EFFICIENT CLASSIFICATION ON THE BASIS OF

DECISION TRESS

A DISSERTATION

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE

AWARD OF THE DEGREE OF

MASTER OF TECHNOLOGY

IN

SIGNAL PROCESSING AND DIGITAL DESIGN

Submitted by:

MUKUL

2K17/SPD/08

Under the supervision of

PROF. RAJESH ROHILLA

DEPARTMENT OF ELECTRONICS AND COMMUNICATION
DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
Bawana Road, Delhi-110042

OCTOBER , 2019

CANDIDATE'S DECLARATION

I, MUKUL, 2K17/SPD/08, of M.Tech, hereby declare that the project Dissertation Titled "EFFICIENT CLASSIFICATION ON THE BASIS OF DECISION TREES" which is submitted by me to the Department of Electronics and Communication, Delhi Technological University, Delhi in partial fulfilment of the requirement for the award of the degree of Master of Technology, is original and not copied from any source without proper citation. This work has not previously formed the basis for the award of any Degree, Diploma Associateship, Fellowship or other similar title or recognition.

Place: Delhi MUKUL
Date: 31-10-2019

CERTIFICATE

I hereby certify that the Project Dissertation titled "EFFICIENT CLASSIFICATION ON THE BASIS OF DECISION TREES" which is submitted by MUKUL, Roll No 2K17/SPD/08, Delhi Technological University, Delhi in partial fulfilment of the requirement for the award of the degree of Master of Technology, is a record of the project work carried out by the students under my supervision. To the best of my knowledge this work has not been submitted in part or full for any Degree or Diploma to this University or elsewhere.

Place: Delhi

Date: 31-10-2019

PROF. RAJESH ROHILLA

Professor

Department of electronics and communication

Delhi Technological University, Delhi

SUPERVISOR

ACKNOWLEDGEMENT

I thank almighty GOD for guiding me throughout the semester. I would like to thank all those who have contributed to the completion of my project and helped me with valuable suggestions for improvement.

I am extremely grateful to Prof. RAJESH ROHILLA, Division of Electronics and Communication, for providing me with best facilities and atmosphere for the creative work guidance and encouragement.

Above all I would like to thank my parents without whose blessings, I would not have been able to accomplish my goal.

……

MUKUL

ABSTRACT

Machine learning has gained a great interest in last decade among the researchers. Having such a great community which provide a continuously growing list of proposed algorithms, it is rapidly finding solutions to its problem. Therefore more and more people are entering the field of machine learning to make the idea of machine learning algorithms more useful and reliable for the digital world.

In this thesis , efficient and accurate classification performances by various machine learning algorithms on the proposed heart disease prediction dataset is being discussed.The dataset cosists of 14 cloumns and 1026 rows. First 13 columns consists of Predictor variables and the last column is our target variable, which consists of Categorical values (0 and 1). Our main focus is to find the best fit model on the above proposed dataset. The machine learning algorithms that are applied and compared are KNN, SVM and DECISION TREES CLASSIFIERS. The programming tool used is python 3.7 and jupyter notebook installed in our systems. The libraries that are installed from external source to our system for plotting decision trees is graphviz and pydotplus.

This thesis comprises of six chapters. Chapter 1 is our introduction part to have brief review about the used technologies in our research work process. Chapter 2 is our literature review section which basically focuses about all the past works that have been made till date on the above mentioned dataset and research ideology. Chapter 3 consists of proposed machine learning algorithms workings and their respective Methodologies in detail.

Chapter 4 is our results and discussion part in which we have evaluated the Performances of various machine learning algorithms used in our research work and finding out best performing Algorithm.

Chapter 5 is our conclusion section in which we have concluded the best fitting Machine learning algorithm model on the basis of results and discussions in chapter 4.

Chapter 6 is our refrences section. The brief review of this thesis is , to find the best fit model With highest accuracy and with minimum number of misclassification on the given heart disease Prediction dataset.

CONTENTS

LIST OF FIGURES:

LIST OF TABLES:

CHAPTER 1

INTRODUCTION

Heart is the next major organ comparing to brain which has more priority in Human body. It pumps the blood and supplies to all organs of the whole body. Prediction of occurrences of heart diseases in medical field is significant work. Data analytics is useful for prediction from more information and it helps medical centre to predict of various disease. Huge amount of patient related data is maintained on monthly basis. The stored data can be useful for source of predicting the occurrence of future disease. Some of the data mining and machine learning techniques are used to predict the heart disease, such as Artificial Neural Network (ANN), Fuzzy Logic, K-Nearest Neighbour(KNN), Naïve Bayes and Support Vector Machine (SVM).

Heart disease is one of the prevalent disease that can lead to reduce the lifespan of human beings nowadays. Each year 17.5 million people are dying due to heart disease [6]. Life is dependent on component functioning of heart, because heart is necessary part of our body. Heart disease is a disease that affects on the function of heart [7]. An estimate of a person's risk for coronary heart disease is important for many aspects of health promotion and clinical medicine. A risk prediction model may be obtained through multivariate regression analysis of a longitudinal study [8]. Due to digital technologies are rapidly growing, healthcare centres store huge amount of data in their database that is very complex and challenging to analysis. Data mining techniques and machine learning algorithms play vital roles in analysis of different data in medical centres. The techniques and algorithms can be directly used on a dataset for creating some models or to draw vital conclusions, and inferences from the dataset. Common attributes

used for heart disease are Age, Sex, Fasting Blood Pressure, Chest Pain type, Resting ECG(test) that measures the electrical activity of the heart), Number of major vessels colored by fluoroscopy, Threst Blood Pressure (high blood pressure), Serum Cholestrol (determine the risk for developing heart disease), Thalach (maximum heart rate achieved), ST depression (finding on an electrocardiogram, trace in the ST segment is abnormally low below the baseline), painloc (chest pain location (substernal=1, otherwise=0)), Fasting blood sugar, Exang (exercise included angina), smoke, Hypertension, Food habits, weight, height and obesity. The prominent heart disease now a days are :-

Arrhythmia -The heart beat is improper whether it may irregular, too slow or too fast.

Cardiac arrest -An unexpected loss of heart function, consciousness and breathing occur suddenly.

Congestive heart failure- The heart does not pump blood as well as it should, it is the condition of chronic.

Congenital heart disease -The heart's abnormality which develops before birth.

Coronary artery disease .The heart's major blood vessels can damage or any disease occurs in the blood vessels.

High Blood Pressure- It has a condition that the force of the blood against the artery walls is too high.

Peripheral artery disease -The narrowed blood vessels which reduce flow of blood in the limbs, is the circulatory condition.

Stroke -Interruption of blood supply occur damage to the brain.

The widespread use of computers in today's society means that large quantities of data are stored electronically. This data relates to virtually all the sides of modern life and is a valuable resource if the right tools are available for putting it to use.

Machine learning algorithms are a set of techniques that automatically build models describing the structure at the heart of a set of data.

Above stated prototypes has two important applications . First, if they precisely show the model representing the data, it can be used to predict the behaviour of future data sets . Second, if they recapitulate the significant information in human-readable and writable form, people can use them to investigate the domain from which it is originating. These two demonstrations are not jointly exclusive. For useful for analysis, a model should be a precise illustration of the domain, and that making it useful for prediction purpose as well.

1.1 MACHINE LEARNING:

The concept of Machine Learning deals with the design of programs that can learn rules from data, adapt to changes, and improve performance with experience. In addition with one of the initial dreams of Computer Science, Machine Learning has now become more crucial as computers are projected to solve increased intricate problems and complications and has now become more integrated into our day to day lives.

Machine Learning Theory

Machine Learning Theory, also known as Computational Learning Theory, goals to understand the essential principles of learning as a computational process. This field looks to recognize on a precise and accurate mathematical grounds ,what skils and information are basically required

to pick up various kinds of jobs and tasks successfully, and to realise the fundamental algorithmic principles tangled in making computers to learn and recognize from given data and to analyse and improve the presentation performance with process of feedback. The goals of above stated theory are both to benefit in the design of improved programmed learning methods and to realize necessary concerns in the learning process itself.

The theory of computation and statistics forms the basis of machine learning and consists of following tasks such as:

• Mathematical models are being created and key aspects of machine learning are being captured, which helps in analyzing different types of learning problems ,how inherently easy and difficult they are going to be.

• Helps in defining favourable conditions for different machine learning algorithms under what conditions they all will perform better and how much computational time all machine learning algorithms are going to take and meeting the desired conditions.

Applications of machine learning:

Most industries have realised the value of machine learning and use various machine algorithms to process large and heavy data problems. By gleaning insights from this data – often in real time – organizations are able to work more efficiently or gain an advantage over competitors. Various important applications are as :

1. Financial services

2. Government agencies

3. Marketing and sales

4. Health and care institutions

5. Tranportation

6. Oil and gases department

1.2 MACHINE LEARNING ALGORITHMS:

Broadly, there are three types of Machine Learning Algorithms

1. Supervised Learning:

How it works: Under this algorithm we already have predetermined target or outcome variable which is called independent variables .These independent variable are to be predicted from dependent varibles which are called predictors Using these set of variables, we generate a function that map inputs to desired outputs. The training process continues until the model achieves a desired level of accuracy on the training data. Examples of Supervised Learning: Regression, **Decision Trees, Random Forest, KNN**, Logistic Regression.

.

2. Unsupervised Learning:

How it works: : In this algorithm, no target or outcome variable is present initially that is to be predicted. The independent variables or predictor variable are passed through these unsupervised algorithms to build a random model in order to predict the dependent varibles or target variable randomly. It is used for clustering population in different groups, which is

widely used for segmenting customers in different groups for specific intervention. Examples of Unsupervised Learning: Clustering algorithm like K-means,spectral clustering , PCA etc

3. Reinforcement Learning:

How it works: : Using this algorithm, the machine is trained to make specific decisions. It works this way: the machine is exposed to an environment where it trains itself continually using trial and error. This machine learns from past experience and tries to capture the best possible knowledge to make accurate business decisions. Example of Reinforcement Learning: Markov Decision Process

Machine learning algorithms that are going to be used are :

1. Decision tree classifiers
2. K-nearest neighbours (KNN)
3. Support vector machines(SVM)

ALGORITHMS USED:

DECISION TREES: A decision support tool that uses a tree-like graph or model of decisions and their possible consequences signifying chance event outcomes, resource costs and utility. It is an algorithm that is purely based on conditional control statements.

