Journal of Chemical Health Risks

www.jchr.org

JCHR (2023) 13(4s), 371-379 | ISSN:2251-6727

Conditions For Cardiovascular Diseases Prediction

Ms. Jhansi Rani Ganapa[1], Mr .Sudheer Choudari[2], R.Venkata Ramana[3]

[1]Assistant Professor, Department of Computer Science and Engineering, Nalanda group of Institutions, Kantepudi, Sattenapalli, Palnadu, Andhra Pradesh.
[2]Mr .Sudheer Choudari, Assistant Professor, Department of Civil Engineering, Centurion University of Technology & Management, Vizianagaram, Andhra Pradesh
[3]Assistant Professor, Department of CSSE, Lendi Institute of Engineering and Technology.

(Received: 02 September 2023 Revised: 14 October Accepted: 07 November)

KEYWORDS

Heart Disease, Health Care Organization, Data mining, Random Forest, Decision Tree, Heart Disease Prediction System,

ABSTRACT:

data mining technology. Today data mining has lots of application in every aspects of human life. Applications of data mining are wide and diverse. Among this health care is a major application of data mining. Medical field has get benefited more from data mining. Heart Disease is the most dangerous life-threatening chronic disease globally. The objective of the work is to predicts the occurrence of heart disease of a patient using random forest algorithm. The dataset was accessed from Kaggle site. The dataset contains 303 samples and 14 attributes are taken for features of the dataset.

Then it was processed using python open access software in jupyter notebook. The datasets are classified and processed using machine learning algorithm Random forest. The outcomes of the dataset are expressed in terms of accuracy, sensitivity and specificity in percentage. Using random forest algorithm, we obtained accuracy of 86.9% for prediction of heart disease with sensitivity value 90.6% and specificity value 82.7%. From the receiver operating characteristics, we obtained the diagnosis rate for prediction of heart disease using random forest is 93.3%. The random forest algorithm has proven to be the most efficient algorithm for classification of heart disease and therefore it is used in the proposed system.

Large amount of data can be extracted by a technique known as Data Mining Technology. In day to day human life, Data mining is widely used. It has created a vertical and significant role in field of health care. Data mining has been increasingly advantageous for the medical industry. The deadliest chronic condition in the world that can cause death is heart disease. The death rate can be decreased by conducting the early diagnosis of heart disease. This programme aids in the early diagnosis and prediction of heart disease. Healthcare organisations nowadays produce enormous amounts of data, yet those data are incredibly disorganised. This data can be used to forecast cardiac illnesses with simplicity if it is properly organised using data mining techniques. The goal of this research is to use the Random Forest and Decision Tree algorithms to forecast if a patient may develop cardiac disease. From GitHub, the dataset was accessed. The data was then processed in a Jupyter notebook using open-source Python tools. Using the machine learning algorithms Random Forest and Decision Tree, the datasets are categorised and processed.

1. INTRODUCTION:

Cardiovascular illnesses, sometimes known as CVDs, affect the blood arteries that supply the heart and other body components. Fat deposits at the arteries cause this to happen, and blood clotting is a possibility. In addition to heart damage, other organs like the brain, heart, kidneys, and eyes all fail.

Figure 1*:X-RAY OF HUMAN HEART*

The various forms of CVD include: 1. coronary heart disease 2. vascular disease of the brain 3. Diseases of the peripheral arteries 4. Arthritis of the heart 5. Congenital cardiac condition 6. Pulmonary embolism and deep vein thrombosis.

Chest pain or Discomfort is the primary symptom for Heart Disease. The other symptoms are syncope, dizziness, heartbeats speed, light headedness. These irregular heartbeats are indicative of a heart condition.

One of the leading causes of impairments and mortality worldwide is CVD. A healthy lifestyle, a balanced food, and metal stability can stop this.

This research article uses Python programming and two separate algorithms (decision tree and random forest) to focus on CVD risk factors, blood pressure, glucose levels, weight and lifestyle habits. Life style habits includes the daily routine food, time and type of exercise, smoking. All the above parameters are considered for risk reduction. Since it can help us to protect people in danger and give them a little more time to live, early detection of CVD is crucial.

