**A**

**Seminar Report**

On

**“Heart Disease Prediction System”**

Submitted

In partial fulfilment

For the award of the Degree of

**BACHELOR OF TECHNOLOGY**

In

** COMPUTER ENGINEERING**

|  |  |
| --- | --- |
| **Submitted to:** | **Submitted By:** |
| Ms. Dhara Upadhyay | Aman Panchal  CS(7th Sem) |
| Assistant Professor | (17EAICS019) |
|  |  |

**Department of Computer Science &Engineering**

**Arya Institute of Engineering & Technology, Jaipur**

**Rajasthan Technical University, Kota [2020]**

***Candidate’s Declaration***

I hereby declare that the work, which is being presented in the Seminar Reportin partial fulfilment for the award of Degree of “Bachelor of Technology” in Computer Engineering, and submitted to the **Department of Computer Science & Engineering,** ARYA Institute of Engineering & Technology, Affiliated to Rajasthan Technical University is a record of my own work carried out under the Guidance of **Ms.** Dhara Upadhyay, Assistant Professor, Department of Computer Science & Engineering.

**Student Name**

Aman Panchal (17EAICS019)

***Acknowledgement***

We wish to express our deep sense of gratitude to our Project Guide & Coordinator **Ms.** Dhara Upadhyayfor guiding me from the inception till the completion of the project. We sincerely acknowledge her for giving her valuable guidance, support for literature survey, critical reviews and comments for our Project.

We would like to first of all express our thanks to **Dr. Arvind Agarwal**, Chairman of Arya Main Campus, for providing us such a great infrastructure and environment for our overall development.

We express sincere thanks tothe Principal of AIET, for his kind cooperation and extendible support towards the completion of our project. Words are inadequate in offering our thanks to **Er. Manish Mukhija**, Head of CSE Department, for consistent encouragement and support for shaping our project in the presentable form.

We also like to express our thanks to all supporting CSE faculty members who have been a constant source of encouragement for successful completion of the project.

Also, our warm thanks to **Arya Institute of Engineering & Technology**, who provided us this opportunity to carryout, this prestigious Project and enhance our learning in various technical fields.

**Student Name**

Aman Panchal(17EAICS019)

**ABSTRACT**

In this project, we were experiment with a real world dataset, and to explore how machine learning algorithms can be used to predict health disease inpatient. We were expected to gain experience using a common data-mining and machine learning library, and were expected to submit a report about the dataset and the algorithms used. After performing the required tasks on a dataset of my choice, herein lies in this report.

**Keyword :** Machine Learning, Numpy, Pandas, Matplotlib, Logistic Regression, K-nearest neighbour, Support Vector Machine, Decision Tree Classifier.

**Table of contents**

|  |  |  |
| --- | --- | --- |
|  | Candidate’s Declaration | ii |
|  | Acknowledgement | iii |
|  | Abstract | iv |
|  | Table of contents | v |
|  | List of Figures | vi |
|  |  |  |
| 1. | Introduction | 1 |
| 2. | The Project | 1 |
|  | 2.1 Data | 1 |
|  | 2.2 Machine Learning | 4 |
| 3. | Result and Analysis | 15 |
| 4. | Conclusion and Future Works | 16 |

**List of Figures**

|  |  |  |
| --- | --- | --- |
| 1. | Correlations between the attributes using heat map.................... | 4 |
| 2. | Sampling over the dataset............................................................ | 5 |
| 3. | Logistic Regression example represented graphically................ | 6 |
| 4. | Understanding K.......................................................................... | 8 |
| 5. | Euclidean Distance……………………………………………... | 9 |
| 6. | Approach of kNN Algorithm…………………………………... | 10 |
| 7. | Example of Decision Tree……………………………………… | 11 |
| 8. | Classifying Grasshopper using Decision Tree………………… | 13 |
| 9. | Random Forest............................................................................. | 14 |
| 10. | Example of How does Random Forest algorithm work……… | 15 |

**1. Introduction:**

This project has been done as part of my course for the B.tech Computer Science in Arya Institute of Engineering and Technology. Supervised by Ms. Dhara Upadhyay. Being interested in everything having a relation with the Machine Learning,the independant project was a great occasion to give me the time to learn and confirm my interest for this field. We can use Machine Learning in Finance, Medicine, almost everywhere. That’swhy I decided to conduct my project around the Machine Learning.

