



BREAST CANCER DIAGNOSIS USING DECISION TREE , K-NN AND RANDOM FOREST

SUBMITTED BY -

04101032016 - MALVIKA BHALLA

04501032016 - APOORVA DHAWAN

07801032016 - DEEKSHA ARORA

INTRODUCTION

Machine learning is widely used in bioinformatics and particularly in breast cancer diagnosis. The number and size of medical databases are increasing rapidly but most of these data are not analyzed for finding the valuable and hidden knowledge. Advanced data mining techniques can be used to discover hidden patterns and relationships. Models developed from these techniques are useful for medical practitioners to make right decisions.

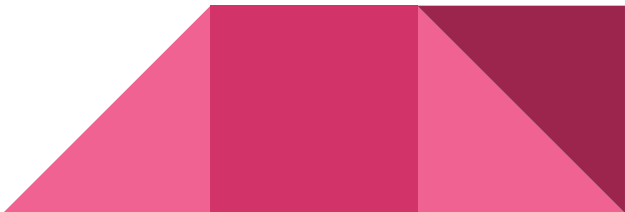


AIM

This is a project on Disease (Breast Cancer) Prediction, in which we use three different algorithms for classifying between the Malignant and Benign cases.

Our Goal of this project will be to analyze breast cancer data using three classification techniques to predict the occurrence of the cancer and then compare the results.


In this project, we have used certain classification methods like :

1. Decision Trees
 2. K-nearest neighbors (K-NN)
 3. Random Forests
- 

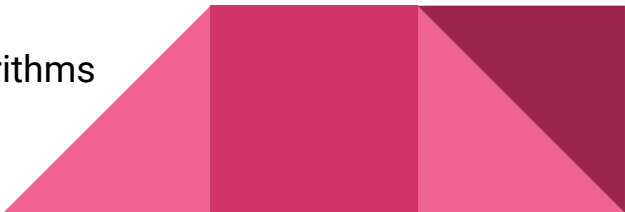
BREAST CANCER DISEASE

Breast cancer has become a common disease among women around the world and considered as the second largest prevalent type of cancer which cause deaths among women.

A group of rapidly dividing cells may form a lump or mass of extra tissue which are known as tumors. Tumors can be categorized either as cancerous (malignant) or non-cancerous (benign). Based on the World Health Organization statistics, there are more than 1.2 billion women around the world which are diagnosed with breast cancer. However, in recent years, this trend has been reduced due to the effective diagnostic techniques which can cure the cancer if it is diagnosed in an appropriate time.



METHODOLOGY

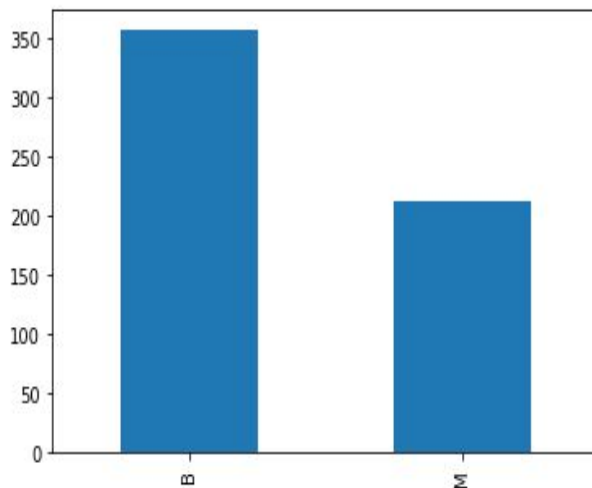
- Select the Breast cancer dataset from UCI repository containing 32 columns and 569 rows.
 - Perform data processing:
 - Import the libraries
 - Import the data-set
 - Check out the missing values
 - Splitting the data-set into Training and Test Set
 - Perform feature Extraction by applying Principal Component Analysis
 - Perform Classification using three techniques:
 - Decision Tree
 - K-nn
 - Random Forest
 - Create a confusion matrix to check the accuracy in respective algorithms and compare the results.
- 

DATASET

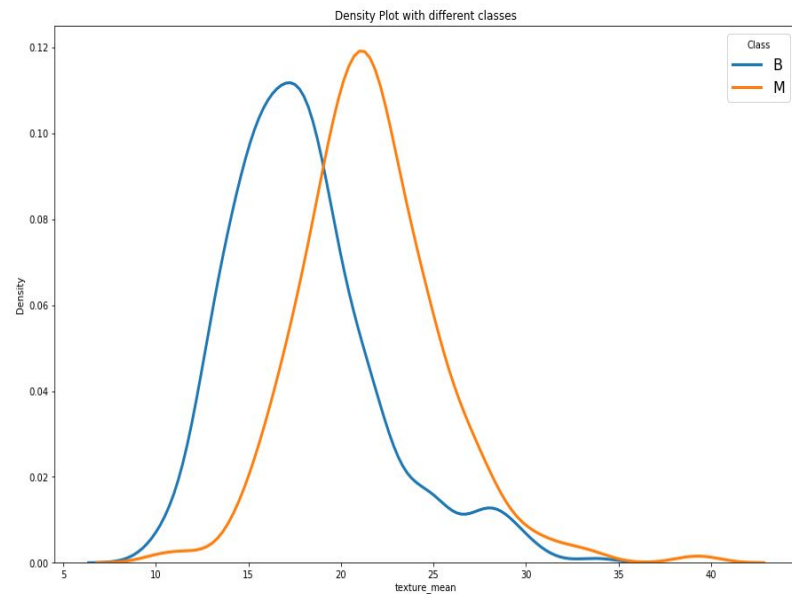
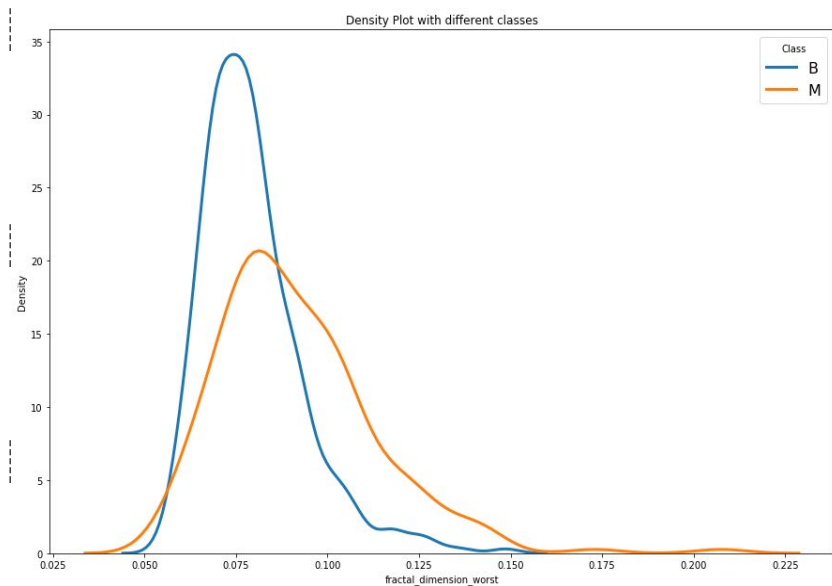
The dataset is taken from the UCI Machine Learning Repository named as **Breast Cancer Wisconsin (Diagnostic) Data Set**. The dataset characteristics is multivariate and had 569 instances and 32 attributes. There was no missing data in the dataset.