2.0 Objective of the Study:

Having a system that can predict CVD is the main goal of this paper. which could help to lower the death rate from CVD and would be useful for the early detection of heart diseases. This system collects user information such name, age, gender, chest discomfort, cholesterol, etc. If a person has heart disease or not, the system will make a prediction. Additionally, the goal of this paper is to demonstrate the prevalence of heart disease by age and gender. The dashboard that is shown in the accompanying paper was used to visualise our system.

3.0 Decision Tree:

This particular form of graph uses the branched method to calculate potential choice outcomes. Since the data set is labelled, this classification is supervised. Graph visualisation software can be used to generate this tree. It has a similar structure to a flowchart, with each node representing test case of an problem and the results of the previous nodes are represented by branch. The class label is identified by final node, or leaf node.

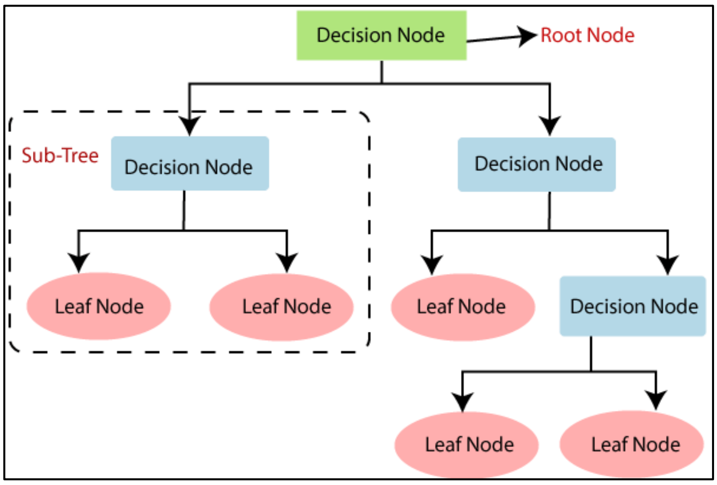


Figure 2*: Decision tree*

3.1 System Description Using Decision Tree:

The cardiac disease is predicted by this system. Heart disease is predicted using a decision tree. Making decisions is aided by decision trees, which offer sound decision-making methods. The examples are split up into different trees. It has leaf nodes, branch nodes, and root nodes.

3.2 Random Forest:

Primarily depending exclusively on one decision tree, the random forest takes the prediction from each tree and based its prediction of the ultimate output on the majority vote of predictions. It is a classifier that consists of a number of decision trees on various subsets of the provided dataset and takes the average to enhance the predicted accuracy of that dataset.