This model will predict that patient has heart disease or not on the basis of age, sex, chest pain type, cholestoral, maximum heart rate , etc.

**2. The Project:**

**2.1 Data**

The crucial element in machine learning task for which a particular attention should beclearly taken is the data. Indeed the results will be highly influenced by the data based onwhere did we find them, how are they formatted, are they consistent, is there any outlier andso on. At this step, many questions should be answered in order to guarantee that thelearning algorithm will be efficient and accurate.

Many sub steps are taken to get, clean and transform the data. I am going to explain each one of them to show how they have been applied on my project why they are useful forthe machine learning part.

**2.1.1 Getting the data:**

The first problem was where I can get the data to build a large enough dataset since I want to be able to predict the price for a given apartment according to the heart disease features.

And we easily get this from kaggle.

**●Cleaning:** It is the first module called to clean the item and verify that all the information in it correspond to the pattern used to extract it The cleaning module removes the noise, and check that all the values are not empty, otherwise the item is dropped. This is done for simplicity,

indeed, it could be better to try to inference them later. After the cleaning part done, the item is sent to the formatting module.

**●Formatting:** the second module is used to format the item’s values as we want. A basic example could be for the price, initially got being string type, is converted as float. This is done for every numeric values. The item formatted is then sent to the last module called Integrating.

**●Integrating:** this module, the last one, is basically the one in charges of saving the items in the format that we want. It also checks that there is no redundancies between the tuples . In my case, I decided to save them in an excel sheet for each website

**The six features ​extracted for each estate were the following**

1. age in years
2. sex
3. cp-chest pain type
4. trestbps- resting blood pressure
5. chol-serum cholesterol in mg/dl
6. fbs-(fasting blood sugar &gt; 120 mg/dl) (1 = true; 0 = false)
7. restecg-resting electrocardiographic results
8. thalach-maximum heart rate achieved
9. exang-exercise induced angina (1 = yes; 0 = no)
10. oldpeak-ST depression induced by exercise relative to rest