It is basically a flowchart-like model in which each node represents a experimental test on a feature or attribute (eg.whether a person is having cancer or not) , outcome of the test is being

represented by the branches and each leaf node is being associated with a class label(target variable). The pathways connecting root to leaf represents classification rules. It is one of the most powerfully used algorithms in the category of supervised algorithms.[14]

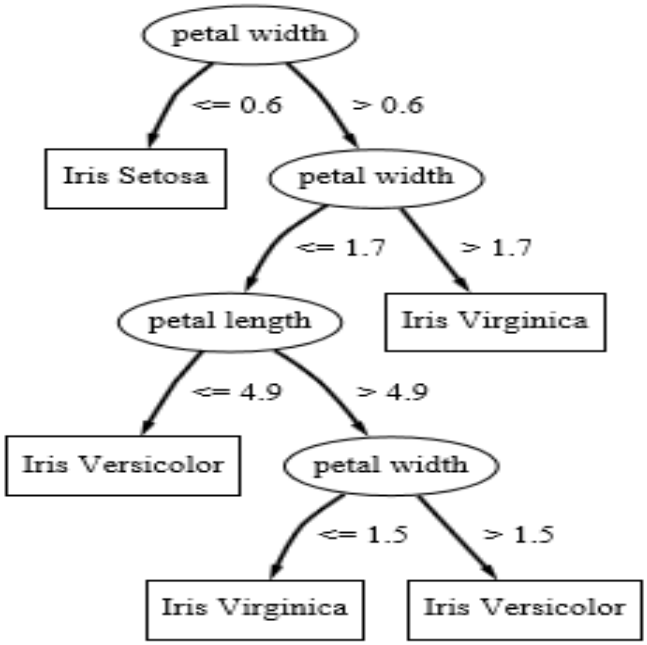


Figure 1.1: Classification of iris flowers with the help of decision trees.

	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa
2	4.7	3.2	1.3	0.2	setosa
3	4.6	3.1	1.5	0.2	setosa
4	5.0	3.6	1.4	0.2	setosa
5	5.4	3.9	1.7	0.4	setosa
6	4.6	3.4	1.4	0.3	setosa
7	5.0	3.4	1.5	0.2	setosa
8	4.4	2.9	1.4	0.2	setosa
9	4.9	3.1	1.5	0.1	setosa

Figure 1.2 : head of iris data set

KNN (K-NEAREST NEIGHBOURS) :

KNN , for pattern classification and regression one of the most oldest and efficient supervised algorithm is k nearst neighbour. It has the ability of creating a powerful distinguishable classifier. The kNN rule classifies each unlabeled example by the majority label of its k-nearest neighbors in the training dataset. Despite its simplicity, the kNN rule often yields competitive results. A recent work on prototype reduction, called Weighted Distance Nearest Neighbor (WDNN) is based on retaining the informative instances and learning their weights for classification. The algorithm assigns a non negative weight to each training instance tuple at the training phase. Only the training instances with positive weight are retained (as the prototypes) in the test phase. Although the WDNN algorithm is well formulated and shows encouraging performance, in practice it can only work with K = 1. A more recent approach WDkNN tries to reduce the time complexity of WDNN and extend it to work for values of K greater than 1.

In the recent past, a lot of research centered at nearest neighbor methodology has been done. However one of the major drawbacks of kNN is that, it is a lazy learner i.e. it uses all the training data at the runtime. The accuracy of kNN highly depends upon the distance metric used. Euclidean distance is a simple and efficient method for computing distance between two reference data points. More complex distance functions may provide better results depending on the dataset and domain. But user may refrain from using a better, generally computationally more complex, distance metric due to high run time of the algorithm. This motivated us to strive for an algorithm which has a significantly low run time and hence can incorporate expensive distance metrics with ease.

SVM (SUPPORT VECTOR MACHINE):

Support Vector Machine is a new data mining paradigm applied for regression. Viewing input data as two sets of vectors in an n-dimensional space, an SVM will construct a separating hyperplane in that space that maximizes the margin between the two datasets. To calculate the margin, two parallel hyperplanes are constructed, one on each side of the separating hyperplane, which are pushed up against the two data sets. Intuitively, a good separation is achieved by the hyperplane that has the largest distance to the neighboring data points of both classes, since in general the larger the margin the lower the generalization error of the classifier.

It is a classification method. In this algorithm, we plot each data item as a point in n-dimensional space (where n is number of features we have) with the value of each feature being the value of a particular coordinate.

For example, if we only had two features like Height and Hair length of an individual, we'd first plot these two variables in two dimensional space where each point has two co-ordinates (these co-ordinates are known as **Support Vectors**)

Now, we will find some *line* that splits the data between the two differently classified groups of data. This will be the line such that the distances from the closest point in each of the two groups will be farthest away

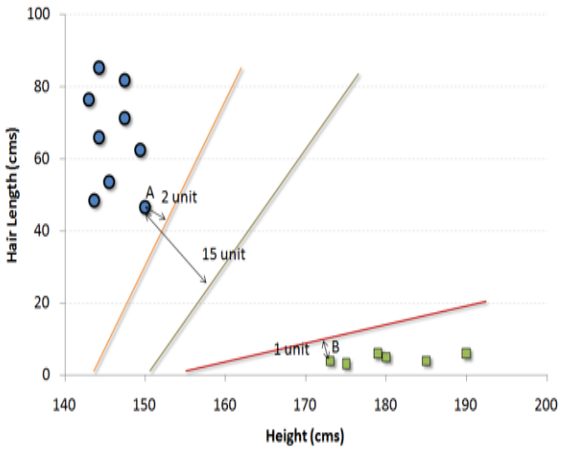


Figure 1.3 : A binary(2 class) SVM

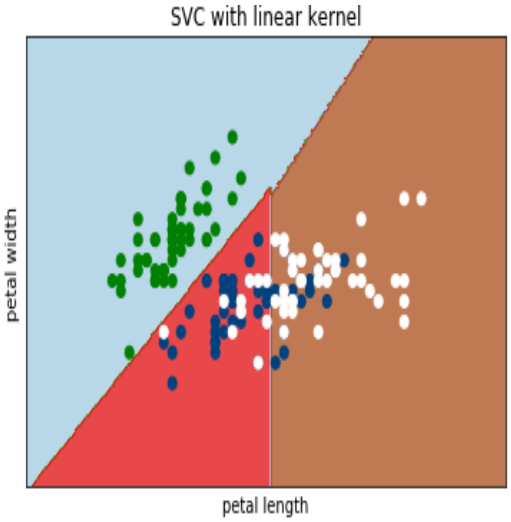


Figure 1.4: A multiclass(3 class) SVM

1.3 MOTIVATION:

In todays busy world heart disease are becoming very much prominent due to unhealthy life Style. The consequences are that every other person is facing the risk of having heart disease Problems in future. The dataset that is being used in our study is giving us proper information About all the important aspects that can play a major role in predicting the possibility of , Whether a given person is going to have a risk of heart disease in his/her future life.

The motivation behind the implementation various machine learning algorithms is to find the Best fit model among various machine learning algorithms models in order to predict the risk Having heart disease among the given human beings with highest precision and accuracies in In their future lives. The idea is to implement various machine learning algorithms (KNN,SVM and decision trees) and comparing their best possible accuracies.

1.4 Related work :

In recent past researchs is being conducted on heart disease prediction dataset with the help of KNN and neural networks methods. Some of the data mining and machine learning techniques are used to predict the heart disease, such as Artificial Neural Network (ANN), Fuzzy Logic, K-Nearest Neighbour(KNN), Naïve Bayes and Support Vector Machine (SVM). Howeever decision trees has not been implemented yet so we are going to look at the possible outcomes after implementing decision tree classifiers and will see whether its going to perform better as compared to other supervised algorithm that is KNN(k-nearest neighbour). KNN is the most recent work that is being conducted in the recent past.

1.5 Organization of Thesis :

- In chapter 1, we have a brief discussuion about machine learning and its algorithms that we are going to discuss in detail in upcoming chapters.

- Chapter 2 is mainly focussed on literature review section. This chapter represents all the past works that had been conducted already on our heart disease dataset.

- Chapter 3 is basically our methodology part .In chapter 3 working methods of various algorithhms are being dicussed that is going to be used on **heart disease prediction dataset .**Algorithms which are going to be implemented and discussed are **decision tree classifiers** and **KNN (k-nearest neighbours.** Our main aim is to prdedict the possibility of having heart heart diseases in the coming future for a particular human being with help of given heart disease prediction dataset.

- Chapter 4 is our results and discussions part. In chapter 4 performances of various machine learning algorithms on the proposed heart disease dataset are discussed. In this following chapter we discuss the experimental results which are being obtained after working on python in jupyter notebook software. Jupyter notebook is a powerful tool which mainly focuses on python. The version used is python 3.7.

- Chapter 5 is our conclusion section. After looking all the previous results that is being obtained in chapter 4 , we have concluded our best fit model that is going to be decision tree classifiers as compared to KNN and SVM.

- Chapter 6 is basically our refrences section with all the refrences that we have gone through to represent our study.

CHAPTER 2

LITERATURE REVIEW:

There are abundant works have been done related to disease prediction systems using different data mining techniques and machine learning algorithms in medical centres. K. Polaraju et al, [11] proposed Prediction of Heart Disease using Multiple Regression Model and it proves that Multiple Linear Regression is appropriate for predicting heart disease chance. The work is performed using training data set consists of 3000 instances with 13 different attributes which has mentioned earlier. The data set is divided into two parts that is 70% of the data are used for training and 30% used for testing. Based on the results, it is clear that the classification accuracy of Regression algorithm is better compared to other algorithms. Marjia et al, [12] developed heart disease prediction using KStar, j48, SMO, and Bayes Net and Multilayer perception using WEKA software. Based on performance from different factor SMO and Bayes Net achieve optimum performance than KStar, Multilayer perception and J48 techniques using kfold cross validation. The accuracy performances achieved by those algorithms are still not satisfactory. Therefore, the accuracy's performance is improved more to give better decision to diagnosis disease. S. Seema et al,[13] focuses on techniques that can predict chronic disease by mining the data containing in historical health records using Naïve Bayes, Decision tree, Support Vector Machine(SVM) and Artificial Neural Network(ANN). A comparative study is performed on classifiers to measure the better performance on an accurate rate. From this experiment, SVM gives highest accuracy rate, whereas for diabetes Naïve Bayes gives the highest accuracy. Ashok Kumar Dwivedi et al, [14] recommended different algorithms like Naive Bayes, Classification Tree, KNN, Logistic Regression, SVM and ANN. The Logistic