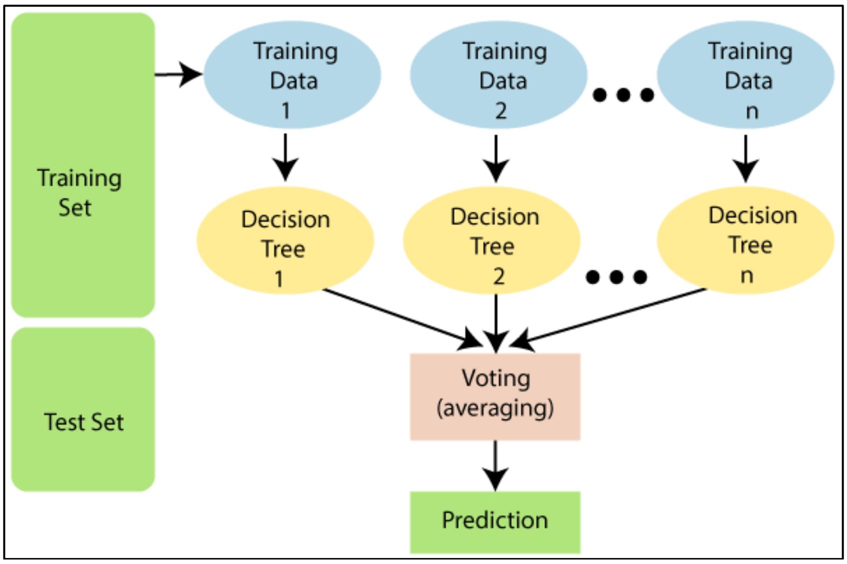


Figure 3*:Random Forest*

3.3 Relationship between Random Forest and Decision Tree:

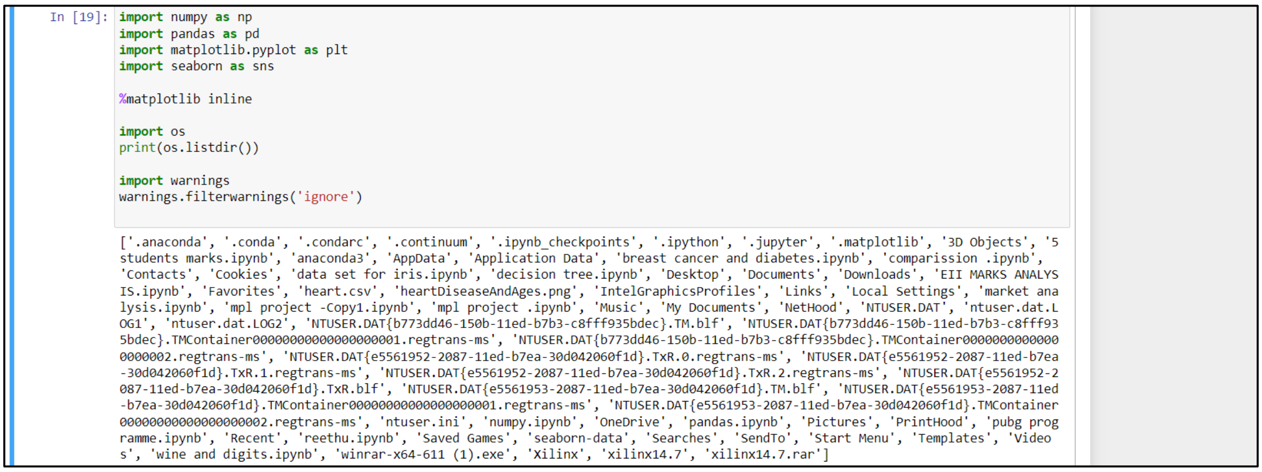
The random forest is nothing more than a collection of decision trees, the outcomes of which are aggregated into a single final result. They can limit over fitting without significantly increasing error due to bias. Random forest and decision tree have nearly same hyper parameters. A decision tree is created by randomly splitting data.

Table 1: Difference between Random Forest and Decision Tree

Sno.	**Random forest**	**Decision tree**
1.	combines several decisions	Combines certain decision only
2.	Interpretation cannot be done easily	This can be interpreted easily
3.	Overfitting takes time	Can be overfit easily
4.	More accurate results are obtained	Less accurate results are obtained

4.0 Dataset Structure and Description for the system:

4.1 Importing Libraries:

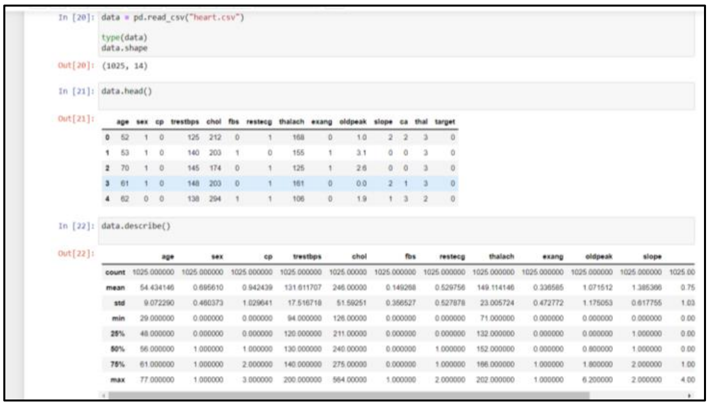


```
In [19]: import numpy as np
         import pandas as pd
         import matplotlib.pyplot as plt
         import seaborn as sns

         %matplotlib inline

         import os
         print(os.listdir())

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

