BACHELOR OF SCIENCE IN
COMPUTER SCIENCE AND ENGINEERING

BRAC
UNIVERSITY

Inspiring Excellence

Heart Disease Prediction Using Machine Learning Algorithms

Authors
Musfiq Ali
MD. Iftiyar Khan
Masud Al Imran
Musnath Siddiki

Supervisor
Hossain Arif
Assistant Professor
Department of CSE

A thesis submitted to the Department of CSE
in partial fulfillment of the requirements for the degree of
B.Sc. Engineering in CSE

Department of Computer Science and Engineering
BRAC University, Dhaka - 1212, Bangladesh

April 2019

I would like to dedicate this thesis to my loving parents . . .

Declaration

It is hereby declared that this thesis report or any part of it has not been submitted elsewhere for the award of any Degree or Diploma.

Authors:

MD. Iftiyar Khan
Student ID: 14101246

Masud Al imran
Student ID: 14301018

Musfiq Ali
Student ID: 14101220

Musnath Siddiki
Student ID: 1221048

Supervisor:

Hossain Arif
Assistant Professor, Department of Computer Science and Engineering
BRAC University

April 2019

Acknowledgements

We wish to express our sincere gratitude to Hossain Arif sir, our thesis supervisor for providing us with all the necessary facilities that was required for our research. We would also like to thank Hossain Arif sir for his sincere and valuable guidance and encouragement. We will forever be indebted to Hossain Arif sir for providing all the necessary help throughout the journey. Moreover, we would also like to thank all the Faculty Staffs of BRAC University for guiding us throughout the study period in BRAC University. Finally, we thank our loving parents, siblings, friends for being our support system. They were always helping us directly or indirectly into making this project a success.

The thesis titled Heart Disease Prediction Using Machine Learning Algorithms
Submitted by:
MD. Iftiyar Khan Student ID: 14101246
Masud Al Imran Student ID: 14301018
Musfiq Ali Student ID: 14101220
Musnath Siddiki Student ID: 1221048
of Academic Year Spring 2019 has been found as satisfactory and accepted as partial fulfillment of the requirement for the Degree of Computer Science and Engineering.

1.	Hossain Arif Assistant Professor BRAC UNIVERSITY	Assistant Professor
2.	Dr. MD. Abdul Mottalib Chairperson BRAC UNIVERSITY	Chairperson
3.	MD. Iftiyar Khan 14101246	Student
4.	Masud Alimran 14301018 Address	Student
5.	MD. Musfiq Ali 14101220	Student
6.	Musnath Siddiki 1221048	Student

Abstract

According to recent survey by WHO (World health organization) 17.9 million people die each year because of heart related diseases and it is increasing rapidly. With the increasing population and disease, it is become a challenge to diagnosing disease and providing the appropriate treatment at the right time. But there is a light of hope that recent advances in technology have accelerated the public health sector by developing advanced functional biomedical solutions. This paper aims at analyzing the various data mining techniques namely Naive Bayes, Random Forest Classification, Decision tree and Support Vector Machine by using a qualified dataset for Heart disease prediction which is consist of various attributes like gender, age, chest pain type, blood pressure, blood sugar etc. The research includes finding the correlations between the various attributes of the dataset by utilizing the standard data mining techniques and hence using the attributes suitably to predict the chances of a heart disease. These machine learning techniques take less time for the prediction of the disease with more accuracy which will reduce the dispose of valuable lives all over the world.

Table of contents

List of figures

List of tables

Nomenclature

Roman Symbols

F	complex function

Greek Symbols

γ	a simply closed curve on a complex plane
$\imath$	unit imaginary number $\sqrt{-1}$
π	$\simeq 3.14\ldots$

Superscripts

j	superscript index

Subscripts

0	subscript index
crit	Critical state

Other Symbols

$\oint_\gamma$	integration around a curve γ

Acronyms / Abbreviations

ALU	Arithmetic Logic Unit
BEM	Boundary Element Method
CD	Contact Dynamics
CFD	Computational Fluid Dynamics

CIF	Cauchy's Integral Formula
CK	Carman - Kozeny
DEM	Discrete Element Method
DKT	Draft Kiss Tumble
DNS	Direct Numerical Simulation
EFG	Element-Free Galerkin
FEM	Finite Element Method
FLOP	Floating Point Operations
FPU	Floating Point Unit
FVM	Finite Volume Method
GPU	Graphics Processing Unit
LBM	Lattice Boltzmann Method
LES	Large Eddy Simulation
MPM	Material Point Method
MRT	Multi-Relaxation Time
PCI	Peripheral Component Interconnect
PFEM	Particle Finite Element Method
PIC	Particle-in-cell
PPC	Particles per cell
RVE	Representative Elemental Volume
SH	Savage Hutter
SM	Streaming Multiprocessors
USF	Update Stress First
USL	Update Stress Last

Chapter 1

Getting started

1.1 Introduction

Health is one among the world challenges for humanity. World health organization (WHO) has mentioned that for an Individual proper health is the fundamental right. So to keep people fit and healthy proper health care services should be provided. 31 percentage of all deaths worldwide are because of heart related problems [6]. Diagnosis and treatment of heart disease is very complex, particularly in developing countries, due to the lack of diagnostic devices and a shortage of physicians and other resources affecting proper prediction and treatment of cardiac patients. With this concern in the recent times computer technology and machine learning techniques are being used to develop software to assist doctors in making decision of heart disease in the preliminary stage. Early stage detection of the disease and predicting the probability of a person to be at risk of heart disease can reduce the death rate. Medical data mining techniques are used in medical data to extract meaningful patterns and knowledge. Medical information has redundancy, multi-attribution, incompleteness and a close relationship with time. The problem of using the massive volumes of data effectively becomes a major problem for the health sector. Data mining provides the methodology and technology to convert these data mounds into useful decision-making information. This predication system for heart disease would facilitate Cardiologists in taking quicker decisions so that more patients can receive treatments within a shorter period of time, resulting in saving millions of life [1].

1.2 Methods

Data mining provides the methodology and technology to convert data mounds into useful decision-making information. In this research the comparison of different machine learning techniques like- Support Vector Machine, Decision Tree, Random Forest, Naive Bayes are implemented to predict heart disease [9]. Naïve mathematician used probability for predicating heart disease, SVM used on classification and regression technique, Random Forest works with varied decision Tree. These algorithms show different accuracy. We will try to tuning our techniques to obtain better accuracy which will be beneficial for more accurate prediction [2].

1.3 Objectives

The main objective of this study is to predict weather a patient is affected with heart disease or not using different machine learning algorithms on a qualified dataset. Find out the co-relations between different attributes [3]. Obtaining clear idea of our proposed data mining techniques and analyze the result and comparing between the results of different data mining techniques. We will analyze our techniques if there is any possibility to bring improvement for our results [4].

