf,Y_test)*100,2)

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

```
The accuracy score achieved using Decision Tree is: 95.08 %
```

Fig 10 – Random Forest Algorithm

Random forests is a supervised learning algorithm. It can be used both for classification and regression. It is also the most flexible and easy to use algorithm. A forest is comprised of trees. It is said that the more trees it has, the more robust a forest is. Random forests creates decision trees on randomly selected data samples, gets prediction from each tree and selects the best solution by means of voting. It also provides a pretty good indicator of the feature importance.

Random forests has 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.

4.6.1 Random Forests vs Decision Trees

- Random forests is a set of multiple decision trees.
- Deep decision trees may suffer from overfitting, but random forests prevents overfitting by creating trees on random subsets.
- Decision trees are computationally faster.
- Random forests is difficult to interpret, while a decision tree is easily interpretable and can be converted to rules.

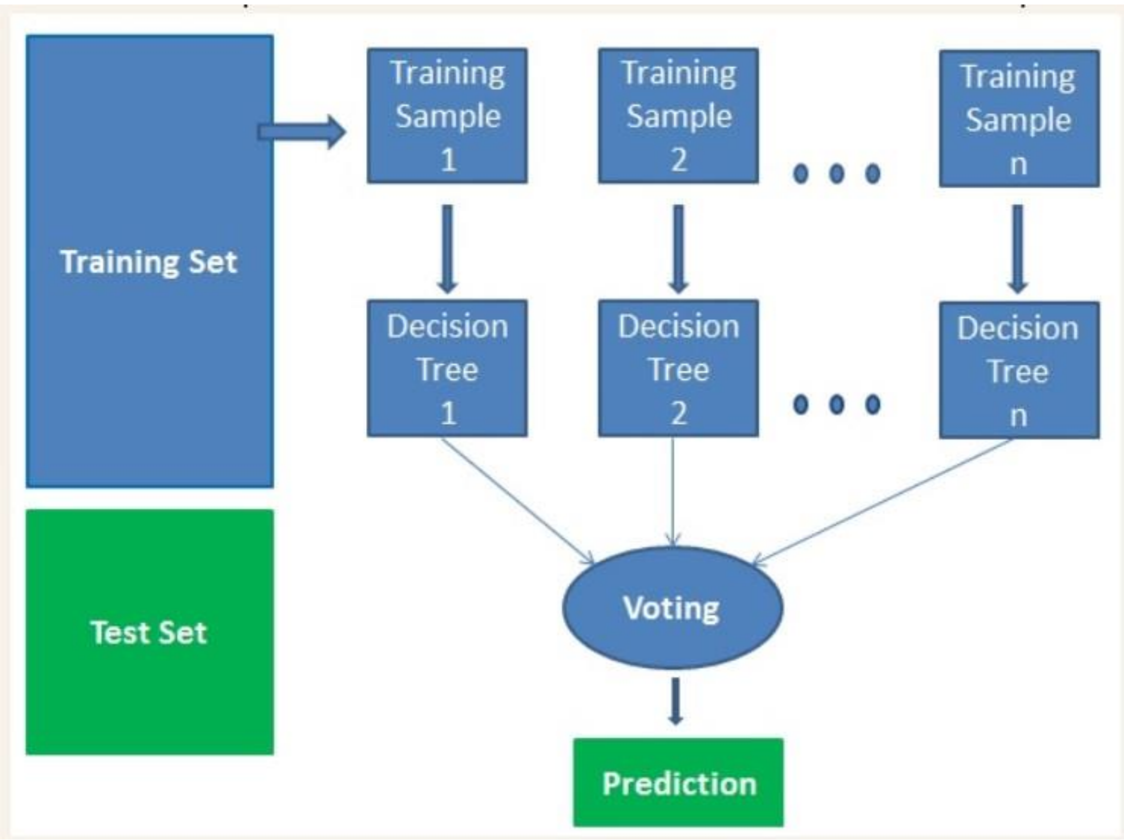


Fig 11 – Random Tree Process

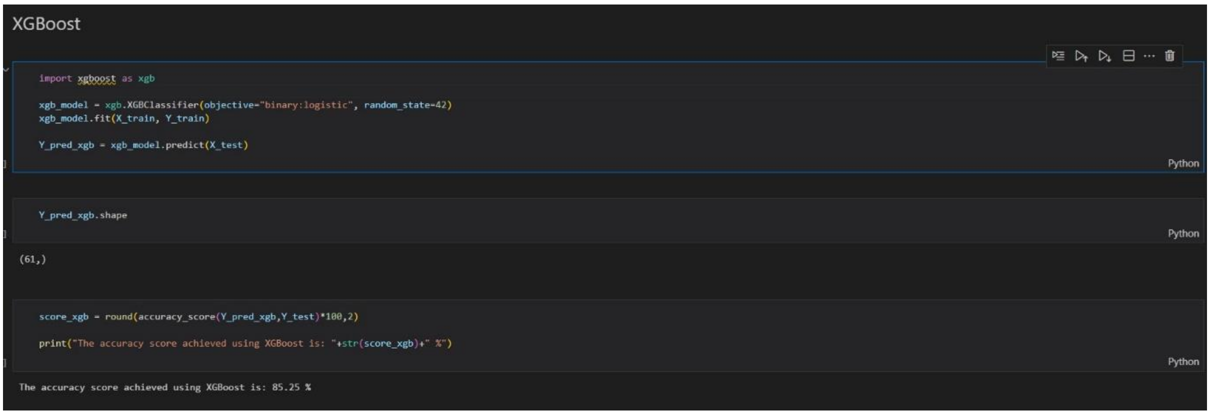
Advantages:

- Random forests is considered as a highly accurate and robust method because of the number of decision trees participating in the process.
- It does not suffer from the overfitting problem. The main reason is that it takes the average of all the predictions, which cancels out the biases.
- The algorithm can be used in both classification and regression problems.
- Random forests can also handle missing values. There are two ways to handle these: using median values to replace continuous variables, and computing the proximity-weighted average of missing values.

Disadvantages:

- Random forests is slow in generating predictions because it has multiple decision trees.
- This whole process is time-consuming.

4.7 XG Boost



```
import xgboost as xgb