B- Benign

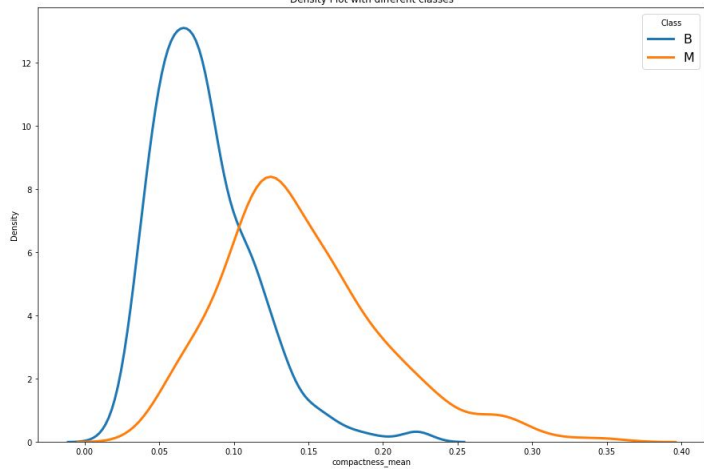
M - Malignant



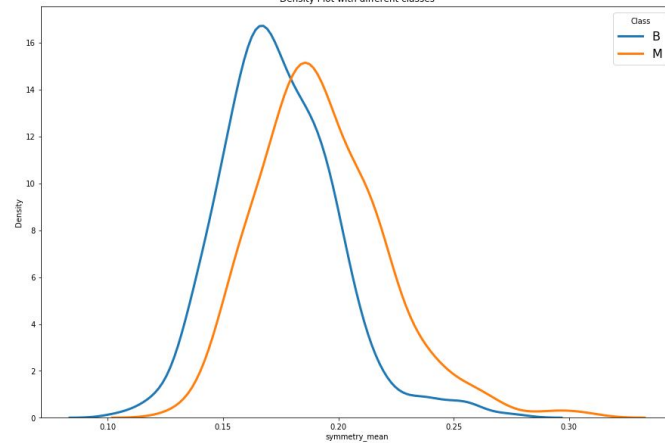
Density Plots showing variation of classes with different attributes