Chapter 2

Literature Review

Here in this chapter we will be discussing about various machine learning classifiers and previous work on the heart disease. In machine learning we can use different algorithms otherwise known as classifiers to help us predict

For our project. Here in our project we are looking forward to predict the number of patient that have heart disease and the number of patient that do not have heart disease running four algorithm to our data set. The reason we are going to use four is that it will allow us to get better and more reliable prediction [5]. Because if we are using one algorithm or classifier and do not have anything else to compare it with then we cannot say that it a reliable prediction because it might be giving us a very good accuracy but this algorithm might not be the best or more appropriate one to use for our scenario. Whereas if we use more than one algorithm or classifier in our case four of them, we can compare them with one another and if we find one classifier is giving us accuracy that is not even in the ball park of the other algorithm provided accuracy we can understand that something is going wrong. It can be that the algorithm itself is not suitable for the job or we made a mistake in our coding [6]. So using more than one algorithm is essential for any prediction based system. Now the algorithms that we have chosen to use in our project are: 1. Decision tree [7], 2. Naïve Bayes [7], 3.SVM (support vector machine) and lastly 4. Random Forest. We will be discussing each of those algorithms below. And finally we are also going to discuss about the previous work that has been done and show how it improved over time and what improvements we were able to bring in our project. Now we are going to talk about some of the work done before on heart disease prediction. Many research has been done on blood test in order to predict heart disease. Our blood offer us with many clues about our heart condition. For example, if our cholesterol in our blood is high that is a clear sign that we are at the increased risk of having a heart attack. Other substances in our blood can also help our doctor to determine if we have heart failure or are at risk of developing plaque deposits in our arteries also known as atherosclerosis. So it is very important to remember that one blood test alone is not enough to determine our risk of heart disease. The most vital risk factors for cardiopathy square measure smoking, high blood pressure high cholesterol and diabetes [7]. Now let us look at some of the blood test that we can do to diagnosis and manage heart disease. First of all we can do the cholesterol test. A cholesterol test also known as lipid panel or lipid profile, measures the flats (lipids) in our blood. The measurements can indicate our risk of having a heart attack or other heart disease. The test is typically including measurements of – (1) total cholesterol. This is a sum of our blood cholesterol content. If it is high than it puts us at a high level risk of having a heart attack. In an ideal state, the total cholesterol should be below 200 mg per deciliter (mg/dL) or 5.2 mill moles per liter (mmol/L). (2) Low-density lipoprotein (LDL) cholesterol. This is sometimes called the ‘bad’ cholesterol. Too much of it in blood causes the accumulation of fatty deposits in our arteries, which reduces blood flow. These plaque deposits typically rupture and cause major heart and tube issues. Our LDL cholesterol level should be less than 130 mg/dL in order for us the stay fit. More desirable level should be under 100 mg/dL, especially if we have diabetes or a history of heart attack, heart stents, heart bypass surgery or other heart/vascular conditions. (3) High-density lipoprotein (HDL) cholesterol. This is typically referred to as the ‘good’ cholesterin as a result of it helps take away cholesterin, keeping arteries open and your blood flowing more freely. Ideally, your HDL cholesterol level should be over 40 mg/dL for a man, and over 50 mg/dL for a woman. (4) Triglycerides. It is another type of fat in the blood [7].

High lipoid levels typically mean you frequently eat a lot of calories than you burn. If it go too high it can increase our risk of heart disease. Ideally, our triglyceride level should be less than 150 mg/dL. The American heart association (AHA) states that a triglyceride level of 100 mg/dL or lower is considered ‘optimal’. (5) Non-HDL cholesterol. Non- high density compound protein cholesterol is that the distinction between total cholesterol and HDL cholesterol (HDL-C). Non-HDL-C contains of cholesterol in lipoprotein particles that are involved in hardening of the arteries (atherosclerosis). This includes beta-lipoprotein (LDL), compound protein (a), intermediate-density compound protein and very-low-density compound protein. In some cases Non-HDL-C fraction can be considered a better marker of risk than LDL cholesterol [7]. Now let us talk a little bit further about High-sensitivity C-reactive protein. C-reactive protein otherwise known as CRP is a protein your liver produces as part of your body’s response to injury or infection (inflammatory response). CRP is a sign of inflammation somewhere in the body. But high sensitivity CRP tests cannot figure out where exactly in the body this may be happening or why it is happening. Inflammation plays a central role in the process of atherosclerosis where fatty deposits clog our arteries. Now measuring CRP alone will not tell our doctor our risk of heart disease. But if we factor in

HS-CRP then test results with other blood test results ad risk factors for heart disease will help to create an overall picture of our heart health. Because these is a variability in CRP levels that the test should be done twice and two weeks apart in order to get a good and acceptable result. An HS-CRP level above 2.0 mg/L indicates a higher risk of heart disease. This test screening is not currently recommended for people without symptoms or a known risk of heart disease. Cholesterol- lowering statin medications may reduce CRP levels and decrease your heart disease risk [7]. Next thing we are going to discuss will be the lipoprotein (a). Lipoprotein (a), or Lp (a), is a type of LDL cholesterol. Our Lp(a) level is determined by our genes and is not generally affected by lifestyle. However high level of Lp(a) may be a sign of increased risk of heart disease, though it is not clear how much risk [11]. Our doctor might order and Lp(a) test if we already have atherosclerosis or heart disease but appear to have otherwise normal cholesterol levels. Lp(a) is often tested if we have a family history of early-onset heart disease or sudden death. Here one thing we should also add is drugs are in development to lower Lp(a), but it is not yet clear what effect lowering Lp(a) will have on heart disease risk. People with high Lp(a) are generally advised to keep a low LDL cholesterol level [8]. Up next we are going to talk about Plasma ceramides. This is a new type of test that measures levels of ceramides in the blood. Ceramides are produced by all of our cells and play a significant role in the growth, function and ultimately death of many types of tissue. Ceramides square measure transported through the blood by lipoproteins and square measure related to coronary artery disease. Three specific ceramides have been liked to plaque buildup in the arteries and insulin resistance. This elevates the level of these ceramides in the blood indicates a higher risk of cardiovascular disease within on to five years [8]. And now finally we are going to talk about Natriuretic peptides. Before talking about Natriuretic peptides we need to realize that brain natriuretic peptide, also called B-type

Natriuretic peptide (BNP), is a protein that our heart and blood vessels produce. BNP can help us by eliminating our body fluids and relaxing our blood vessels and funnels sodium into our urine [10]. When our heart is damaged our body secretes high levels of BNP into our blood stream to try to ease the strain on our heart [13]. BNP levels may also rise if we have a new or increased chest pain (unstable angina) or after a heart attack. Now our BNP level can also help in the diagnosis and evaluation of heart failure and other heart conditions. Normal levels do vary according to age and gender and whether we are overweight [12]. One of the foremost vital uses of BNP is to do to map out whether or not shortness of breath is because of failure. Now for people who have heart failure, establishing a baseline BNP can be helpful and future tests can be used to help gauge how well our treatment works. In addition to that a variation of BNP called N-terminal BNP also is useful in diagnosing heart failure and in some laboratories is used instead of BNP N- terminal BNP may also be useful in evaluating our risk of a heart attack and other problems if we already have heart disease. Finally A high level of BNP alone is not enough to diagnose a heart problem in that case our doctor will also consider our risk factors and other blood test results [15]. So as we can see due to the demand of having a system that can predict heart disease prediction many studies have taken place.

