



(Established under Galgotias University Uttar Pradesh Act No. 14 of 2011)

HEART HEALTH PREDICTION

A Report for the Evaluation 3 of Project 2

Submitted by

SHANU KUMAR

(1613101667)

in partial fulfilment for the award of the degree

of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

SCHOOL OF COMPUTING SCIENCE AND ENGINEERING

**Under the Supervision of
Mr. GAUTAM KUMAR
Assistant Professor**

APRIL / MAY- 2020



**SCHOOL OF COMPUTING AND SCIENCE AND
ENGINEERING**

BONAFIDE CERTIFICATE

Certified that this project report **“HEART HEALTH PREDICTION”** is the bonafide work of **“SHANU KUMAR (1613101667)”** who carried out the project work under my supervision.

SIGNATURE OF HEAD

Dr. MUNISH SHABARWAL,
PhD (Management), PhD (CS)
**Professor & Dean,
School of Computing Science &
Engineering**

SIGNATURE OF SUPERVISOR

Mr. GAUTAM KUMAR
**Assistant Professor
School of Computing Science &
Engineering**

TABLE OF CONTENTS

| CHAPTER NO. | TITLE | PAGE NO. |
|-------------|---------------------------------|-----------|
| | ABSTRACT | 4 |
| 1. | INTRODUCTION | 5 |
| | 1.1 GENERAL | 5 |
| | 1.2 MACHINE LEARNING | 5 |
| | 1.2.1 General | 5 |
| | 1.2.2 Machine Learning Types | 10 |
| | 1.2.2.1 General | 10 |
| | 1.2.2.2 Supervised | 11 |
| | 1.2.2.3 Semi-Supervised | 12 |
| | 1.2.2.4 Unsupervised | 13 |
| | 1.2.2.5 Reinforcement | 14 |
| | 1.3 WHAT IS HEART DISEASE | 16 |
| 2. | LITERATURE REVIEW | 21 |
| | 2.1 GENERAL | 21 |
| 3. | IMPLEMENTATION OF MODEL | 24 |
| | 3.1 Existing System | 27 |
| | 3.2 Proposed System | 27 |
| | 3.2.1 Datasets | 28 |
| | 3.3 Implementation | 29 |
| | 3.3.1 Data Exploratory Analysis | 29 |
| | 3.3.2 Source code | 33 |
| 4. | RESULTS | 37 |
| 5. | CONCLUSION | 40 |
| | 5.1 Future Scope | 41 |
| | 5.2 Conclusion | 41 |
| 6. | REFERENCES | 42 |

ABSTRACT

The health care industry produces a huge amount of data. This data is not always made use to the full extent and is often underutilized. Using this huge amount of data, a disease can be detected, predicted or even cured. A huge threat to human kind is caused by diseases like heart disease, cancer, tumour and Alzheimer's disease. In this , we try to concentrate on heart disease prediction. Using machine learning techniques, the heart disease can be predicted. The medical data such as Blood pressure, hypertension, diabetes and so on is taken as input and then these features are modelled for prediction. This model can then be used to predict future medical data. The algorithms like K-nearest neighbour, Naïve Bayes, and decision tree are used. The accuracy of the model using each of the algorithms is calculated. Then the one with a good accuracy is taken as the model for predicting the heart disease.

Keywords: Coronary heart disease, Decision tree, K nearest neighbour, Machine Learning, Naïve Bayes.

1. INTRODUCTION

1.1 General

In our day to day life, people are undergoing a routine and busy schedule which leads to stress and anxiety. In addition to this, the percentage of people who are obese and addicted to cigarette goes up drastically. This leads to diseases like heart disease, cancer, etc. The challenge behind these diseases is its prediction. Each person has different values of pulse rate and blood pressure. But medically proven, the pulse rate must be 60 to 100 beats per minute and the blood pressure must be in the range of 120/80 to 140/90. Heart disease is one of the major cause of death in the world. The number of people affected by heart disease increases irrespective of age in both men and women. But other factors like gender, diabetes, BMI also contribute to this disease. In this, we have tried prediction and analysis of heart disease by considering the parameters like age, gender, blood pressure, heart rate, diabetes and so on. Since numerous factors are involved in heart disease, the prediction of this disease is challenging. Some of major symptoms of heart attack are:

- Chest tightness.
- Shortness of breath.
- Nausea, Indigestion, Heartburn, or stomach pain.
- Sweating and Fatigue.
- Pressure in the upper back Pain that spreads to the arm.

The following are the type of heart disease: Heart means “cardio”. Hence all heart diseases concern to category of cardiovascular diseases. The different kinds of heart disease are:

- Coronary heart diseases.
- Angina pectoris
- Congestive heart failure.
- Cardiomyopathy
- Congenital heart diseases.

Coronary heart disease or coronary artery disease is the narrowing of the coronary arteries. The coronary arteries supply oxygen and blood to the heart. It causes a large number of people to become ill or to face death. It is one of the popular type of heart disease. High blood glucose from diabetes can damage blood vessels and nerves that control heart and blood vessels. If a person has diabetes for a longer time, there are high chances for that person to have heart disease in future. With diabetes, there are other reasons which contribute to heart disease. They are smoking which raises the risk of developing heart disease, high blood pressure makes the heart work harder to pump blood and it can strain heart and damage blood vessels, abnormal cholesterol levels also contribute to heart disease and obesity. Also, family history of heart disease can be a cause of having heart disease. But this history is not considered in this paper for prediction of heart disease. The other risk factors include age, gender, stress and unhealthy diet. Chance of having a heart disease increases when a person is getting older. Men have a greater risk of heart disease. However, women also have the same risk after menopause. Leading a stressed life can also damage the arteries and increase the chance of coronary heart disease. So, in this paper based on the factors mentioned above we try to predict the risk of heart disease. A large amount of work has been done related to heart prediction system by using various techniques and algorithms by many authors. These techniques may be based on deep-learning, machine-learning, data mining and so on. The aim of all those papers is to achieve better accuracy and to make the system more efficient so that it can predict the chances of heart attack.

1.2 MACHINE LEARNING

1.2.1 General

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



Fig 1. Machine Learning

The term Machine Learning was coined by Arthur Samuel in 1959, an American pioneer in the field of computer gaming and artificial intelligence and stated that “it gives computers the ability to learn without being explicitly programmed”.

And in 1997, Tom Mitchell gave a “well-posed” mathematical and relational definition that “A computer program is said to learn from experience E with respect to some task T and some performance measure P , if its performance on T , as measured by P , improves with experience E .”

Machine learning involves a computer to be trained using a given data set, and use this training to predict the properties of a given new data. For example, we can train a computer by feeding it 1000 images of cats and 1000 more images which are not of a cat, and tell each time to the computer whether a picture is cat or not. Then if we show the computer a new image, then from the above training, the computer should be able to tell whether this new image is a cat or not.

Let’s try to understand Machine Learning in layman terms. Consider you are trying to toss a paper to a dustbin.

After first attempt, you realize that you have put too much force in it. After second attempt, you realize you are closer to target but you need to increase your throw angle. What is happening here is basically after every throw we are learning something and improving the end result. We are programmed to learn from our experience.

This implies that the tasks in which machine learning is concerned offers a fundamentally operational definition rather than defining the field in cognitive terms. This follows Alan Turing's proposal in his paper "Computing Machinery and Intelligence", in which the question "Can machines think?" is replaced with the question "Can machines do what we (as thinking entities) can do?" Within the field of data analytics, machine learning is used to devise complex models and algorithms that lend themselves to prediction; in commercial use, this is known as predictive analytics. These analytical models allow researchers, data scientists, engineers, and analysts to "produce reliable, repeatable decisions and results" and uncover "hidden insights" through learning from historical relationships and trends in the data set(input).