**2.1.2 Attribute Types**

**# Column Non-Null Count Dtype**

**--- ------ -------------- -----**

0 age 303 non-null int64

1 sex 303 non-null int64

2 cp 303 non-null int64

3 trestbps 303 non-null int64

4 chol 303 non-null int64

5 fbs 303 non-null int64

6 restecg 303 non-null int64

7 thalach 303 non-null int64

8 exang 303 non-null int64

9 oldpeak 303 non-null float64

10 slope 303 non-null int64

11 ca 303 non-null int64

12 thal 303 non-null int64

13 target 303 non-null int64

**2.1.3 Basic Statistical Measures**

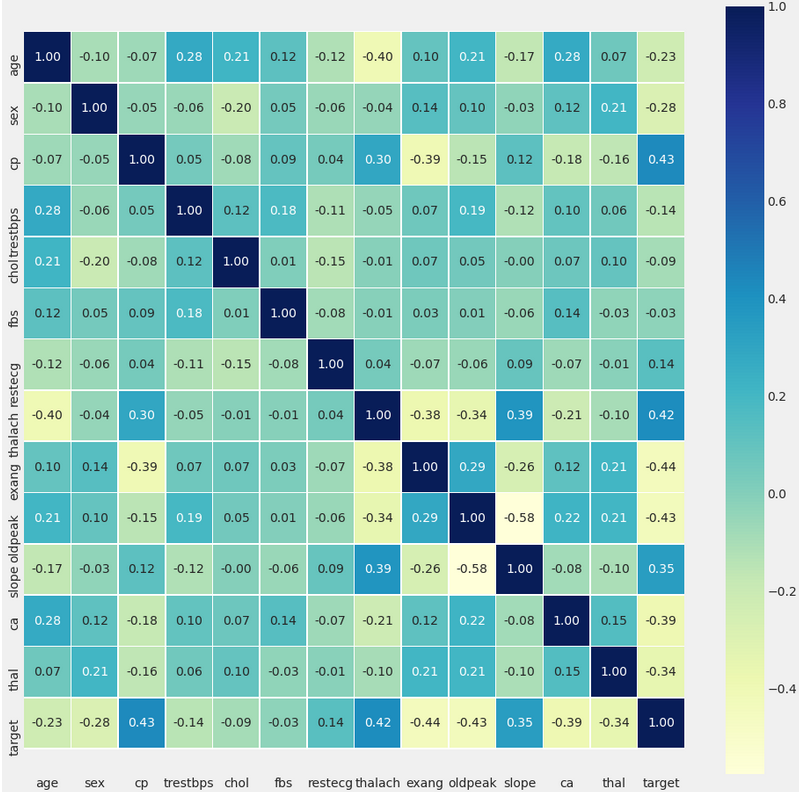
|  | **age** | **sex** | **cp** | **trestbps** | | **chol** | **fbs** | | **restecg** | **thalach** | **exang** | **oldpeak** | | **slope** | **ca** | **thal** | **target** | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **count** | 303 | 303 | 303 | | 303 | 303 | | 303 | 303 | 303 | 303 | | 303 | 303 | 303 | 303 | | 303 |
| **mean** | 54.37 | 0.68 | 0.97 | | 131.62 | 246.26 | | 0.15 | 0.53 | 149.65 | 0.33 | | 1.04 | 1.40 | 0.73 | 2.31 | | 0.54 |
| **Std** | 9.08 | 0.47 | 1.03 | | 17.54 | 51.83 | | 0.36 | 0.53 | 22.91 | 0.47 | | 1.16 | 0.62 | 1.02 | 0.61 | | 0.50 |
| **Min** | 29.00 | 0.00 | 0.00 | | 94.00 | 126.00 | | 0.00 | 0.00 | 71.00 | 0.00 | | 0.00 | 0.00 | 0.00 | 0.00 | | 0.00 |
| **25%** | 47.50 | 0.00 | 0.00 | | 120.00 | 211.00 | | 0.00 | 0.00 | 133.50 | 0.00 | | 0.00 | 1.00 | 0.00 | 2.00 | | 0.00 |
| **50%** | 55.00 | 1.00 | 1.00 | | 130.00 | 240.00 | | 0.00 | 1.00 | 153.00 | 0.00 | | 0.80 | 1.00 | 0.00 | 2.00 | | 1.00 |
| **75%** | 61.00 | 1.00 | 2.00 | | 140.00 | 274.50 | | 0.00 | 1.00 | 166.00 | 1.00 | | 1.60 | 2.00 | 1.00 | 3.00 | | 1.00 |
| **Max** | 77.00 | 1.00 | 3.00 | | 200.00 | 564.00 | | 1.00 | 2.00 | 202.00 | 1.00 | | 6.20 | 2.00 | 4.00 | 3.00 | | 1.00 |

With the type of my attributes in mind, I used basic statistical measures on

my datasuch as the mean, the mode, the standard deviation, etc.

**2.1.4 Correlation Analysis And Redundancies**

After seeing the distribution of my variables, I computed the correlation between thedifferent attributes and how each one is correlated to the price. Indeed, two attributes highlycorrelated (not using price) could be useless because they will not have a great impact on theregression result and should be reduced to one (cf: Principal Component Analysis). On theother hand, an attribute low correlated with the dependent variable (Price) could not be reallyinfluent on the result.