Chapter 3

Algorithm Discussion

3.1 Algorithms

In machine learning we can use different algorithms otherwise known as classifiers to help us predict for our project. Here in our project we are looking forward to predict the number of patient that have heart disease and the number of patient that do not have heart disease running four algorithm to our data set. The reason we are going to use four is that it will allow us to get better and more reliable prediction [21]. Because if we are using one algorithm or classifier and do not have anything else to compare it with then we cannot say that it a reliable prediction because it might be giving us a very good accuracy but this algorithm might not be the best or more appropriate one to use for our scenario [22]. Whereas if we use more than one algorithm or classifier in our case four of them, we can compare them with one another and if we find one classifier is giving us accuracy that is not even in the ball park of the other algorithm provided accuracy we can understand that something is going wrong. It can be that the algorithm itself is not suitable for the job or we made a mistake in our coding [24]. So using more than one algorithm is essential for any prediction based system. Now the algorithms that we have chosen to use in our project are: 1. Decision tree, 2. Naïve Bayes, 3.SVM (support vector machine) and lastly 4. Random Forest. We will be discussing each of those algorithms below.

3.1.1 DECISION TREE (DT)

For our first algorithm we will be using Decision Tree classifier. It is one of the most popular machine learning algorithms to this date [25]. They are used for both classification and regression problems. Now a question might arise why we are willing to use Decision tree classifier over other classifiers. To answer that question we can bring about two reasons. One being, Decision trees often tries to mimic the same way human brain thinks so it is quite simple to understand the data and come

to some good conclusions or interpretations. Second reason can be, Decision trees actually allows us to see the logic for the data to interpret rather than being a black box algorithm like SVM, NN and others. It has the specialty of being simple and clear easily becoming one of the favorite among programmers of this generation [1]. Now that we have discussed why Decision tree is good let us look further into what actually is Decision tree classifier. To start a decision tree is a tree where there are a bunch of nodes and each node represent a feature (attribute), each link (branch) represent a decision otherwise known as rule and each leaf of the tree represent an outcome otherwise known as categorical or continues value. The idea is to create a tree for the entire data and get an outcome at every leaf [2]. Now we are a bit more familiar with what is decision tree. Let us go ahead and discuss about how we can build a decision tree classifier. Decision tree can be made based on two different algorithm. One being the CART (classification and Regression Trees) and the other being ID3 (iterative Dichotomiser 3).

For ID3 first we take the x value in the column and a y value which stays at the last position of the column and only has “YES” or “NO” value. For the chart above we have (outlook, temp, humidity, windy) as our x values and play which

Only has two option either ‘YES’ or “NO” is at the last position of the column or is our y value. Now we need to do the mapping of x and y. As we can see it is a binary classification problem, so let us build the tree using the ID3 algorithm. Now in order to create a tree we need a root node at first and we need to pick one

first to be the root node [4]. A general rule of thumb is to choose the feature which has the most influence on the value y first as the root node. Then we move on and choose the next most influential feature to be the next node. Here we are going to use the concept of entropy which is the measure of the amount of uncertainty in the data set. We need to calculate the entropy for all categorical values for the binary classification problem [5]. So to sum it all up we can say that we need to compute the entropy for the data set first. Then for every attributes/features we need to first of all calculate entropy for all the categorical values then take the average value information entropy for the current attribute and finally calculate how much we have gained for the current attribute. After that we need to pick the highest gain attribute and repeat until we get our desired tree. Now that is the process of ID3. As we have discussed above Decision tree classifier has been made on another algorithm know as CART short for classification and Regression Trees. In this algorithm we use Gini Index as our cost function used to evaluate splits in the dataset. Here our target variable is indeed a binary variable so it will take two values (yes and no). And as we all know there can be 4 combinations [6]. Now we need to figure out the Gini score which will give us a good idea of how we can split the data. If we can get the Gini score of 0 we can consider it to be a perfect separation whereas worst case scenario would a split of 50/50. Now the question arises how we can calculate Gini index value.

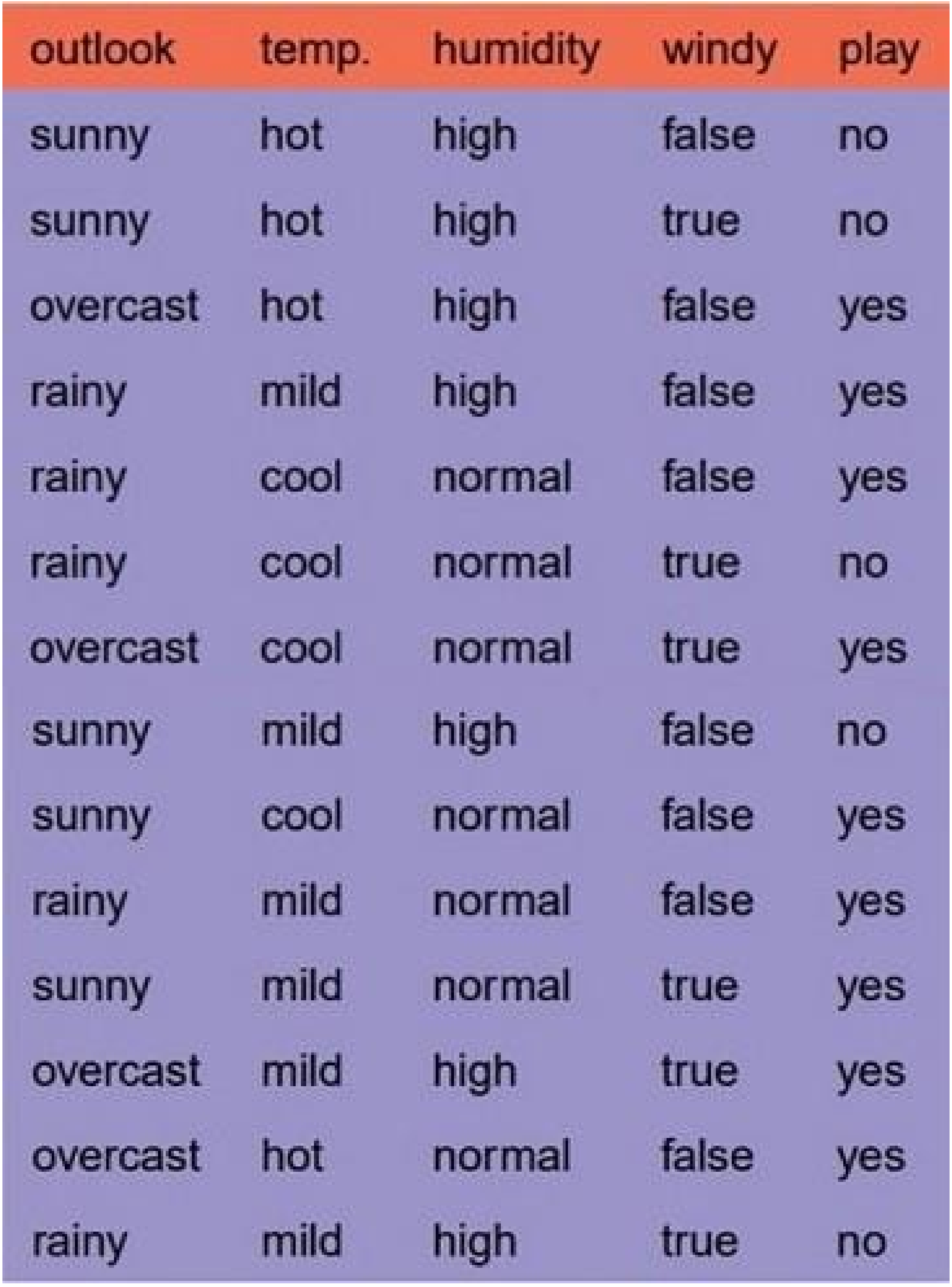
Now if the target variable is categorical variable with multiple levels the gini index will be still similar. So the steps for this method is the first compute the gini index for data-set [7]. Then for every feature we need to calculate gini index for all categorical values and take average information entropy for the current attribute and in the end calculate the gini gain. After we are done with that we can pick the best gini gain attribute and we need to repeat until we get our desired tree. And that is how decision tree algorithm works [7].