```
['.anaconda', '.conda', '.condarc', '.continuum', '.ipynb_checkpoints', '.ipython', '.jupyter', '.matplotlib', '3D Objects', '5
students marks.ipynb', 'anaconda3', 'AppData', 'Application Data', 'breast cancer and diabetes.ipynb', 'comparission .ipynb',
'Contacts', 'Cookies', 'data set for iris.ipynb', 'decision tree.ipynb', 'Desktop', 'Documents', 'Downloads', 'EII MARKS ANALYS
IS.ipynb', 'Favorites', 'heart.csv', 'heartDiseaseAndAges.png', 'IntelGraphicsProfiles', 'Links', 'Local Settings', 'market ana
lysis.ipynb', 'mpl project -Copy1.ipynb', 'mpl project .ipynb', 'Music', 'My Documents', 'NetHood', 'NTUSER.DAT', 'ntuser.dat.L
OG1', 'ntuser.dat.LOG2', 'NTUSER.DAT{b773dd46-150b-11ed-b7b3-c8fff935bdec}.TM.blf', 'NTUSER.DAT{b773dd46-150b-11ed-b7b3-c8fff93
5bdec}.TMContainer00000000000000000001.regtrans-ms', 'NTUSER.DAT{b773dd46-150b-11ed-b7b3-c8fff935bdec}.TMContainer0000000000000
0000002.regtrans-ms', 'NTUSER.DAT{e5561952-2087-11ed-b7ea-30d042060f1d}.TxR.0.regtrans-ms', 'NTUSER.DAT{e5561952-2087-11ed-b7ea
-30d042060f1d}.TxR.1.regtrans-ms', 'NTUSER.DAT{e5561952-2087-11ed-b7ea-30d042060f1d}.TxR.2.regtrans-ms', 'NTUSER.DAT{e5561952-2
087-11ed-b7ea-30d042060f1d}.TxR.blf', 'NTUSER.DAT{e5561953-2087-11ed-b7ea-30d042060f1d}.TM.blf', 'NTUSER.DAT{e5561953-2087-11ed
-b7ea-30d042060f1d}.TMContainer00000000000000000001.regtrans-ms', 'NTUSER.DAT{e5561953-2087-11ed-b7ea-30d042060f1d}.TMContainer
00000000000000000002.regtrans-ms', 'ntuser.ini', 'numpy.ipynb', 'OneDrive', 'pandas.ipynb', 'Pictures', 'PrintHood', 'pubg prog
ramme.ipynb', 'Recent', 'reethu.ipynb', 'Saved Games', 'seaborn-data', 'Searches', 'SendTo', 'Start Menu', 'Templates', 'Video
s', 'wine and digits.ipynb', 'winrar-x64-611 (1).exe', 'Xilinx', 'xilinx14.7', 'xilinx14.7.rar']
```

Figure 4:*Libraries in python*

4.2 Basic Information from Dataset:



```
In [20]: data = pd.read_csv("heart.csv")

         type(data)
         data.shape

Out[20]: (1025, 14)

In [21]: data.head()

Out[21]:
```

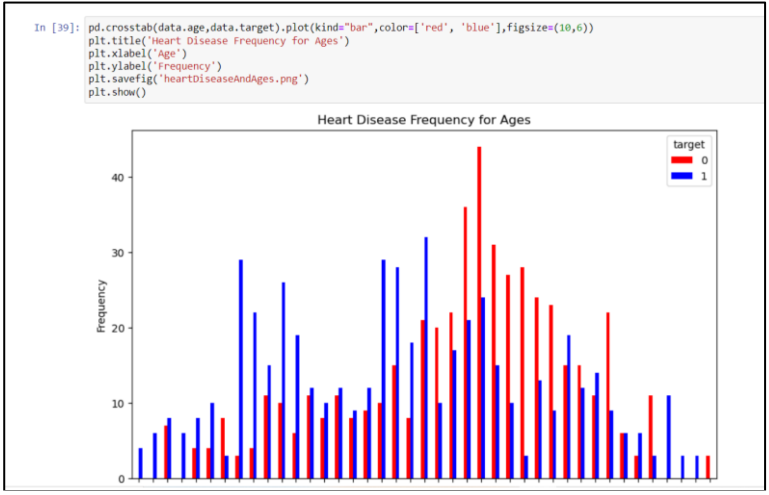
	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
0	52	1	0	125	212	0	1	168	0	1.0	2	2	3	0
1	53	1	0	140	203	1	0	155	1	3.1	0	0	3	0
2	70	1	0	145	174	0	1	125	1	2.6	0	0	3	0
3	61	1	0	148	203	0	1	161	0	0.0	2	1	3	0
4	62	0	0	138	294	1	1	106	0	1.9	1	3	2	0