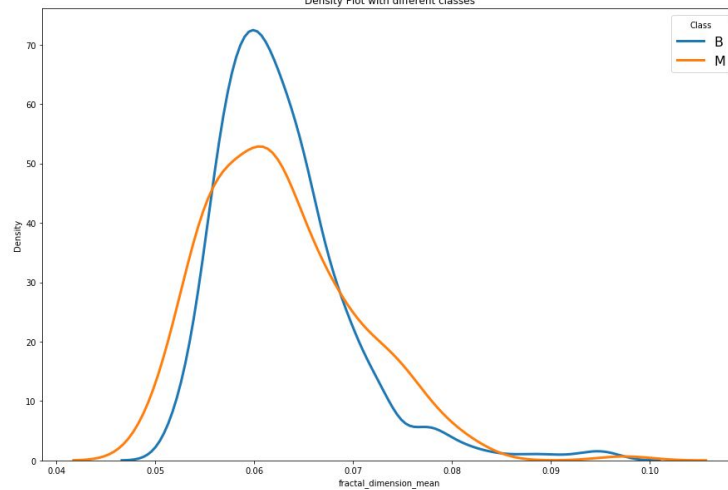
Density Plot with different classes



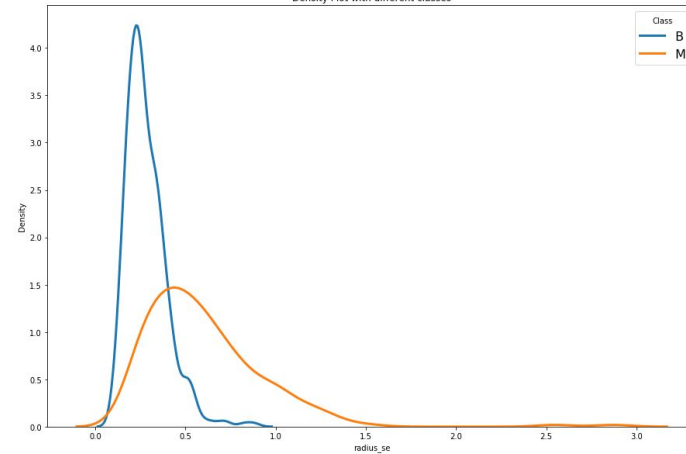
Density Plot with different classes



Density Plot with different classes

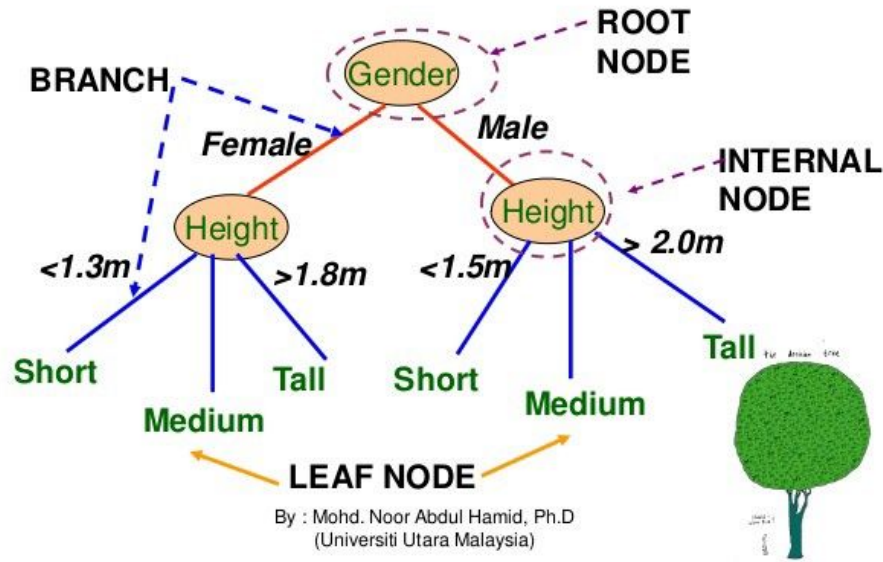


Density Plot with different classes



DECISION TREE

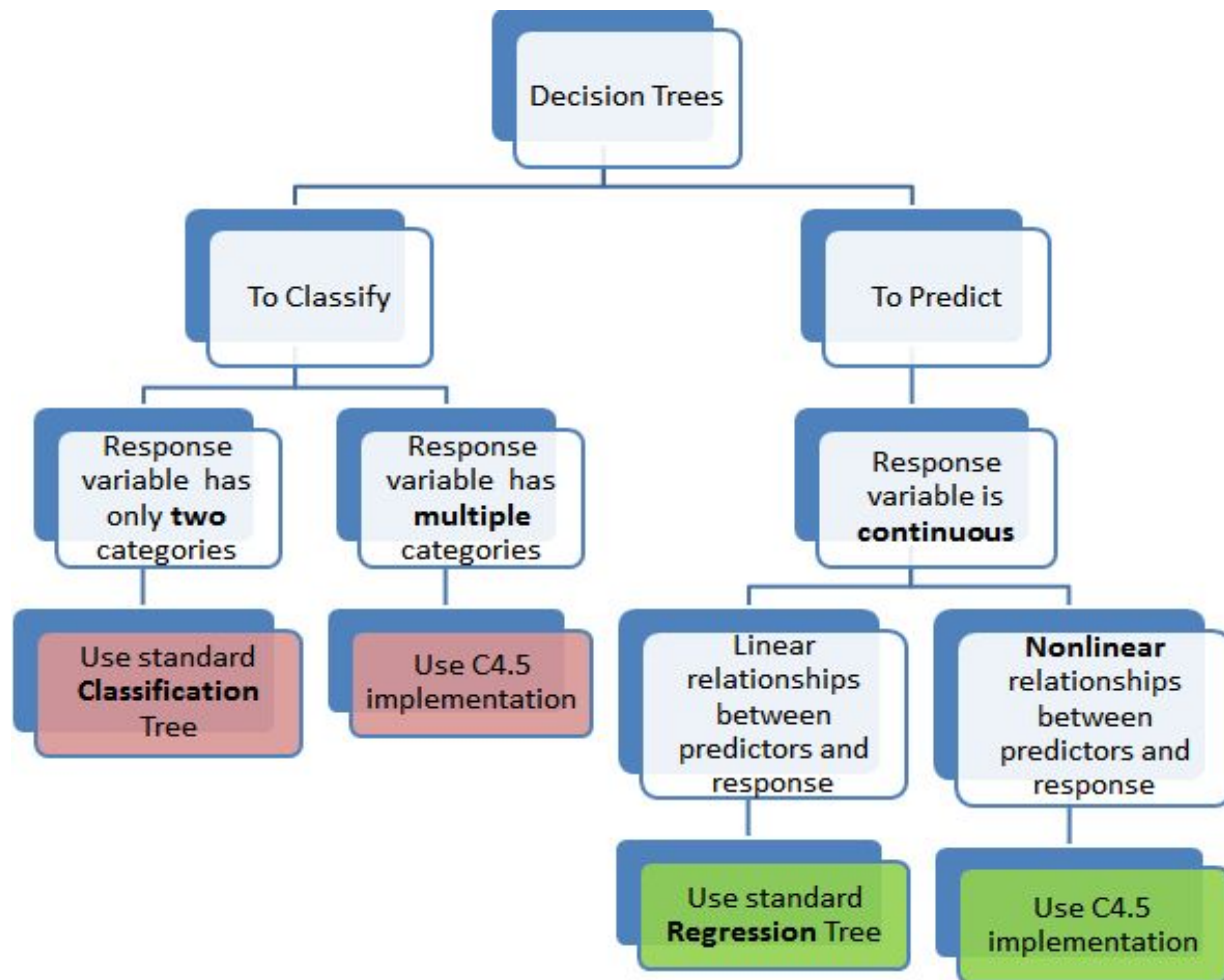
Decision Tree Diagram



A decision tree is a flowchart-like structure in which each internal node represents a “test” on an attribute, each branch represents the outcome of the test, and each leaf node represents a class label. The paths from root to leaf represent classification rules.