Figure 3.1 shows Weather Dataset presented in order for us to understand Decision tree in a much better way.

3.1.2 NAIVE BAYES (NB)

We have already talked about our first machine learning algorithm, the decision tree classifier. Now we are going to talk about our second machine algorithm which we are going to use for our prediction purposes which is named Naive Bayes classifier. To give a brief introduction of naïve bayes algorithm we can say that it is a collection of classification algorithms based on Bayes Theorem [7]. We cannot call it a single algorithm because it is a family of algorithms where all of them share a common principle [7]. Here every pair of features that are being classified is independent from each other. Now in order to understand it better we are going to use our previous data set of weather and if we can play or not. We are going to analyze the features – outlook, temperature, humidity, windy and predict Play golf with yes or no.

Before we start we can see that the dataset is divided into two parts. One being the feature matrix and the other one being response vector [**?**]. Here the feature matrix contains all the dependent features value which will be used to figure out the response vector. These features are – Outlook, Temperature, Humidity and Windy. We also have Play Golf as our response vector which contains the value of class variable otherwise known as output value for each row of feature matrix. Now in naïve bayes we assume that each feature makes an independent and equal contribution to the outcome. Now if we think about it our naïve bayes algorithm's relation to our dataset, we understood that we are assuming that no pair of feature are dependent [14]. As an example, if the temperature is 'Hot' it has nothing to do with the other feature humidity or the outlook being 'Rainy' has no influence on the winds. That's how we can assume that the features are independent. Secondly, we can say that each feature has the same weight of importance. For example knowing only temperature and humidity without knowing the other feature we will not be able to predict the outcome accurately.



outlook	temp.	humidity	windy	play
sunny	hot	high	false	no
sunny	hot	high	true	no
overcast	hot	high	false	yes
rainy	mild	high	false	yes
rainy	cool	normal	false	yes
rainy	cool	normal	true	no
overcast	cool	normal	true	yes
sunny	mild	high	false	no
sunny	cool	normal	false	yes
rainy	mild	normal	false	yes
sunny	mild	normal	true	yes
overcast	mild	high	true	yes
overcast	hot	normal	false	yes
rainy	mild	high	true	no

Fig. 3.1 Weather Dataset presented in order for us to understand Decision tree in a much better way.

There is not a single feature that we can call irrelevant and assume it to contribute equally to the outcome. One thing we should note is that the assumptions that we have made by naïve bayes are not generally correct in terms of real life situations. In-fact, there is no such thing as independent assumption in real life [8]. Now are going to look at the formula of naïve bayes but before that we need to know about Bayes Theorem from which we got our naïve bayes theorem.

The formula above is considered to be Bayes Theorem Formula. Here A and B are events. Here we are trying to find probability of event A, given the event B is true [17]. We can name event B as evidence. P(A) is called priori of A and the evidence is and attribute value of an unknown instance. One thing we also need to note that here it is event B. P(A|B) is called posteriori probability of B. Here probability of event after evidence is seen. Now with regards to our dataset we are going to apply Bayes Theorem as is shown below:

Here y is class variable and x is a dependent feature vector which can be shown as:

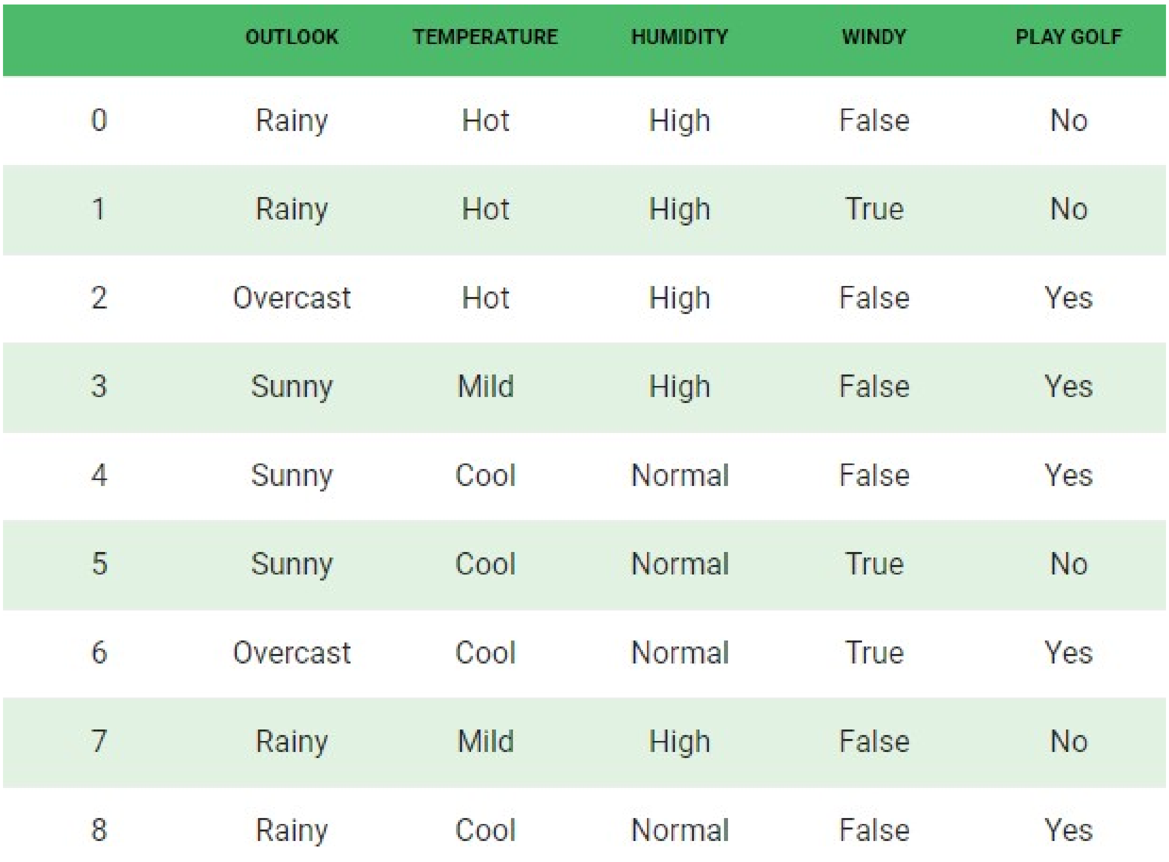
Here we would like to clear that a feature vector and corresponding class variable can be:

So we can say P(X|Y) here means, the probability of 'Playing golf' being no give that the weather conditions are 'Rainy outlook', 'Temperature is hot', 'High humidity' and 'No wind' [8]. Now for the Naïve assumption it is time to put a naïve assumption to bayes theorem which is independence among the features. So now, we split evidence into the independent parts. Now if any two events A and B can be considered as independent, then,

Hence we can reach the result:

Now we need to create the classifier model [18]. For doing this we find the probability of given set of inputs for all possible values of the class variable y and pick up the output with maximum probability. Now we can express it mathematically by:

So in the end we are left with the task of calculating P(y) and P(X|Y) Please note that P(Y) is also being called class probability and P(X|Y) is also being called conditional probability [16]. One thing we would like to add to this would be the different naïve bayes classifiers differ mainly by the assumptions they make regarding the distribution of P(X|Y). Now let us try to apply the above formula manually on our weather dataset. For this one we need to do some precipitations on our dataset. Here we need to find P(X|Y) for each X in X and Y in Y. all these calculations have been demonstrated in the tables below; So as we can see from the figure above, we have calculated P(X|Y) for each X in X and Y in Y manually in the table 1 to 4. As an example, probability of playing golf given that the temperature is cool is P(temp. = cool | play golf = Yes) = 3/9. Also we need to find class probabilities P(Y) which has been calculated in the table 6 which we can give an example of P(play gold = yes) = 9/14. So now, we tend to ar finished our pre-computations and also the classifier is prepared. So that is how a general Naïve Bayes classifier work [21]].