Suppose that you decide to check out that offer for a vacation . You browse through the travel agency website and search for a hotel. When you look at a specific hotel, just below the hotel description there is a section titled "You might also like these hotels". This is a common use case of Machine Learning called "Recommendation Engine". Again, many data points were used to train a model in order to predict what will be the best hotels to show you under that section, based on a lot of information they already know about you.

So if you want your program to predict, for example, traffic patterns at a busy intersection (task T), you can run it through a machine learning algorithm with data about past traffic patterns (experience E) and, if it has successfully "learned", it will then do better at predicting future traffic patterns (performance measure P).

The highly complex nature of many real-world problems, though, often means that inventing specialized algorithms that will solve them perfectly every time is impractical, if not impossible. Examples of machine learning problems include, "Is this cancer?",

“Which of these people are good friends with each other?”, “Will this person like this movie?” such problems are excellent targets for Machine Learning, and in fact machine learning has been applied such problems with great success.

When do we need Machine Learning?

When do we need machine learning rather than directly program our computers to carry out the task at hand? Two aspects of a given problem may call for the use of programs that learn and improve on the basis of their “experience”: the problem’s complexity and the need for adaptivity.

Tasks That Are Too Complex to Program.

- **Tasks Performed by Animals/Humans:** There are numerous tasks that we human beings perform routinely, yet our introspection concerning how we do them is not sufficiently elaborate to extract a well-defined program. Examples of such tasks include driving, speech recognition, and image understanding. In all of these tasks, state of the art machine learning programs, programs that “learn from their experience,” achieve quite satisfactory results, once exposed to sufficiently many training examples.

- **Tasks beyond Human Capabilities:** Another wide family of tasks that benefit from machine learning techniques are related to the analysis of very large and complex data sets: astronomical data, turning medical archives into medical knowledge, weather prediction, analysis of genomic data, Web search engines, and electronic commerce. With more and more available digitally recorded data, it becomes obvious that there are treasures of meaningful information buried in data archives that are way too large and too complex for humans to make sense of. Learning to detect meaningful patterns in large and complex data sets is a promising domain in which the combination of programs that learn with the almost unlimited memory capacity and ever increasing processing speed of computers opens up new horizons. Adaptivity. One limiting feature of programmed tools is their rigidity – once the program has been written down and installed, it stays unchanged.

However, many tasks change over time or from one user to another. Machine learning tools – programs whose behavior adapts to their input data – offer a solution to such issues; they are, by nature, adaptive to changes in the environment they interact with. Typical successful applications of machine learning to such problems include programs that decode handwritten text, where a fixed program can adapt to variations between the handwriting of different users; spam detection programs, adapting automatically to changes in the nature of spam e-mails; and speech recognition programs.

Terminologies of Machine Learning

- **Model**

A model is a specific representation learned from data by applying some machine learning algorithm. A model is also called hypothesis.

- **Feature**

A feature is an individual measurable property of our data. A set of numeric features can be conveniently described by a feature vector. Feature vectors are fed as input to the model. For example, in order to predict a fruit, there may be features like color, smell, taste, etc.

Note: Choosing informative, discriminating and independent features is a crucial step for effective algorithms. We generally employ a feature extractor to extract the relevant features from the raw data.

- **Target (Label)**

A target variable or label is the value to be predicted by our model. For the fruit example discussed in the features section, the label with each set of input would be the name of the fruit like apple, orange, banana, etc.

- **Training**

The idea is to give a set of inputs(features) and it's expected outputs(labels), so after training, we will have a model (hypothesis) that will then map new data to one of the categories trained on.

- **Prediction**

Once our model is ready, it can be fed a set of inputs to which it will provide a predicted output(label).

The figure shown below clears the above concepts:

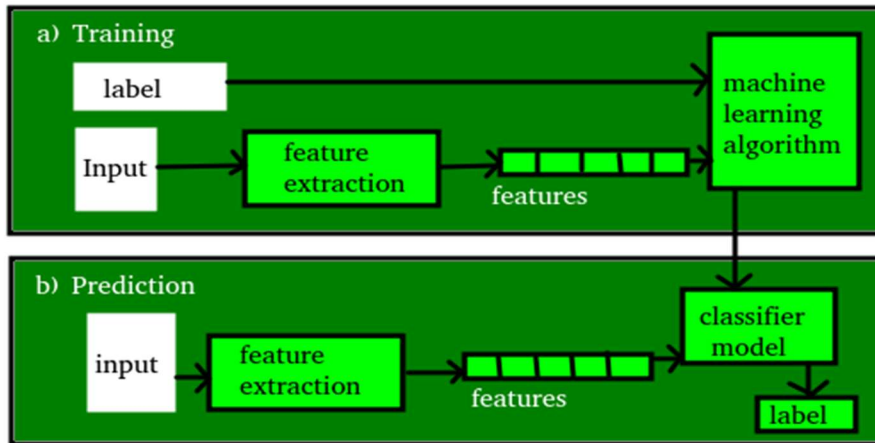


Fig-2 Training and Prediction

1.2.2 Machine Learning Types

1.2.2.1 General

Learning is, of course, a very wide domain. Consequently, the field of machine learning has branched into several subfields dealing with different types of learning tasks. We give a rough taxonomy of learning paradigms, aiming to provide some perspective of where the content sits within the wide field of machine learning.

Terms frequently used are:

- **Labeled data:** Data consisting of a set of training examples, where each example is a pair consisting of an input and a desired output value (also called the supervisory signal, labels, etc)
- **Classification:** The goal is to predict discrete values, e.g. {1,0}, {True, False}, {spam, not spam}.
- **Regression:** The goal is to predict continuous values, e.g. home prices.

There are some variations of how to define the types of Machine Learning Algorithms but commonly they can be divided into categories according to their purpose and the main categories are the following:

- Supervised learning
- Unsupervised Learning
- Semi-supervised Learning
- Reinforcement Learning

1.2.2.2 Supervised Learning

- I like to think of supervised learning with the concept of function approximation, where basically we train an algorithm and in the end of the process we pick the function that best describes the input data, the one that for a given X makes the best estimation of y ($X \rightarrow y$). Most of the time we are not able to figure out the true function that always makes the correct predictions and other reason is that the algorithm relies upon an assumption made by humans about how the computer should learn and these assumptions introduce a bias.
- Here the human experts act as the teacher where we feed the computer with training data containing the input/predictors and we show it the correct answers (output) and from the data the computer should be able to learn the patterns.
- Supervised learning algorithms try to model relationships and dependencies between the target prediction output and the input features such that we can predict the output values for new data based on those relationships which it learned from the previous data sets.

