HEART DISEASES MODEL ACCURACY AND ANALYSIS

CONTENTS

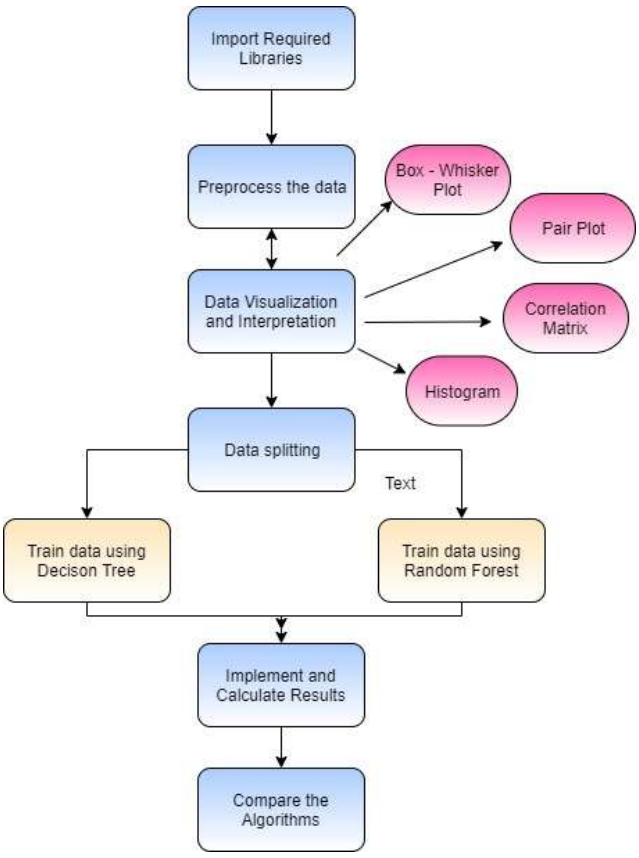
1. Introduction
2. About Our Dataset
3. Libraries Used
4. Data Preprocessing
5. Data Interpretation and Analysis
6. Data Splitting
7. Evaluation Parameters
8. Algorithms Used
    - 8.1. Decision Tree
        - 8.1.1. About the Algorithm
        - 8.1.2. Implementation and Results
    - 8.2. Random Forest
        - 8.2.1. About the Algorithm
        - 8.2.2. Implementation and Results
    - 8.3. Decision Tree X Random Forest
9. Conclusion
10. References

Introduction

Among all fatal diseases, heart diseases are considered as the most prevalent. Medical practitioners conduct different surveys on heart diseases and gather information of heart patients, their symptoms and disease progression.

In this study, a Heart Disease Prediction Analysis is developed using Random Forest and Decision Tree algorithms for predicting the risk level of heart disease. The system uses 14 medical parameters such as age, sex, blood pressure, cholesterol, and obesity for prediction.

The code predicts the likelihood of patients getting heart disease. It helps us establish a relationship between medical factors related to heart disease and form patterns. Through this project we aim to exploit data mining techniques on medical dataset to assist in the prediction of the heart diseases. A process flowchart is shown below:



About our Dataset

The data on Heart Diseases helps us understand how the pandemic is progressing so as to take counter measures to curb the disease spread.
The dataset used in this project is the Cleveland Heart Disease dataset taken from the UCI repository.

Index	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
0	63	1	1	145	233	1	2	150	0	2.3	3	0	6	0
1	67	1	4	160	286	0	2	108	1	1.5	2	3	3	2
2	67	1	4	120	229	0	2	129	1	2.6	2	2	7	1
3	37	1	3	130	250	0	0	187	0	3.5	3	0	3	0
4	41	0	2	130	204	0	2	172	0	1.4	1	0	3	0
5	56	1	2	120	236	0	0	178	0	0.8	1	0	3	0
6	62	0	4	140	268	0	2	160	0	3.6	3	2	3	3
7	57	0	4	120	354	0	0	163	1	0.6	1	0	3	0
8	63	1	4	130	254	0	2	147	0	1.4	2	1	7	2
9	53	1	4	140	203	1	2	155	1	3.1	3	0	7	1
10	57	1	4	140	192	0	0	148	0	0.4	2	0	6	0
11	56	0	2	140	294	0	2	153	0	1.3	2	0	3	0
12	56	1	3	130	256	1	2	142	1	0.6	2	1	6	2
13	44	1	2	120	263	0	0	173	0	0	1	0	7	0
14	52	1	3	172	199	1	0	162	0	0.5	1	0	7	0
15	57	1	3	150	168	0	0	174	0	1.6	1	0	3	0
16	48	1	2	110	229	0	0	168	0	1	3	0	7	1
17	54	1	4	140	239	0	0	160	0	1.2	1	0	3	0
18	48	0	3	130	275	0	0	139	0	0.2	1	0	3	0

Source: https://www.kaggle.com/ronitf/heart-disease-uci

The dataset consists of 303 individual's data. There are 14 columns in our dataset, which are described below:

Age: displays the age of the individual.

Sex: displays the gender of the individual using the following format :
1 = male , 0 = female

Chest-pain type: displays the type of chest-pain experienced by the individual using the following format: 1 = typical angina, 2 = atypical angina, 3 = non — angina pain , 4 = asymptotic

Resting Blood Pressure: displays the resting blood pressure value of an individual in mmHg (unit)

Serum Cholesterol: displays the serum cholesterol in mg/dl (unit)

Fasting Blood Sugar: compares the fasting blood sugar value of an individual with 120mg/dl.
If fasting blood sugar > 120mg/dl then : 1 (true) else : 0 (false)

Resting ECG : displays resting electrocardiographic results :
0 = normal
1 = having ST-T wave abnormality
2 = left ventricular hypertrophy

Max heart rate achieved: displays the max heart rate achieved by an individual

Exercise induced angina: 1 = yes 0 = no

ST depression induced by exercise relative to rest: displays the value which is an integer or float.

Peak exercise ST segment: 1 = up sloping ,2 = flat, 3 = down sloping

Number of major vessels (0–3) colored by fluoroscopy: displays the value as integer or float.

Thal: displays the thalassemia : 3 = normal, 6 = fixed defect , 7 = reversible defect

Diagnosis of heart disease: Displays whether the individual is suffering from heart disease or not : 0 = absence , 1, 2, 3, 4 = present.

Libraries Used

Numpy: Used for adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays.

Pandas: Used for data manipulation and analysis.

Matplotlib.pyplot: Collection of command style functions that make matplotlib work like MATLAB.

Seaborn: It is a Python data visualization library based on matplotlib.

Scikit-learn: Sklearn library contains a lot of efficient tools for machine learning and statistical modeling including classification, regression, clustering and dimensionality reduction.

IPython.display: Public API for display tools in IPython.

Pydot : It is an interface to Graphviz and can parse and dump into the DOT language used by GraphViz.

Data Preprocessing

Steps Involved in Data Preprocessing:

Data Cleaning: The data can have many irrelevant and missing parts. To handle this part, data cleaning is done. It involves handling of missing data, noisy data etc.

Data Transformation: This step is taken in order to transform the data in appropriate forms suitable for mining process.