Regression gives better accuracy compared to other algorithms. MeghaShahi et al, [15] suggested Heart Disease Prediction System using Data Mining Techniques. WEKA software used for automatic diagnosis of disease and to give qualities of services in healthcare centres. The paper used various algorithms like SVM, Naïve Bayes, Association rule, KNN, ANN, and Decision Tree. The paper recommended SVM is effective and provides more accuracy as compared with other data mining algorithms. Chala Beyene et al, [16] recommended Prediction and Analysis the occurrence of Heart Disease Using Data Mining Techniques. The main objective is to predict the occurrence of heart disease for early automatic diagnosis of the disease within result in short time. The proposed methodology is also critical in healthcare organisation with experts that have no more knowledge and skill. It uses different medical attributes such as blood sugar and heart rate, age, sex are some of the attributes are included to identify if the person has heart disease or not. Analyses of dataset are computed using WEKA software. R. Sharmila et al, [17] proposed to use non- linear classification algorithm for heart disease prediction. It is proposed to use bigdata tools such as Hadoop Distributed File System (HDFS), Mapreduce along with SVM for prediction of heart disease with optimized attribute set. This work made an investigation on the use of different data mining techniques for predicting heart diseases. It suggests to use HDFS for storing large data in different nodes and executing the prediction algorithm using SVM in more than one node simultaneously using SVM. SVM is used in parallel fashion which yielded better computation time than sequential SVM. Jayami Patel et al, [18] suggested heart disease prediction using data mining and machine learning algorithm. The goal of this study is to extract hidden patterns by applying data mining techniques. The best algorithm J48 based on UCI data has the highest accuracy rate compared to LMT. Purushottam et al, [19] proposed an efficient heart disease prediction system using data mining. This system helps medical practitioner to make effective decision

making based on the certain parameter. By testing and training phase a certain parameter, it provides 86.3% accuracy in testing phase and 87.3% in training phase. K.Gomathi et al, [20] suggested multi disease prediction using data mining techniques.Nowadays, data mining plays vital role in predicting multiple disease. By using data mining techniques the number of tests can be reduced. This paper mainly concentrates on predicting the heart disease, diabetes and breast cancer etc., P.Sai Chandrasekhar Reddy et al, [21] proposed Heart disease prediction using ANN algorithm in data mining. Due to increasing expenses of heart disease diagnosis disease, there was a need to develop new system which can predict heart disease. Prediction model is used to predict the condition of the patient after evaluation on the basis of various parameters like heart beat rate, blood pressure, cholesterol etc. The accuracy of the system is proved in java. Ashwini shetty et al, [22] recommended to develop the prediction system which will diagnosis the heart disease from patient's medical dataset. 13 risk factors of input attributes have taken into account to build the system. After analysis of the data from the dataset, data cleaning and data integration was performed. Jaymin Patel et al, [23] suggested data mining techniques and machine learning to predict heart disease. There are two objectives to predict the heart system. 1. This system not assume any knowledge in prior about the patient's records. 2. The system which chosen must be scalar to run against the large number of records. This system can be implemented using WEKA software. For testing, the classification tools and explorer mode of WEKA are used.. Noura Ajam [25] recommended artificial neural network for heart disease diagnosis. Based on their ability, Feed forward Back propogation learning algorithms have used to test the model. By considering appropriate function, classification accuracy reached to 88% and 20 neurons in hidden layer. ANN shows result significantly for heart disease prediction. Prajakta Ghadge et al, [26] suggested big data for heart attack prediction. The objective of this paper is to provide prototype using big data and data modelling

techniques.It can be also used to extract patterns and relationships from database which associated with heart disease. This system consists of two databases namely, original big dataset and another is updated one. A java-file system named HDFS used to provide a user with reliable. This system can assist the healthcare practitioners to make intelligent decisions. The automation in this system would be advantageous. Sairabi H.Mujawar et al, [28] used k-means and naïve bayes to predict heart disease. This paper is to build the system using historical heart database that gives diagnosis. 13 attributes have considered for building the system. To extract knowledge from database, data mining techniques such as clustering, classification methods can be used. 13 attributes with total of 300 records were used from the Cleveland Heart Database. This model is to predict whether the patient have heart disease or not based on the values of 13 attributes. KNN is basically the most recent work that have been conducted on the proposed heart disease prediction dataset.

KNN is properly based on determining the value of parameter K which signifies the no. of nearest neighbours. KNN is basically a distance based learning which doesn't give us appropriate and best results in various cases. Also computation cost is quite high for KNN because we need to compute the distance of each querry instance to all training examples which makes it a slow learning algorithm as well.

On the other hand decision tress is a fast learning classifier algorithm as it uses entropy, information gain as its classifying criterion. As Compared to other basic algorithms,decision trees needs less effort for data interpretation during pre-processing.

A decision tree can deal with any form of data whether it is categorical type or continuous type. Basically, it does not require normalization and scaling of data so the data remain in its actual

form and we get the best representation of our outcome model. If the data have some missing values,the process of building the decision tree will not be affected to any significant extent.The best part of implementing decision trees is its simple way of representing the decision tree model. They are easy to explain to the technical teams as well to the big firms owners and stakeholders . However , decision trees involve higher time to train a model and they are computational expensive as well for large datasets where complexity is more. Decision trees also sometimes become inadequate to regression problems and for predictiong continuous values. Another algorithm that is focussed and implemented on the proposed heart disease dataset is support vector machines(SVM). Support vector machines works relly well when we are having clearly separable classes and we have greater dimensional spaces than our no of samples. SVM does not perform very well on large datasets as compared to decision trees which are relatively more efficient . The basic strength of using SVM is its ideology of using different kernels . Usage of different kernels provides support vector machines an extra benefit to analyse the data with different approaches.

Chapter 3

WORKING PROCEDURES OF PROPOSED ALGORITHM:

3.1 CLASSIFICATION:

In this subsection, we describe the problem of classification and notation used to model the dataset. The problem of classification is to estimate the value of the class variable based on the values of one or more independent variables (known as feature variables). We model the tuple as {x, y} where x is an ordered set of attribute values like $\{X_1, X_2, ..., X_d\}$ and y is the class variable to be predicted. Here xi is the value of the ith attribute and there are d attributes overall corresponding to a d-dimensional space. Formally, the problem has the following inputs:

• A set of n tuples called the training dataset, D = $\{(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)\}$.

• A query tuple X_t.

The output is an estimated value of the class variable for the given query xt, mathematically it can be expressed as:

$$y_t = f(X_t, D, \text{parameters}) \qquad \text{Eq (1.1)}$$

Where parameters are the arguments that the function f() takes. These are generally set by the user or are learned by some method.[5]

3.2 METHODOLOGY:

In this section we are going to discuss the working methodology of used machine machine learning algorithms.

MATHEMATICAL MODEL OF KNN

In this subsection, we present a mathematical model for KNN algorithma and show that KNN only makes use of local prior probabilities for classification.

For a given query instance Xt, KNN algorithm works as follows:

$$y_t = \underset{c \in \{c_1, c_2, \ldots, c_m\}}{\arg\max} \sum_{x_i \in N(x_t, k)} E(y_i, c)$$

Eq:(1.2)

Where yt is the predicted class for the query instance Xt and m is the number of classes present in the data. Also

$$E(a,b) = \begin{cases} 1 & if a = b \\ 0 & else \end{cases}$$

$$N(x,k) = Set\ of\ k\ nearest\ neighbor\ of\ x$$

Eq:(1.3)

Eq (1.3) can also be written as

$$y_t = \arg\max \left\{ \sum_{x_i \in N(x_t,k)} E(y_i, c_1), \sum_{x_i \in N(x_t,k)} E(y_i, c_2), \quad \dots \quad , \sum_{x_i \in N(x_t,k)} E(y_i, c_m) \right\}$$

$$y_t = \arg\max \left\{ \sum_{x_i \in N(x_t,k)} \frac{E(y_i, c_1)}{k}, \sum_{x_i \in N(x_t,k)} \frac{E(y_i, c_2)}{k}, \quad \dots \quad , \sum_{x_i \in N(x_t,k)} \frac{E(y_i, c_m)}{k} \right\}$$

And we know that

$$p(c_j)_{(x_t,k)} = \sum_{x_i \in N(x_t,k)} \frac{E(y_i, c_j)}{k} \qquad \text{Eq:(1.4)}$$

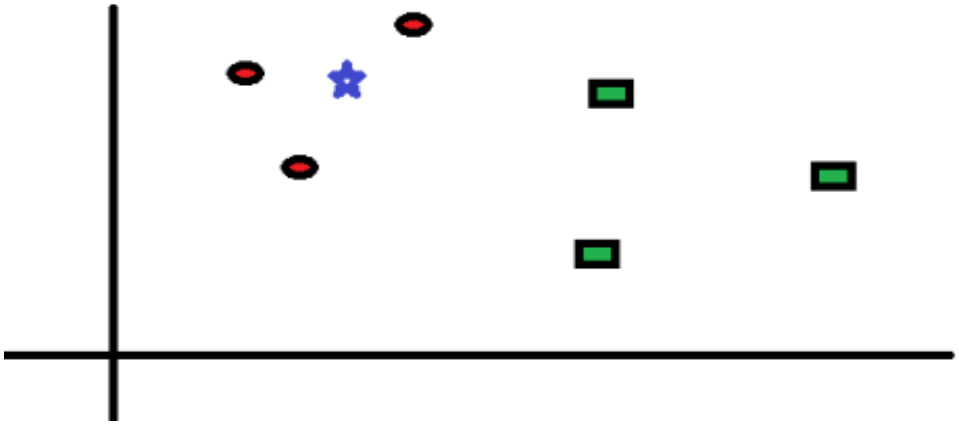
Where $p(cj)_{(xt,k)}$ is the probability of occurrence of jth class in the neighborhood of Xt . Hence Eq. 1.3 turns out to be :

$$y_t = \arg\max \{ p(c_1)_{(x_t,k)}, p(c_2)_{(x_t,k)}, \dots, p(c_m)_{(x_t,k)} \} \qquad \text{Eq:(1.5)}$$

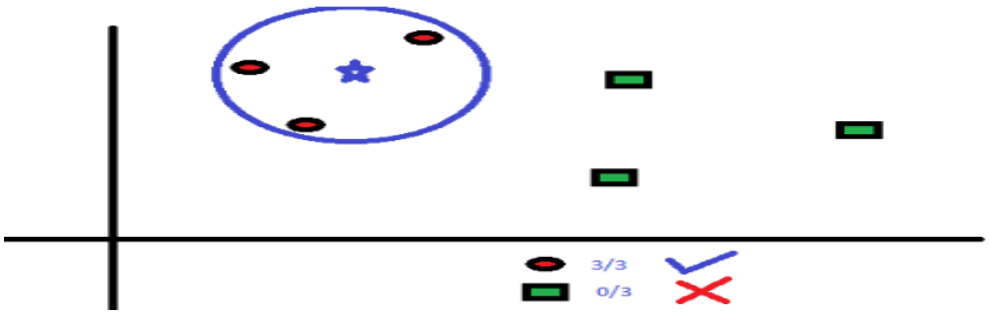
It is clear from Eq. 1.5, that KNN algorithm uses only prior probabilities to calculate the class of the query instance. It ignores the class distribution around the neighborhood of query point.[24]

How does the KNN algo works:

Let's take a simple case to understand this algorithm. Following is a spread of red circles (RC) and green squares (GS) :



You intend to find out the class of the blue star (BS). BS can either be RC or GS and nothing else. The "K" is KNN algorithm is the nearest neighbours we wish to take vote from. Let's say K = 3. Hence, we will now make a circle with BS as center just as big as to enclose only three data points on the plane.