```
In [22]: data.describe()

Out[22]:
```

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	
count	1025.000000	1025.000000	1025.000000	1025.000000	1025.00000	1025.000000	1025.000000	1025.000000	1025.000000	1025.000000	1025.000000	1025.00
mean	54.434146	0.695610	0.942439	131.611707	246.00000	0.149268	0.529756	149.114146	0.336585	1.071512	1.385366	0.75
std	9.072290	0.460373	1.029641	17.516718	51.59251	0.356527	0.527878	23.005724	0.472772	1.175053	0.617755	1.03
min	29.000000	0.000000	0.000000	94.000000	126.00000	0.000000	0.000000	71.000000	0.000000	0.000000	0.000000	0.00
25%	48.000000	0.000000	0.000000	120.000000	211.00000	0.000000	0.000000	132.000000	0.000000	0.000000	1.000000	0.00
50%	56.000000	1.000000	1.000000	130.000000	240.00000	0.000000	1.000000	152.000000	0.000000	0.800000	1.000000	0.00
75%	61.000000	1.000000	2.000000	140.000000	275.00000	0.000000	1.000000	166.000000	1.000000	1.800000	2.000000	1.00
max	77.000000	1.000000	3.000000	200.000000	564.00000	1.000000	2.000000	202.000000	1.000000	6.200000	2.000000	4.00

***Figure** 5:Dataset information*

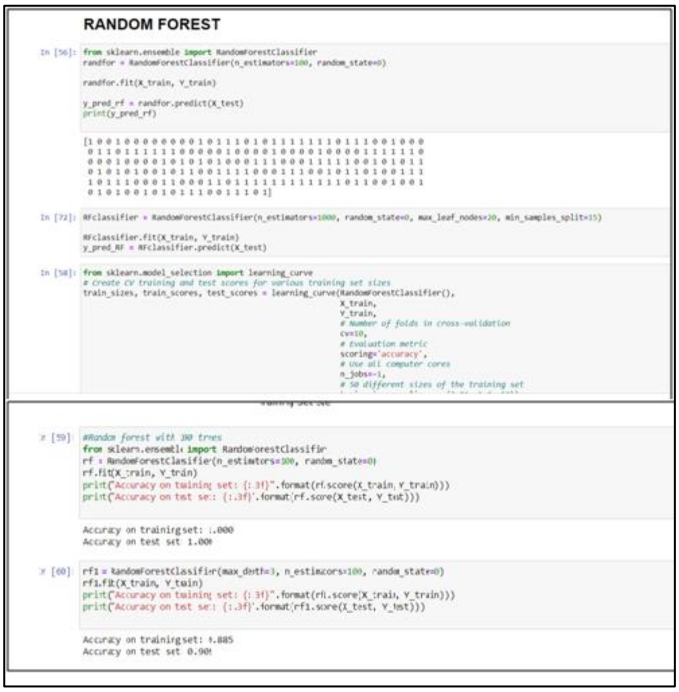
4.3 Heart Disease Frequency for Ages:



```
In [39]: pd.crosstab(data.age,data.target).plot(kind="bar",color=['red', 'blue'],figsize=(10,6))
         plt.title('Heart Disease Frequency for Ages')
         plt.xlabel('Age')
         plt.ylabel('Frequency')
         plt.savefig('heartDiseaseAndAges.png')
         plt.show()
```

***Figure 6:** Frequencies of Heart disease*

4.4 Random Forest:



RANDOM FOREST

```
In [56]: from sklearn.ensemble import RandomForestClassifier
         randfor = RandomForestClassifier(n_estimators=100, random_state=0)

         randfor.fit(X_train, Y_train)

         y_pred_rf = randfor.predict(X_test)
         print(y_pred_rf)