Data Reduction: Since data mining is a technique that is used to handle huge amount of data. While working with huge volume of data, analysis became harder in such cases. In order to get rid of this, we uses data reduction technique. It aims to increase the storage efficiency and reduce data storage and analysis costs.

sklearn.preprocessing.StandardScaler() function():

This function Standardize features by removing the mean and scaling to unit variance. The standard score of a sample x is calculated as:

z = (x – u) / s, where,

x = variable

u = mean

s = standard deviation

Dataset before preprocessing :

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
0	63	1	3	145	233	1	0	150	0	2.3	0	0	1	1
1	37	1	2	130	250	0	1	187	0	3.5	0	0	2	1
2	41	0	1	130	204	0	0	172	0	1.4	2	0	2	1
3	56	1	1	120	236	0	1	178	0	0.8	2	0	2	1
4	57	0	0	120	354	0	1	163	1	0.6	2	0	2	1

Dataset after preprocessing :

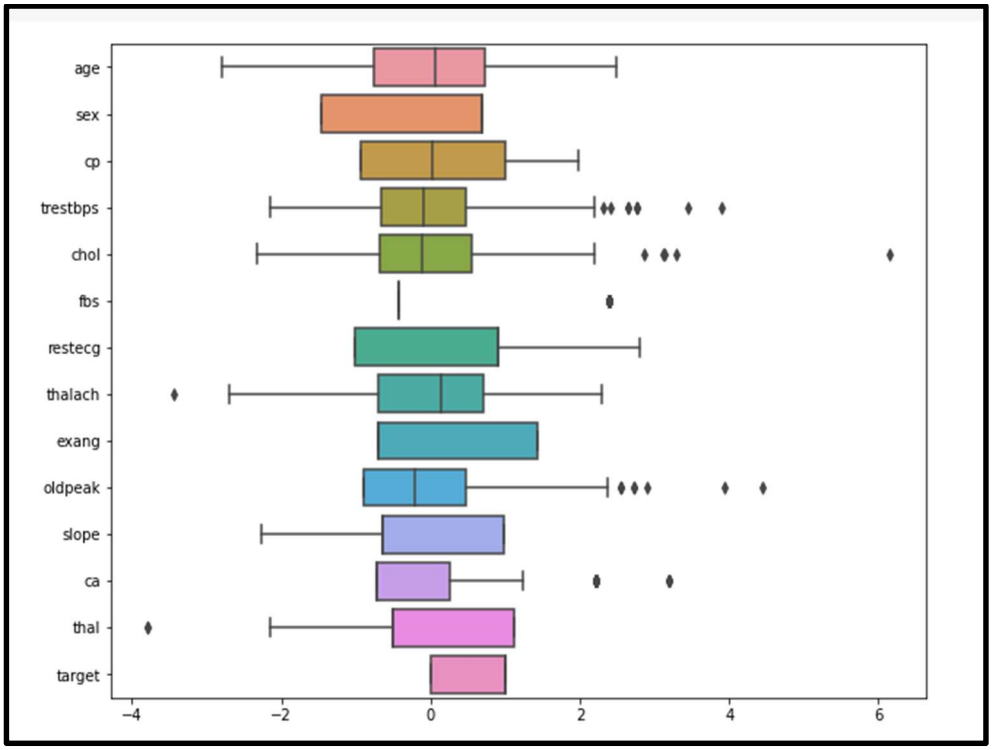
	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
0	0.952197	0.681005	1.973123	0.763956	-0.256334	2.394438	-1.005832	0.015443	-0.696631	1.087338	-2.274579	-0.714429	-2.148873	1
1	-1.915313	0.681005	1.002577	-0.092738	0.072199	-0.417635	0.898962	1.633471	-0.696631	2.122573	-2.274579	-0.714429	-0.512922	1
2	-1.474158	-1.468418	0.032031	-0.092738	-0.816773	-0.417635	-1.005832	0.977514	-0.696631	0.310912	0.976352	-0.714429	-0.512922	1
3	0.180175	0.681005	0.032031	-0.663867	-0.198357	-0.417635	0.898962	1.239897	-0.696631	-0.206705	0.976352	-0.714429	-0.512922	1
4	0.290464	-1.468418	-0.938515	-0.663867	2.082050	-0.417635	0.898962	0.583939	1.435481	-0.379244	0.976352	-0.714429	-0.512922	1

Data Interpretation and Analysis

Target Count of each Target Class

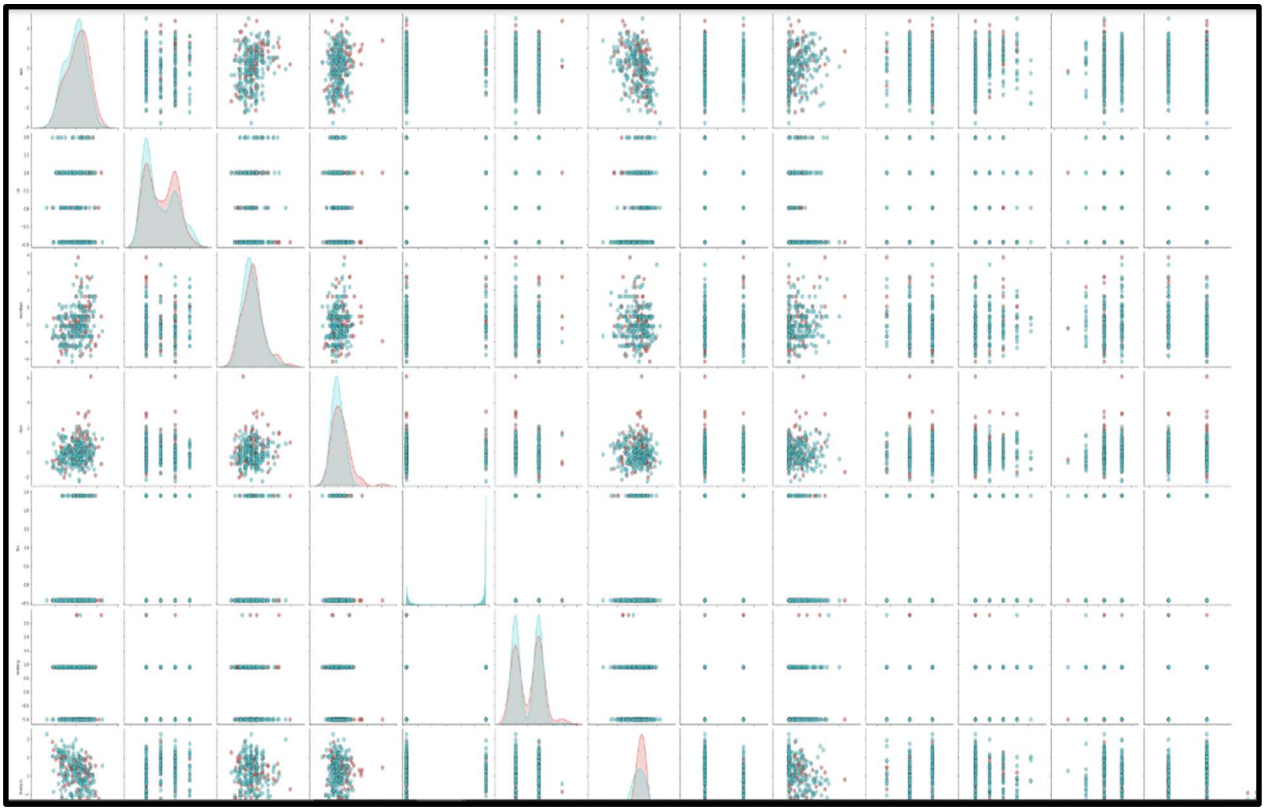
Here, target = 1 implies that the person is suffering from heart disease and target = 0 implies the person is not suffering.

Box and Whisker Plot



A box and whisker plot helps us visualize how the values in the data are spread out. It tells us about the outliers and what their values are. It can also tell us whether data is symmetrical and how tightly the data is grouped.