	OUTLOOK	TEMPERATURE	HUMIDITY	WINDY	PLAY GOLF
0	Rainy	Hot	High	False	No
1	Rainy	Hot	High	True	No
2	Overcast	Hot	High	False	Yes
3	Sunny	Mild	High	False	Yes
4	Sunny	Cool	Normal	False	Yes
5	Sunny	Cool	Normal	True	No
6	Overcast	Cool	Normal	True	Yes
7	Rainy	Mild	High	False	No
8	Rainy	Cool	Normal	False	Yes

Fig. 3.2 Weather Dataset used in order to explain Naïve bayes classifier.

Figure 3.2 shows Weather Dataset used in order to explain Naïve bayes classifier.

3.1.3 Support Vector Machine (SVM)

Before we start discussing about SVM (support vector machine) we need to be accustomed with linear regression and

Logistic regression algorithms. If not it is suggested to look at them before moving on to support vector machine. Support vector machine is another simple algorithm that every machine learning expert should have in his or her arsenal. In this scenario Support vector machine is highly preferred by many as it produces significant accuracy with less computation power. Moreover support vector machine, abbreviated as SVM can be used for both regression and classification tasks. But it is widely used in classification objectives [10]. Now let us a bit more about what is support vector machine. The objective of the support vector machine algorithm is to find a hyper plane in an N-dimension space (N – the number of features) that distinctly classifies the data points [10]. Now in order to separate the two classes of data points there are many possible hyper planes that could be chosen. In our objective though we should find a plane that has the maximum margin for example the maximum distance between data points of both classes. Maximizing the margin distance providing us with some reinforcement so that future data points can be classified with more confidence. Here hyper planes are decision boundaries that help classify the data points [19]. Data points falling on either aspect of the hyper plane will be attributed to completely different categories. Here one more thing we would like to add is that the dimension of the hyper plane depends upon the number of features. If we can find the number of input feature is 2 then the hyper plane is just a line. If the number of input feature is 3 then the hyper plane becomes a two dimensional plane [18]. From here what we can understand is that it becomes very difficult to imagine when the number of feature exceeds 3. We should always remember that support vectors are data points that are close to the hyper plane and influence the position and orientation of the hyper plane. Using these support vectors, we maximize the margin of the classifier [20]. Now if we delete the support vectors it will change the position of the hyper plane. These are the points that will eventually help us build our SVM. Now we are going to talk a bit about the large margin intuition. To start we can say that in logistic regression we take the output of the linear function and squash the value within the range of [0, 1] using the sigmoid function. If the squashed value is greater than a threshold value (.5) we assign it as label 1, else we assign it a label 0. In SVM, we take the output of the linear function and if that output is greater than 1, we identify it with one class and if the output is -1, we identify is with another class. Since the threshold value are changed to 1 and -1 in SVM, by doing this we can obtain this reinforcement range of values ([-1, 1])

which acts as margin. Next we are going to talk about cost function and gradient updates. To start, in the SVM algorithm, we are looking to maximize the margin between the data points and the hyper plane. The loss operate that helps maximize the margin is hinge loss [10]. The cost is 0 if the predicted value and the actual value are of the same sign. If they're not, we then calculate the loss value. We conjointly add a regularization parameter the value operate. The objective of the regularization parameter is to balance the margin maximization and loss. After adding the regularization parameter, the cost functions looks as below. Now that we have the loss function, we take partial derivatives with respect to the weights to find the gradients. Using the gradients, we can update our weights. Now when there is no misclassification, we can consider our

Model correctly predicting the class of our data point, we only have to update the gradient from the regularization parameter. And if the situation do occur when there is indeed a misclassification. For example our model makes a mistake on the prediction of the class of our data point, we include the loss along with the regularization parameter to perform gradient update [10]. SVM implementation in python can be shown as following: Since the iris dataset has 3 categories, we will remove one of the classes [13]. This will leave us with a binary class classification problem. Also we would also like to add that there are four features available for us to use. We will be using only two features. For example Sepal length and Petal length. We take these two features and plot them to visualize, from the above graph, you can infer that a linear line can be used to separate from the data points. We extract the specified options and split it into coaching and testing knowledge. 90 Percentage of the data is used for training and the rest 10 percentage is used for testing. So it would be a good time to build our SVM model using the numpy library. a (0.0001) is the learning rate and the regularization parameter lambda is set to 1/epochs. Therefore, the regularizing value reduces the number of epoch's increases. Now it is time that we clip the weights as the test data contains only 10 data points. We extract the features from the test data and predict the values. We acquire the predictions and compare it with the particular values and print the accuracy of our model [13]. There is another straightforward thanks to implement the SVM formula. We can use the SCikit learn library and simply decision the connected functions to implement the SVM model. The number of lines of code reduces significantly too few lines. At the end of the day we can conclude by saying that Support vector machine is an elegant and powerful algorithm. We should use it wisely.

3.1.4 RANDOM FOREST CLASSIFIER (RF)

Before we start let us give a brief information about the random forest classifier. It is a flexible, easy to use machine learning algorithm that produces, even without hyper- parameter tuning, a great result most of the time [13]. Moreover, It is also known to be one of the most used algorithms, because its simplicity and the fact that it can be used for both classification and regression tasks. Here down below we are going to show how the random forest algorithm works and several other important things about it. First of all let us discuss about how it actually works. For starters random forest is a supervised learning algorithm. Like you will already see from its name, it creates a forest and makes it somehow random. The 'forest' it builds, is an ensemble of decision trees, most of the time trained with the 'bagging' method. The general plan of the textile technique is that a mixture of learning models will increase the result. If you want to say it in simple words: random forest builds multiple decision trees and merges them together to get a more accurate and stable prediction [13]. One massive advantage of random forest is that it is used for each classification and regression issues, which form the majority of current machine learning systems. I will refer random forest in classification, since classification is sometimes considered the building block of machine learning. Below you'll be able to see however a random forest would seem like with 2 trees: Random forest has nearly the same hyper parameters as a decision tree or a bagging classifier. Fortunately, you do not have to combine a decision tree with a bagging classifier and