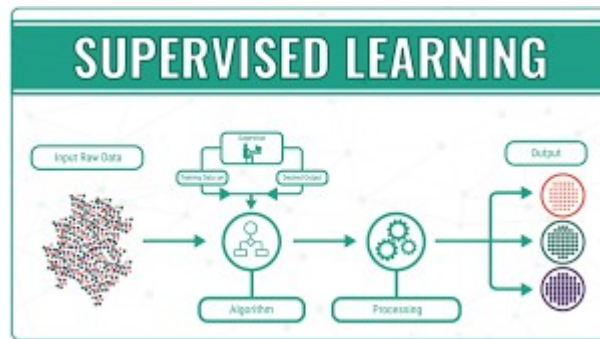


Fig 3: Supervised Learning

Draft

- Predictive Model
- we have labelled data
- The main types of supervised learning problems include regression and classification problems

List of Common Algorithms

- Nearest Neighbour
- Naive Bayes
- Decision Trees
- Linear Regression
- Support Vector Machines (SVM)
- Neural Networks

1.2.2.3 Unsupervised Learning

- The computer is trained with unlabelled data.
- Here there's no teacher at all, actually the computer might be able to teach you new things after it learns patterns in data, these algorithms are particularly useful in cases where the human expert doesn't know what to look for in the data.
- are the family of machine learning algorithms which are mainly used in pattern detection and descriptive modelling. However, there are no output categories or labels here based on which the algorithm can try to model relationships. These algorithms try to use techniques on the input data to mine for rules, detect patterns, and summarize and group the data points which help in deriving meaningful insights and describe the data better to the users.

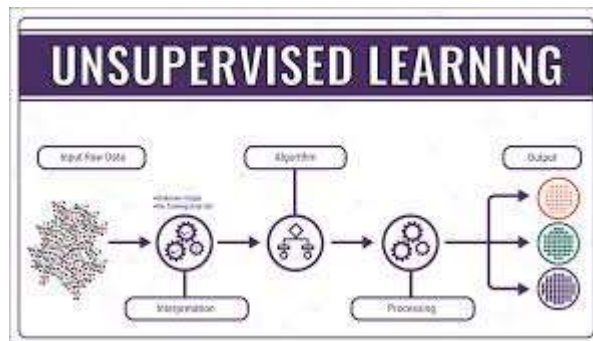


Fig 4: Unsupervised Learning

Draft

- Descriptive Model
- The main types of unsupervised learning algorithms include Clustering algorithms and Association rule learning algorithms.

List of Common Algorithms

- k-means clustering, Association Rules

1.2.2.4 Semi-Supervised Learning

In the previous two types, either there are no labels for all the observation in the dataset or labels are present for all the observations. Semi-supervised learning falls in between these two. In many practical situations, the cost to label is quite high, since it requires skilled human experts to do that. So, in the absence of labels in the majority of the observations but present in few, semi-supervised algorithms are the best candidates for the model building. These methods exploit the idea that even though the group memberships of the unlabeled data are unknown, this data carries important information about the group parameters.

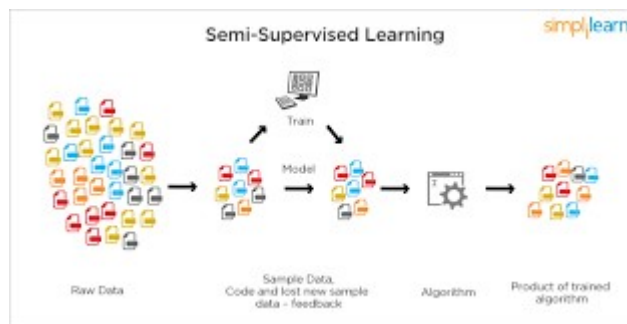


Fig 5: Semi-Supervised Learning

1.2.2.5 Reinforcement Learning

method aims at using observations gathered from the interaction with the environment to take actions that would maximize the reward or minimize the risk. Reinforcement learning algorithm (called the agent) continuously learns from the environment in an iterative fashion. In the process, the agent learns from its experiences of the environment until it explores the full range of possible states.

Reinforcement Learning is a type of Machine Learning, and thereby also a branch of Artificial Intelligence. It allows machines and software agents to automatically determine the ideal behaviour within a specific context, in order to maximize its

performance. Simple reward feedback is required for the agent to learn its behaviour; this is known as the reinforcement signal.

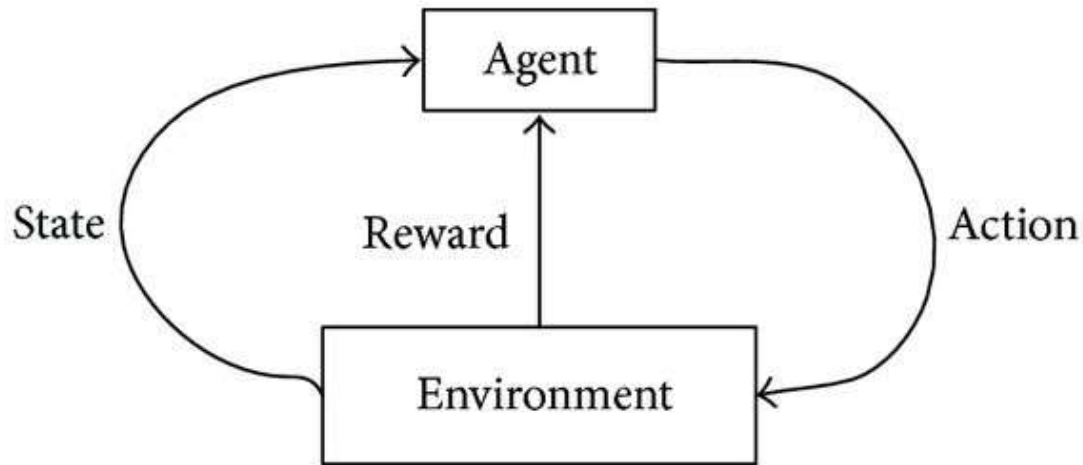


Fig 6: Reinforcement Learning

There are many different algorithms that tackle this issue. As a matter of fact, Reinforcement Learning is defined by a specific type of problem, and all its solutions are classed as Reinforcement Learning algorithms. In the problem, an agent is supposed to decide the best action to select based on his current state. When this step is repeated, the problem is known as a Markov Decision Process.

In order to produce intelligent programs (also called agents), reinforcement learning goes through the following steps:

1. Input state is observed by the agent.
2. Decision making function is used to make the agent perform an action.
3. After the action is performed, the agent receives reward or reinforcement from the environment.
4. The state-action pair information about the reward is stored.

List of Common Algorithms

- Q-Learning
- Temporal Difference (TD)
- Deep Adversarial Networks

Use cases:

Some applications of the reinforcement learning algorithms are computer played board games (Chess, Go), robotic hands, and self-driving cars.

1.3 What is Heart Disease?

Heart disease describes a range of conditions that affect your heart. Diseases under the heart disease umbrella include blood vessel diseases, such as coronary artery disease; heart rhythm problems (arrhythmias); and heart defects you're born with (congenital heart defects), among others. The term "heart disease" is often used interchangeably with the term "cardiovascular disease." Cardiovascular disease generally refers to conditions that involve narrowed or blocked blood vessels that can lead to a heart attack, chest pain (angina) or stroke. Other heart conditions, such as those that affect your heart's muscle, valves or rhythm, also are considered forms of heart disease. Heart failure is a serious condition with high prevalence (about 2% in the adult population in developed countries, and more than 8% in patients older than 75 years). About 3 – 5% of hospital admissions are linked with heart failure incidents. Heart failure is the first cause of admission by healthcare professionals in their clinical practice.

The costs are very high, reaching up to 2% of the total health costs in the developed countries. Building an effective disease management strategy requires analysis of large amount of data, early detection of the disease, assessment of the severity and early prediction of adverse events. This will inhibit the progression of the disease, will improve the quality of life of the patients and will reduce the associated medical costs. Toward this direction machine learning techniques have been employed. The aim of this paper is to present the state-of-the-art of the machine learning methodologies applied for the assessment of heart failure. More specifically, models predicting the presence, estimating the subtype, assessing the severity of heart failure and predicting the presence of adverse events, such as destabilizations, re-hospitalizations, and mortality are presented. According to the authors' knowledge, it is the first time that such a comprehensive review, focusing on all aspects of the management of heart failure, is presented.