Pair-Plot

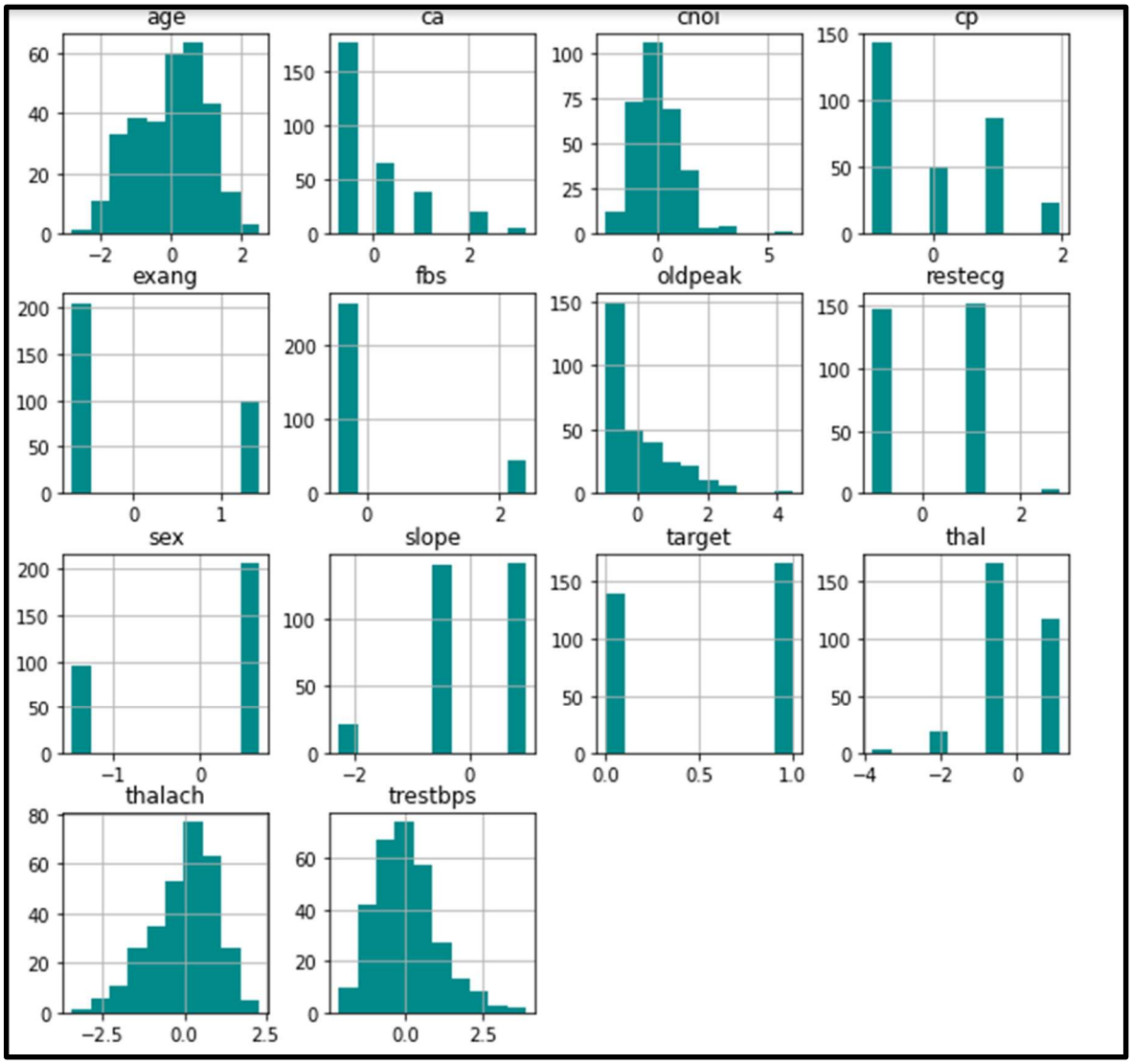


Since our dataset contains many variables, the pairs plot built on the KDE plot and the scatter plot , is used. The KDE plot i.e Kernel Density Estimate Plot on the diagonal allows us to see the Probability Density of a continuous variable while the scatter plots on the upper and lower triangles show the relationship (or lack thereof) between two variables. Hence, the relation between each and every variable can be analyzed.

Heat Map

The correlation matrix in the form of heatmap shows us the correlation coefficients between variables. Each cell in the table shows the correlation between two variables. A correlation matrix is used to summarize data in order to identify patterns.

Histograms



The histograms for each parameter have been plotted where each bar groups numbers into ranges. Taller bars show that more data falls in that range. The histograms display the shape and spread of continuous sample data for each parameter.

Data Splitting

To help us split the data in a random manner, the **SciKit library** provides a tool, called the **Model Selection** library which contained a class named 'train_test_split.' Using this we can easily split the dataset into the training and the testing datasets in various proportions. There are three parameters as follows:

test_size — This parameter decides the size of the data that has to be split as the test dataset. This is given as a fraction.

train_size — Specify this parameter only if you're not specifying the test_size. This is the same as test_size, but instead it tells the class what percent of the dataset one wants to split as the training set.

random_state — This parameter will act as the seed for the random number generator during the split. It is given as an integer.

Evaluation Parameters

To evaluate the effectiveness of the Machine Learning algorithms, the following measures have been used , for which the confusion matrix is the basis.

Confusion Matrix

The confusion matrix is a table used to describe the performance of a classification model on a set of test data for which the true values are known. It helps visualization of the performance of an algorithm. The confusion matrix can be represented as follows:

Confusion Matrix

	Actually Positive (1)	Actually Negative (0)
Predicted Positive (1)	True Positives (TPs)	False Positives (FPs)
Predicted Negative (0)	False Negatives (FNs)	True Negatives (TNs)

Where –
True +ve = Number of positive instances correctly classified as positive.
False +ve = Number of positive instances incorrectly classified as negative.
True -ve = Number of negative instances correctly classified as negative.
False -ve = Number of negative instances incorrectly classified as positive

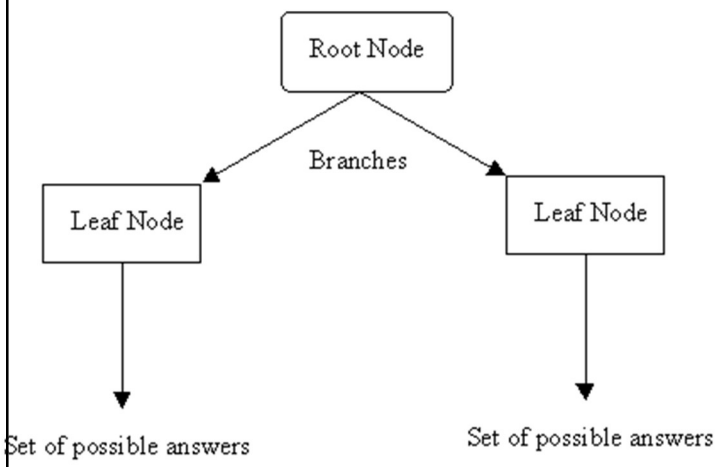
Based on the values of TP, FP, TN and FN, the following measures can be determined:

Measure	Definition	Formula
Accuracy	It indicates the closeness of a predicted or classified value to its real value.	$\frac{TP + TN}{TP + TN + FP + FN}$
Precision	It represents the probability that an item is relevant. It is the measure of exactness.	$\frac{TP}{TP + FP}$
Recall	It represents a probability that a relevant item is selected. It measures completeness.	$\frac{TP}{TP + FN}$
F1 - Measure	It is the harmonic mean between Precision and Recall.	$\frac{2 * Precision * Recall}{Precision + Recall}$
Sensitivity	Used to find out the proportion of positive samples that are correctly identified also called a true positive rate	$\frac{TP}{TP + FN}$
Specificity	Used to find out the proportion of negative samples that are correctly identified and also called a true negative rate.	$\frac{TN}{FP + TN}$
False Positive Rate	Used to find out the proportion of negative samples that are misclassified as positive samples.	$\frac{FP}{FP + TN}$
False Negative Rate	Used to find out the proportion of positive samples which are misclassified as negative samples.	$\frac{FN}{TP + FN}$
Negative Predictive Value	Used to find out the number of samples which are true negative.	$\frac{TN}{FN+TN}$
False Discovery Rate (FDR)	Used to find out a proportion of false positive among all the samples that are classified as positive	$\frac{FP}{TP+FP}$
Matthews correln coeff. (MCC)	It is a correlation coefficient between the actual classes and predicted classes.	$\frac{TP \times TN - FN \times FP}{\sqrt{(TP + FN)(TP + FP)(TN + FN)(TN + FP)}}$

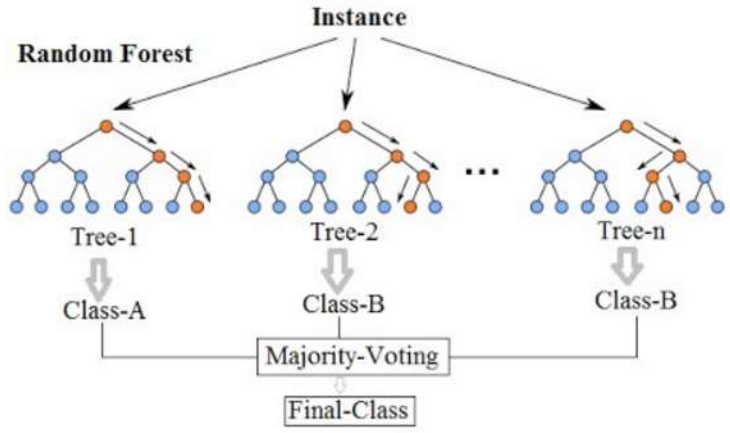
Algorithms

In this project, we have implemented two machine learning algorithms and have compared both their accuracies at the end. The algorithms are as follows:

Decision Tree



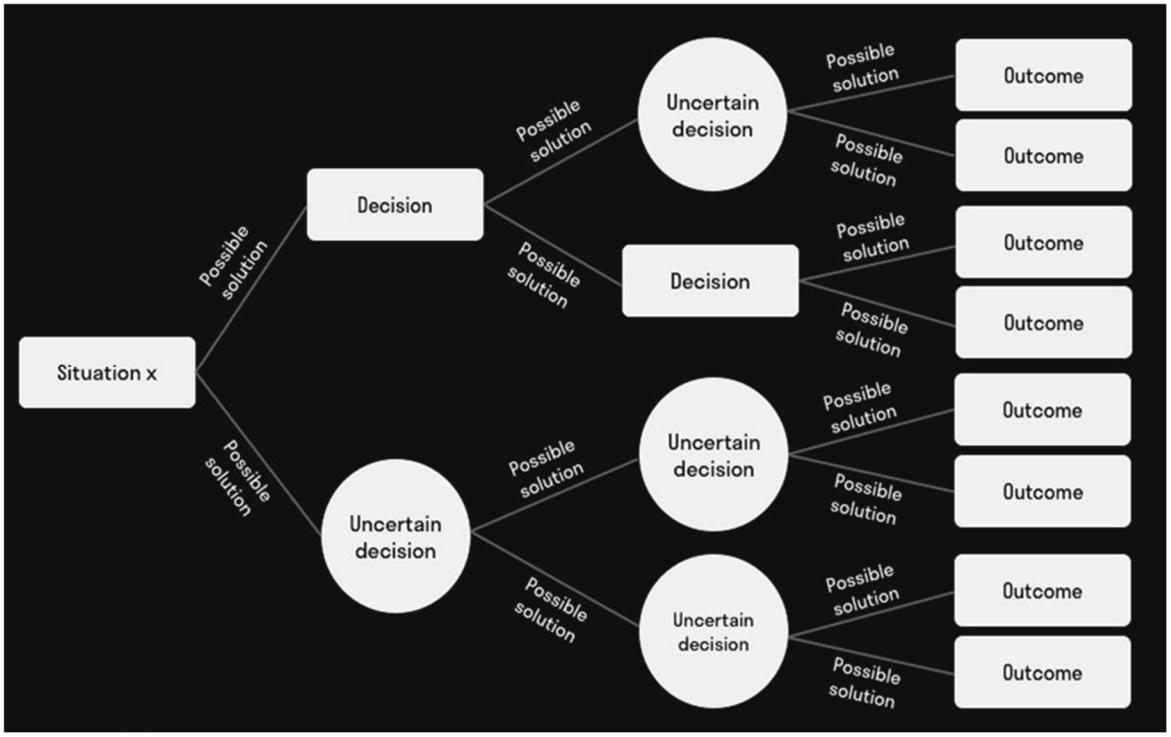
Random Forest



Decision Tree

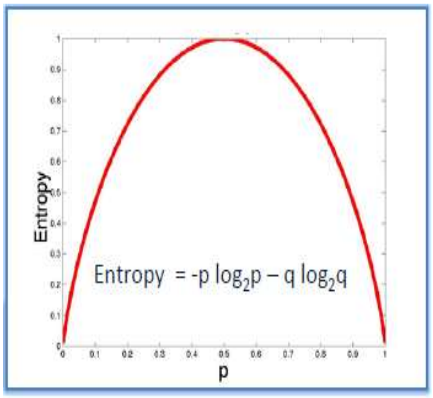
About our Algorithm

A decision tree is a map of the possible outcomes of a series of related choices. It allows an individual or organization to weigh possible actions against one another based on their costs, probabilities, and benefits. They can be used to map out an algorithm that predicts the best choice mathematically.



Decision trees two major parameters to decide to split a node into two or more sub-nodes:

a) ***Entropy:*** To build a decision tree, we need to calculate two types of entropy using frequency tables as follows:



- Entropy using the frequency table of one attribute:

$$E(S) = \sum_{i=1}^{c} - p_i \log_2 p_i$$

- Entropy using the frequency table of two attributes:

$$E(T, X) = \sum_{c \in X} P(c) E(c)$$

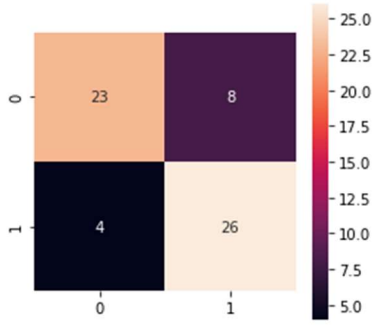
b) ***Information Gain:*** The information gain is based on the decrease in entropy after a dataset is split on an attribute

$$Gain(T, X) = Entropy(T) - Entropy(T, X)$$

Implementation And Results

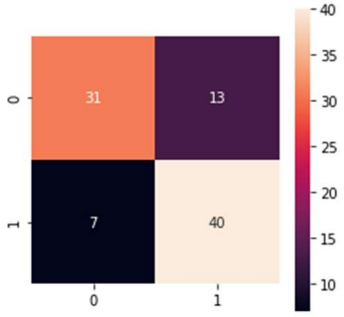
Step1: After the data has been split and trained based on different testing and training ratios using the decision classifier, the confusion matrix and the classification report is calculated for each training ratio.