Tree based learning algorithms are considered to be one of the best and mostly used **supervised** learning methods. Tree based methods empower predictive models with high accuracy, stability and ease of interpretation. Unlike linear models, they map nonlinear relationships quite well. They are adaptable at solving any kind of problem at hand (classification or regression). Decision Tree algorithms are referred to as **CART (Classification and Regression Trees)**.





DECISION TREE ALGORITHM

The decision tree algorithm tries to solve the problem, by using tree representation. Each internal node of the tree corresponds to an attribute, and each leaf node corresponds to a class label.

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.
- 

ADVANTAGES

- Easy to Understand
- Useful in Data exploration
- Less data cleaning required
- Data type is not a constraint
- Easy to generate rules

DISADVANTAGES

- Over fitting
 - Not fit for continuous variables
 - Calculations become complex when there are many classes.
 - Decision tree learners create *biased* trees if some classes dominate.
- 

APPLYING DECISION TREE ON OUR PRE-PROCESSED DATA

While applying decision tree on the pre processed data we used Gini Index algorithm for the split.

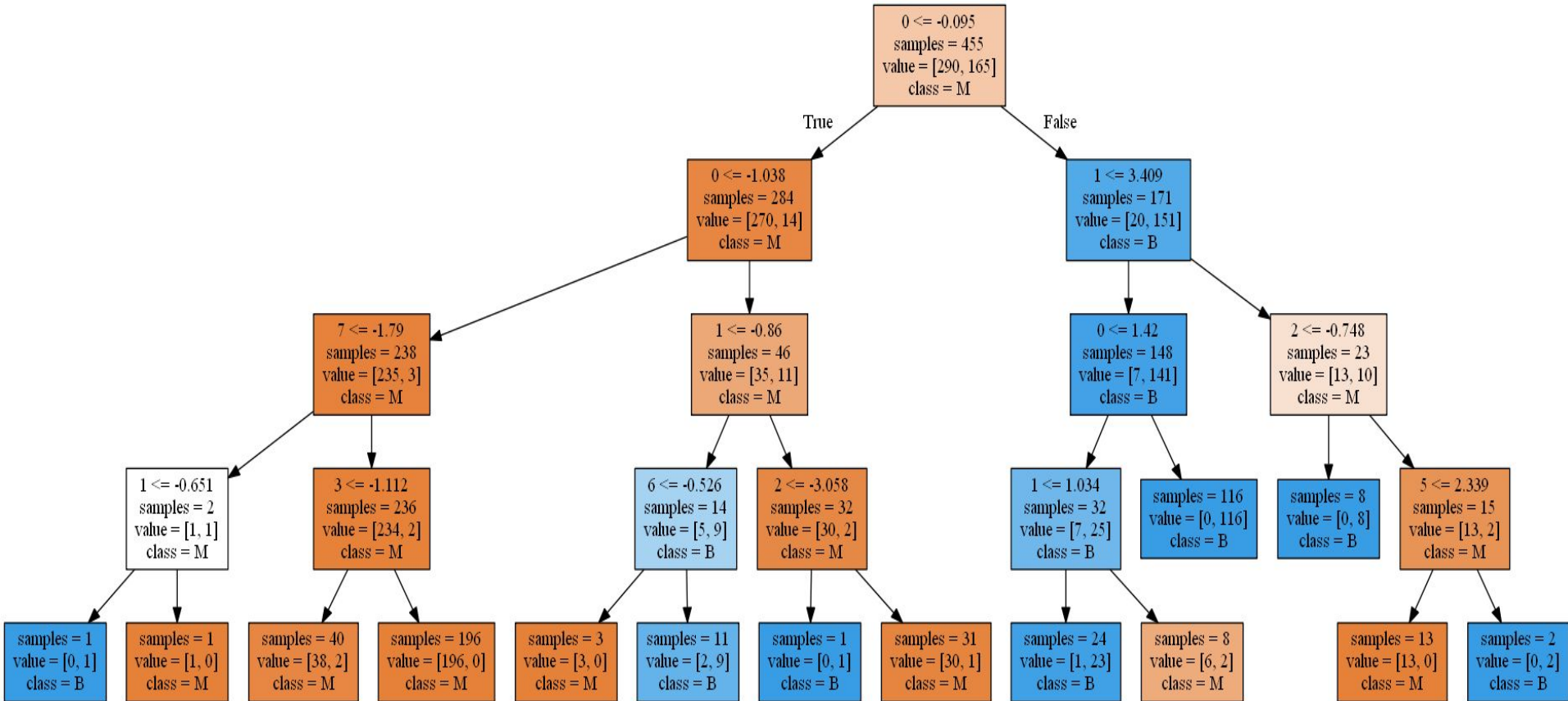
Gini Index :

Sklearn supports “Gini” criteria for Gini Index and by default.

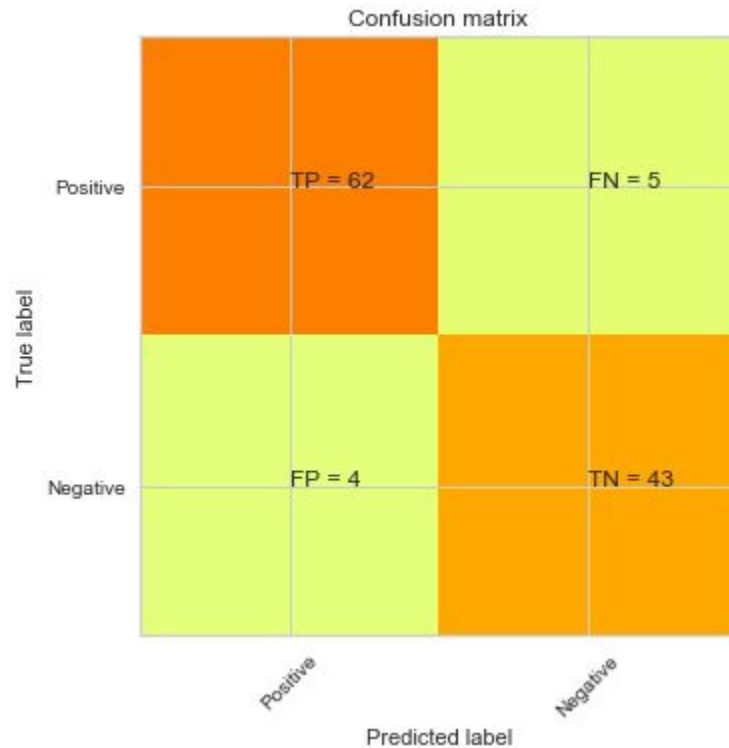
$$GiniIndex = 1 - \sum_j p_j^2$$

- It works with categorical target variable “Success” or “Failure”.
- It performs only Binary splits.
- Higher the value of Gini higher the homogeneity.
- CART (Classification and Regression Tree) uses Gini method to create binary splits.