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

Y_pred_xgb = xgb_model.predict(X_test)
```

```
Y_pred_xgb.shape
```

```
(61,)
```

```
score_xgb = round(accuracy_score(Y_pred_xgb,Y_test)*100,2)

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

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

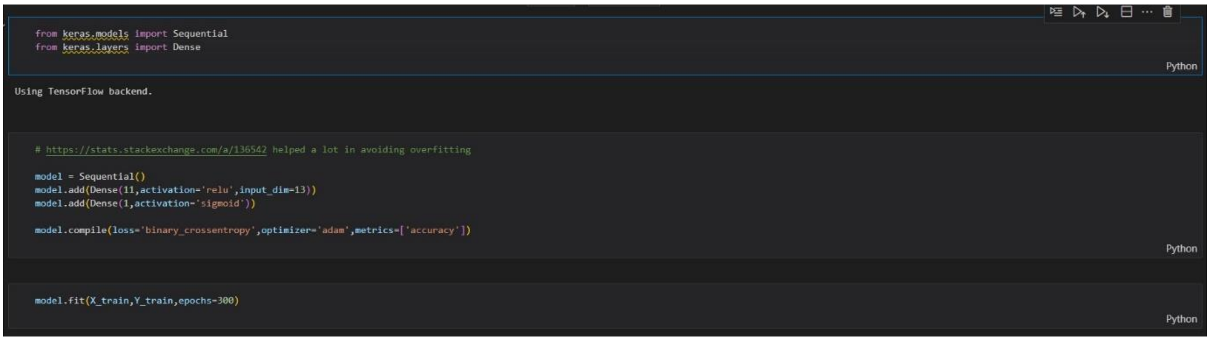
Fig 12 – XG Boost Model

- It was initially developed by Tianqi Chen and was described by Chen and Carlos Guestrin in their 2016 paper titled “XGBoost: A Scalable Tree Boosting System.”
- Extreme Gradient Boosting, or XG Boost for short is an efficient open-source implementation of the gradient boosting algorithm. As such, XGBoost is an algorithm, an open-source project, and a Python library
- Gradient boosting refers to a class of ensemble machine learning algorithms that can be used for classification or regression predictive modeling problems.
- Ensembles are constructed from decision tree models. Trees are added one at a time to the ensemble and fit to correct the prediction errors made by prior models. This is a type of ensemble machine learning model referred to as boosting.
- Models are fit using any arbitrary differentiable loss function and gradient descent optimization algorithm. This gives the technique its name, “gradient boosting,” as the loss gradient is minimized as the model is fit, much like a neural network.

The two main reasons to use XG Boost are execution speed and model performance.

- Generally, XG Boost is fast when compared to other implementations of gradient boosting. Szilard Pafka performed some objective benchmarks comparing the performance of XG Boost to other implementations of gradient boosting and bagged decision trees. He wrote up his results in May 2015 in the blog post titled “Benchmarking Random Forest Implementations.”
- His results showed that XGBoost was almost always faster than the other benchmarked implementations from R, Python Spark, and H2O.

4.8 NEURAL NETWORK