Training Ratio 80:20



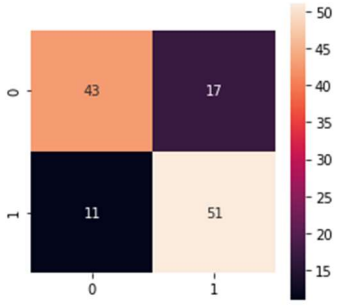
	precision	recall	f1-score	support
0	0.82	0.73	0.77	44
1	0.77	0.85	0.81	47
accuracy			0.79	91
macro avg	0.79	0.79	0.79	91
weighted avg	0.79	0.79	0.79	91

Training Ratio 70:30



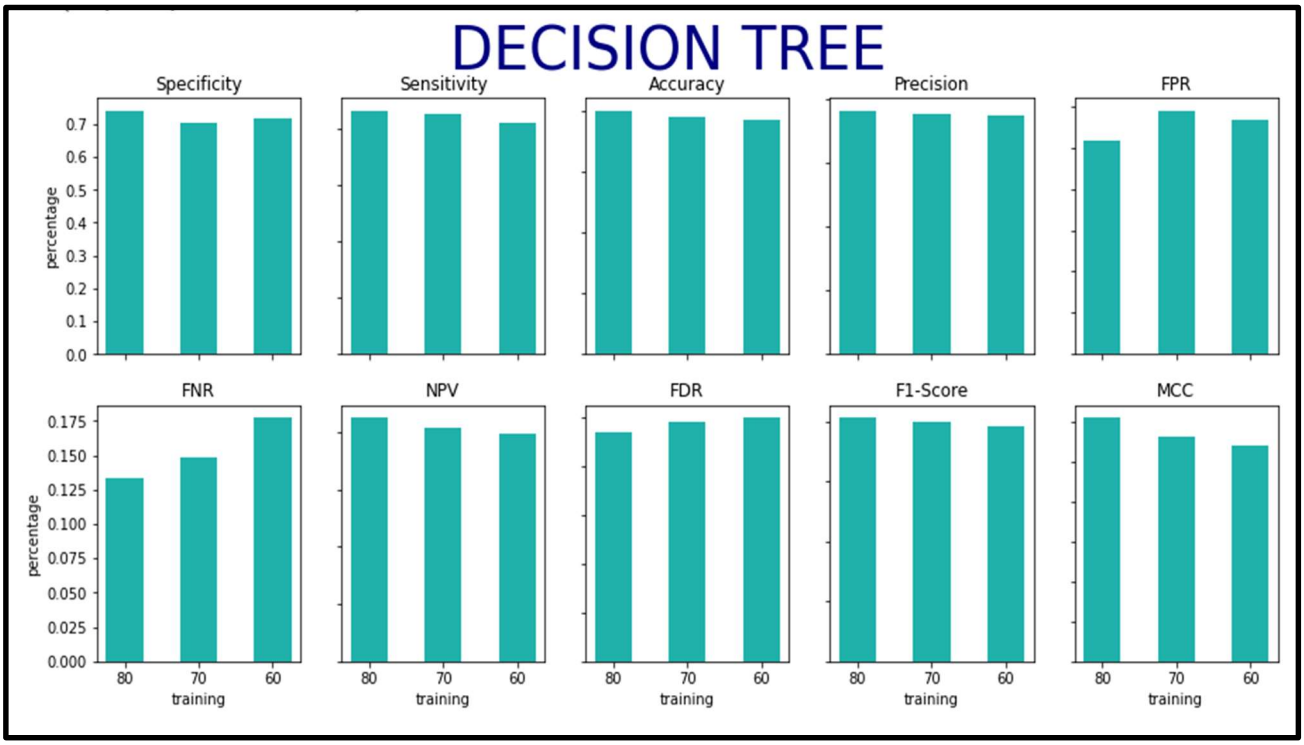
	precision	recall	f1-score	support
0	0.85	0.74	0.79	31
1	0.76	0.87	0.81	30
accuracy			0.80	61
macro avg	0.81	0.80	0.80	61
weighted avg	0.81	0.80	0.80	61

Training Ratio 60:40

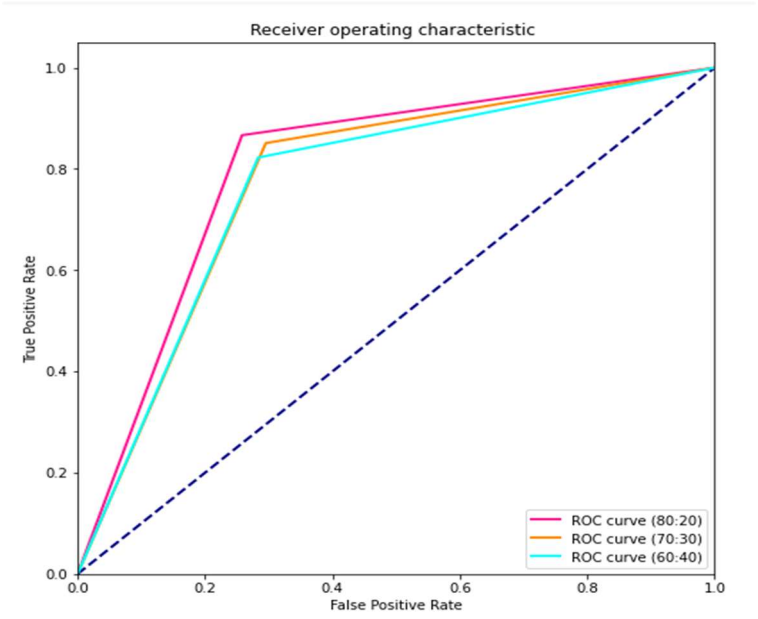


	precision	recall	f1-score	support
0	0.80	0.73	0.77	60
1	0.76	0.82	0.79	62
accuracy			0.78	122
macro avg	0.78	0.78	0.78	122
weighted avg	0.78	0.78	0.78	122

Step2: Now the evaluation parameters for each training ratio is calculated based on the formulae provided above and results are visualized graphically to compare results based on different ratios.



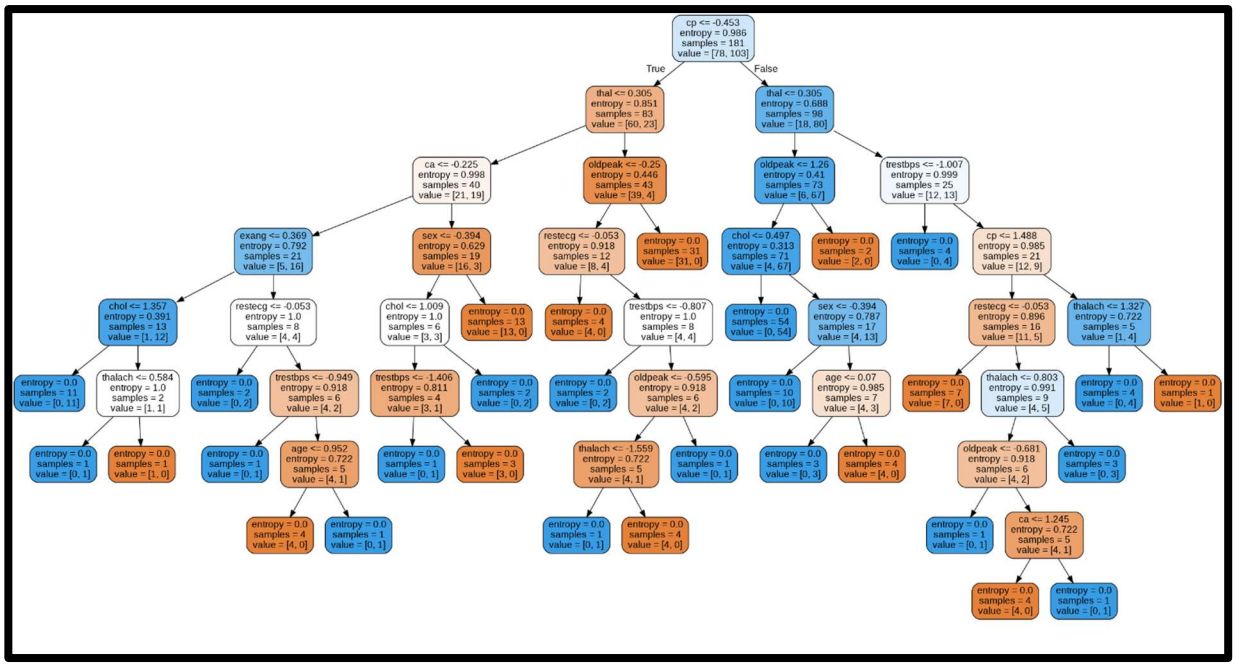
Step3: Now plotting the Receiver Operating Curve (ROC) which shows the performance of the decision tree model by plotting TPR vs. FPR at different classification thresholds for all ratios.