The three closest points to BS is all RC. Hence, with good confidence level we can say that the BS should belong to the class RC. Here, the choice became very obvious as all three votes from the closest neighbor went to RC. The choice of the parameter K is very crucial in this algorithm.

IMPLEMENTATION IN PYTHON:

Before going through python implementation first we should look at the heart disease dataset onto which algorithms are being applied.

Data set contain following features:

Age- age in years

Sex – (1= male ; 0=female)

Cp = chest pain type

Trestbps = resting blood pressure (in mm hg on admission to the hospital)

Chol – serum cholesterol in mg/dl

Fbs – (fasting blood sugar>120mg/dl) (1= true ; 0=false)

Restecg – resting electrocardiographic results

Thalach – maximum heart rate achieved

Exang – exercise induced antiga (1=yes; 0=no)

Oldpeak – ST depression induced by exercise relative to rest

Slope – the slope of the peak exercise ST segment

Ca – number of major vessels (0-3) colored by flourosopy

Thal – 3=normal; 6= fixed defect; 7=reversible defect

Target – have disease or not (1=yes, 0=no)

The above dataset contains 14 features in particular. The **target** variable provide us the information about the presence of heart disease in the patient

PYTHON IMPLEMENTAION FOR KNN:

We will look into this process in following steps:

1. LOAD ALL THE REQUIRED LIBRARIES

```
Import pandas as pd

Import  numpy as np

Import  matplotlib.pyplot as plt

Import  seaborn as sns

%matplotlib inline
```

from sklearn.cross_validation import train_test_split

from sklearn import metrics

from sklearn.metrics import confusion matrix ,accuracy score(for accuracy comparison purposes)

from sklearn.neighbours import KNeighborsClassifier (for importing KNN classifier from scikit learn).

2. **LOADING OUR DATASET** :

```
df = pd.read_csv('C:/Users/DELL/Downloads/heart.csv',header=0)
```

The above command is basically loading and converting our dataset into a dataframe. The head of our dataframe can be seen as

```
df.head()
```

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
0	52	1	0	125	212	0	1	168	0	1.0	2	2	3	0
1	53	1	0	140	203	1	0	155	1	3.1	0	0	3	0
2	70	1	0	145	174	0	1	125	1	2.6	0	0	3	0
3	61	1	0	148	203	0	1	161	0	0.0	2	1	3	0
4	62	0	0	138	294	1	1	106	0	1.9	1	3	2	0

3. Next we split our dataset in the ratio of 70:30. It means 70% dataset is being taken for training purpose and 30% of the data is being reserved for test purpose. Independent variables (x) is

assigned with 13 features columns except the target variable column and dependent variable(y) is assigned with the classification variable ie. Target variable which is basically is a categorical variable (0,1).

```
X = df.drop('target',axis=1)
y = df['target']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.30, random_state=101)
```

4. Our next step is our **Training part** , We are training our KNN model with the help of assigned 70% of the dataset values(X_train,Y_train). This can be achieved with the help of following commands.

```
knn = KNeighborsClassifier(n_neighbors=3)
knn.fit(X_train, y_train)
```

5. Next step is our **prediction part**. It involves prediction of values of test dataset(pred) with the help of assigned 30% of the assigned independent test dataset values(X_test).

```
pred = knn.predict(X_test)
```

6. This step includes clalculation of confusion matrix and accuracy score so as to determine up to what extent our model is performing well.

```
print(classification_report(y_test,pred))
```

```
print(confusion_matrix(y_test,pred))
```

```
accuracy_score(y_test,pred)
```

7. This step includes predicting values of k which plays an important role in determining the accuracy of KNN based classifier model. Under this step we plot a graph b/w error rate and k so as to determine the optimal value oh k for which lower error rate value is achieved. Here k is basically nothing but no of neighbours for determing the class of given feature.

```
error_rate = []
```

```
for i in range(1,40):

    knn = KNeighborsClassifier(n_neighbors=i)
    knn.fit(X_train,y_train)
    pred_i = knn.predict(X_test)
    error_rate.append(np.mean(pred_i != y_test))
```

```
plt.figure(figsize=(10,6))
plt.plot(range(1,40),error_rate,color='blue', linestyle='dashed', marker='o',
         markerfacecolor='red', markersize=10)
plt.title('Error Rate vs. K Value')
plt.xlabel('K')
plt.ylabel('Error Rate')
```

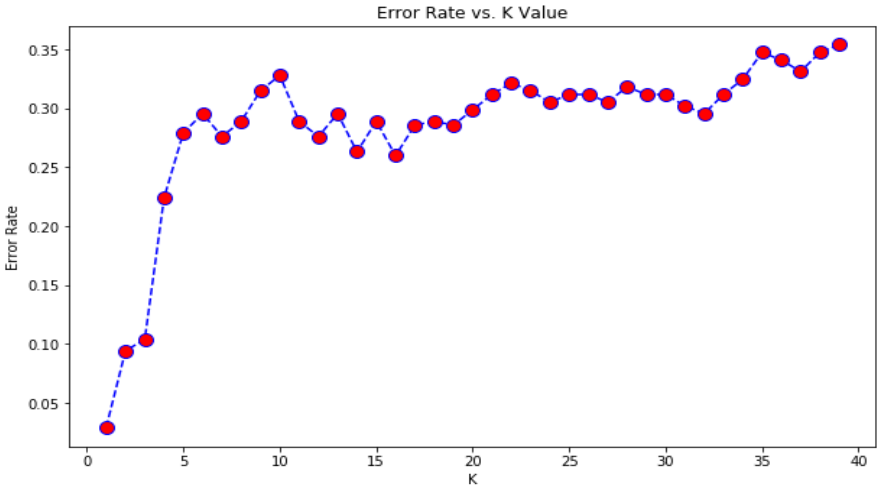


Figure 1.5 error rate vs k graph for KNN

So here in above graph we can see the lowest value of error rate is achieved for lower values of K . A value of 2 or 3 will be optimal for K so as to achieve lower error rate.

Decision trees classifier

Decision tree is a type of supervised learning algorithm (having a pre-defined target variable) that is mostly used in classification problems. It works for both categorical and continuous input and output variables. In this technique, we split the population or sample into two or more homogeneous sets (or sub-populations) based on most significant splitter / differentiator in input variables[4].

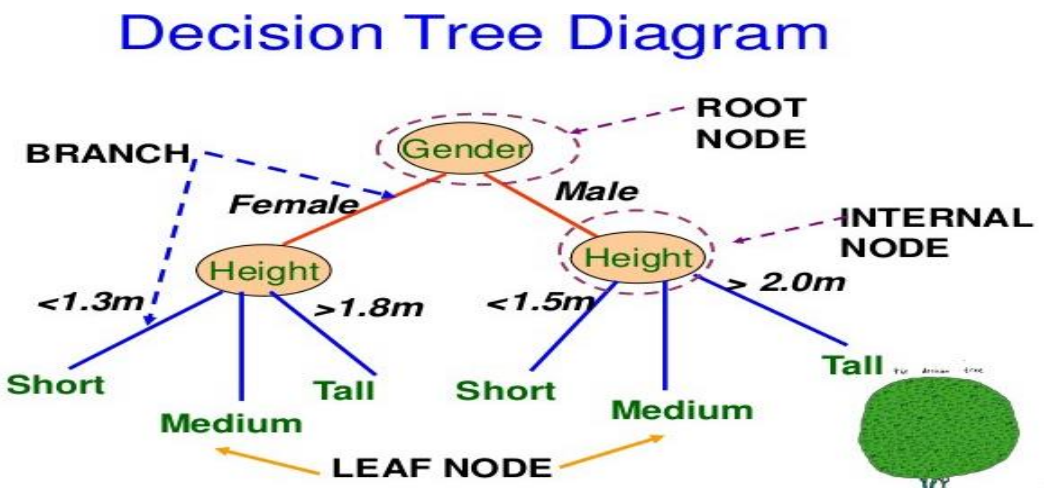


Figure 2.2 a sample decision tree

Types of Decision Trees:

Types of decision tree is based on the type of target variable we have. It can be of two types:

1. **Categorical Variable Decision Tree:** Decision Tree which has categorical target variable then it called as categorical variable decision tree. E.g.:- In above scenario of student problem, where the target variable was "Student will play cricket or not" i.e. YES or NO.

2. **Continuous Variable Decision Tree:** Decision Tree has continuous target variable then it is called as Continuous Variable Decision Tree.

Steps for creating decision trees:

1. Place the best attribute of the dataset at the **root** of the tree.

2. Split the training set into **subsets**. Subsets should be made in such a way that each subset contains data with the same value for an attribute.

3. Repeat step 1 and step 2 on each subset until you find **leaf nodes** in all the branches of the tree.

Assumptions while creating Decision Tree:

Some of the assumptions we make while using Decision tree:

- At the beginning, the whole training set is considered as the **root.**
- Feature values are preferred to be categorical. If the values are continuous then they are discretized prior to building the model.
- Records are **distributed recursively** on the basis of attribute values.
- Order to place attributes as root or internal node of the tree is done by using some statistical approach**.**
- Statistical approaches that are to be used are gini index , information gain(based on entropy) , chi square and reduction in variance. Gini index and information gain criterion is widely used for classifiacation purpose in decision tree classifiers.

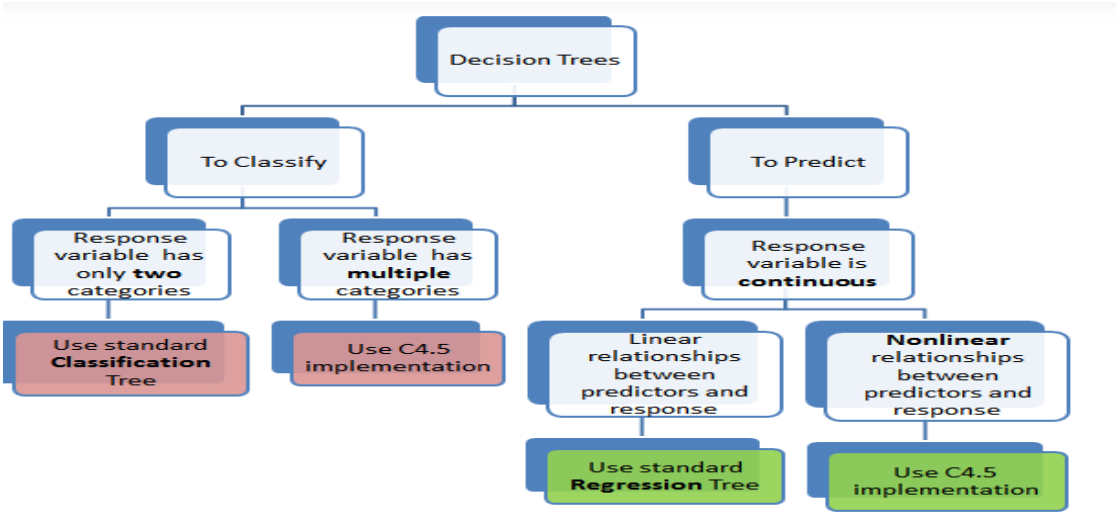


Figure 2.3 flow chart showing different levels of decision tree in different cases

Basic splitting criterions in decision trees:

The two most widely used criterion are **gini index** and **information gain** that we are going to discuss with the help of a sample example. Below we have a sample data set.