Naïve Bayes:

Bayes' Theorem is stated as:

$$P(h|d) = (P(d|h) * P(h)) / P(d)$$

- $P(h|d)$ is the probability of hypothesis h given the data d . This is called the posterior probability.
- $P(d|h)$ is the probability of data d given that the hypothesis h was true.
- $P(h)$ is the probability of hypothesis h being true (regardless of the data). This is called the prior probability of h .
- $P(d)$ is the probability of the data (regardless of the hypothesis). we are interested in calculating the posterior probability of $P(h|d)$ from the prior probability $p(h)$ with $P(D)$ and $P(d|h)$. After calculating the posterior probability for a number of different hypotheses, we will select the hypothesis with the highest probability. This is the maximum probable hypothesis and may formally be called the (MAP) hypothesis.

This can be written as:

$$\text{MAP}(h) = \max(\text{P}(h|d))$$

or

$$\text{MAP}(h) = \max((\text{P}(d|h) * \text{P}(h)) / \text{P}(d))$$

or

$$\text{MAP}(h) = \max(\text{P}(d|h) * \text{P}(h))$$

The $\text{P}(d)$ is a normalizing term which allows us to calculate the probability. We can drop it when we are interested in the most probable hypothesis as it is constant and only used to normalize. Back to classification, if we have an even number of instances in each class in our training data, then the probability of each class (e.g. $\text{P}(h)$) will be equal. Again, this would be a constant term in our equation, and we could drop it so that we end up with:

$$\text{MAP}(h) = \max(\text{P}(d|h))$$

K-Nearest Neighbor

We can implement a KNN model by following the below steps:

1. Load the data
2. Initialize the value of k
3. For getting the predicted class, iterate from 1 to total number of training data points
 - Calculate the distance between test data and each row of training data. Here we will use Euclidean distance as our distance metric since it's the most popular method. The other metrics that can be used are Chebyshev, cosine, etc.

- Sort the calculated distances in ascending order based on distance values
- Get top k rows from the sorted array
- Get the most frequent class of these rows
- Return the predicted class

Decision Tree

Pseudocode

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

- 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 placing attributes as root or internal node of the tree is done by using some statistical approach.

The popular attribute selection measures

- Information gain
- Gini index

2. LITERATURE SURVEY

2.1 General

Monika Gandhi et.al, [1] used Naïve Bayes, Decision tree and neural network algorithms and analysed the medical dataset. There are an enormous number of features involved. So, there's a requirement to scale back the amount of features. this will be done by feature selection. On doing this, they assert that point is reduced.They made use of decision tree and neural networks. J Thomas, R Theresa Princy [2] made use of K nearest neighbour algorithm, neural network, naïve Bayes and decision tree for heart disease prediction. They made use of data mining techniques to detect the heart disease risk rate. Sana Bharti, Shailendra Narayan Singh [3] made use of Particle Swarm Optimization, Artificial neural network, Genetic algorithm for prediction. Associative classification is a new and efficient technique which integrates association rule mining and classification to a model for prediction and achieved good accuracy. Purushottam et.al, [4] proposed “An automated system in medical diagnosis would enhance medical care and it can also reduce costs. In this study, we have designed a system that can efficiently discover the rules to predict the risk level of patients based on the given parameter about their health. The rules can be prioritized based on the user's requirement. The performance of the system is evaluated in terms of classification accuracy and the results shows that the system has great potential in predicting the heart disease risk level more accurately”. Sellappan Palaniyappan, Rafiah Awang [5] made use of decision tree Naïve Bayes, Decision tree, Artificial Neural Networks to build Intelligent Heart Disease Prediction Systems (IHDPS).To enhance visualization and ease of interpretation, it displays the results both in tabular and graphical forms. By providing effective treatments, it also helps to reduce treatment costs. Discovery of hidden patterns and relationships often has gone unexploited. Advanced data mining techniques helped remedy this situation.

Himanshu Sharma, M A Rizvi [6] made use of Decision tree, support vector machine, deep learning, K nearest neighbour algorithms. Since the datasets contain noise, they tried to reduce the noise by cleaning and pre-processing the dataset and also tried to reduce the dimensionality of the dataset. They found that good accuracy can be achieved with neural networks. Animesh Hazra et.al, [7] discussed in detail the cardiovascular disease and different symptoms of heart attack. The different types of classification and clustering algorithms and tools were used. V.Krishnaiah, G.Narsimha, N.Subhash Chandra [8] presented an analysis using data mining. The analysis showed that using different techniques and taking different number of attributes gives different accuracies for predicting heart diseases. Ramandeep Kaur, Er.Prabhsharn Kaur [9] have showed that the heart disease data contains unnecessary, duplicate information. This has to be pre processed. Also, they say that feature selection has to be done on the dataset for achieving better results. J.Vijayashree and N.Ch.SrimanNarayanaIyengar [10] used data mining. A huge amount of data is produced on a daily basis. As such, it cannot be interpreted manually. Data mining can be effectively used to predict diseases from these datasets. In this paper, different data mining techniques are analysed on heart disease database. In conclusion, this paper analyses and compares how different classification algorithms work on a heart disease database. Benjamin EJ et.al [11] says that there are seven key factors for heart disease such as smoking, physical inactivity, nutrition, obesity, cholesterol, diabetes and high blood pressure. They also discussed the statistics of heart disease including stroke and cardio vascular disease. Abhay Kishore et.al [12] on their experimentation showed that recurrent neural network gives good accuracy when compared to other algorithms like CNN, Naïve Bayes and SVM. Hence, neural networks perform well in heart disease prediction. They also achieved a system that could predict silent heart attacks and inform the user as earliest possible.

M.Nikhil Kumar et.al [13] used various algorithms – Decision tree, random forest, Naïve Bayes, KNN, Support vector machine, logistic model tree algorithm. Naïve Bayes algorithm gave good results when compared to other algorithms. They made use of UCI repository of heart disease dataset. Also, J48 algorithm took less time to build and gave good results. Amandeep Kaur et.al [14] compared various algorithms such as artificial neural network, K – nearest neighbour, Naïve Bayes, Support vector machine on heart disease prediction. Stephen F Weng et.al [15] used four machine learning algorithms such as logistic regression, random forest, gradient boosting machines and neural networks. They showed that machine learning algorithms perform well at predicting the heart disease cases correctly. They say that this is the first experimentation using machine learning techniques to routine patient data in electronic records. The source of the dataset is the Clinical Practice Research Datalink (CPRD). These are the electronic medical records which contains all the medical related data such as statistics of human population, medical history, specialists. It also contains details of medicine intake, outcomes and details of hospital admissions. Sahaya Arthy et.al [16] analyse the existing works on heart disease prediction which uses data mining. The data mining techniques are commonly used in heart disease prediction. They also discuss the databases used such as the heart disease dataset from UCI repository, tools used such as Weka, Rapid Miner, Data melt, Apache Mahout, Rattle, KEEL, R data mining and so on. They conclude that use of single algorithm results in better accuracy in prediction. But use of hybridization of two or more algorithms can enhance and improve the heart disease prediction

3. IMPLEMENTATION OF MODEL

3.1 EXISTING SYSTEM

The healthcare environment is still „information rich“ but „knowledge poor“ There is a wealth of data available within the healthcare systems. However, there is a lack of effective analysis tools to discover hidden relationships and trends in the data for African. Medical diagnosis is considered as a significant yet intricate task that needs to be carried out precisely and efficiently. The automation of the same would be highly beneficial. Clinical decisions are often made based on doctor’s intuition and experience rather than on the knowledge rich data hidden in the database. This practice leads to unwanted biases, errors and excessive medical costs which affects the quality of service provided to patients. Data mining have the potential to generate a knowledge-rich environment which can help to significantly improve the quality of clinical decisions.