DECISION TREE OUTPUT



CONFUSION MATRIX



CLASSIFICATION REPORT AFTER APPLYING DECISION TREE ALGORITHM

	precision	recall	f1-score	support
0	0.94	0.93	0.93	67
1	0.90	0.91	0.91	47
micro avg	0.92	0.92	0.92	114
macro avg	0.92	0.92	0.92	114
weighted avg	0.92	0.92	0.92	114

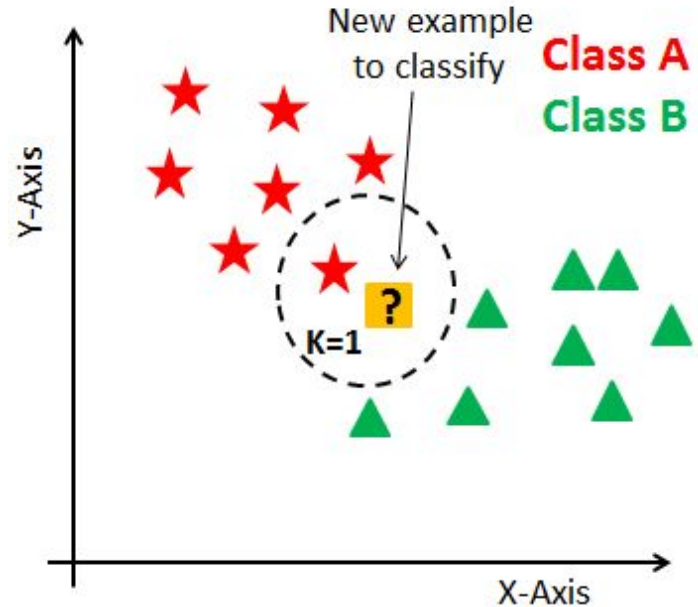
ACCURACY AFTER APPLYING DECISION TREE ALGORITHM

ACCURACY : 92.105%

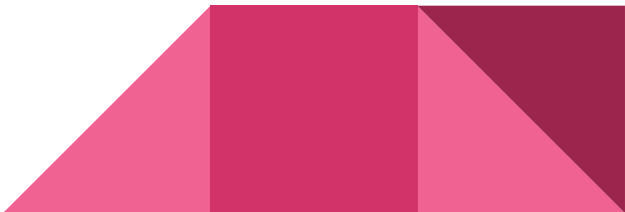


K-NN (K-Nearest Neighbours)

The k-nearest neighbors algorithm is one of the most used algorithms in machine learning. Before classifying a new element, we must compare it to other elements using a similarity measure. Its k-nearest neighbors are then considered, the class that appears most among the neighbors is assigned to the element to be classified. The neighbors are weighted by the distance that separate it to the new elements to classify.



KNN ALGORITHM

1. Load the data
 2. Initialise the value of k
 3. For getting the predicted class, iterate from 1 to total number of training data points
 1. 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.
 2. Sort the calculated distances in ascending order based on distance values
 3. Get top k rows from the sorted array
 4. Get the most frequent class of these rows
 5. Return the predicted class
- 

A case is classified by a majority vote of its neighbors, with the case being assigned to the class most common amongst its K nearest neighbors measured by a distance function.

If $K = 1$, then the case is simply assigned to the class of its nearest neighbor.

It should also be noted that all three distance measures are only valid for continuous variables.

Distance functions

Euclidean

$$\sqrt{\sum_{i=1}^k (x_i - y_i)^2}$$

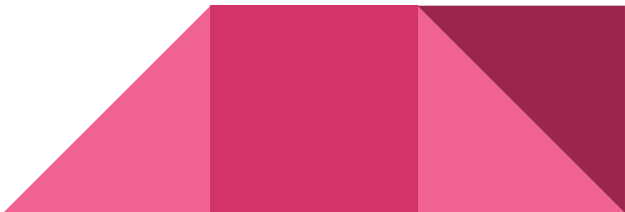
Manhattan

$$\sum_{i=1}^k |x_i - y_i|$$

Minkowski

$$\left(\sum_{i=1}^k (|x_i - y_i|)^q \right)^{1/q}$$

ADVANTAGES

1. **K-NN is pretty intuitive and simple**
 2. **K-NN has no assumptions:** K-NN is a non-parametric algorithm which means there are assumptions to be met to implement K-NN.
 3. **No Training Step:** K-NN does not explicitly build any model, it simply tags the new data entry based learning from historical data.
- 

4. **Can be used both for Classification and Regression**

5. **Variety of distance criteria to be choose from:**

K-NN algorithm gives user the flexibility to choose distance while building K-NN model.

Euclidean Distance

Hamming Distance

Manhattan Distance

Minkowski Distance




DISADVANTAGES

1) K-NN slow algorithm: K-NN might be very easy to implement but as dataset grows efficiency or speed of algorithm declines very fast.

2)Optimal number of neighbors: One of the biggest issues with K-NN is to choose the optimal number of neighbors to be consider while classifying the new data entry.