	A	B	C	D	E
1	4.8	3.4	1.9	0.2	positive
2	5	3	1.6	0.2	positive
3	5	3.4	1.6	0.4	positive
4	5.2	3.5	1.5	0.2	positive
5	5.2	3.4	1.4	0.2	positive
6	4.7	3.2	1.6	0.2	positive
7	4.8	3.1	1.6	0.2	positive
8	5.4	3.4	1.5	0.4	positive
9	7	3.2	4.7	1.4	negative
10	6.4	3.2	4.5	1.5	negative
11	6.9	3.1	4.9	1.5	negative
12	5.5	2.3	4	1.3	negative
13	6.5	2.8	4.6	1.5	negative
14	5.7	2.8	4.5	1.3	negative
15	6.3	3.3	4.7	1.6	negative
16	4.9	2.4	3.3	1	negative

Fig 2.4 Sample data set

First we will analyze the above data with the help of **information gain** and will find the best split variable for higher accuracy.

INFORMATION GAIN:

A , B , C , D attributes can be considered as predictor variables and E column class label can be considered as target variable.

First ,we have to choose some random variable to categorize each variable.

A	B	C	D
>= 5	>= 3	>= 4.2	>= 1.4
< 5	< 3	< 4.2	<=1.4

Figure 2.5 : categorizing each variable limits

There are basically 2 steps for calculating information gain for each attribute:

1. Calculate ENTROPY of the TARGET.

2. ENTROPY for every attribute A, B, C, D needs to be calculated using information gain formula we will subtract this enropy from the entropy of target. This result is information gain.

The entropy of the target **variable E** is given as :

Variable E	
Positive	Negative
8	8

We have 8 records with negative class and 8 records with positive class . So we can diretly estimate the entropy of target as 1.

$E(8,8) = -1*P(+VE)*\log_2(P_{(+VE)}) + P(-VE)*\log_2(P_{(-VE)})$

$= -1*((8/16)* \log_2(8/16)) + (8/16)* \log_2(8/16))$

$= 1$

ABOVE WE HAVE ENTROPY FOR OUR TARGET VARIABLE

Now we will see about how to obtain information gain for given variables other than E.

Information gain for Var A

Var A has value >=5 for 12 records out of 16 and 4 records with value <5 value.

- For Var A >= 5 & class == positive: 5/12
- For Var A >= 5 & class == negative: 7/12
    - Entropy(5,7) = -1 * ((5/12)*log2(5/12) + (7/12)*log2(7/12)) = 0.9799
- For Var A <5 & class == positive: 3/4
- For Var A <5 & class == negative: 1/4
    - Entropy(3,1) = -1 * ((3/4)*log2(3/4) + (1/4)*log2(1/4)) = 0.81128

Entropy(Target , A) = P(>=5) * E(5,7) + P(<5) * E(3,1)
= (12/16) * 0.9799 + (4/16) * 0.81128 = 0.937745

$$\text{Information Gain(IG)} = \text{E(Target)} - \text{E(Target,A)} = 1 - 0.9337745 = 0.062255$$

		Target	
		Positive	Negative
A	>= 5.0	5	7
	<5	3	1
Information Gain of A = 0.062255			

		Target	
		Positive	Negative
B	>= 3.0	8	4
	< 3.0	0	4
Information Gain of B= 0.7070795			

		Target	
		Positive	Negative
C	>= 4.2	0	6
	< 4.2	8	2
Information Gain of C= 0.5488			

		Target	
		Positive	Negative
D	>= 1.4	0	5
	< 1.4	8	3
Information Gain of D= 0.41189			

From the above information gain calculations we can build a decision tree . We should place the tree according to their values.

An attribute with better value than other should position as root and a branch with entropy close to zero should be converted to a leaf node . A branch with entropy more than 0 needs further splitting.

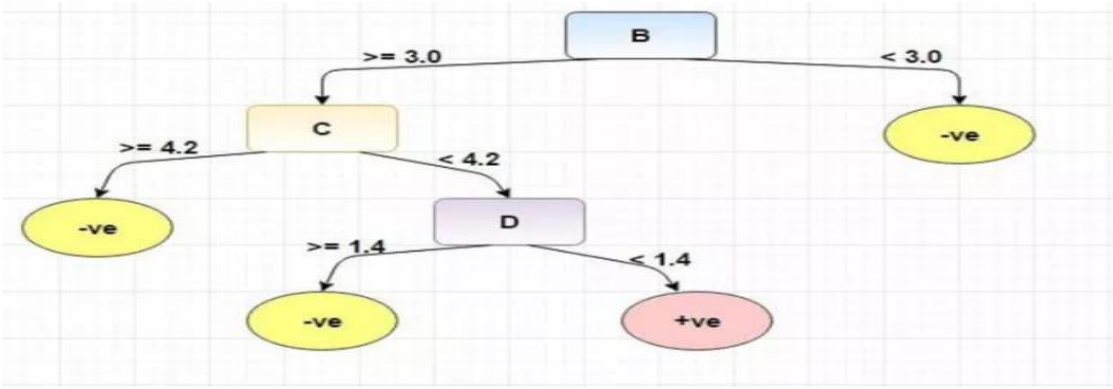


Figure 2.5 Representation of decision tree on the basis of information gain

GINI INDEX:

Gini index is a metric to measure as to how often a randomly chosen element would be incorrectly identified. It means an attribute with lower gini index should be preferred.

Lets take following sample data:

	A	B	C	D	E
1	4.8	3.4	1.9	0.2	positive
2	5	3	1.6	0.2	positive
3	5	3.4	1.6	0.4	positive
4	5.2	3.5	1.5	0.2	positive
5	5.2	3.4	1.4	0.2	positive
6	4.7	3.2	1.6	0.2	positive
7	4.8	3.1	1.6	0.2	positive
8	5.4	3.4	1.5	0.4	positive
9	7	3.2	4.7	1.4	negative
10	6.4	3.2	4.5	1.5	negative
11	6.9	3.1	4.9	1.5	negative
12	5.5	2.3	4	1.3	negative
13	6.5	2.8	4.6	1.5	negative
14	5.7	2.8	4.5	1.3	negative
15	6.3	3.3	4.7	1.6	negative
16	4.9	2.4	3.3	1	negative

Fig 2.6 sample data for gini index

We are going to use above sample data for gini index calculation. In above sample data we have 5 columns out of which 4 columns are continuous data and 5^{th} column consists of class labele.

A , B, C, D attributes can be considered as a predictors and E column class labels can be considered as a target variable. For constructing a decision tree from this data , we have to convert continuos data into categorical data.

We will choose some random values to categorize each attribute.

A	B	C	D
>= 5	>= 3	>= 4.2	>= 1.4
< 5	< 3	< 4.2	<=1.4

Fig 2. 6 Gini index formula representation

Gini Index for Var A

Var A has value >=5 for 12 records out of 16 and 4 records with value <5 value.

- For Var A >= 5 & class == positive: 5/12
- For Var A >= 5 & class == negative: 7/12
    - gini(5,7) = 1- ((5/12)2 + (7/12)2) = 0.4860
- For Var A <5 & class == positive: 3/4
- For Var A <5 & class == negative: 1/4
    - gini(3,1) = 1- ((3/4)2 + (1/4)2) = 0.375

By adding weight and sum each of the gini indices:

$$\text{gini(Target, A)} = (12/16) * (0.486) + (4/16) * (0.375) = 0.45825$$

		wTarget	
		Positive	Negative
A	>= 5.0	5	7
	<5	3	1
Ginin Index of A = 0.45825			

		Target	
		Positive	Negative
B	>= 3.0	8	4
	< 3.0	0	4
Gini Index of B= 0.3345			

		Target	
		Positive	Negative
C	>= 4.2	0	6
	< 4.2	8	2
Gini Index of C= 0.2			

		Target	
		Positive	Negative
D	>= 1.4	0	5
	< 1.4	8	3
Gini Index of D= 0.273			

Fig 2.7 Represntation of gini index for continuous variables

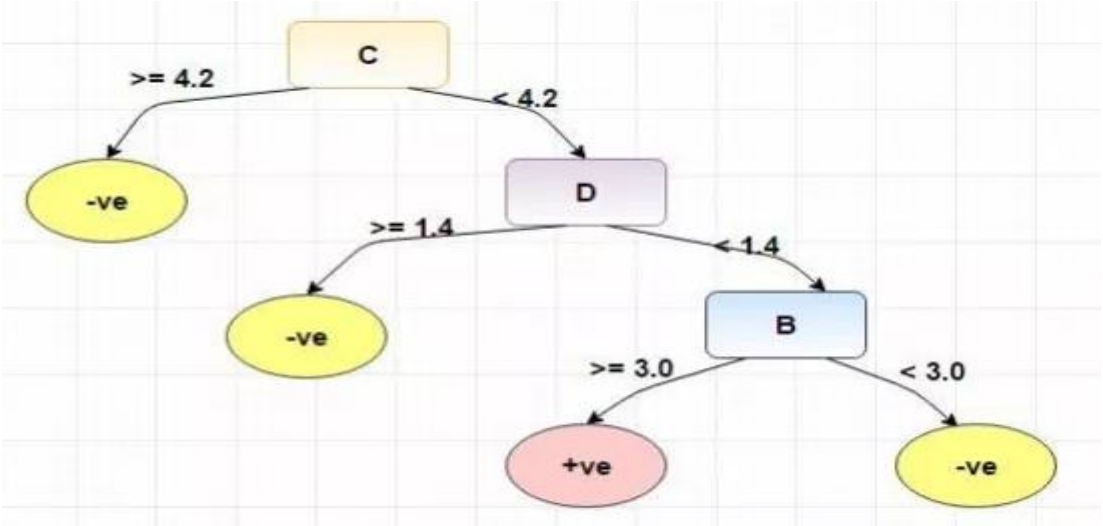


Fig 2.8 decision tree on the basis of gini index

As we can see above the continuous variable with lower gini indexes is being considered as the root node, as gini index is a measure to how often a chosen variable would be incorrectly identified, so a variable with lowest gini index value exhibits higher probability of correctly identifying our target variable.