         [1 0 0 1 0 0 0 0 0 0 0 0 1 0 1 1 1 0 1 0 1 1 1 1 1 1 1 0 1 1 1 0 0 1 0 0 0
          0 1 1 0 1 1 1 1 1 1 0 0 0 0 0 1 0 0 0 0 1 0 0 0 0 1 0 0 0 0 1 1 1 1 1 1 0
          0 0 0 1 0 0 0 0 1 0 1 0 1 0 1 0 0 0 1 1 1 0 0 0 1 1 1 1 1 0 0 1 0 1 0 1 1
          0 1 0 1 0 1 0 0 1 0 1 1 0 0 1 1 1 1 0 0 0 1 1 1 0 0 1 0 1 1 0 1 0 0 1 1 1
          1 0 1 1 1 0 0 0 1 1 0 0 0 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 0 0 1 0 0 1
          0 1 0 1 0 0 1 0 1 0 1 1 1 0 0 1 1 1 0 1]

In [72]: RFclassifier = RandomForestClassifier(n_estimators=1000, random_state=0, max_leaf_nodes=20, min_samples_split=15)

         RFclassifier.fit(X_train, Y_train)
         y_pred_RF = RFclassifier.predict(X_test)

In [58]: from sklearn.model_selection import learning_curve
         # Create CV training and test scores for various training set sizes
         train_sizes, train_scores, test_scores = learning_curve(RandomForestClassifier(),
                                                                 X_train,
                                                                 Y_train,
                                                                 # Number of folds in cross-validation
                                                                 cv=10,
                                                                 # Evaluation metric
                                                                 scoring='accuracy',
                                                                 # Use all computer cores
                                                                 n_jobs=-1,
                                                                 # 50 different sizes of the training set

 [59]:   #Random forest with 100 trees
         from sklearn.ensemble import RandomForestClassifier
         rf = RandomForestClasifier(n_estimators=100, random_state=0)
         rf.fit(X_train, Y_train)
         print("Accuracy on training set: {:.3f}".format(rf.score(X_train, Y_train)))
         print("Accuracy on test set: {:.3f}".format(rf.score(X_test, Y_test)))

         Accuracy on training set: 1.000
         Accuracy on test set: 1.000

 [60]:   rf1 = RandomForestClassifier(max_depth=3, n_estimators=100, random_state=0)
         rf1.fit(X_train, Y_train)
         print("Accuracy on training set: {:.3f}".format(rf1.score(X_train, Y_train)))
         print("Accuracy on test set: {:.3f}".format(rf1.score(X_test, Y_test)))

         Accuracy on training set: 0.885
         Accuracy on test set: 0.908
```

Figure 7:*Code for Random Forest*

4.5 Decision Tree:

Pseudocode Decision Tree

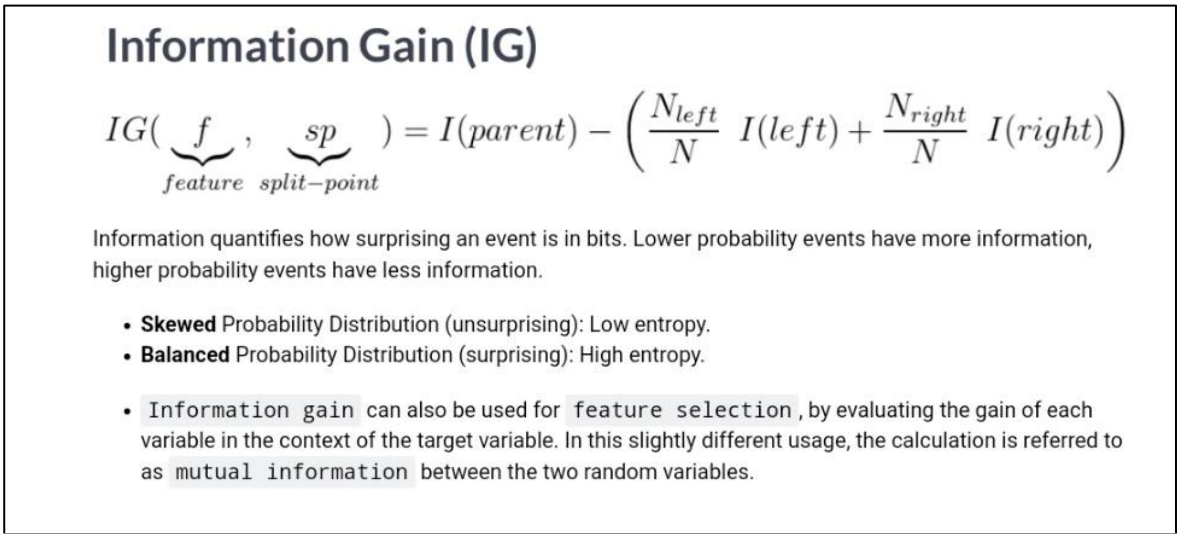
```
1. Put the dataset's best attribute at the top of the tree.