Step4: We will now compare the performance of the model at different training ratios in tabular form to evaluate the overall performance of the model.

	Algorithm and Measures	80:20	70:30	60:40
0	Specificity	0.741935	0.727273	0.733333
1	Sensitivity	0.866667	0.851064	0.822581
2	Accuracy	0.803279	0.791209	0.778689
3	Precision	0.764706	0.769231	0.761194
4	FPR	0.258065	0.272727	0.266667
5	FNR	0.133333	0.148936	0.177419
6	NPV	0.851852	0.820513	0.800000
7	FDR	0.235294	0.230769	0.238806
8	F1-Score	0.812500	0.808081	0.790698
9	MCC	0.612567	0.584012	0.558548

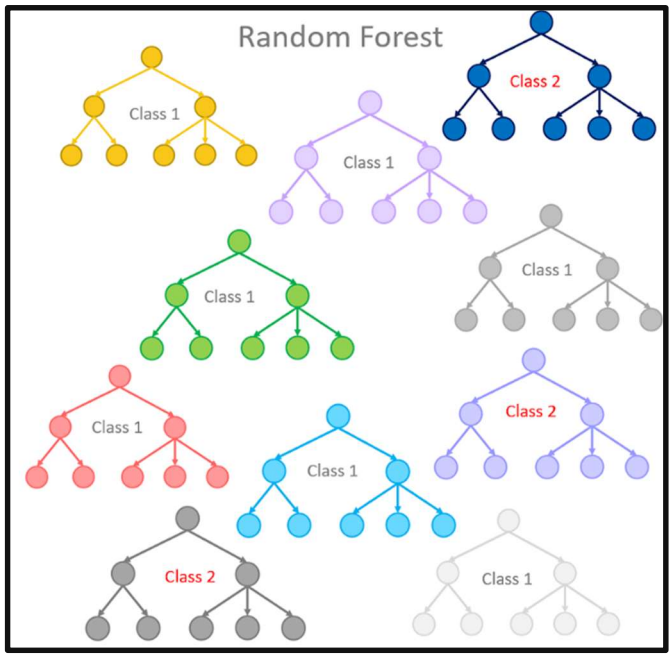
Step5: Finally, we will plot the decision tree for our dataset for visualization.



Random Forest Classifier

About our Algorithm

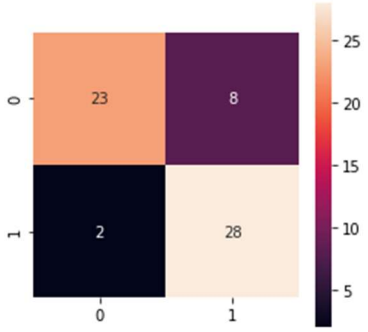
Random forest is a supervised learning algorithm which is used for both classification as well as regression. But however, it is mainly used for classification problems. As we know that a forest is made up of trees and more trees means more robust forest. Similarly, random forest algorithm creates decision trees on data samples and then gets the prediction from each of them and finally selects the best solution by means of voting. It is an ensemble method which is better than a single decision tree because it reduces the over-fitting by averaging the result.



Implementation And Results

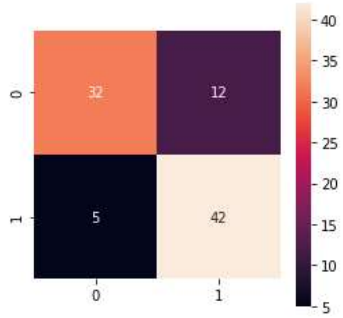
Step1: After the data has been split and trained based on different testing and training ratios using the random forest classifier, the confusion matrix and the classification report is calculated for each training ratio.

Training Ratio 80:20



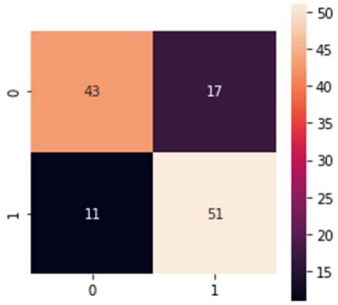
	precision	recall	f1-score	support
0	0.92	0.74	0.82	31
1	0.78	0.93	0.85	30
accuracy			0.84	61
macro avg	0.85	0.84	0.83	61
weighted avg	0.85	0.84	0.83	61

Training Ratio 70:30



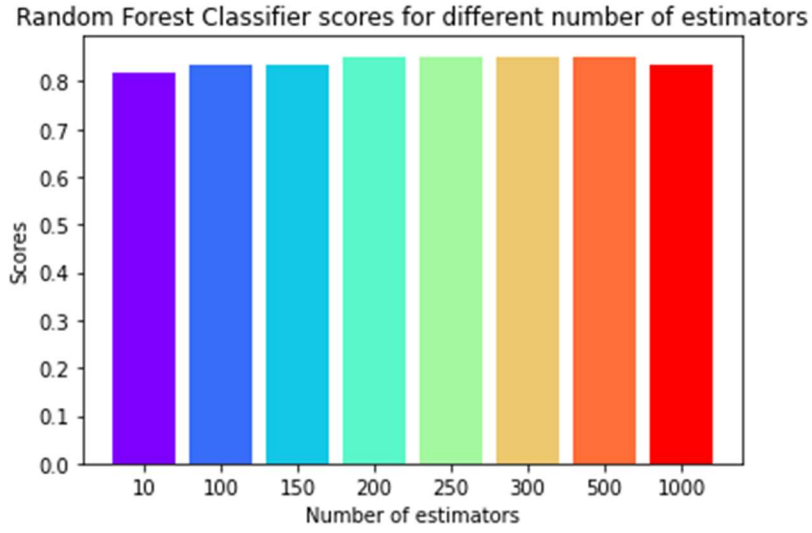
	precision	recall	f1-score	support
0	0.86	0.73	0.79	44
1	0.78	0.89	0.83	47
accuracy			0.81	91
macro avg	0.82	0.81	0.81	91
weighted avg	0.82	0.81	0.81	91