**Figure 1** - Correlations between the different attributes using heat map.

**2.2 Machine Learning:**

The Machine Learning part is about trying to find the best learning algorithm for agiven problem even if it is highly conditioned by how well the data has been processed andtune some parameters to improve it. Depending on the problem, if it is supervised (meaningwe build a model from labeled training set, the value of the dependent variable is known) or ifit unsupervised (the model is built on unstructured and unlabeled data), if it is a regression orclassification problem, many learning algorithm exist each with their benefits and drawbacks.

**2.2.1 Sampling**

Given my dataset, I applied a sampling technique in order to divide it into differentsubset having each its own utility. It is commonly assumed that more we have data to build amodel more it will have tend to give good results. Usually the dataset is divided as follow withtheir respective utility :



Figure 2: Sampling over the dataset.

**2.2.2 Learning Algorithm**

**Logistic Regression**

**“Logistic Regression is based on this principle: it expresses the multiple logistic regression equation in logarithmic terms(called the logit) and thus overcomes the problem of violating the assumption of Linearity.”**

In order to understand the difference between logistic and linear regression, we need to first understand the difference between a continuous and a categoric variable.

Continuous variables are numeric values. They have an infinite number of values between any two given values. Examples include the length of a video or the time a payment is received or the population of a city.

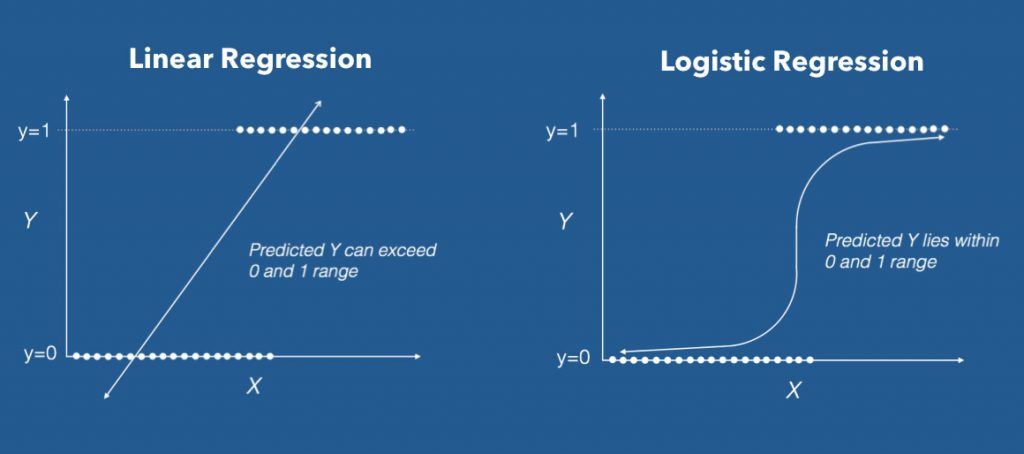
Categorical variables, on the other hand, have distinct groups or categories. They may or may not have a logical order. Examples include gender, payment method, age bracket and so on.

In linear regression, the dependent variable Y is always a continuous variable. If the variable Y is a categorical variable, then linear regression cannot be applied.

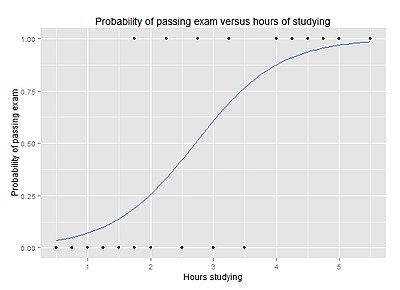
In case Y is a categorical variable that has only 2 classes, logistic regression can be used to overcome this problem. Such problems are also known as binary classification problems.

It’s also important to understand that standard logistic regression can only be used for binary classification problems. If Y has more than 2 classes, it becomes a multi-class classification and standard logistic regression cannot be applied.

One of the biggest advantages of logistic regression analysis is that it can compute a prediction probability score for an event. This makes it an invaluable predictive modeling technique for data analytics.