Can just easily use the classifier-class of Random Forest. We already have told that with random forest we can also deal with regression tasks by using the random forest repressor. Random Forest adds further randomness to the model whereas growing the trees. Instead of sorting out the foremost necessary feature whereas rending a node, it searches for the best feature among a random subset of features. This ends up in a good diversity that typically ends up in a more robust model. So, in random forest, only a random subset of the features is taken into consideration by the algorithm for splitting a node. You can even make trees more random [15]. In addition to that using random thresholds for each feature rather than searching for the best possible thresholds (like a normal decision tree does). Now let us shed some light on the real life analogy of random forest classifier. Imagine a guy named Andrew that wants to decide to which places he would travel during a one year vacation trip. He asks people who know him for advice. First of all he goes to find a friend than he asks Andrew where he traveled to in the past and if he liked it or not. Based on the answers he can offer St. Andrew some recommendation. This is a general decision tree algorithm approach. Andrews's friend created rules to guide his decision about what he should recommend by using the answers of Andrew. Now after that, Andrew starts asking more and more of his friends to advise him and they again ask him different questions, where they can derive some recommendations from. Then he rather chose to go to places that where recommend the most to him which is the general or otherwise known as typical Random Forest algorithm approach [15]. Now we are going to talk about different features of Random Forest. First of all we can say that another great quality of the random forest algorithm is that it is very easy to measure the relative importance of each feature on the prediction. Sklearn provides a good tool for this that measures a options importance by viewing what proportion the tree nodes that use that feature cut back impurity across all trees in the forest. It computes this score mechanically for every feature when coaching and scales the results so the add of all importance is up to one [15]. Now in any case if you do not know about how a decision tree works and if we do not know what a leaf or node is here is a good description from Wikipedia Is. In a decision tree each internal node represents a "test" on an attribute (for example whether a coin flip come up heads or tails) each branch represents the outcome of the check and every leaf node represents a category label (decision taken when computing all attributes). A node that has no kids could be a leaf. Now if we look through at the feature importance we can decide which features we may want to drop because they do not contribute enough or nothing to the prediction process. This is important because a general rule in machine learning is that the more features you have the more likely our model will suffer from over fitting and vice versa. Down below we can see a table and a visualization that show the importance of 13 features which we used during the supervised classification project with the famous titanic dataset which we were able to find on kaggle [16]. Now here we are going to talk about the different aspects of decision trees and random forests. We already mentioned that random forest is a collection of decision trees, but there are some differences. For example, if we input a training dataset with features and labels into a decision tree. It will formulate some set of rules which can be wont to build the predictions. For example if you would like to predict whether or not someone can click on an internet advert you

May collect the ads the person clicked within the past and a few features that describe the decision he has made. If we want to put the features and labels into a decision tree it will generate some rules [23]. Which later can be used to predict whether the advertisement will be clicked or not. In comparison the Random Forest rule willy-Nelly selects observations and options to make many call trees then averages the results. We were able to see another difference is that "Deep" decision trees might suffer from over fitting. Random Forest prevents over fitting most of the time, by creating random subjects of the features and building smaller trees using these subsets. Afterwards it combines the sub trees. Note that this does not work every time and that it also makes the computation slower depending on how many trees your random forest builds. Important Hyper parameters that we can discuss here will be shown below. But before we get to that we need to know that the hyper parameters in random forest are either used to increase the predictive power of the model or to make the model faster. I will here talk about the hyper parameters of sklearns built-in random forest function [21]. Increasing the predictive power – First of all there is the n estimators hyper parameter which is just the number of trees the algorithms builds before taking the maximum voting or taking averages of predictions. In general a better variety of trees will increase the performance and makes the predictions additional stable however it conjointly slows down the computation. Another important hyper parameter is “max feature” which is the maximum number of features random forest considers to split a node. Sklearn provides several options described in their documentation. The last important hyper-parameter we will talk in terms of speed is “min sample leaf”. This determines whether its name already says the minimum number of leafs that are required to split an internal node. Increasing the models speed – The “n jobs” hyper- parameter tells the engine how many processors it is allowed to use. If it’s a worth of one it will solely use one processor. A value of “-1” means there’s no limit. “random state” makes the models output replicable. The model will always produce the same results when it has a definite value of random state and if it has been given the same hyper parameters and the same training data. Lastly, there’s the “oob score” (also known as oob sampling) that could be a random forest cross validation methodology [21]. In this sampling regarding common fraction of the information isn’t wont to train the model and might be wont to value its performance. There samples are called the out of bag samples. It is terribly just like the leave-one-out cross- validation methodology, but almost no additional computational burden goes along with it. As for above we discussed about Random Forest classifiers in details. Now let us take a quick look at the advantages and disadvantages of Random Forest. First of all as we already mentioned an advantage of random forest is that it can be used for both regression and classification tasks and that it is easy to view the relative importance it assigns to the input options. Random Forest is additionally thought of as a awfully handy and straightforward to use rule. The main reason behind it is default hyper parameters often produce a good prediction result. The number of hyper parameters is additionally not that top and that they ar uncomplicated to grasp [21]. One of the big problems in machine learning is over fitting, but most of the time this will not happen that easy to a random forest classifier. That is because if there are enough trees in the forest the classifier will not over fit the model. But in our opinion the main limitation of random forest is that a large number of trees can make the algorithm too slow and ineffective for real-time predictions. In general these algorithms are fast to train but quite slow to create prediction once they are trained. A

additional correct prediction needs additional trees which ends in a very slower model in most real- world applications the random forest rule is quick enough [22]. But in that case there can certainly be situations where run-time performance is important and other approaches would be preferred. And in fact random forest could be a prognosticative modeling tool and not a descriptive tool. That being said if you are looking a description of the relationships in our data other approaches would be preferred [22].

3.2 Design of system

In this portion of our report we are going to discuss how we prepared or designed the whole system. In terms of how we executed the system it will be discussed later in the book.

3.2.1 Dataset

We found our data set that has been used in our book from kaggle (https://www.kaggle.com/ronitf/heart-disease- uci/version/1) [22]. The dataset that we used in our thesis has in total 14 columns and 303 rows. First 13 of those columns are the features that we will be using later on in order to predict the final column 'diagnosis' which will tell us if the patient is going to be affected by heart disease or not. The 303 rows represents data of 303 patients that we found from the dataset.

3.2.2 Preprocessing

Before we start let us give a brief information about what data preprocessing actually is. Data preprocessing may be a data processing technique that involves remodeling data into a lucid format. Real-world data is often incomplete, inconsistent and lacking in certain behaviors or trends and is likely to contain many errors [22]. Data preprocessing may be a tried technique of partitioning such problems. Data preprocessing prepares raw data for further processing. Data preprocessing is used in database-driven applications such as customer relationship management and rule-based applications. For our thesis we are using standard scaler from the sklearn library for preprocessing our data. We choose this one over the many other ones because it suits very well with our system.

3.2.3 Load data

We created an array called col names and put down all our columns on that array. Then we read the csv file also known as the dataset file.

3.2.4 Analyze features

In this section we are going to distribute the target value is vital for choosing appropriate accuracy metrics and consequently properly assess different machine learning models. First of all we are going to count values of explained variable otherwise known as the determining variable which is going to give us the prediction of a patient being affected by heart disease or not [22]. Second of all we are going to separate numeric features from categorical features. Then we are going to show the relation between the categorical features in various plots and try to figure out or rather observe the influence of those categorical features in the actual determining variable “diagnosis”.

3.2.5 Modeling and predicting with machine learning

The main goal of the entire project is to predict heart disease occurrence with the highest accuracy. In order to achieve this we will test several classification algorithms. This section includes all results obtained from the study and introduces the best performer according to accuracy metric. I have chosen several algorithms typical for solving supervised learning problems throughout classification methods. First of all, let’s equip ourselves with a handy tool that benefits from the cohesion of SciKit Learn library and formulate a general function for training our models. The reason for displaying accuracy on both, train and test sets, is to allow us to evaluate whether the model over fits or under fits the data (so-called bias/variance tradeoff). Then we are going to split the data then test and train them in the ratio of 70:30. Then we are going to create a model where we are going to run all our algorithms [24].