.

3.2 PROPOSED SYSTEM

In this paper, comparison of various machine learning methods is done for predicting the 10 year risk of coronary heart disease of the patients from their medical data. The algorithms like K- nearest neighbour, Naïve Bayes, and decision tree are used. The accuracy of the model using each of the algorithms is calculated. Then the one with a good accuracy is taken as the model for predicting the heart disease.

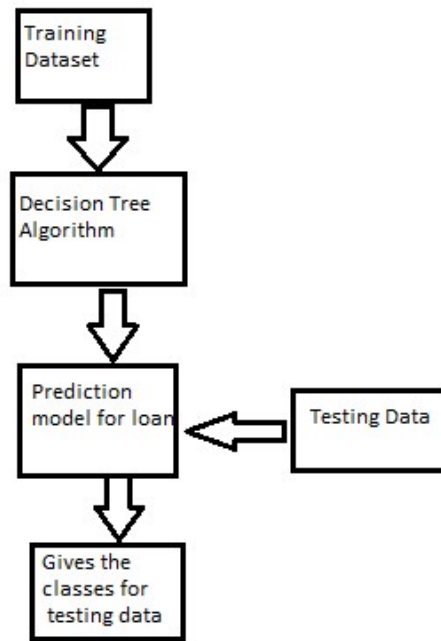


Fig 9: Architecture of proposed system

In the proposed model for diseases prediction, Dataset is split into training and testing data. After then training datasets are trained using algorithm and a prediction model is developed using the algorithm. Testing datasets are then given to model for the prediction of loan. Various libraries like pandas, numpy have been used. After the loading of datasets, Data Preprocessing like missing value treatment of numerical and categorical is done by checking the values. Numerical and categorical values are segregated. Outliers and frequency analysis are done, outliers are checked by getting the boxplot diagram of attributes.

3.2.1 DATA SET:

Datasets are gathered from Kaggle. Data set is now provided to Machine learning models on the basis of this facts this version is trained. The dataset used in this project contains 14 variables. The independent variable that needs to be predicted, 'diagnosis', determines whether a person is healthy or suffer from heart disease.

Experiments with the Cleveland database have concentrated on endeavors to distinguish disease presence (values 1, 2, 3, 4) from absence (value 0). There are several missing attribute values, distinguished with symbol '?'. The header row is missing in this dataset, so the column names have to be inserted manually.

Features information:

- age - age in years
- sex - sex (1 = male; 0 = female)
- chest pain - chest pain type (1 = typical angina; 2 = atypical angina; 3 = nonanginal pain; 4 = asymptomatic)
- blood pressure - resting blood pressure (in mm Hg on admission to the hospital)
- serum cholesterol - serum cholesterol in mg/dl
- fasting blood sugar - fasting blood sugar > 120 mg/dl (1 = true; 0 = false)
- electrocardiographic - resting electrocardiographic results (0 = normal; 1 = having ST-T; 2 = hypertrophy)
- max heart rate - maximum heart rate achieved
- induced angina - exercise induced angina (1 = yes; 0 = no)
- ST depression - ST depression induced by exercise relative to rest
- slope - the slope of the peak exercise ST segment (1 = upsloping; 2 = flat; 3 = down sloping)
- no of vessels - number of major vessels (0-3) colored by fluoroscopy
- thalassemia - 3 = normal; 6 = fixed defect; 7 = reversable defect
- diagnosis - the predicted attribute - diagnosis of heart disease (angiographic disease status) (Value 0 = < 50% diameter narrowing; Value 1 = > 50% diameter narrowing)

| age | sex | cp | trestbps | chol | fbs | restecg | thalach | exang | oldpeak | slope | ca | thal | num |
|-----|-----|----|----------|------|-----|---------|---------|-------|---------|-------|----|------|-----|
| 63 | 1 | 1 | 145 | 233 | 1 | 2 | 150 | 0 | 2.3 | 3 | 0 | 6 | 0 |
| 67 | 1 | 4 | 160 | 286 | 0 | 2 | 108 | 1 | 1.5 | 2 | 3 | 3 | 2 |
| 67 | 1 | 4 | 120 | 229 | 0 | 2 | 129 | 1 | 2.6 | 2 | 2 | 7 | 1 |
| 37 | 1 | 3 | 130 | 250 | 0 | 0 | 187 | 0 | 3.5 | 3 | 0 | 3 | 0 |
| 41 | 0 | 2 | 130 | 204 | 0 | 2 | 172 | 0 | 1.4 | 1 | 0 | 3 | 0 |
| 56 | 1 | 2 | 120 | 236 | 0 | 0 | 178 | 0 | 0.8 | 1 | 0 | 3 | 0 |
| 62 | 0 | 4 | 140 | 268 | 0 | 2 | 160 | 0 | 3.6 | 3 | 2 | 3 | 3 |
| 57 | 0 | 4 | 120 | 354 | 0 | 0 | 163 | 1 | 0.6 | 1 | 0 | 3 | 0 |
| 63 | 1 | 4 | 130 | 254 | 0 | 2 | 147 | 0 | 1.4 | 2 | 1 | 7 | 2 |
| 53 | 1 | 4 | 140 | 203 | 1 | 2 | 155 | 1 | 3.1 | 3 | 0 | 7 | 1 |
| 57 | 1 | 4 | 140 | 192 | 0 | 0 | 148 | 0 | 0.4 | 2 | 0 | 6 | 0 |
| 56 | 0 | 2 | 140 | 294 | 0 | 2 | 153 | 0 | 1.3 | 2 | 0 | 3 | 0 |
| 56 | 1 | 3 | 130 | 256 | 1 | 2 | 142 | 1 | 0.6 | 2 | 1 | 6 | 2 |
| 44 | 1 | 2 | 120 | 263 | 0 | 0 | 173 | 0 | 0 | 1 | 0 | 7 | 0 |
| 52 | 1 | 3 | 172 | 199 | 1 | 0 | 162 | 0 | 0.5 | 1 | 0 | 7 | 0 |
| 57 | 1 | 3 | 150 | 168 | 0 | 0 | 174 | 0 | 1.6 | 1 | 0 | 3 | 0 |
| 48 | 1 | 2 | 110 | 229 | 0 | 0 | 168 | 0 | 1 | 3 | 0 | 7 | 1 |
| 54 | 1 | 4 | 140 | 239 | 0 | 0 | 160 | 0 | 1.2 | 1 | 0 | 3 | 0 |
| 48 | 0 | 3 | 130 | 275 | 0 | 0 | 139 | 0 | 0.2 | 1 | 0 | 3 | 0 |
| 49 | 1 | 2 | 130 | 266 | 0 | 0 | 171 | 0 | 0.6 | 1 | 0 | 3 | 0 |
| 64 | 1 | 1 | 110 | 211 | 0 | 2 | 144 | 1 | 1.8 | 2 | 0 | 3 | 0 |
| 58 | 0 | 1 | 150 | 283 | 1 | 2 | 162 | 0 | 1 | 1 | 0 | 3 | 0 |