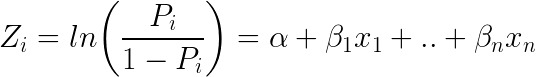


**Figure 3: Logistic Regression example represented graphically**



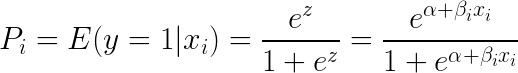
**How does Logistic Regression Work?**

Here’s what the logistic equation looks like:



Logistic Regression Equation

Taking e (exponent) on both sides of the equation results in:



Equation with E Exponent

Here’s how the equation can be implemented in R:

# Template code

# Step 1: Build Logit Model on Training Dataset

logitMod <- glm(Y ~ X1 + X2, family=“binomial”, data = trainingData)

# Step 2: Predict Y on Test Dataset

predictedY <- predict(logitMod, testData, type=“response”)

# K-nearest neighbours

K Nearest neighbor falls in the category of the supervised machine learning algorithm. Like [Logistic Regression](https://automateintellect.com/logistic-regression/) and [Support Vector machines](https://automateintellect.com/support-vector-machines/), it is usually used for classification problems. Although it can be used for both regression and classification tasks.

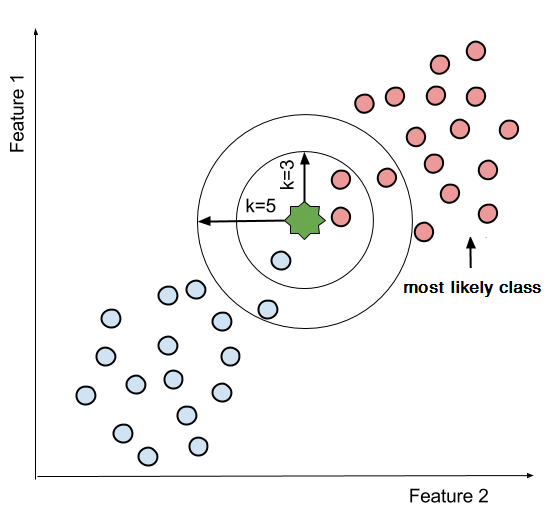


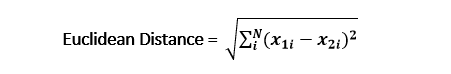
Figure 4: Understanding K

**How K Nearest Neighbor Works?**

As the name mentions ‘neighbor’ term plays an important role in the working of K nearest neighbor. This algorithm calculates the distance between the new data point and the neighbors. The neighbors with the shortest distances are selected. The new data variable is then assigned to the class with the most number of close neighbors.

In other words, when a new data point enters for prediction. K nearest model goes through the memorized or stored data. It looks for instances that are close to the new data point and assigns to the class with the most closely distanced neighbor instances.

Different techniques are employed to calculate the distance between new and trained instances. The basic technique used is [the Euclidean distance](http://rosalind.info/glossary/euclidean-distance/). Calculated as the square root of the sum of the squared distances between two points. And given by:



Other methods may include Manhattan or Hamming distance.

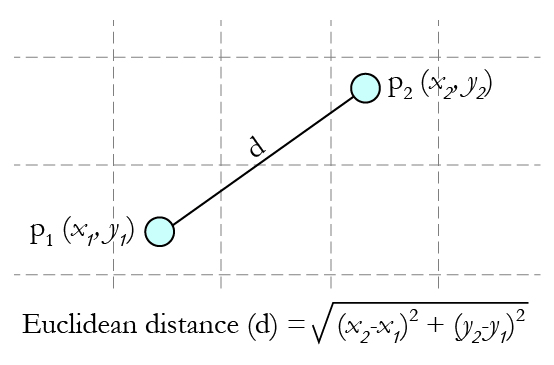
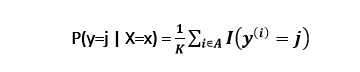


Figure 5: Euclidean Distance

Steps for K Nearest Neighbor Algorithm Working:

1. Select the number K to start working with.
2. Calculate the distance between the new data point and the K neighbors.
3. Select the neighbor data points that are closest, minimum distance.
4. Count the number of nearest neighbors in each class.
5. Or calculate the conditional probability for the assignment of the class.
6. Assign the new data point to the class with the maximum number of nearest neighbors. Or the new data point goes to the class with the largest probability.