3.2.6 Finding the result

At the end we are going to create a summery table where we are going to show the different accuracy percentage of different algorithms [25]. Where we are going to find out that it does not come as a surprise that the more complex algorithms like SVM and Random Forests generated better results compared to the basic ones. It is worth to emphasize that in most cases hyper parameter tuning is essential to achieve robust results out of these techniques. By producing decent results, simpler methods proved to be useful as well. Machine learning has absolutely bright future in medical field. Just imagine a place where heart disease experts are not available. With just basic information about a certain patient’s medical history, we may quite accurately predict whether a disease will occur or not [25]. We are going to discuss them more in details in the later section.

Chapter 4

Results and Analysis

4.1 Results and Analysis

In our previous chapters we have discussed about different algorithms, previous works in this field and the dataset we used for our experiments [25]. All those were the foundation for this chapter. In this chapter we discussed about results that we found after implementing the algorithms and analyzed them.

4.2 Accuracy of Models with All Features

The results have been obtained by applying different classification algorithms. In our first experiment we used the whole dataset with all features and applied Support Vector Machine, Decision Tree, Random Forest, Gaussian Naive Bayes. Table 4.1 contains accuracy of the different algorithms that we applied on our dataset. Fig 4.1 shows the graphical representation of the table [25].

Classifier	**Accuracy (%)**
SVM	86.81
Decision Tree	70.33
Random Forest	89.01
Gaussian Naive Bayes	91.21

Table 4.1 Accuracy of our algorithms

4.3 Feature Engineering

1. A lot of features can affect the accuracy of the algorithm. So working with the features is very important. There are few reasons for which some may want to work with some selected features. 2. Choosing less features helps us to train faster. 3. By picking up the most important features, we can use interactions between them as new features. Sometimes this gives surprising improvement [7]. 4. Some features are linearly related to others. This might put a strain on the model. 5. Feature Selection means to select only the important features in-order to improve the accuracy of the algorithm. 6. It reduces training time and reduces over fitting.

4.4 Feature Importance

A very basic question that we might ask of a model is what features have the biggest impact on predictions? This concept is called feature importance. In dataset there may be some attributes which don't effect the prediction that much [1]. In some cases, few attributes may decrease the accuracy level of a model. So, it is important to work with the correct attributes. So far we have worked with all the features of the dataset and listed [4.1] the accuracy of different models. Now, we want to see the change of accuracies of different classifiers after selecting a subset of the attributes. We can see the importance of a feature via Decision Tree [Table 4.2] and Random Forest [Table 4.3].

Figure 4.1 shows Decision tree importance checker.

Figure 4.2 shows Random forest importance checker.

4.5 Accuracy of Models with Selected Features

After seeing the feature importance of Table 4.2 and Table 4.3, we selected the below features to see the difference in prediction [2] – CA, Thal, Thalach, Cp, Old peak, age

4.2 Table 4.4 shows the changes in the accuracy after selecting features. And fig shows the graphical representation of changes in the accuracy [2].

4.6 Cross Validation

Cross validation is an essential step in model training. It tells us whether our model is at high risk of overfitting. In many competitions, public LB scores are not very reliable. Often when we improve the model and get a better local CV score, the LB score becomes worse. It is