Training Ratio 60:40

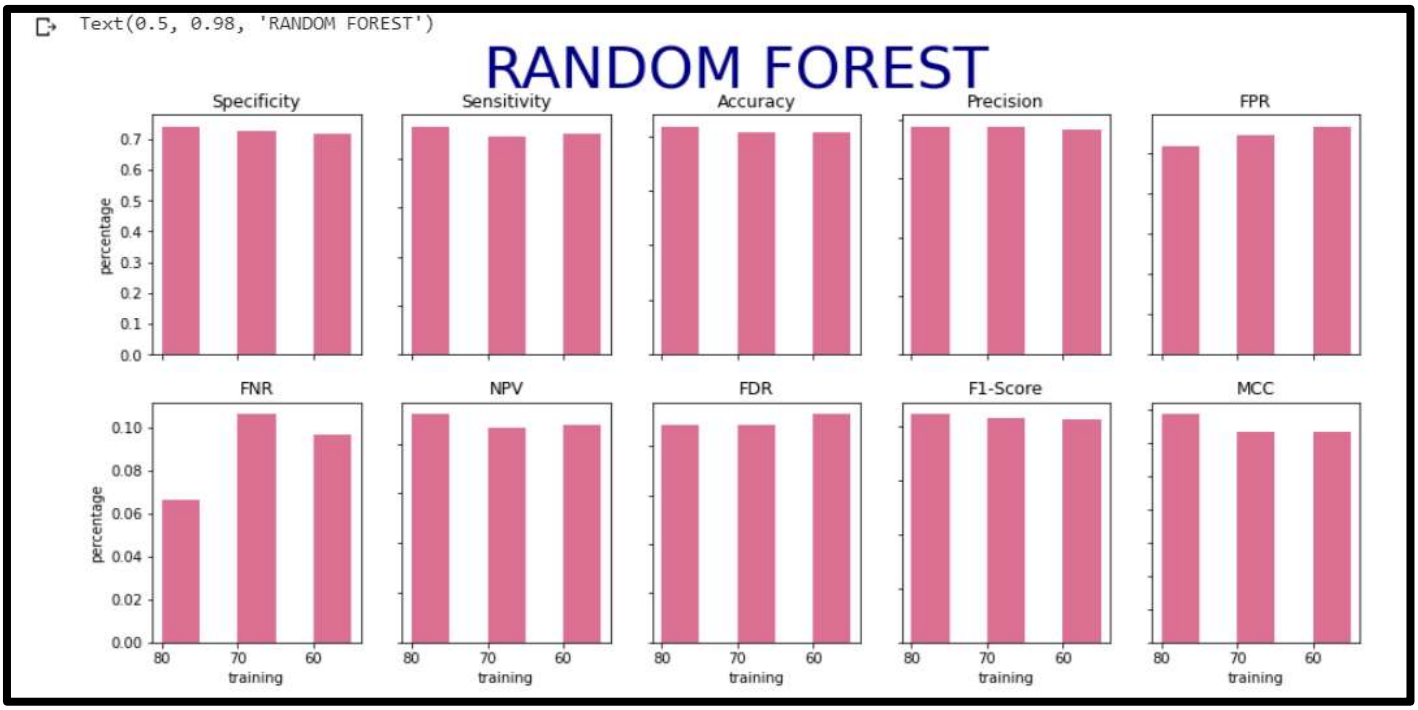


	precision	recall	f1-score	support
0	0.88	0.72	0.79	60
1	0.77	0.90	0.83	62
accuracy			0.81	122
macro avg	0.82	0.81	0.81	122
weighted avg	0.82	0.81	0.81	122

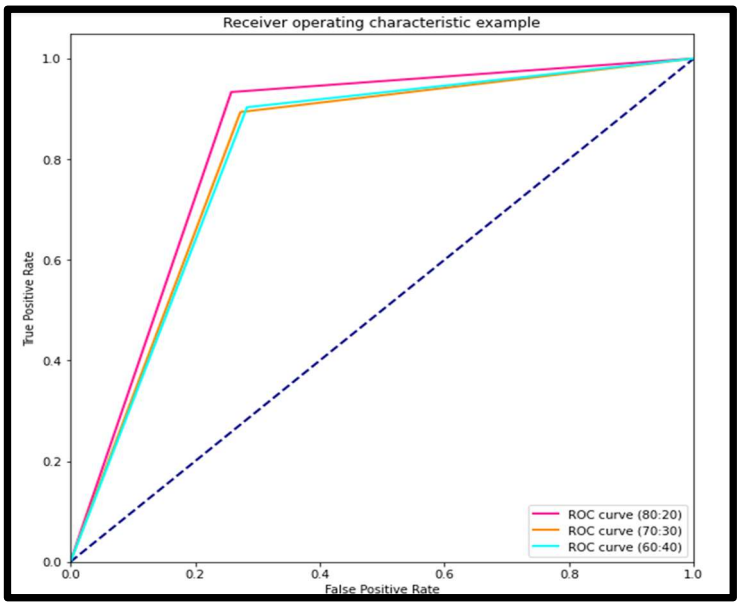
Step2: Now the random forest classifier scores for different number of estimators is shown



Step3: Now the evaluation parameters for each training ratio is calculated based on the formulae provided above and results are visualized graphically to compare results based on different ratios.



Step4: Now plotting the Receiver Operating Curve (ROC) showing the performance of the decision tree model by plotting TPR vs. FPR at different classification thresholds for all ratios.



Step5: We will now compare the performance of the model at different training ratios in tabular form to evaluate the overall performance of the model.

	Algorithm and Measures	80:20	70:30	60:40
0	Specificity	0.741935	0.727273	0.716667
1	Sensitivity	0.933333	0.893617	0.903226
2	Accuracy	0.836066	0.813187	0.811475
3	Precision	0.777778	0.777778	0.767123
4	FPR	0.258065	0.272727	0.283333
5	FNR	0.066667	0.106383	0.096774
6	NPV	0.920000	0.864865	0.877551
7	FDR	0.222222	0.222222	0.232877
8	F1-Score	0.848485	0.831683	0.829630
9	MCC	0.686431	0.631673	0.632162

Decision Tree X Random Forest

Now finally we will compare the results obtained from both the algorithms , so as to evaluate the performance of the models and understand which model works better for our dataset. Tabular data has been shown below for each training ratio.

Training Ratio 70:30

RATIO 70:30

	Algorithm and Measures	Decision Tree	Random Forest
0	Specificity	0.727273	0.727273
1	Sensitivity	0.872340	0.893617
2	Accuracy	0.802198	0.813187
3	Precision	0.773585	0.777778
4	FPR	0.272727	0.272727
5	FNR	0.127660	0.106383
6	NPV	0.842105	0.864865
7	FDR	0.226415	0.222222
8	F1-Score	0.820000	0.831683
9	MCC	0.607598	0.631673