For regression, the method remains the same but instead of classes, we go for a continuous target value by considering methods such as mean, median, etc.

**What is K?**

Up till now we have been talking about using K but have not explicitly defined this parameter. K is a hypermeter. It is defined by the data engineer. You working as a data engineer will define K depending upon the type of model you wish to train.

The smaller values of K define a rigid model. K = 2 means two neighboring points to consider. It will reduce the bias but the variance will be high. The model will not perform that well for new data as we have limited the points to consider.

The larger values of K will entertain more outliers and you will get a more flexible model. It means more neighboring points are participating leading to a more generalized prediction of new data points. Larger K parameter implies lower variance but higher bias.

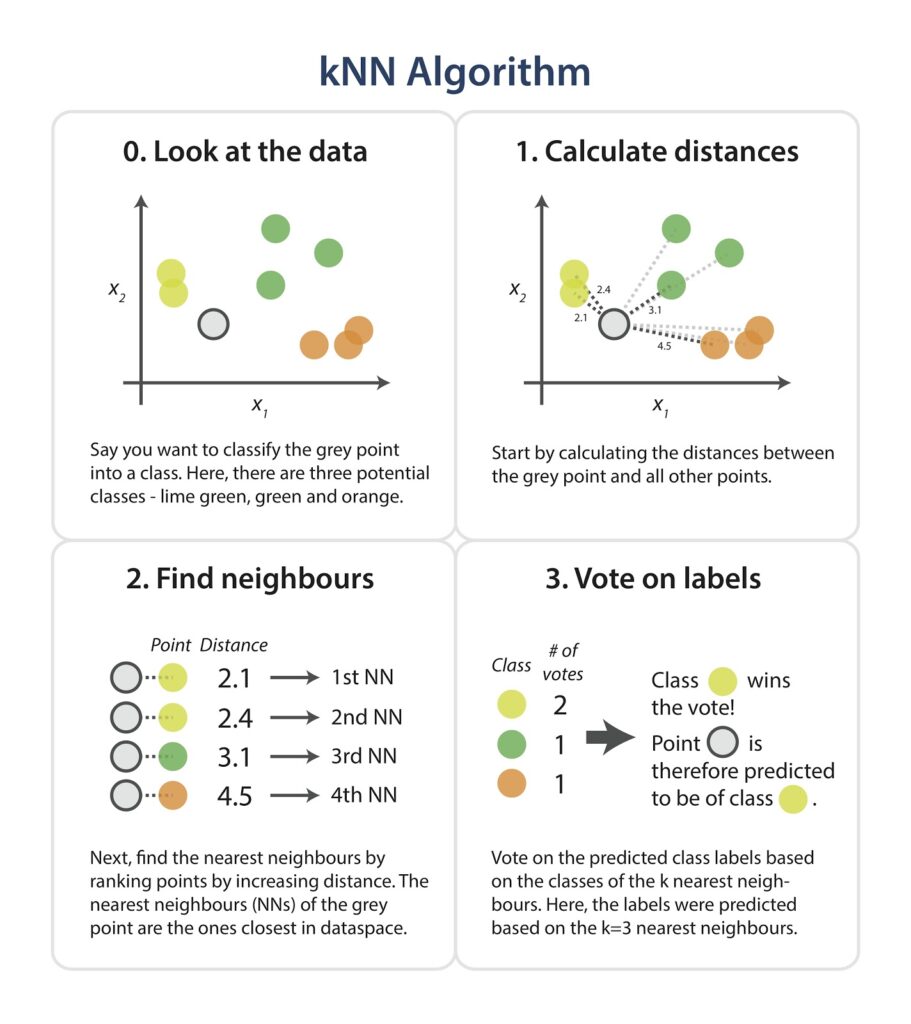


Figure 6: Approach of kNN Algorithm

**Decision Tree Classification**

A Decision Tree is a simple representation for classifying examples. It is a Supervised Machine Learning where the data is continuously split according to a certain parameter.