During the implementation in python code , we are going to use GINI INDEX as our splitting criterion.

Implementation in python:

1. First step consists of loading our decision tree model and then setting up the criterion on which we are going to form our decision tree classifier. Here in this case we are going with **gini index** criterion for better understanding and accuracy.

```
from sklearn import tree
from sklearn.tree import DecisionTreeClassifier
clftree=tree.DecisionTreeClassifier(criterion='gini',splitter='best',max_depth=8,random_state=0)
```

2. This step refers to loading our dataset and converting it to a dataframe with the help of pandas library.

```
df = pd.read_csv('C:/Users/DELL/Downloads/heart.csv',header=0)
```

4. This step is basically about splitting process. First we assign columns to independent variables (x) and dependent variable (y). The columns containing features which act as predictors are assigned to as independent variables (x) and the columns containing class labels or target variables that are to be predicted are being assigned to dependent variable(Y).

After all this the data set is being splitted into test and train format in the ratio (30:70) depending upon the format of dataset. X_TRAIN, X_PRED , Y_TRAIN,Y_PRED will be the variables to look out for.

```
x=df.loc[:,df.columns!="target"]
type(x)
```

```
y=df["target"]
type(y)
```

```
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.30, random_state=0)
```

5. Now next step includes training our decision tree model that we have imported in step 1.

```
clftree.fit(x_train,y_train)
```

6. This steps consists of Predicting the class labels of the test dataset on basis on training that is being done by assigning 70% of the dataset entries to the training model.

```
y_train_pred=clftree.predict(x_train)
y_test_pred=clftree.predict(x_test)
```

```
y_test_pred
```

```
array([1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 1, 1, 0, 1, 0, 1, 1,
       1, 1, 1, 1, 1, 0, 1, 1, 1, 0, 0, 1, 0, 0, 0, 0, 1, 1, 0, 1, 1, 1,
       1, 1, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0,
       0, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 1, 0, 1, 0,
       1, 0, 0, 0, 1, 1, 1, 0, 0, 0, 1, 1, 1, 1, 1, 0, 0, 1, 0, 1, 0, 1,
       0, 0, 1, 0, 1, 0, 1, 0, 0, 1, 0, 1, 1, 0, 0, 1, 1, 1, 1, 0, 0, 0,
       1, 1, 1, 0, 0, 1, 0, 1, 1, 0, 1, 0, 0, 1, 1, 1, 0, 0, 1, 1, 1, 0,
       0, 0, 1, 1, 0, 0, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       0, 1, 1, 0, 0, 1, 0, 0, 1, 0, 1, 0, 1, 0, 0, 1, 0, 1, 0, 1, 1, 1,
       0, 0, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 0, 1, 0, 0, 0, 1, 0, 1, 1,
       0, 1, 1, 0, 1, 0, 0, 1, 1, 0, 0, 1, 1, 0, 1, 0, 1, 1, 0, 0, 0, 0,
       0, 0, 1, 1, 0, 1, 0, 0, 0, 0, 1, 1, 0, 1, 1, 0, 0, 0, 1, 1, 1, 1,
       1, 0, 0, 1, 1, 1, 0, 0, 1, 0, 1, 0, 0, 1, 1, 0, 0, 1, 1, 0, 1, 0,
       0, 1, 1, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 1, 0, 0, 1, 1, 0, 0, 0],
      dtype=int64)
```

7. Next step is our model performance part. This can be done by importing confusion matrix and accuracy_score paremeters with the help of scikit learn.

```
from sklearn.metrics import accuracy_score,confusion_matrix
```

```
confusion_matrix(y_test,y_test_pred)
```

```
accuracy_score(y_test,y_test_pred)
```

8. Last and the most important step is to plot our decision tree. For this we need pydotplus and graphviz library installed in our system. Following python commands are:

```
dot_data=tree.export_graphviz(clftree,out_file=None,feature_names=x_train.columns,filled=True)
```

```
graph=pydotplus.graph_from_dot_data(dot_data)
```

```
Image(graph.create_png())
```

And the decision so formed look like as:

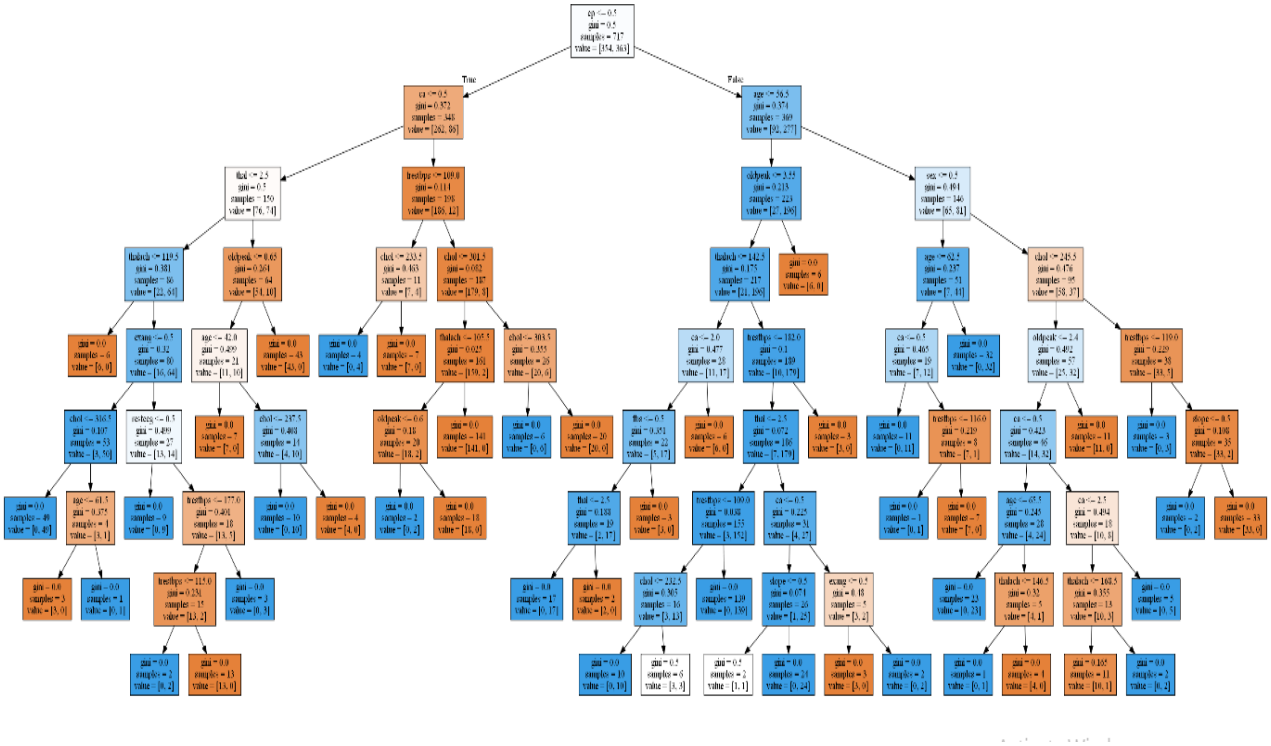


Fig 2.9 : Decision tree with max depth=8, with gini index as splitting criterion

9. To further enhance our model accuracy and prevent overfitting we can import random forest classifier through following commands.

```
from sklearn.ensemble import RandomForestClassifier
```

```
rf_clf=RandomForestClassifier(n_estimators=11,n_jobs=-1,random_state=0)
```

SVM (SUPORT VECTOR MACHINES):

It is a classification method. In this algorithm, we plot each data item as a point in n-dimensional space (where n is number of features we have) with the value of each feature being the value of a particular coordinate.

For example, if we only had two features like Height and Hair length of an individual, we'd first plot these two variables in two dimensional space where each point has two co-ordinates (these co-ordinates are known as **Support Vectors**)

Now, we will find some *line* that splits the data between the two differently classified groups of data. This will be the line such that the distances from the closest point in each of the two groups will be farthest away.

IMPLEMENTATION IN PYTHON:

1. Load all the important libraries.

2. Load the dataset and then represent it in the form of continuous and categorial variables(X,Y).

3. Next step is to split the data into TRAIN and TEST dataset in the ratio 70:30.

4. Now import the model and train it for respective KERNELS. The following KERNELS are **LINEAR , POLYNOMIAL** and **RBF.**

FOR LINEAR

```
clf_svm=svm.SVC(kernel='linear',C=0.5)
clf_svm.fit(x_train,y_train)
```

FOR POLYNOMIAL

```
clf_svm_p3=svm.SVC(kernel='poly',degree=2,C=10)
clf_svm_p3.fit(x_train,y_train)
```

FOR RADIAL

```
clf_svm_r=svm.SVC(kernel='rbf',gamma=0.7,C=.2)
clf_svm_r.fit(x_train,y_train)
```

5. The next step is to draw the hyperplanes and measuring the accuracy score and confusion matrix so as to decide which kernel is going to perform better in terms of classification for a particular dataset. Below is hyperplane design for iris dataset for all the three kernels.