```
from keras.models import Sequential
from keras.layers import Dense
```

```
Using TensorFlow backend.
```

```
# https://stats.stackexchange.com/a/136542 helped a lot in avoiding overfitting

model = Sequential()
model.add(Dense(11,activation='relu',input_dim=13))
model.add(Dense(1,activation='sigmoid'))

model.compile(loss='binary_crossentropy',optimizer='adam',metrics=['accuracy'])
```

```
model.fit(X_train,Y_train,epochs=300)
```

Fig 13 – Neural Network Algorithm

Vectors, layers, and linear regression are some of the building blocks of neural networks. The data is stored as vectors, and with Python you store these vectors in arrays. Each layer transforms the data that comes from the previous layer. You can think of each layer as a feature engineering step, because each layer extracts some representation of the data that came previously.

One cool thing about neural network layers is that the same computations can extract information from any kind of data. This means that it doesn’t matter if you’re using image data or text data. The process to extract meaningful information and train the deep learning model is the same for both scenarios.

TYPES:

1. Convolutional neural network (CNN)
2. Recurrent neural network (RNN)
3. Deep Neural Network (DNN)

Advantages:

- They store information on the entire network, meaning that the neural network can continue functioning even if some information is lost from one part of the neural network.
- Once neural networks are trained with a quality data set, they save on costs and time as they take a shorter time to analyse data and present results.
- Neural networks provide quality and accuracy in results.

CHAPTER 5

SOFTWARE AND HARDWARE REQUIREMENTS

Hardware Requirements:

1.	System Processor	:	i5 / i7
2.	Hard Disk	:	500 GB.
3.	RAM	:	8 GB / 12 GB.

Any desktop / Laptop system with above configuration or higher level.

Software Requirements:

1.	Operating system	:	Windows 8/10 (64 bits)
2.	Coding language	:	Python 3
3.	Environment	:	Anaconda framework
4.	Tools	:	OpenCV

IDE : Jupyter Notebook

CHAPTER 6

SOURCE CODE

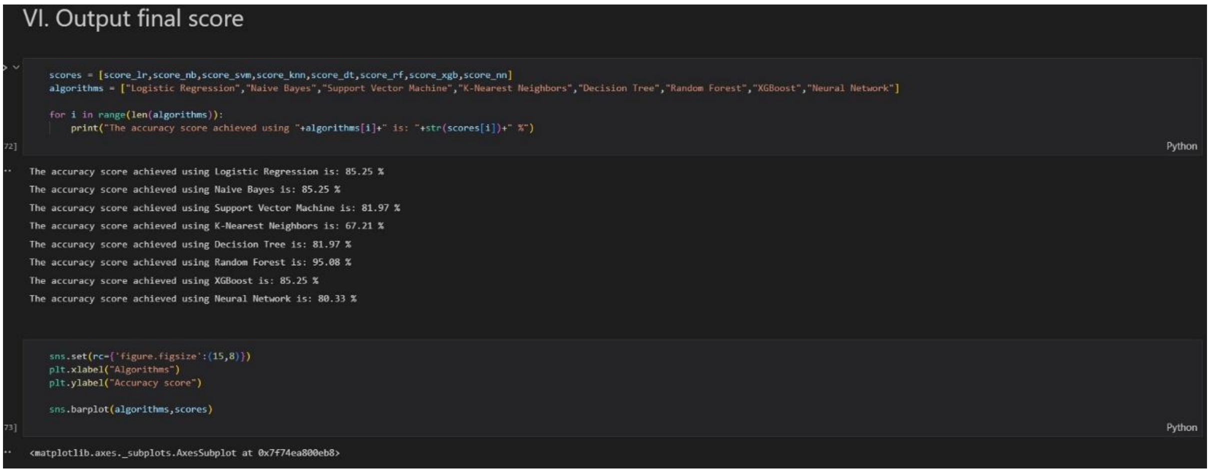
```
importing essential libraries

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns


%matplotlib inline

import os

print(os.listdir())


import warnings

warnings.filterwarnings('ignore')

importing and understanding our dataset

dataset = pd.read_csv("heart.csv")

type(dataset)

dataset.shape

dataset.shape

dataset.sample(5)

dataset.describe()

dataset.info()