**Decision Tree consists of :**

1. **Nodes** : Test for the value of a certain attribute.
2. **Edges/ Branch** : Correspond to the outcome of a test and connect to the next node or leaf.
3. **Leaf nodes** : Terminal nodes that predict the outcome (represent class labels or class distribution).

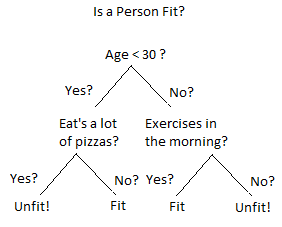
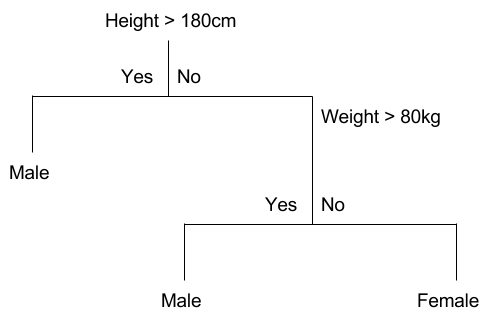


Figure 7: Example of Decision Tree

**1. Classification trees (Yes/No types) :**

What we’ve seen above is an example of classification tree, where the outcome was a variable like ‘fit’ or ‘unfit’. Here the decision variable is **Categorical/ discrete**.

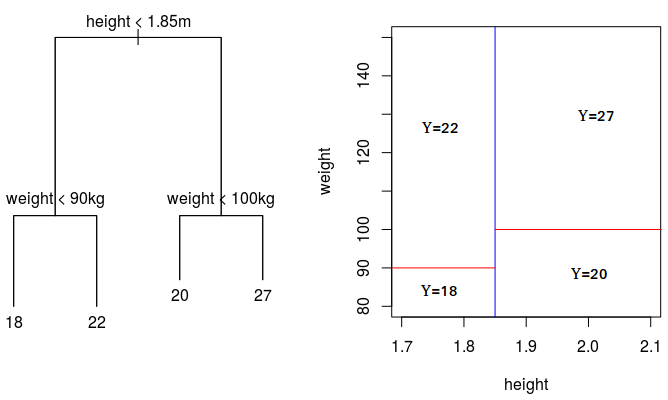
Such a tree is built through a process known as **binary recursive partitioning**. This is an iterative process of **splitting the data into partitions**, and then splitting it up further on each of the branches.



Example of a Classification Tree

**2. Regression trees (Continuous data types) :**

Decision trees where the target variable can take **continuous values** (typically real numbers) are called **regression trees**. (e.g. the price of a house, or a patient’s length of stay in a hospital)



Example of Regression Tree

**Basic Divide-and-Conquer Algorithm :**

1. Select a test for root node. Create branch for each possible outcome of the test.
2. Split instances into subsets. One for each branch extending from the node.
3. Repeat recursively for each branch, using only instances that reach the branch.
4. Stop recursion for a branch if all its instances have the same class.

**Decision Tree Classifier**

* Using the decision algorithm, we start at the tree root and split the data on the feature that results in the **largest information gain (IG)** (reduction in uncertainty towards the final decision).
* In an iterative process, we can then repeat this splitting procedure at each child node **until the leaves are pure**. This means that the samples at each leaf node all belong to the same class.
* In practice, we may set a **limit on the depth of the tree to prevent overfitting**. We compromise on purity here somewhat as the final leaves may still have some impurity.