3)Imbalanced data causes problems: k-NN doesn't perform well on imbalanced data. If we consider two classes, A and B, and the majority of the training data is labeled as A, then the model will ultimately give a lot of preference to A. This might result in getting the less common class B wrongly classified.

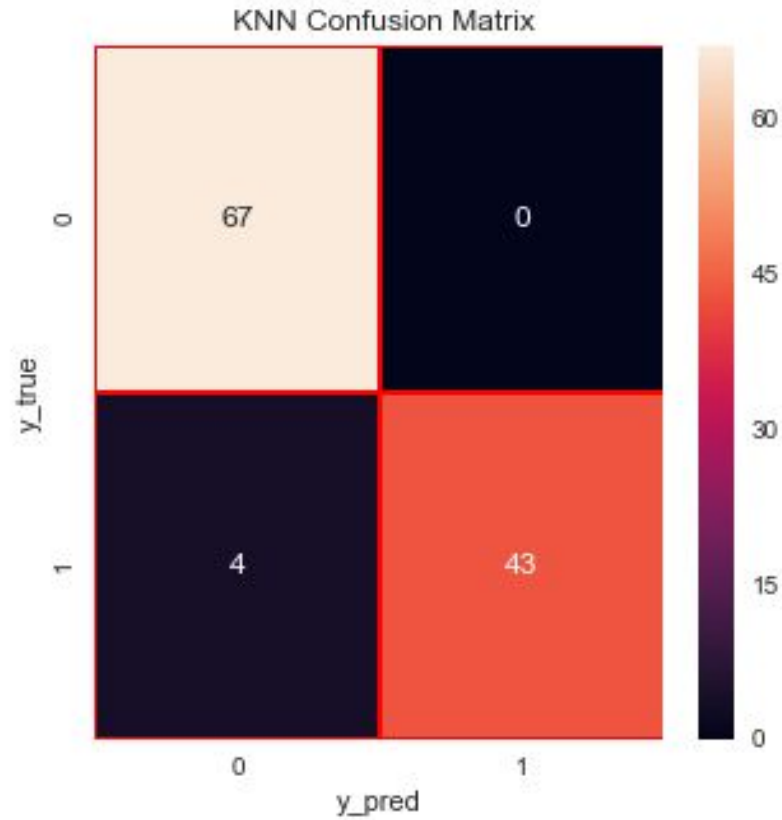


RESULT

The following table gives the values of precision, recall ,f-score,support and accuracy when the KNN algorithm was run at k=5.

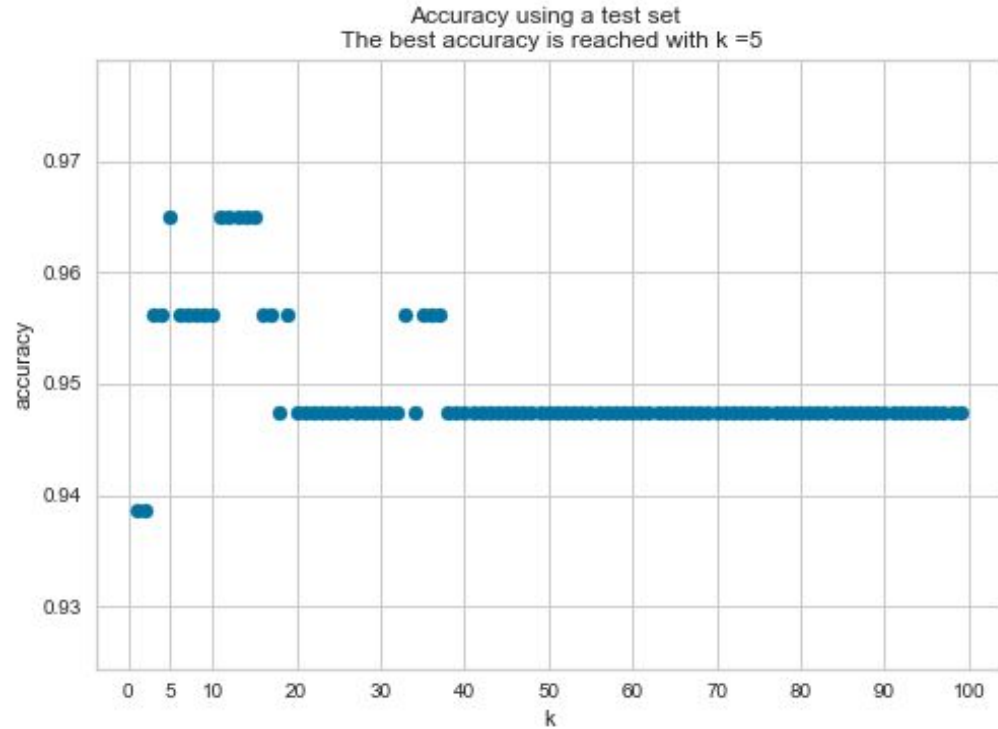
	precision	recall	f1-score	support
0	0.94	1.00	0.97	67
1	1.00	0.91	0.96	47
micro avg	0.96	0.96	0.96	114
macro avg	0.97	0.96	0.96	114
weighted avg	0.97	0.96	0.96	114