Fig 10: Datasets from Kaggle

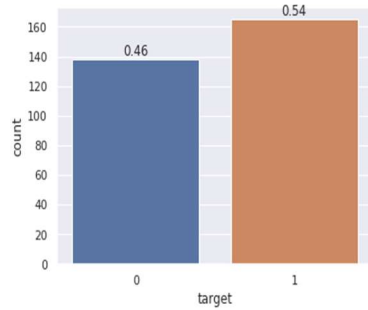
3.3 IMPLEMENTATION OF MODEL

3.3.1 Data Exploratory Analysis

Data Exploratory Analysis is done through Bivariate Analysis by Numeric (TTest) or Categorical (Chisquare). Visualization of Attributes are also done by Bivariate Analysis. We use a BivariateAnalysisPlot which is used to analyze the impact of features on the target variables.

```
# for showing the percentage
for p in ax.patches:
    height = p.get_height()
    ax.text(p.get_x()+p.get_width()/2.,
           height + 3,
           '{:1.2f}'.format(height/total),
           ha="center")
```

```
1    165
0    138
Name: target, dtype: int64
```



From the total dataset of 303 patients, 165 (54%) have a heart disease (target=1)

(1 is who have Heart Disease and 0 is who don't have Heart Disease). No. of Heart Disease patients is 165. No. of patients who don't have a heart disease is 138. [Which is a good balance of target data.]

```
In [0]: print("Percentage of patience without heart problems: "+str(round(target_temp[0]*100/303,2)))
        print("Percentage of patience with heart problems: "+str(round(target_temp[1]*100/303,2)))
```

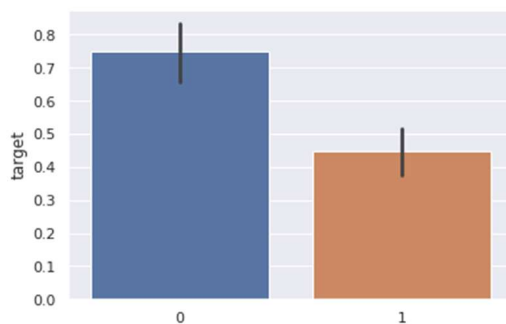
```
Percentage of patience without heart problems: 45.54
Percentage of patience with heart problems: 54.46
```

```
In [0]: data["sex"].unique()
```

```
Out[0]: array([1, 0])
```

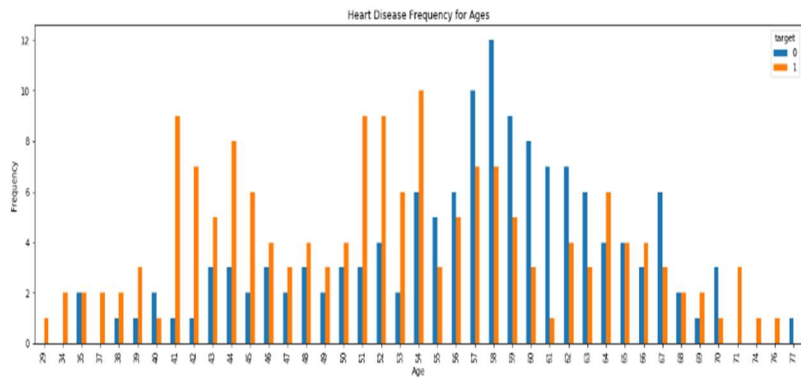
```
In [0]: sns.barplot(data["sex"],data["target"])
```

```
Out[0]: <matplotlib.axes._subplots.AxesSubplot at 0x7f6288091b70>
```



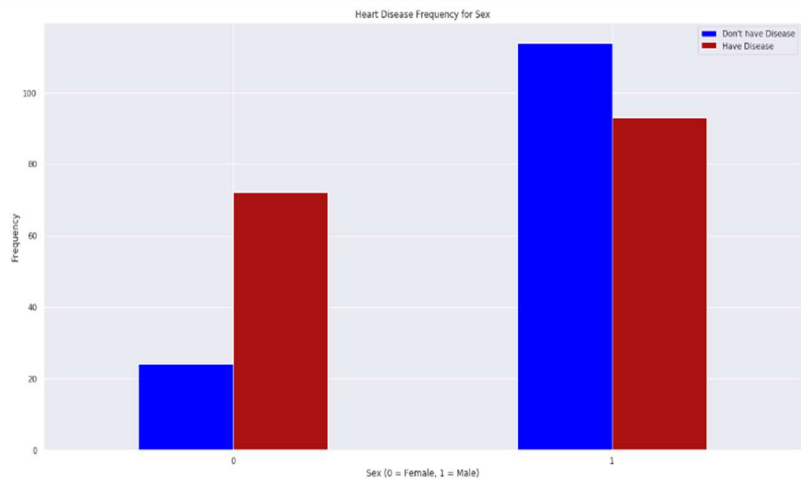
(percentage of patient with or without heart problem in the dataset)

```
In [0]: pd.crosstab(data.age,data.target).plot(kind="bar",figsize=(20,6))
plt.title('Heart Disease Frequency for Ages')
plt.xlabel('Age')
plt.ylabel('Frequency')
plt.savefig('heartDiseaseAndAges.png')
plt.show()
```



Heart Disease Frequency for ages

```
In [0]: pd.crosstab(data.sex,data.target).plot(kind="bar",figsize=(20,10),color=['blue', '#AA1111' ])
plt.title('Heart Disease Frequency for Sex')
plt.xlabel('Sex (0 = Female, 1 = Male)')
plt.xticks(rotation=0)
plt.legend(["Don't have Disease", "Have Disease"])
plt.ylabel('Frequency')
plt.show()
```

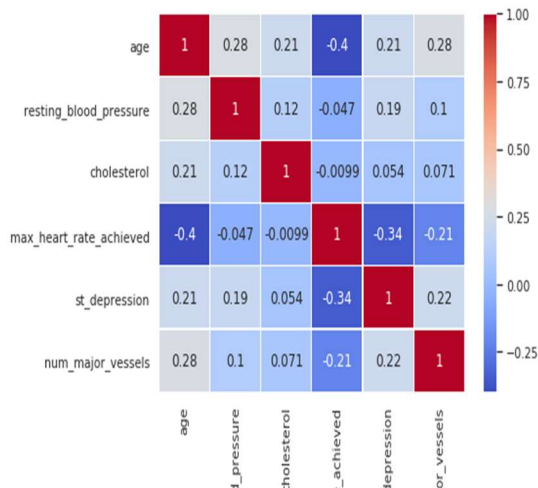


Heart Disease Frequency for sex

```
In [0]: #Set the width and height of the plot
f, ax = plt.subplots(figsize=(7, 5))

#Correlation plot
df_corr = data.loc[:,cnames]
#Generate correlation matrix
corr = df_corr.corr()

#Plot using seaborn library
sns.heatmap(corr, annot = True, cmap='coolwarm',linewidths=.1)
plt.show()
```



Correlation analysis is a method of statistical evaluation used to study the strength of a relationship between two, numerically measured, continuous variables (e.g. height and weight).