2. Make subsets of the training set. Subsets should be created in such a way that each subset has data with the same value for an attribute.
3. Repeat steps 1 and 2 for each subgroup until you locate leaf nodes in all of the tree's branches.
```

Decision Tree Assumptions:

- At first, the entire training set is regarded as the root.
- Categorical feature values are desired. If the values are continuous, they are discretized before the model is built.
- Recursively, records are dispersed based on attribute values.
- A statistical approach is used to place characteristics as the tree's root or internal node.

4.5.1 Information gain

It is the basic criterion which is used for splitting a node. It can be used with both continuous and discrete variables.



Information Gain (IG)

$$IG(\underbrace{f}_{feature}, \underbrace{sp}_{split-point}) = I(parent) - \left(\frac{N_{left}}{N} I(left) + \frac{N_{right}}{N} I(right)\right)$$

Information quantifies how surprising an event is in bits. Lower probability events have more information, higher probability events have less information.

- **Skewed** Probability Distribution (unsurprising): Low entropy.
- **Balanced** Probability Distribution (surprising): High entropy.

- `Information gain` can also be used for `feature selection`, by evaluating the gain of each variable in the context of the target variable. In this slightly different usage, the calculation is referred to as `mutual information` between the two random variables.

Figure 8: *Information Gain*

4.5.2 Gini index:

This is an effective measure of randomness in a dataset's values. It seeks to reduce contaminants in a tree model from its root nodes (at the top of the decision tree) to its leaf nodes (vertical branches down the decision tree). This is used to describe a country's inequality. For example, the greater the value of the index, the greater the inequality.

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

Figure 9*: Formula for Gini index*

Similarly using visualisation codes for the given data decision tree is being obtained

ADVANTAGES:

When it comes to classifying labelled data, decision trees are simple to make. It makes complex facts easier for us to interpret. Our project shows that there are several data imports concerning the disease's prediction in the modern world. Therefore, we can easily understand the data by using a decision tree. The process of cleansing data can be sped up and investigated. This can also be applied to relationships that are not linear.

DISADVANTAGES:

Even though decision trees have many benefits, one disadvantage is that they are unstable by nature. When new information is added, this becomes complicated. Compared to other classifications, it takes a while. When the outcome is a continuous variable, this is less effective.