Accuracy is 96.49122807017544 % for K-Value:5

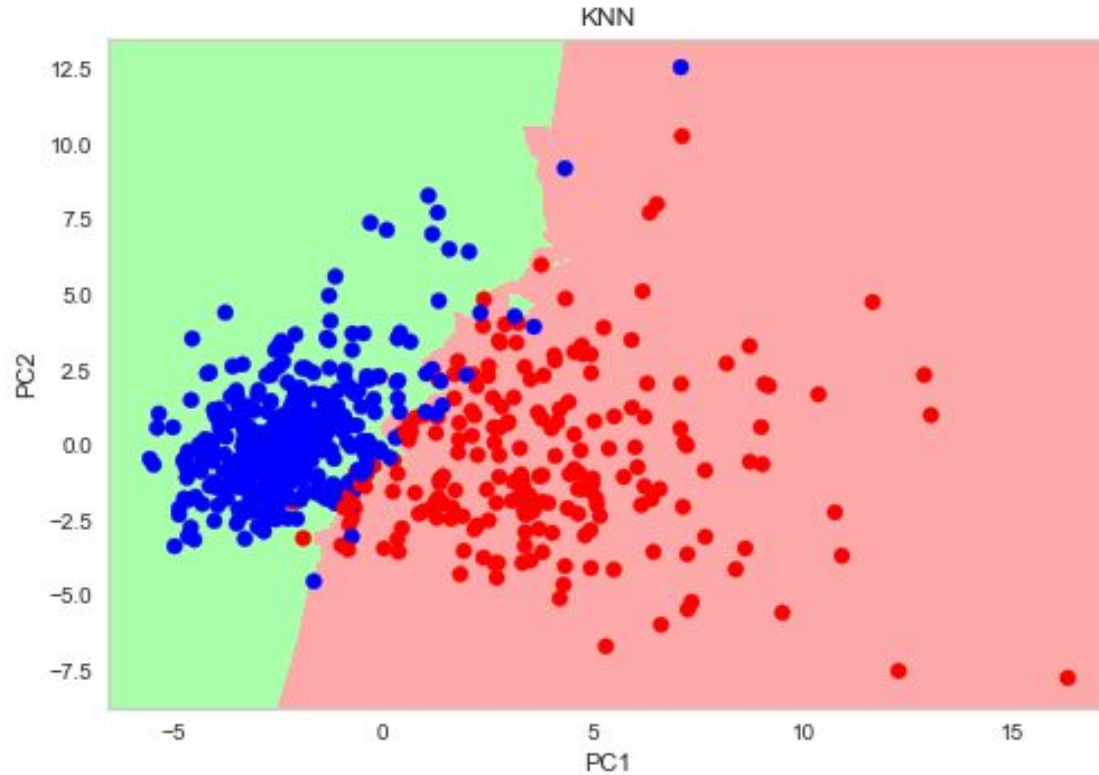


A confusion matrix plotted for the Knn algorithm is as follows:

The best value of K which maximizes the accuracy was found by running the algorithm on a different number of values of K ranging from 1 to 100. The best results were found at K=5



Finally a plot for the decision boundary for all data was made (both training and test data):



ACCURACY AFTER APPLYING KNN ALGORITHM

ACCURACY :96.49%

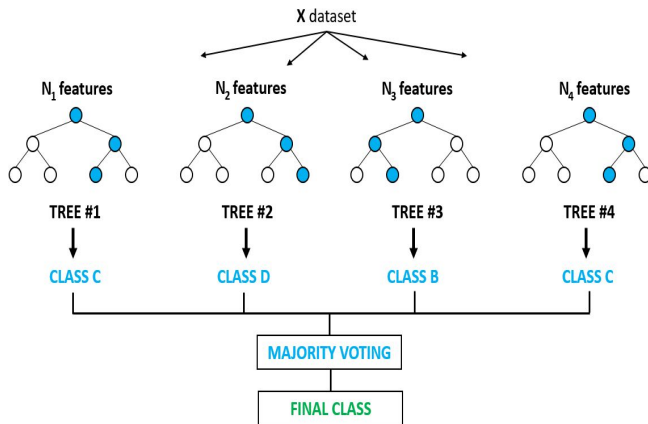


RANDOM FOREST

Random Forest is a supervised learning algorithm. It creates a forest and makes it somehow random. The 'forest' it builds, is an ensemble of Decision Trees.

Random forest builds multiple decision trees and merges them together to get a more accurate and stable prediction.


Random Forest adds additional randomness to the model, while growing the trees. Instead of searching for the most important feature while splitting a node, it searches for the best feature among a random subset of features. This results in a wide diversity that generally results in a better model.



RANDOM FOREST ALGORITHM

1. Randomly select k features from total m features.
Where $k \ll m$
2. Among the k features, calculate the node d using the best split point.
3. Split the node into daughter nodes using the best split.
4. Repeat 1 to 3 steps until “ l ” number of nodes has been reached.
5. Build forest by repeating steps 1 to 4 for n number times to create n number of trees.


To perform prediction using the trained random forest algorithm uses the below pseudocode.

1. Takes the test features and use the rules of each randomly created decision tree to predict the outcome and stores the predicted outcome (target)
 2. Calculate the votes for each predicted target.
 3. Consider the high voted predicted target as the final prediction from the random forest algorithm.
- 