Store numeric variables in cname variable

3.3.2 Source Code

Import Libraries

```
import numpy as np
```

```
import pandas as pd
```

```
import matplotlib.pyplot as plt
```

```
import seaborn as sns
```

```
%matplotlib inline
```

```
import os
```

```
print(os.listdir())
```

```
import warnings
```

```
warnings.filterwarnings('ignore')
```

Import dataset

```
data = pd.read_csv("heart.csv")
```

splitting the dataset into train and test

```
from sklearn.model_selection import train_test_split
```

```
predictors = data.drop("target",axis=1)
```

```
target = data["target"]
```

```
X_train,X_test,Y_train,Y_test
```

=

```
train_test_split(predictors,target,test_size=0.20,random_state=0)
```

```
print("Training features have {0} records and Testing features have {1} records." \
```

```
      format(X_train.shape[0], X_test.shape[0]))
```

```
X_train.shape
```

```
X_test.shape
```

```
Y_train.shape
```

```
Y_test.shape
```

Importing accuracy score

```
from sklearn.metrics import accuracy_score
```

Modeling and predicting with Machine learning

```
def train_model(X_train, y_train, X_test, y_test, classifier, **kwargs):
```

```
    """
```

Fit the chosen model and print out the score.

```
"""  
  
# instantiate model  
model = classifier(**kwargs)  
  
# train model  
model.fit(X_train,y_train)  
  
# check accuracy and print out the results  
fit_accuracy = model.score(X_train, y_train)  
test_accuracy = model.score(X_test, y_test)  
  
print(f"Train accuracy: {fit_accuracy:0.2%}")  
print(f"Test accuracy: {test_accuracy:0.2%}")  
return model
```

Naïve bayes

```
from sklearn.naive_bayes import GaussianNB  
nb = train_model(X_train, Y_train, X_test, Y_test, GaussianNB)  
nb.fit(X_train, Y_train)  
y_pred_nb = nb.predict(X_test)  
print(y_pred_nb)  
score_nb = round(accuracy_score(y_pred_nb,Y_test)*100,2)  
print("The accuracy score achieved using Naive Bayes is: "+str(score_nb)+" %")
```



```
#Gaussian Naive Bayes
```

```
from sklearn.naive_bayes import GaussianNB
```

```
model = train_model(X_train, Y_train, X_test, Y_test, GaussianNB)
```

```
from sklearn.metrics import confusion_matrix
```

```
matrix= confusion_matrix(Y_test, y_pred_nb)
```

```
sns.heatmap(matrix,annot = True, fmt = "d")
```

```
from sklearn.metrics import precision_score
```

```
precision = precision_score(Y_test, y_pred_nb)
```

```
print("Precision: ",precision)
```

```
from sklearn.metrics import recall_score
```

```
recall = recall_score(Y_test, y_pred_nb)
```

```
print("Recall is: ",recall)
```

```
print((2*precision*recall)/(precision+recall))
```

```
fnr = FN*100/(FN+TP)
```

```
fnr
```

KNN

```
from sklearn.neighbors import KNeighborsClassifier
```

```
knn = train_model(X_train, Y_train, X_test, Y_test, KNeighborsClassifier,
```

```
n_neighbors=8)
```

```
knn.fit(X_train, Y_train)
```

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

```
print(y_pred_knn)
```

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

```

print("The accuracy score achieved using KNN is: "+str(score_knn)+" %")
# KNN
from sklearn.neighbors import KNeighborsClassifier
model = train_model(X_train, Y_train, X_test, Y_test, KNeighborsClassifier)
# Seek optimal 'n_neighbours' parameter
for i in range(1,10):
    print("n_neighbors = "+str(i))
    train_model(X_train, Y_train, X_test, Y_test, KNeighborsClassifier, n_neighbors=i)
from sklearn.metrics import confusion_matrix
matrix= confusion_matrix(Y_test, y_pred_knn)
sns.heatmap(matrix,annot = True, fmt = "d")
from sklearn.metrics import precision_score
precision = precision_score(Y_test, y_pred_knn)
print("Precision: ",precision)
from sklearn.metrics import recall_score
recall = recall_score(Y_test, y_pred_knn)
print("Recall is: ",recall)
print((2*precision*recall)/(precision+recall))
CM = pd.crosstab(Y_test, y_pred_knn)
CM
fnr = FN*100/(FN+TP)
fnr
CM = pd.crosstab(Y_test, Y_pred_knn4)
TN=CM.iloc[0,0]
FP=CM.iloc[0,1]
FN=CM.iloc[1,0]

```

```
TP=CM.iloc[1,1]
```

```
fnr = FN*100/(FN+TP)
```

```
fnr
```

Decision Tree

```
from sklearn.tree import DecisionTreeClassifier
```

```
dt = DecisionTreeClassifier(max_depth=3, random_state=0)
```

```
dt.fit(X_train, Y_train)
```

```
y_pred_dt = dt.predict(X_test)
```

```
print(y_pred_dt)
```

```
score_dt = round(accuracy_score(y_pred_dt, Y_test)*100,2)
```

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

```
from sklearn.tree import DecisionTreeClassifier
```

```
tree1 = DecisionTreeClassifier(random_state=0)
```

```
tree1.fit(X_train, Y_train)
```

```
print("Accuracy on training set: {:.3f}".format(tree1.score(X_train, Y_train)))
```

```
print("Accuracy on test set: {:.3f}".format(tree1.score(X_test, Y_test)))
```

```
tree1 = DecisionTreeClassifier(max_depth=3, random_state=0)
```

```
tree1.fit(X_train, Y_train)
```

```
print("Accuracy on training set: {:.3f}".format(tree1.score(X_train, Y_train)))
```

```
print("Accuracy on test set: {:.3f}".format(tree1.score(X_test, Y_test)))
```

```
df = pd.read_csv('heart.csv')
```

```
df.head()
```

```
from pandas import DataFrame, Series
```

```
from IPython.display import Image
```

```

from io import StringIO
import pydotplus
from sklearn import preprocessing
def plot_decision_tree(clf,feature_name,target_name):
    dot_data = StringIO()
    tree.export_graphviz(clf, out_file=dot_data,
                        feature_names=feature_name,
                        filled=True, rounded=True,
                        special_characters=True)
    graph = pydotplus.graph_from_dot_data(dot_data.getvalue())
    return Image(graph.create_png())
from sklearn import tree
clf = tree.DecisionTreeClassifier()
clf = clf.fit(X_train,Y_train)
plot_decision_tree(clf, X_train.columns,df.columns[1])

```

Final Score

```
# initialize an empty list
```

```
accuracy = []
```

```
# list of algorithms names
```

```
classifiers = ['KNN', 'Decision Trees', 'Logistic Regression', 'Naive Bayes', 'Random
Forests']
```

```
# list of algorithms with parameters
```

```
models = [KNeighborsClassifier(n_neighbors=8), DecisionTreeClassifier(max_depth=3,
random_state=0), LogisticRegression(),
          GaussianNB(), RandomForestClassifier(n_estimators=100, random_state=0)]
```

```

# loop through algorithms and append the score into the list
for i in models:
    model = i
    model.fit(X_train, Y_train)
    score = model.score(X_test, Y_test)
    accuracy.append(score)
# create a dataframe from accuracy results
summary = pd.DataFrame({'accuracy':accuracy}, index=classifiers)
summary
scores = [score_lr,score_nb,score_knn,score_dt,score_rf]
algorithms = ["Logistic Regression","Naive Bayes","K-Nearest Neighbors","Decision
Tree","Random Forest"]
sns.set(rc={'figure.figsize':(15,8)})
plt.xlabel("Algorithms")
plt.ylabel("Accuracy score")

sns.barplot(algorithms,scores)

```

4.RESULTS

The machine learning models is evaluated using the AUC-ROC metric. This can be used to understand the model performance. The ROC curve is the Receiver Operating Characteristic curve. The AUC is the area under the ROC curve. If the AUC score is high, the model performance is high and vice versa. The ROC curve of the machine learning algorithms.