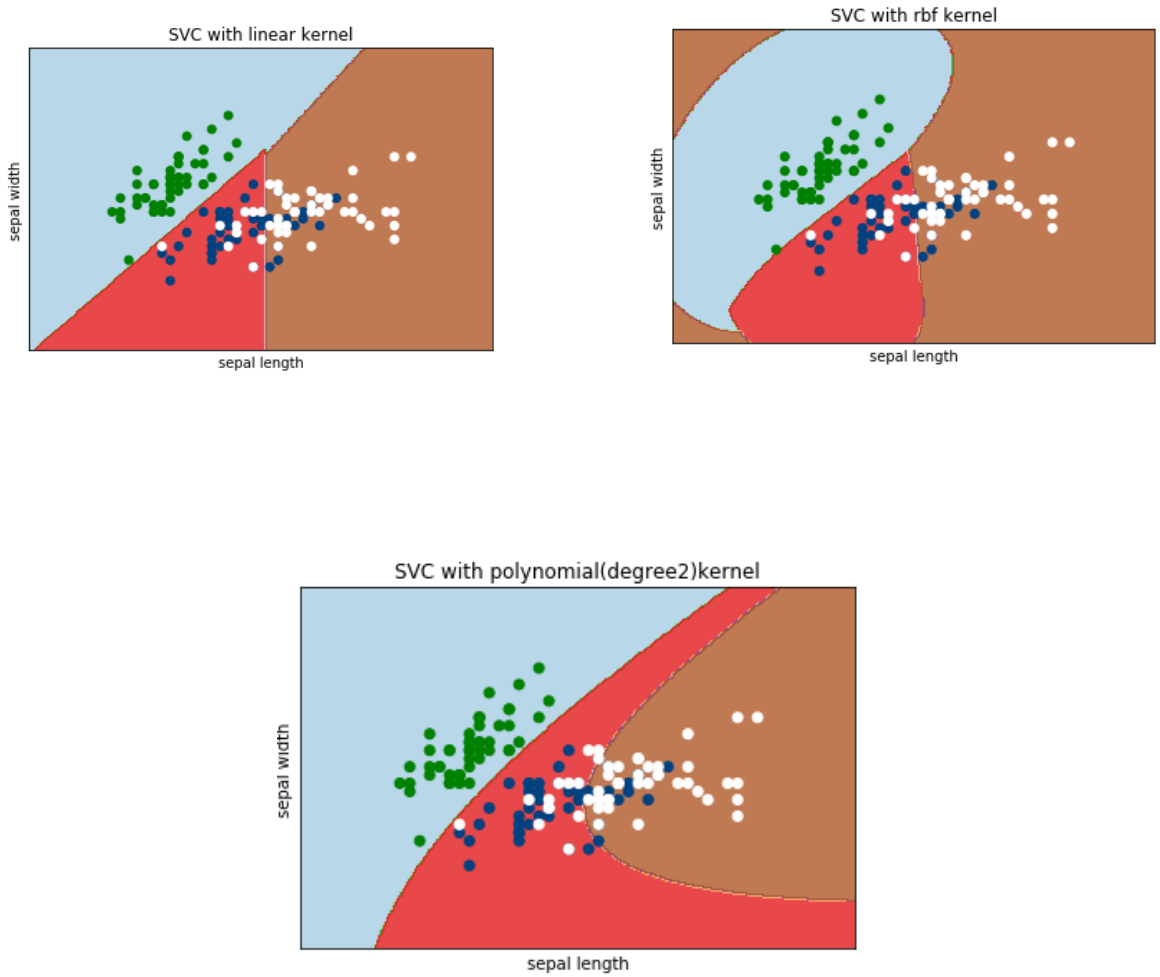


Fig 2.10 Hyperplanes for all the respective kernels on iris dataset.

CHAPTER 4

RESULTS AND DISCUSSION

In this section we are going through the performance analysis of all the three machine learning algorithms used and is going to check the best fitting algorithm on the basis of **CONFUSION MATRICE** and **ACCURACY SCORE.**[2]

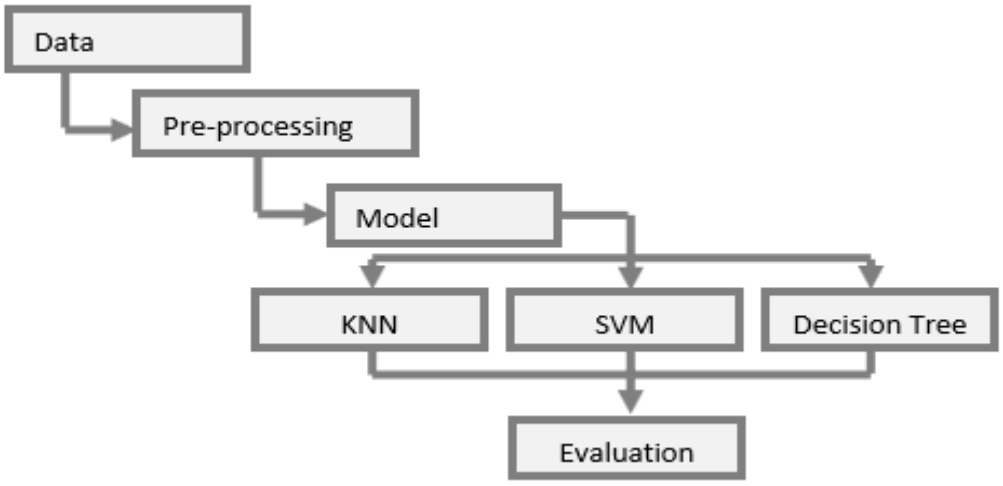


FIG 3.1 : FLOW CHART SHOWING THE COMPARATIVE ANALYSIS B/W OUR THREE ALGORITHM USED.[2]

This research had been done using several Machine Learning algorithms, namely KNN, SVM, and Random Forest. Machine Learning processing through several processes: data collecting, preprocessing, model building, comparison of models, and evaluation.

No	Feature Title	Variable Data Type	Feature Categorization
1	AGE	Continuous	IN YEARS
2	SEX	CATEGORIAL	0-FEMALE , 1-MALE
3	CP	CONTINUOUS	RANDOM WHOLE NO.
4	TRESTBPS	CONTINUOUS	IN mm hg
5	CHOL	CONTINUOUS	IN mg/dl
6	Fbs	Categorical	1=true 0=false
7	Restecg	CATEGORIAL	1=HIGH 0=LOW
8	Thalach	CONTINUOUS	RANDOM READINGS
9	EXANG	CATEGORIAL	1=YES 0=NO
10	OLDPEAK	CONTINUOUS	0 -3
11	SLOPE	CONTINUOUS	0-10
12	THAL	CONTINUOUS	3=NORMAL, 6=FIXED DEFECT 7=REVERSIBLE DEFECT
13	CA	CONTINUOUS	0-3
14	TARGET	CATEGORIAL	1=YES , 0=NO

Table 1: Detail of Dataset

3.1 MODEL RESULTS:

Before the data is being processed the data set is split into two parts by a ratio of 70:30, which 70% for the training purpose and 30 % for the testing purpose .Training data is used to get the model. Training data used was of 717 samples out of the total 1026 samples with 13 predictor variable and 1 target variable . Output of training data is a model used for classification purposes. In testing purpose accuracy score is being calculated for the algorithm models by comparing the predicting labels with the true labels that are already present in 30% of the test dataset. The tools that are used to calculate accuracy in classification process are confusion matrices and accuracy scores.

These two parameters are already present in the library called sklearn in python Jupyter notebook. The accuracies of the model that had been built is shown in Table 2.

Table 2: Model Result

Algorithm	**Result**	**Accuracy**
KNN	k = 3	88.63%
SVM	C = 0.5 for **linear** C = 10 for **poly** Gamma=0.7,C=0.2 for **Rbf kernel**	86.36% 84.41% 86.5%
Decision Tree	Gini index = 0.5 , max depth=8	97.17%
Random forest classifiers	No of estimators=6	98.83%

```
##linear
```

```
clf_svm=svm.SVC(kernel='linear',C=0.5)
clf_svm.fit(x_train,y_train)
```

```
y_train_pred=clf_svm.predict(x_train)
y_test_pred=clf_svm.predict(x_test)
y_test_pred
```

```
from sklearn.metrics import accuracy_score,confusion_matrix
```

```
confusion_matrix(y_test,y_test_pred)
```

```
array([[114,  31],
       [ 11, 152]], dtype=int64)
```

```
accuracy_score(y_test,y_test_pred)
```

```
0.8636363636363636
```

Now we look at the CONFUSION METRICES of all the three algorithms that are being used and evaluate how much percentage of the test data is being classified correctly.

Now we see CONFUSION MATRICES for test data to all the above used algorithms.

TOTAL SAMPLES = 1026

TRAIN SAMPLES = 0.70*1026 = 717(approx. 70%)

TEST SAMPLES= 0.30*1026 = 309(approx. 30% of the total samples)

Confusion matrice for KNN:

Table 3 : confusion matrix for KNN

	Active	**Non active**
Active	**133**	**12**
Non active	**23**	**140**

Active= person is healthy and have no heart disease

Non active = person is at risk of having heart disease in future

Confusion matrice for decision tree and random classifier:

Table 4 : CONFUSION MATRICE FOR DECISION TREE

	Active	**Non active**
Active	**145**	**0**
Non active	**7**	**156**

TABLE 5 : CONFUSION MATRICE FOR RANDOM FOREST

	Active	**Non active**
Active	**145**	**0**
Non active	**3**	**160**

Confusion matrices for SVM :

FOR LINEAR KERNEL:

TABLE 6: CONFUSION MATRICE FOR SVM LINEAR KERNEL

	active	**Non active**
Active	**114**	**31**
Non active	**11**	**152**

CLASSIFICATION RESULT:

Table 7: Comparison of Confusion Matrix

Prediction		KNN	SVM (LINEAR)	DecisionTree
	TRUE	85.8%	78.5%	98.8%
Active	FALSE	14.2%	21.5%	1.2%
Non-	TRUE	85%	93.5%	95.7%
Active	FALSE	15%	6.5%	4.3%

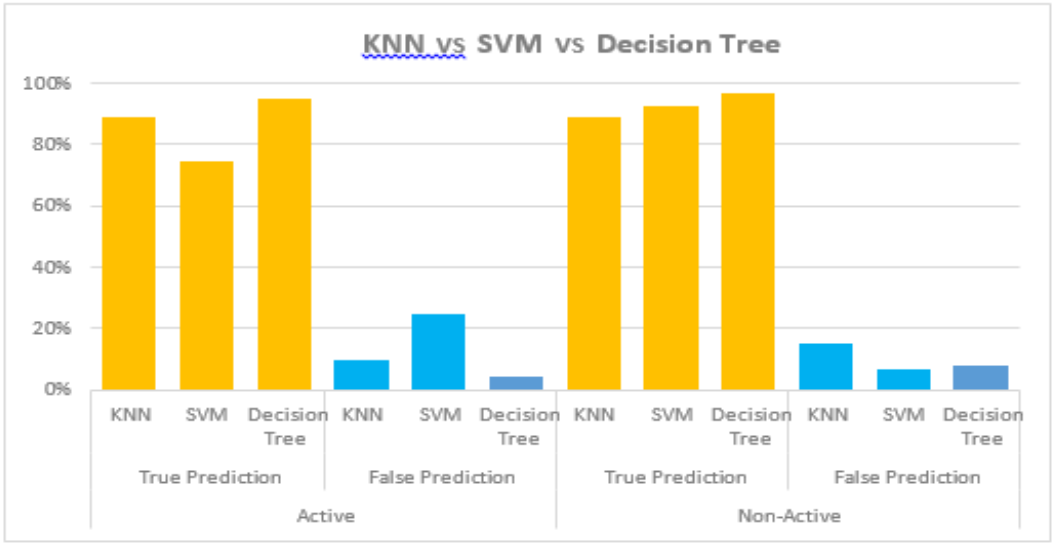


Figure 3.2 Comparing testing accuracies of KNN , SVM , decision tree.