###Luckily, we have no missing values
```

```
info = ["age","1: male, 0: female","chest pain type, 1: typical angina, 2: atypical angina, 3:
non-anginal pain, 4: asymptomatic","resting blood pressure"," serum cholestoral in
mg/dl","fasting blood sugar > 120 mg/dl","resting electrocardiographic results (values
0,1,2)"," maximum heart rate achieved","exercise induced angina","oldpeak = ST depression
induced by exercise relative to rest","the slope of the peak exercise ST segment","number of
major vessels (0-3) colored by flourosopy","thal: 3 = normal; 6 = fixed defect; 7 = reversable
defect"]

for i in range(len(info)):
    print(dataset.columns[i]+":\t\t\t"+info[i])

dataset["target"].describe()

dataset["target"].unique()

print(dataset.corr()["target"].abs().sort_values(ascending=False))

#This shows that most columns are moderately correlated with target, but 'fbs' is very weakly
correlated.

y = dataset["target"]

sns.countplot(y)

target_temp = dataset.target.value_counts()

print(target_temp)

print("Percentage of patience without heart problems:
"+str(round(target_temp[0]*100/303,2)))

print("Percentage of patience with heart problems: "+str(round(target_temp[1]*100/303,2)))

#Alternatively,

# print("Percentage of patience with heart problems: "+str(y.where(y==1).count()*100/303))

# print("Percentage of patience with heart problems: "+str(y.where(y==0).count()*100/303))

# #Or,

# countNoDisease = len(df[df.target == 0])

# countHaveDisease = len(df[df.target == 1])
```

```
dataset["sex"].unique()
sns.barplot(dataset["sex"],y)
dataset["cp"].unique()
sns.barplot(dataset["cp"],y)
dataset["fbs"].describe()
dataset["fbs"].unique()
sns.barplot(dataset["fbs"],y)
dataset["restecg"].unique()
sns.barplot(dataset["restecg"],y)
dataset["exang"].unique()
from sklearn.model_selection import train_test_split

predictors = dataset.drop("target",axis=1)
target = dataset["target"]
X_train,X_test,Y_train,Y_test = train_test_split(predictors,target,test_size=0.20,random_state=0)
from sklearn.tree import DecisionTreeClassifier

max_accuracy = 0
for x in range(200):
    dt = DecisionTreeClassifier(random_state=x)
    dt.fit(X_train,Y_train)
    Y_pred_dt = dt.predict(X_test)
    current_accuracy = round(accuracy_score(Y_pred_dt,Y_test)*100,2)
    if(current_accuracy>max_accuracy):
        max_accuracy = current_accuracy
        best_x = x
#print(max_accuracy)
#print(best_x)

dt = DecisionTreeClassifier(random_state=best_x)
dt.fit(X_train,Y_train)
Y_pred_dt = dt.predict(X_test)
scores = [score_lr,score_nb,score_svm,score_knn,score_dt,score_rf,score_xgb,score_nn]
algorithms = ["Logistic Regression","Naive Bayes","Support Vector Machine","K-Nearest Neighbors","Decision Tree","Random Forest","XGBoost","Neural Network"]

for i in range(len(algorithms)):
    print("The accuracy score achieved using "+algorithms[i]+" is: "+str(scores[i])+" %")
sns.set(rc={'figure.figsize':(15,8)})
plt.xlabel("Algorithms")
plt.ylabel("Accuracy score")
sns.barplot(algorithms,scores)
```

CHAPTER 7

OUTPUT FINAL SCORE



VI. Output final score

```
scores = [score_lr,score_nb,score_svm,score_knn,score_dt,score_rf,score_xgb,score_nn]
algorithms = ["Logistic Regression","Naive Bayes","Support Vector Machine","K-Nearest Neighbors","Decision Tree","Random Forest","XGBoost","Neural Network"]

for i in range(len(algorithms)):
    print("The accuracy score achieved using "+algorithms[i]+" is: "+str(scores[i])+" %")
```

```
The accuracy score achieved using Logistic Regression is: 85.25 %
The accuracy score achieved using Naive Bayes is: 85.25 %
The accuracy score achieved using Support Vector Machine is: 81.97 %
The accuracy score achieved using K-Nearest Neighbors is: 67.21 %
The accuracy score achieved using Decision Tree is: 81.97 %
The accuracy score achieved using Random Forest is: 95.08 %
The accuracy score achieved using XGBoost is: 85.25 %
The accuracy score achieved using Neural Network is: 80.33 %
```