Training Ratio 80:20

RATIO 80:20

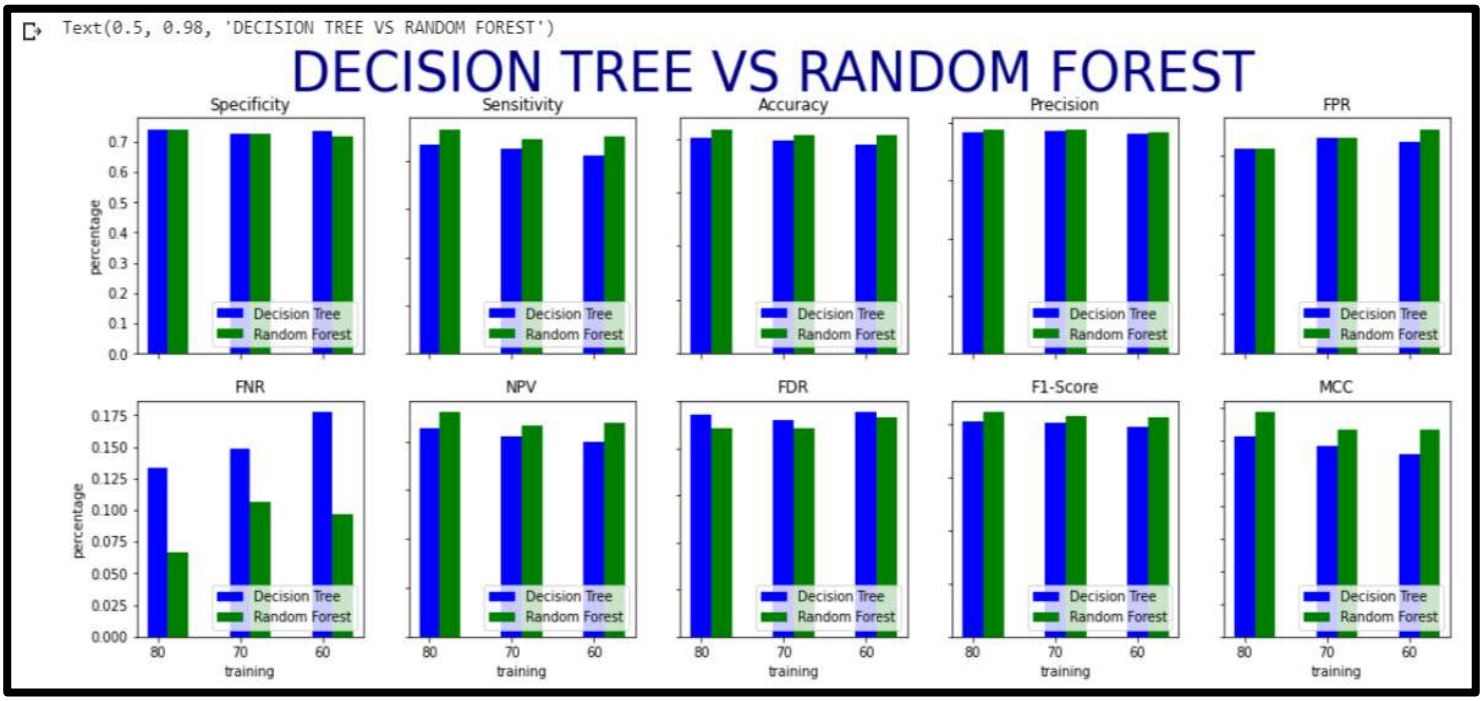
	Algorithm and Measures	Decision Tree	Random Forest
0	Specificity	0.741935	0.741935
1	Sensitivity	0.866667	0.933333
2	Accuracy	0.803279	0.836066
3	Precision	0.764706	0.777778
4	FPR	0.258065	0.258065
5	FNR	0.133333	0.066667
6	NPV	0.851852	0.920000
7	FDR	0.235294	0.222222
8	F1-Score	0.812500	0.848485
9	MCC	0.612567	0.686431

Training Ratio 60:40

RATIO 60:40

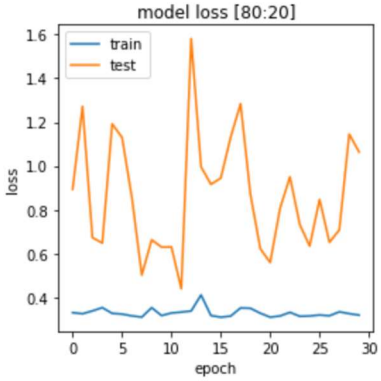
	Algorithm and Measures	Decision Tree	Random Forest
0	Specificity	0.733333	0.716667
1	Sensitivity	0.822581	0.903226
2	Accuracy	0.778689	0.811475
3	Precision	0.761194	0.767123
4	FPR	0.266667	0.283333
5	FNR	0.177419	0.096774
6	NPV	0.800000	0.877551
7	FDR	0.238806	0.232877
8	F1-Score	0.790698	0.829630
9	MCC	0.558548	0.632162

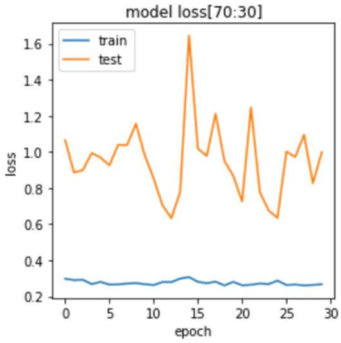
The comparison graphs between decision tree and random forest is plotted below.

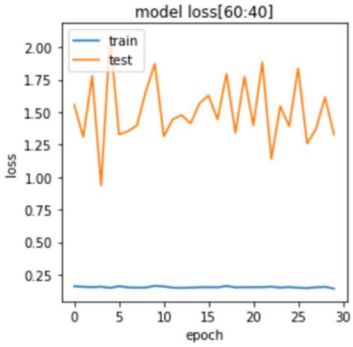


Accuracy X Loss

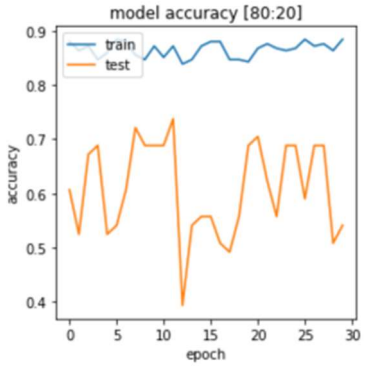
A loss function is used to optimize a machine learning algorithm. It is the sum of errors made for each example in training or validation sets. Loss value implies how poorly or well a model behaves after each iteration of optimization.

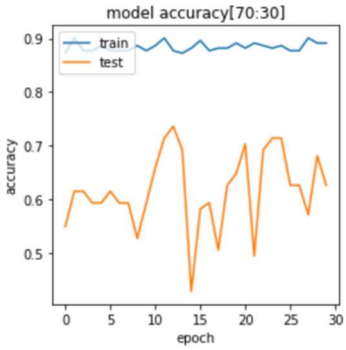


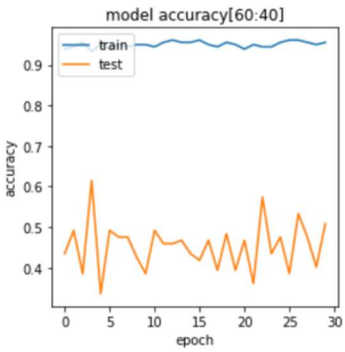




The accuracy of a model is usually determined after the model parameters and is calculated in the form of a percentage. It is the measure of how accurate your model's prediction is compared to the true data.







We plot these metrics against epochs which is one complete presentation of the data set to be learned to a learning machine. To implement loss/accuracy vs epoch, we use Artificial Neural Networks (ANN) which evaluates the data and comes up with output. Then based on that output, the neural network tries to correct itself and tries to improve the accuracy of the output. Each time an output is produced, one epoch is completed.

Conclusion

Our proposed project is user-friendly, scalable, reliable and an expandable analysis which can also help in reducing treatment costs by providing Initial diagnostics in time. The model can also serve the purpose of training tool for medical students and will be a soft diagnostic tool.

There are many possible improvements that could be explored to improve the scalability and accuracy of this prediction system. As we have developed a generalized system, in future we can use this system for the analysis of different data sets.

The performance of the health's diagnosis can be improved significantly by handling numerous class labels in the prediction process, and it can be another positive direction of research.

Heart disease is one of the leading causes of deaths worldwide and the early prediction of heart disease is very important. Some of the Heart Disease classification systems were reviewed in this project and based on different research studies it is concluded that an insight of different data mining techniques that can be employed in automated heart disease prediction systems.

References

- https://towardsdatascience.com/heart-disease-prediction-73468d630cfc
- file:///C:/Users/hp/Desktop/ProposalHeartDeseasePredictionSystem.pdf
- https://www.irjet.net/archives/V3/i8/IRJET-V3I8250.pdf
- https://medium.com/greyatom/decision-trees-a-simple-way-to-visualize-a-decision-dc506a403aeb
- https://medium.com/@contactsunny/how-to-split-your-dataset-to-train-and-test-datasets-using-scikit-learn-e7cf6eb5e0d
- https://towardsdatascience.com/understanding-random-forest-58381e0602d2