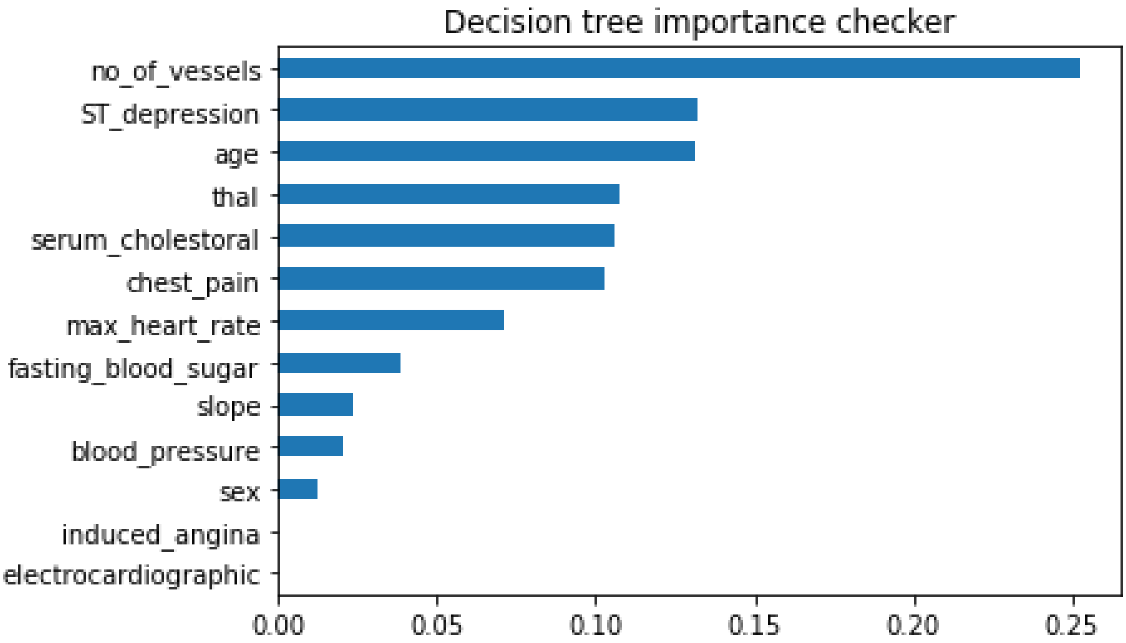


Fig. 4.1 Decision tree importance checker

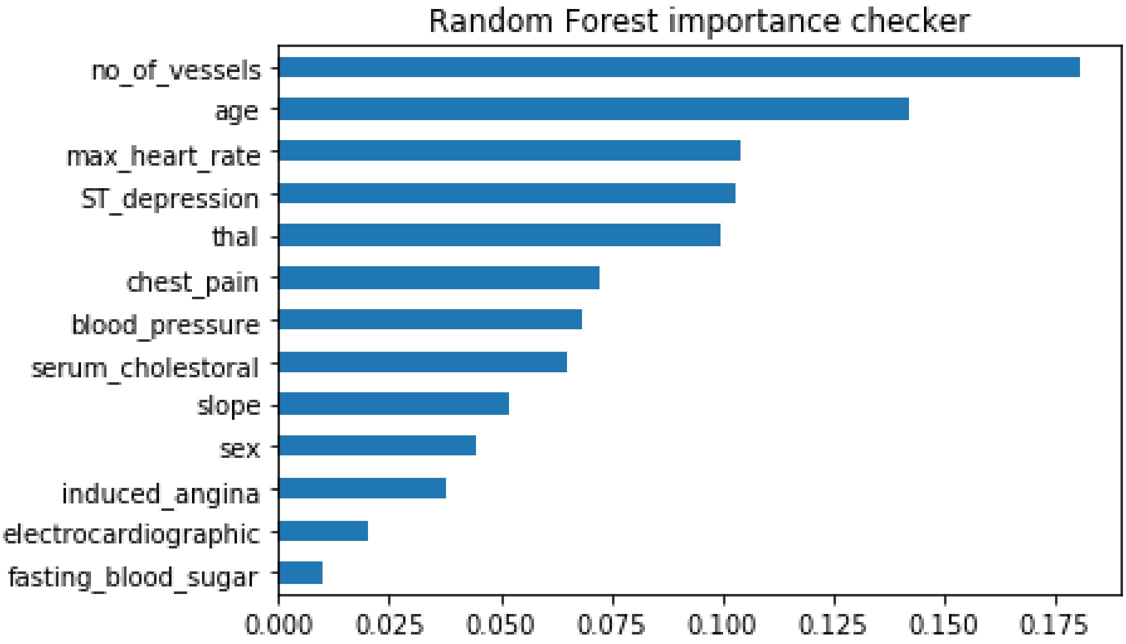


Fig. 4.2 Random forest importance checker

Classifier	New Accuracy (%)	Previous Accuracy (%)	Accuracy increase (%)
SVM	91.11	86.81	3.30
Decision Tree	84.62	70.33	14.29
Random Forest	91.21	89.01	2.20
Gaussian Naive Bayes	91.21	91.21	0

Table 4.2 Accuracy Comparison

widely believed that we should trust our CV scores under such situation. Ideally we would want CV scores obtained by different approaches to improve in sync with each other and with the LB score, but this is not always possible [4]. Usually 5-fold CV is good enough. If we use more folds, the CV score would become more reliable, but the training takes longer to finish as well. However, we shouldn't use too many folds if our training data is limited. Otherwise we would have too few samples in each fold to guarantee statistical significance. Many times the data is imbalanced, i.e. there may be a high number of class1 instances but less number of other class instances. Thus we should train and test our1 algorithm on each and every instance of the dataset [4]. Then we can take an average of all the noted accuracies over the dataset. 1. The k-Fold Cross Validation works by first dividing the dataset into k-subsets. 2. Let's say we divide the dataset into (k=10) parts. We reserve 1 part for testing and train the algorithm over the other 9 parts. 3. We continue the process by changing the testing part in each iteration and training the algorithm over the other parts. The accuracies and errors are then averaged to get an average accuracy of the algorithm. 4. An algorithm may under fit over a dataset for some training data and sometimes also over fit the data for other training set. Thus with cross-validation, we can achieve a generalized model.

4.7 Analysis

From all the tables above, different algorithms performed better depending upon the situation whether cross validation and feature selection is used or not. Every algorithm has its intrinsic capacity to out-perform other algorithm depending upon the situation. For example, Random Forest performs much better with a large number of datasets than when data is small [4]. While support vector machine performs better with a smaller number of datasets. In case of decision tree missing values play an important role. Even after imputing it can't give the result which it can with a perfect dataset. Gaussian naive Bayes is the best classifier on this dataset [4]. The reason of its pre-assumption that all the attributes are independent. If there was a dependency between the attributes in the dataset it would have given less accuracy.

Chapter 5

Results and Analysis

5.1 Conclusion

In this reasearch we have tried to compare different machine learning algorithms and predict if a certain person, given various personal characteristics and symptoms, will get heart disease or not. The main motive of our report was to comparing the accuracy and analyzing the reasons behind the variation of different algorithms. We have used Cleveland dataset for heart diseases which contains 303 instances and used 10-fold Cross Validation to divide the data into two sections which are training and testing datasets [5]. We have considered 13 attributes and implemented four different algorithms to analyze the accuracy. By the end of the implementation part, we have found Gaussian Naive Bayes and Random Forest are giving the maximum accuracy level in our dataset which is 91.21 percent and Decision Tree is performing the lowest level of accuracy which is 84.62 percent. Probably for other instances and other datasets other algorithm may work in better way but in our case we have found this result [6]. Moreover, if we increase the attributes, maybe we can found more accurate result but it will take more time to process and the system will be slower than now as it will be little more complex and will be handling more data's. So considering these possible things we took a decision which is better for us to work with.

5.2 Future Scope

The dataset that is used in our thesis is very small and old. Moreover, no new dataset regarding heart disease has been introduced so far. There is a need of new dataset and we can collect that from various hospitals of Bangladesh [7]. We can also evaluate the efficiency of each individual classifier and also such classifiers in combination, by employing the bagging, boosting and stacking techniques.

References

[1] Arnold, C. (1990). *Heart disease*. Franklin Watts.

[2] Canfield, J., Hansen, M. V., and Rackner, V. (2005). *Heart disease*. Health Communications.

[3] Carney, R. M. and Freedland, K. E. (2010). Psychotherapies for depression in people with heart disease. *Depression and Heart Disease*, page 145–168.

[4] Dittmer, L. (2012). *Heart disease*. Creative Education.

[5] Goetz, T. (2010). *The decision tree*. Rodale.

[6] Gold, J. C. and Cutler, D. J. (2000). *Heart disease*. Enslow Publishers.

[7] Healey, J. (2005). *Heart disease*. Spinney Press.

[8] Hook, S. V. (2001). *Heart disease*. Smart Apple Media.

[9] Jiang, W. and Xiong, G. L. (2010). Epidemiology of the comorbidity between depression and heart disease. *Depression and Heart Disease*, page 1–37.

[10] Johansson, P. (1998). *Heart disease*. Enslow Publishers.

[11] Klapholz, M. (2003). Heart failure in the elderly. *Heart Disease*, 5(4):241–243.

[12] Mensah, G. A. (2009). The burden of valvular heart disease. *Valvular Heart Disease: A Companion to Braunwalds Heart Disease*, page 1–18.

[13] Mervyn, L. and Mervyn, L. (1990). *Heart disease*. Thorsons.

[14] Moodie, D. (2016). The journal congenital heart disease-2016. *Congenital Heart Disease*, 11(1):5–6.

[15] Morris, P., Warriner, D., Morton, A., and Mayhew, P. (2016). *Heart disease*. JP Medical Ltd.

[16] Naff, C. F. (2008). *Heart disease*. Greenhaven Press.

[17] Otto, C. M. (2009). Evaluation of valvular heart disease by echocardiography. *Valvular Heart Disease: A Companion to Braunwalds Heart Disease*, page 62–84.

[18] Rajamannan, N. M. (2009). Cellular, molecular, and genetic mechanisms of valvular heart disease. *Valvular Heart Disease: A Companion to Braunwalds Heart Disease*, page 39–54.

[19] Roberts, W. C. and Ko, J. M. (2009). Clinical pathology of valvular heart disease. *Valvular Heart Disease: A Companion to Braunwalds Heart Disease*, page 19–38.

[20] Shavelle, D. M. (2009). Evaluation of valvular heart disease by cardiac catheterization and angiocardiography. *Valvular Heart Disease: A Companion to Braunwalds Heart Disease*, page 85–100.

[21] Sheen, B. (2004). *Heart disease*. Thomson/Gale.

[22] Silverstein, A., Silverstein, V. B., and Nunn, L. S. (2006). *Heart disease*. Lerner.

[23] Stout, K. (2009). Valvular heart disease in pregnancy. *Valvular Heart Disease: A Companion to Braunwalds Heart Disease*, page 424–436.

[24] Thomas, R. (2002). *Heart disease*. Vega.

[25] Tiger, S. and Reingold, M. (1986). *Heart disease*. J. Messner.