The comparison of AUC score of the various algorithms is as follows:

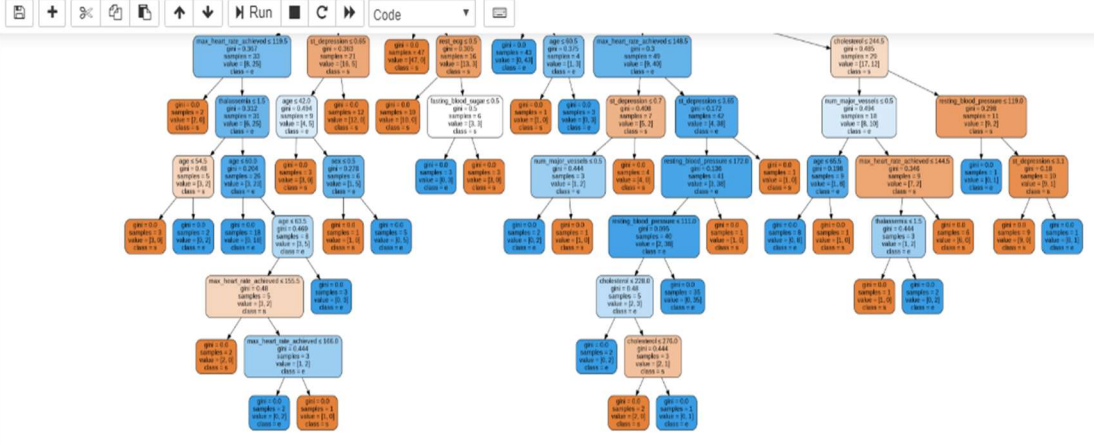
| Algorithm | AUC score |
|------------------|------------------|
| NB | 0.68 |
| KNN | 0.56 |
| Decision tree | 0.53 |

The accuracy of the algorithms is calculated. The accuracy results are tabulated as follows:

| Method | Accuracy |
|---------------|-----------------|
| KNN | 67.21% |
| NB | 85.25% |
| Decision tree | 81.97% |

The accuracy of NAÏVE BAYES algorithm is good when compared to other algorithms

Output



```
accuracy
In [56]: score_dt = round(accuracy_score(Y_pred_dt, Y_test)*100,2)
print("The accuracy score achieved using Decision Tree is: "+str(score_dt)+" %")
The accuracy score achieved using Decision Tree is: 81.97 %
```

Output for Decision tree



```
KNN(K Nearest Neighbors)
In [0]: from sklearn.neighbors import KNeighborsClassifier
knn = KNeighborsClassifier(n_neighbors=7)
knn.fit(X_train, Y_train)
Y_pred_knn=knn.predict(X_test)

In [39]: Y_pred_knn.shape
Out[39]: (61,)

In [40]: score_knn = round(accuracy_score(Y_pred_knn, Y_test)*100,2)
print("The accuracy score achieved using KNN is: "+str(score_knn)+" %")
The accuracy score achieved using KNN is: 67.21 %
```

Output for KNN

```
In [0]: from sklearn.naive_bayes import GaussianNB
        nb = GaussianNB()
        nb.fit(X_train,Y_train)
        Y_pred_nb = nb.predict(X_test)

In [26]: Y_pred_nb.shape
Out[26]: (61,)

In [27]: score_nb = round(accuracy_score(Y_pred_nb,Y_test)*100,2)
        print("The accuracy score achieved using Naive Bayes is: "+str(score_nb)+" %")
The accuracy score achieved using Naive Bayes is: 85.25 %
```

Output for Naïve bayes

5. Conclusion

5.1 Future Work

In future, this model can be used to compare various machine learning algorithm generated prediction models and the model which will give higher accuracy will be chosen as the prediction model. The heart disease prediction can be done using other machine learning algorithms. Logistic regression can also perform well in case of binary classification problems such as heart disease prediction. Random forests can perform well than decision trees. Also, the ensemble methods and artificial neural networks can be applied to the data set. The results can be compared and improvised.

5.2 Conclusion

This project discusses the various machine learning algorithms such as Naïve Bayes, decision tree and k- nearest neighbour which were applied to the data set. It utilizes the data such as cholesterol, diabetes, bp and then tries to predict the possible coronary heart disease patient in next 10 years.

Family history of heart disease can also be a reason for developing a heart disease as mentioned earlier. So, this data of the patient can also be included for further increasing the accuracy of the model. This work will be useful in identifying the possible patients who may suffer from heart disease in the next 10 years. This may help in taking preventive measures and hence try to avoid the possibility of heart disease for the patient. So when a patient is predicted as positive for heart disease, then the medical data for the patient can be closely analysed by the doctors. An example would be - suppose the patient has diabetes which may be the cause for heart disease in future and then the patient can be given treatment to have diabetes in control which in turn may prevent the heart disease.

6. REFERENCES

- [1] Monika Gandhi, Shailendra Narayanan Singh Predictions in heart disease using techniques of data mining (2015)
- [2] J Thomas, R Theresa Princy Human heart disease prediction system using data mining techniques (2016)
- [3] Sana Bharti, Shailendra Narayan Singh, Amity university, Noida, India Analytical study of heart disease prediction comparing with different algorithms (May 2015)
- [4] Purushottam, Kanak Saxena, Richa Sharma Efficient heart disease prediction system using Decision tree (2015)
- [5] Sellappan Palaniyappan, Rafiah Awang Intelligent heart disease prediction using data mining techniques (August 2008)
- [6] Himanshu Sharma, M A Rizvi Prediction of Heart Disease using Machine Learning Algorithms: A Survey (August 2017)
- [7] Animesh Hazra, Subrata Kumar Mandal, Amit Gupta, Arkomita Mukherjee and Asmita Mukherjee Heart Disease Diagnosis and Prediction Using Machine Learning and Data Mining Techniques: A Review (2017)
- [8] V.Krishnaiah, G.Narsimha, N.Subhash Chandra Heart Disease Prediction System using Data Mining Techniques and Intelligent Fuzzy Approach: A Review (February 2016)
- [9] Ramandeep Kaur, 2Er. Prabhsharn Kaur A Review - Heart Disease Forecasting Pattern using Various Data Mining Techniques (June 2016)
- [10] J.Vijayashree and N.Ch. SrimanNarayanaIyengar Heart Disease Prediction System Using Data Mining and Hybrid Intelligent Techniques: A Review (2016)
- [11] Benjamin EJ et.al Heart Disease and Stroke Statistics 2018 At-a-Glance (2018)
- [12] Abhay Kishore, Ajay Kumar, Karan Singh, Maninder Punia, Yogita Hambir Department of Computer Engineering, Army Institute of Technology, Pune, Maharashtra Professor, Department of Computer Engineering, Army Institute of Technology, Pune, Maharashtra Heart Attack Prediction Using Deep Learning (2018)
- [13] M.Nikhil Kumar, K.V.S Koushik, K.Deepak Department of CSE, VR Siddhartha Engineering College, Vijayawada, Andhra Pradesh, India Prediction Heart Diseases using Data mining and machine learning algorithms and tools.(2018)
- [14] Dataset Source, <https://www.kaggle.com>