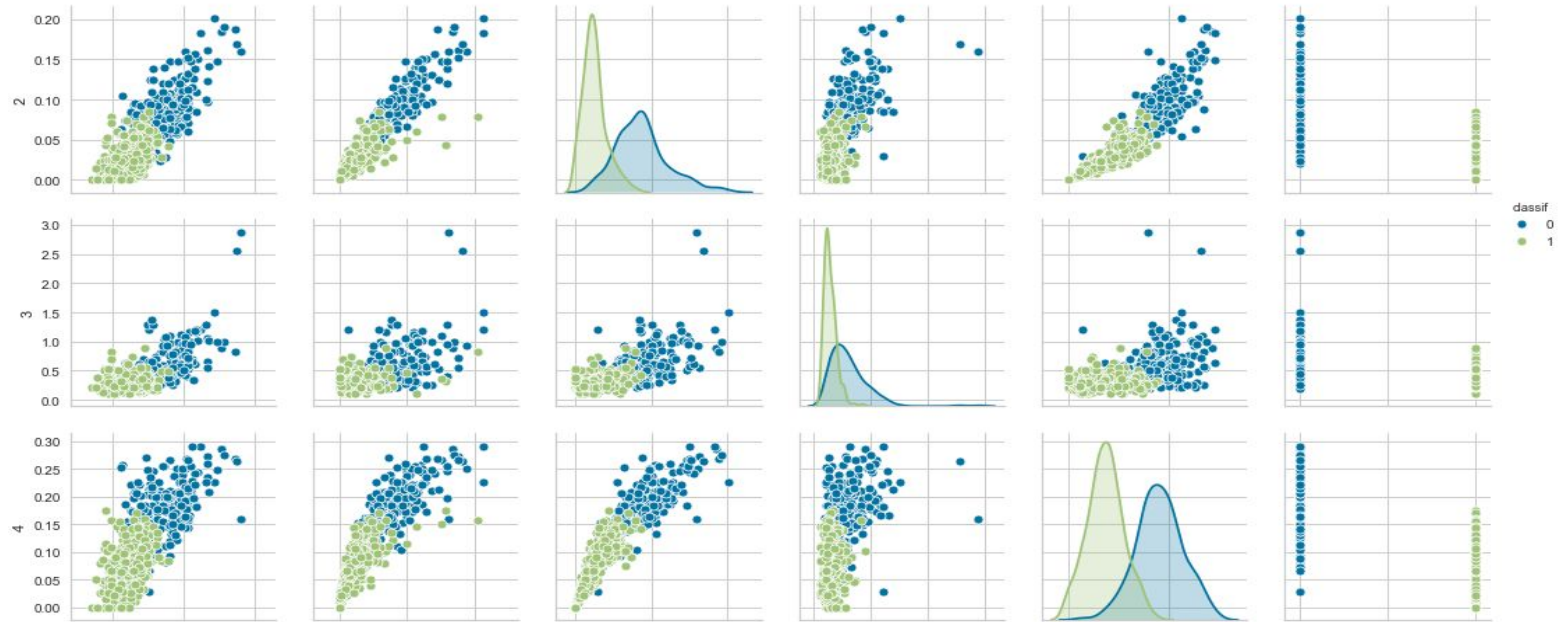
Advantages:

- It is one of the most accurate learning algorithms available. For many data sets, it produces a highly accurate classifier.
- It runs efficiently on large databases.
- It can handle thousands of input variables without variable deletion.
- It gives estimates of what variables are important in the classification.

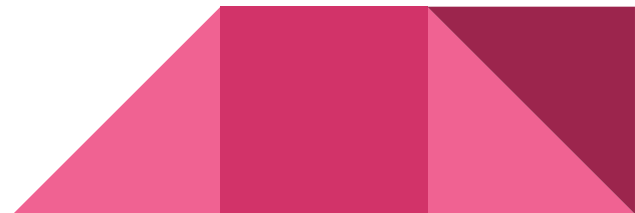
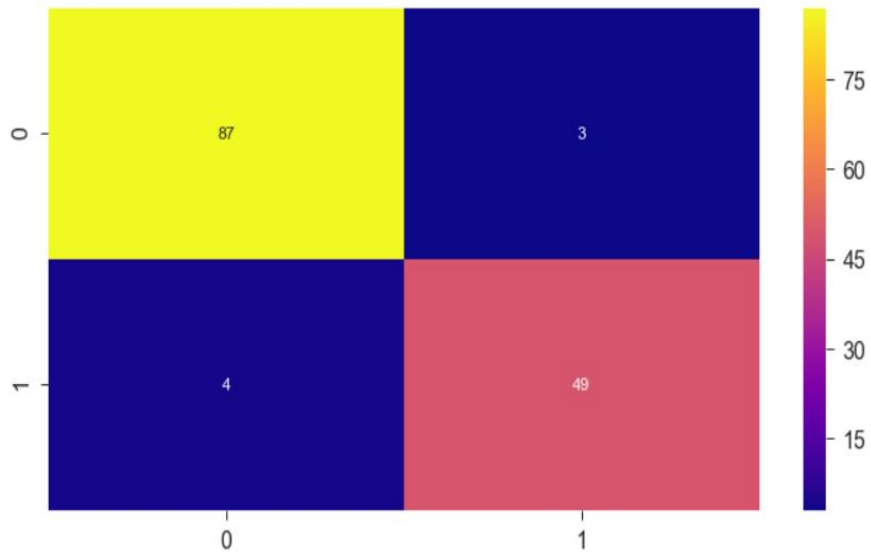
Disadvantages:

- Random forests have been observed to overfit for some datasets with noisy classification/regression tasks.
 - For data including categorical variables with different number of levels, random forests are biased in favor of those attributes with more levels. Therefore, the variable importance scores from random forest are not reliable for this type of data.
- 

OUTPUT OF RANDOM FOREST



CONFUSION MATRIX



CLASSIFICATION REPORT AFTER APPLYING RANDOM FOREST ALGORITHM

	precision	recall	f1-score	support
0	0.90	0.98	0.94	53
1	0.99	0.93	0.96	90
micro avg	0.95	0.95	0.95	143
macro avg	0.94	0.96	0.95	143
weighted avg	0.95	0.95	0.95	143

ACCURACY AFTER APPLYING RANDOM FOREST ALGORITHM

ACCURACY :94.13%



CONCLUSION

Decision tree, k-nn and Random Forest are powerful data mining techniques that can be used to classify cancerous tumors. Decision tree algorithm creates understandable rules, indicates important attributes, Random Forest yields the accuracy(93.43%) and f-measure In this study, all the algorithms have been used as intelligent methods for breast cancer diagnostic. Algorithms were successful in correctly classifying more than 92.64% cases. However, k-nn algorithm had a better predictive accuracy rate on average (rate of correct classification is 96.49%).



REFERENCES

<https://towardsdatascience.com/the-random-forest-algorithm-d457d499ffcd>

<https://www.datascience.com/resources/notebooks/random-forest-intro>

<https://medium.com/greyatom/decision-trees-a-simple-way-to-visualize-a-decision-dc506a403aeb>

<https://towardsdatascience.com/decision-trees-in-machine-learning-641b9c4e8052>

<https://www.analyticsvidhya.com/blog/2018/03/introduction-k-neighbours-algorithm-clustering/>

<https://towardsdatascience.com/knn-k-nearest-neighbors-1-a4707b24bd1d>