```
sns.set(rc={'figure.figsize':(15,8)})
plt.xlabel("Algorithms")
plt.ylabel("Accuracy score")

sns.barplot(algorithms,scores)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f74ea800eb8>
```

Fig 14 – Accurate Output

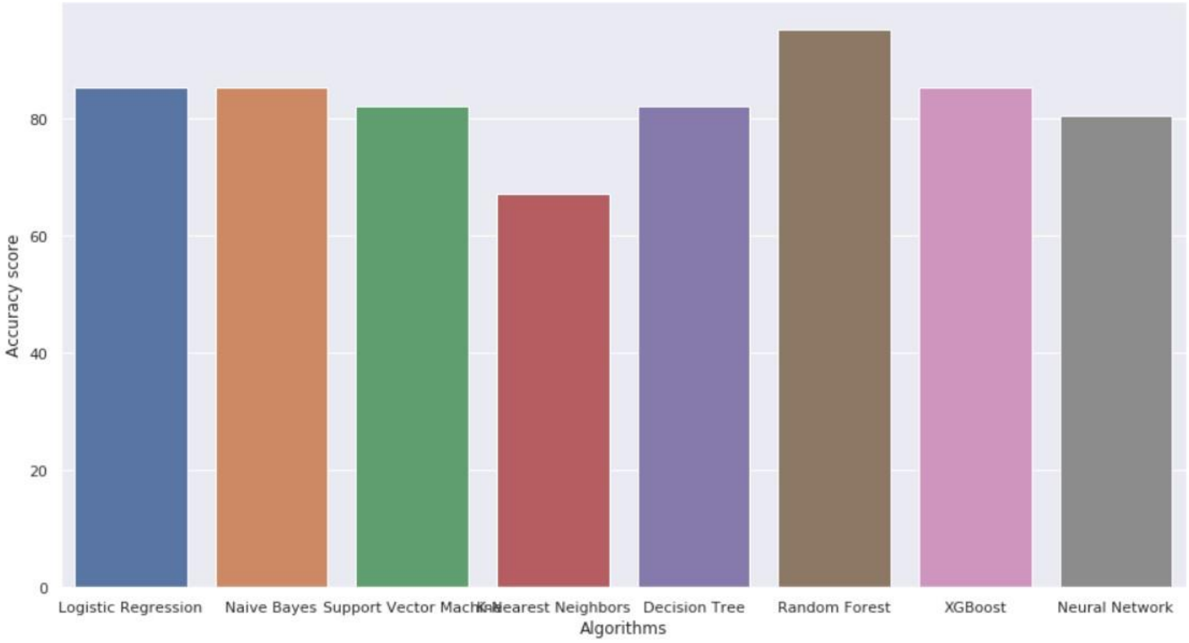


Fig 15 - Results

CONCLUSION

We proposed three methods in which comparative analysis was done and promising results were achieved. The conclusion which we found is that machine learning algorithms performed better in this analysis. Many researchers have previously suggested that we should use ML where the dataset is not that large, which is proved in this paper. The methods which are used for comparison are confusion matrix, precision, specificity, sensitivity, and F1 score. For the 13 features which were in the dataset, K Neighbors classifier performed better in the ML approach when data pre-processing is applied.

The computational time was also reduced which is helpful when deploying a model. It was also found out that the dataset should be normalized; otherwise, the training model gets overfitted sometimes and the accuracy achieved is not sufficient when a model is evaluated for real-world data problems which can vary drastically to the dataset on which the model was trained. It was also found out that the statistical analysis is also important when a dataset is analysed and it should have a Gaussian distribution, and then the outlier’s detection is also important and a technique known as Isolation Forest is used for handling this. The difficulty which came here is that the sample size of the dataset is not large. If a large dataset is present, the results can increase very much in deep learning and ML as well. The algorithm applied by us in ANN architecture increased the accuracy which we compared with the different researchers. The dataset size can be increased and then deep learning with various other optimizations can be used and more promising results can be achieved. Machine learning and various other optimization techniques can also be used so that the evaluation results can again be increased. More different ways of normalizing the data can be used and the results can be compared. And more ways could be found where we could integrate heart-disease-trained ML and DL models with certain multimedia for the ease of patients and doctors.

REFERENCES

1. **Mohammed Abdul Khaleel**

“A survey of data mining techniques on medical data for finding locally frequent diseases.” International Journal of Advanced Research in Computer Science and Software Engineering 2013/8.

2. **Gayathri p and N Jai Sankar**

“Comprehensive Study of Heart Disease Diagnosis Using Data Mining and Soft Computing Techniques”. International Journal of Engineering and Technology 2013/8.

3. **L. Sathish Kumar and A. Padma Priya**

“Prediction for Common Disease using ID3 Algorithm in Mobile Phone and Television” International Journal of Computer Applications (0975 – 8887) Volume 50 – No.4, July 2012.

4. **Ashir Javeed, Shijie Zhou et al.**

(2017) designed “An Intelligent Learning System Base Detection”.

5. www.google.com

6. www.bing.com

7. www.youtube.com

8. www.IEEE.org