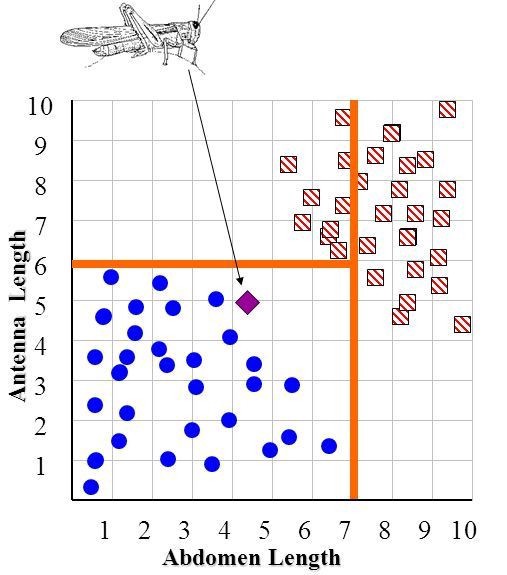
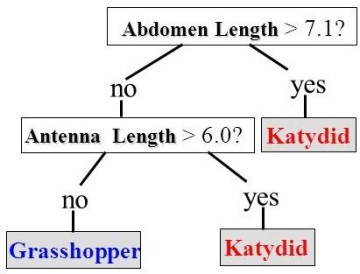


Figure 8: Classifying Grasshopper using Decision Tree

Classifying whether an insect is a Grasshopper or a Katydid based on Antenna Length and Abdomen Length.

# Random Forest Algorithm

Random Forest is a popular machine learning algorithm that belongs to the supervised learning technique. It can be used for both Classification and Regression problems in ML. It is based on the concept of **ensemble learning,**

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

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



Figure 9: Random Forest

**Why use Random Forest?**

Below are some points that explain why we should use the Random Forest algorithm:

* It takes less training time as compared to other algorithms.
* It predicts output with high accuracy, even for the large dataset it runs efficiently.
* It can also maintain accuracy when a large proportion of data is missing.

**How does Random Forest algorithm work?**

Random Forest works in two-phase first is to create the random forest by combining N decision tree, and second is to make predictions for each tree created in the first phase.

The Working process can be explained in the below steps and diagram:

**Step-1:** Select random K data points from the training set.

**Step-2:** Build the decision trees associated with the selected data points (Subsets).

**Step-3:** Choose the number N for decision trees that you want to build.

**Step-4:** Repeat Step 1 & 2.

**Step-5:** For new data points, find the predictions of each decision tree, and assign the new data points to the category that wins the majority votes.

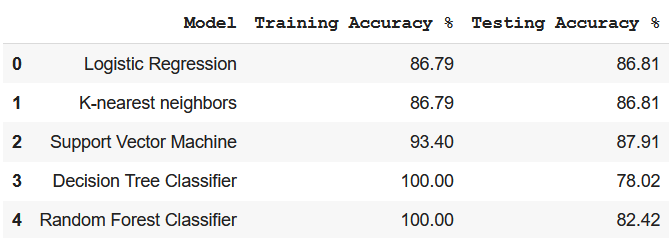
The working of the algorithm can be better understood by the below example:



Figure 10: Example of How does Random Forest algorithm work

**3. Results:**

The results we got after using various machine learning algorithm.



But the model is not working well in the test data so we further perform hyper parameter tuning and the result we get are :

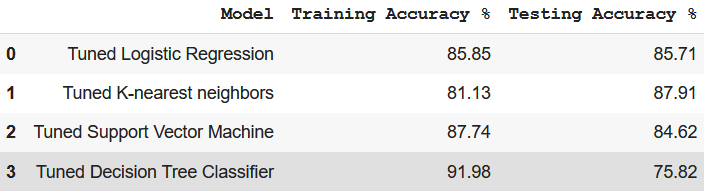


Figure 11: Results

# And the we select the best model which is K-nearest neighbours it give accuracy of 81.13 on training data and accuracy is increase to 87.91 on test data.

**4. Conclusion and Future Work:**

So at last we choose model which was built by using K-nearest neighbours after hyper parameter tuning due to its accuracy and giving best result on test data for prediction of health disease in patient.

In future we will try to build a website and fit this model into and also connect with the database having contact no’s of doctors and suggest those which are near the location of patient.