The final result is a comparison of model classification to see which algorithm has the best accuracy. It is shown in the below table while above table is showing us the comparison of classification accuracies.

Accuracy		
KNN	**SVM**	**Decision Tree**
89%	86% (linear) 84% (polynomial) 86.5% (Rbf kernel)	97% 98%(with random forest)

So the above table clearly shows that our decision tree clssifiers are working better than other classification supervised machine learning algorithm.

DISCUSSION:

The model specifications for different algorithms with best results are as- for KNN algorithm to predict the best performance is k (kernel) = 3 with accuracy 88.63%, value C = 0.5 for SVM linear kernel algorithm with accuracy 86.36%, and GINI INDEX= 0.5 for Decision Tree algorithm with 97.17% accuracy. The comparison of the three algorithms shows that the best accuracy is for Decision Tree algorithm. After testing, it turns out that the DECISION TREE MODEL can predict better than the KNN algorithm and SVM. It can be seen that the SVM algorithm can predict exactly 114 active patients and 152 non-active patients, while the DECISION TREE MODEL predicted exactly 145 active patients and 156 non-active patients, and the KNN algorithm only predicts exactly 133 active patients and 140 non-active patients.

So when the model testing is done, Decision tree algorithm has the best accuracy model compared to SVM and KNN.

Comparison on the basis of CONFUSION MATRICE shows the same things to the results of previous comparisons. DECISION TREE ALGORITHM has the best accuracy to predict active patients (98%) compared to KNN (85.8%) and SVM (78.5%) while using linear kernel.

Also ,the Decision Tree algorithm has the best accuracy for predicting non-active patients count ,(95.7%) compared to SVM (93.5%) and KNN (85%). Moreover, algorithm decision tree has the best accuracy in predicting non-active patients and has the difference of 2% from SVM algorithm. While for predicting the accuracy of active patients, decision tree algorithm has a 12% difference from KNN and a difference of approx. 20% when SVM(with linear kernel) is used . It can be said that the Decision tree algorithm occupies the best position compared to SVM and KNN. This is substantiated after the overall accuracy calculation is performed, it is found that DECISION TREE ALGORITHM has the best classification accuracy of 97% while the SVM has highest 86% accuracy with rbf as well as linear kernel and KNN has 89% accuracy. Thus, the best algorithm for predicting patients heart disease is DECISION TREE algorithm. Also if we want to enhance the accuracy of decision tree classifiers by random forest classifier , an increase of 1% in accuracy is seen.

Chapter 5

Conclusions :

KNN algorithm can predict student performance well with k = 3. The best model of SVM algorithm to predict model performance is by using the value of C = 0.5 for linear kernel and C=0.2 , gamma=0.5 for RBF kernel. Whereas while using the Decision Tree algorithm, the best predictions results is obtained at GINI INDEX = 0.50 . **Comparison of three algorithm machine learning (KNN, SVM, and Decision Tree) shows that DECISION TREE ALGORITHM has the best accuracy (97%) compared to SVM (87%) and KNN (89%) in predicting HEART DISEASE in nearby future for active as well as non -active patients**.

```
from sklearn import tree
from sklearn.tree import DecisionTreeClassifier
clftree=tree.DecisionTreeClassifier(criterion='gini',splitter='best',max_depth=8,random_state=0)
```

```
clftree.fit(x_train,y_train)
```

```
y_train_pred=clftree.predict(x_train)
y_test_pred=clftree.predict(x_test)
```

model performance

```
from sklearn.metrics import accuracy_score,confusion_matrix
```

```
confusion_matrix(y_train,y_train_pred)
```

```
array([[354,   0],
       [  5, 358]], dtype=int64)
```

```
confusion_matrix(y_test,y_test_pred)
```

```
array([[145,   0],
       [  7, 156]], dtype=int64)
```

```
accuracy_score(y_test,y_test_pred)
```

```
0.9772727272727273
```

Furthermore in decision trees the accuracy has been further enhanced by approx (1%) with the help of random forest classifiers. In random forest the no of estimators are taken as 6 which is the optimal one. Large value of estimators can lead to overfitting.

```
from sklearn.ensemble import RandomForestClassifier
```

```
rf_clf=RandomForestClassifier(n_estimators=6,n_jobs=-1,random_state=0)
```

```
rf_clf.fit(x_train,y_train)
```

```
RandomForestClassifier(bootstrap=True, class_weight=None, criterion='gini',
            max_depth=None, max_features='auto', max_leaf_nodes=None,
            min_impurity_decrease=0.0, min_impurity_split=None,
            min_samples_leaf=1, min_samples_split=2,
            min_weight_fraction_leaf=0.0, n_estimators=6, n_jobs=-1,
            oob_score=False, random_state=0, verbose=0, warm_start=False)
```

accuracy measure with randaom forest

```
confusion_matrix(y_test,rf_clf.predict(x_test))
```

```
array([[145,   0],
       [  4, 159]], dtype=int64)
```

```
accuracy_score(y_test,rf_clf.predict(x_test))
```

```
0.987012987012987
```

CHAPTER 6

REFRENCES:

[1]. H. ELAIDI, Y. ELHADDAR, Z. BENABBOU, H. ABBA "An idea of a clustering algorithm using support vector machines based on binary decision tree" , 2018.

[2] .Slamet Wiyono,Taufiq Abidin "Comparative Study of Machine Learning Knn, SVM, and decision tree algorithm to predict students performance" , 2019.

[3]. Manus Ross, Corey A. Graves, John W. Campbell, Jung H. Kim "Using Support Vector Machines to Classify Student Attentiveness for the Development of Personalized Learning Systems" .2013

[4]. Hongtao Xie, Fuhua Shang "The Study of Methods for Post-pruning Decision Trees Based on Comprehensive Evaluation Standard" ,2014.

[5] V.N. Vapnik. The nature of statistical learning theory, Second Edition, Springer, pp. 131-145, 2000.

[6] Animesh Hazra, Arkomita Mukherjee, Amit Gupta, Asmita Mukherjee, "Heart Disease Diagnosis and Prediction Using Machine Learning and Data Mining Techniques: A Review", Research Gate Publications, July 2017, pp.2137-2159.

[7] V. Krishnaiah, G. Narsimha, N. Subhash Chandra, "Heart Disease Prediction System using Data Mining Techniques and Intelligent Fuzzy Approach: A Review", International Journal of Computer Applications, February 2016.

[8] Guizhou Hu, Martin M. Root, “Building Prediction Models for Coronary Heart Disease by Synthesizing Multiple Longitudinal Research Findings”, European Science of Cardiology, 10 May 2005.

[9] T.Mythili, Dev Mukherji, Nikita Padaila and Abhiram Naidu, “A Heart Disease Prediction Model using SVMDecision Trees- Logistic Regression (SDL)”, International Journal of Computer Applications, vol. 68, 16 April 2013. International Journal of Computer Applications (0975 – 8887) Volume 181 – No. 18, September 2018 25

[10] Nimai Chand Das Adhikari, Arpana Alka, and rajat Garg, “HPPS: Heart Problem Prediction System using Machine Learning”.

[11] K. Polaraju, D. Durga Prasad, “Prediction of Heart Disease using Multiple Linear Regression Model”, International Journal of Engineering Development and Research Development, ISSN:2321-9939, 2017.

[12] Marjia Sultana, Afrin Haider, “Heart Disease Prediction using WEKA tool and 10-Fold cross-validation”, The Institute of Electrical and Electronics Engineers, March 2017.

[13] Dr.S.Seema Shedole, Kumari Deepika, “Predictive analytics to prevent and control chronic disease”, https://www.researchgate.net/punlication/316530782, January 2016.

[14] Ashok kumar Dwivedi, “Evaluate the performance of different machine learning techniques for prediction of heart disease using ten-fold cross-validation”, Springer, 17 September 2016.

[15] Megha Shahi, R. Kaur Gurm, “Heart Disease Prediction System using Data Mining Techniques”, Orient J. Computer Science Technology, vol.6 2017, pp.457-466.

[16] Mr. Chala Beyene, Prof. Pooja Kamat, “Survey on Prediction and Analysis the Occurrence of Heart Disease Using Data Mining Techniques”, International Journal of Pure and Applied Mathematics, 2018.

[17] R. Sharmila, S. Chellammal, “A conceptual method to enhance the prediction of heart diseases using the data techniques”, International Journal of Computer Science and Engineering, May 2018.

[18] Jayami Patel, Prof. Tejal Upadhay, Dr. Samir Patel, “Heart disease Prediction using Machine Learning and Data mining Technique”, March 2017.

[19] Purushottam, Prof. (Dr.) Kanak Saxena, Richa Sharma, “Efficient Heart Disease Prediction System”, 2016, pp.962-969.

[20] K.Gomathi, Dr.D.Shanmuga Priyaa, “Multi Disease Prediction using Data Mining Techniques”, International Journal of System and Software Engineering, December 2016, pp.12-14.

[21] Mr.P.Sai Chandrasekhar Reddy, Mr.Puneet Palagi, S.Jaya, “Heart Disease Prediction using ANN Algorithm in Data Mining”, International Journal of Computer Science and Mobile Computing, April 2017, pp.168- 172.

[22] Ashwini Shetty A, Chandra Naik, “Different Data Mining Approaches for Predicting Heart Disease”, International Journal of Innovative in Science Engineering and Technology, Vol.5, May 2016, pp.277- 281.

[23] Jaymin Patel, Prof. Tejal Upadhyay, Dr.Samir Patel, “Heart Disease Prediction using Machine Learning and Data Mining Technique”, International Journal of Computer Science and Communication, September 2015-March 2016, pp.129-137.

[24] Boshra Brahmi, Mirsaeid Hosseini Shirvani, “Prediction and Diagnosis of Heart Disease by Data Mining Techniques”, Journals of Multidisciplinary Engineering Science and Technology, vol.2, 2 February 2015, pp.164- 168.

[25] Noura Ajam, “Heart Disease Diagnoses using Artificial Neural Network”, The International Insitute of Science, Technology and Education, vol.5, No.4, 2015, pp.7-11.

[26] Prajakta Ghadge, Vrushali Girme, Kajal Kokane, Prajakta Deshmukh, “Intelligent Heart Disease Prediction System using Big Data”, International Journal of Recent Research in Mathematics Computer Science and Information Technology, vol.2, October 2015 - March 2016, pp.73-77.

[27] S.Prabhavathi, D.M.Chitra, “Analysis and Prediction of Various Heart Diseases using DNFS Techniques”, International Journal of Innovations in Scientific and Engineering Research, vol.2, 1, January 2016, pp.1-7.

[28] Sairabi H.Mujawar, P.R.Devale, “Prediction of Heart Disease using Modified K-means and by using Naïve Bayes”, International Journal of Innovative research in Computer and Communication Engineering, vol.3, October 2015, pp.10265-10273