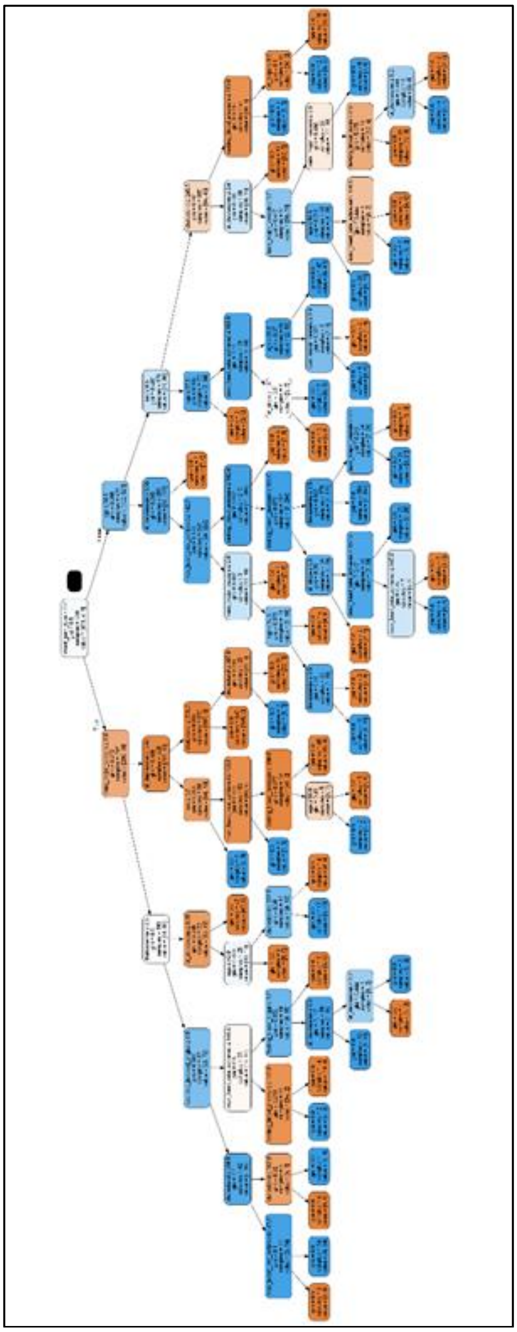


Figure 8 : *Pseudocode for Decision tree*

5.0 CONCLUSION:

We covered decision trees and random forests, two supervised learning techniques utilised for both classification and regression, in the aforementioned academic study. This project's goal is to divide the provided GitHub data set, mostly using a set of guidelines and constraints. While the trees are growing, the random forest adds more randomness to the model. It looks for the best feature among a selection of randomly selected features. We have also studied the operation of a decision tree utilising well-known algorithms like GINI, information gain, etc. Last but not least, we also covered the benefits and drawbacks of decision trees and random forests.

6.0 REFERENCES:

- Noble, W. S. Support vector machine applications in computational biology. *Kernel Methods Comput. Biol.* **71**, 92 (2004).
- Aruna, S. & Rajagopalan, S. A novel SVM based CSSFFS feature selection algorithm for detecting breast cancer. *Int. J. Comput. Appl.* **31**, 20 (2011).

- Lakhani, P. & Sundaram, B. Deep learning at chest radiography: Automated classification of pulmonary tuberculosis by using convolutional neural networks. *Radiology* **284**, 574–582 (2017).
- Yasaka, K. & Akai, H. Deep learning with convolutional neural network for differentiation of liver masses at dynamic contrast-enhanced CT: A preliminary study. *Radiology* **286**, 887–896 (2018).
- Christ, P. F. *et al.* *Automatic Liver and Lesion Segmentation in CT Using Cascaded Fully Convolutional Neural Networks and 3D Conditional Random Fields. International Conference on Medical Image Computing and Computer-Assisted Intervention* 415–423 (Springer, Berlin, 2016).
- Krittanawong, C. *et al.* Deep learning for cardiovascular medicine: A practical primer. *Eur. Heart J.* **40**, 2058–2073 (2019).
- Krittanawong, C., Zhang, H., Wang, Z., Aydar, M. & Kitai, T. Artificial intelligence in precision cardiovascular medicine. *J. Am. Coll. Cardiol.* **69**, 2657–2664 (2017).
- Krittanawong, C. *et al.* Future direction for using artificial intelligence to predict and manage hypertension. *Curr. Hypertens. Rep.* **20**, 75 (2018).
- Rutter, C. M. & Gatsonis, C. A. A hierarchical regression approach to meta-analysis of diagnostic test accuracy evaluations. *Stat. Med.* **20**, 2865–2884 (2001).
- Stroup, D. F. *et al.* Meta-analysis of observational studies in epidemiology: A proposal for reporting. Meta-analysis Of Observational Studies in Epidemiology (MOOSE) group. *JAMA* **283**, 2008–2012 (2000).
- Goff, D. C. Jr. *et al.* 2013 ACC/AHA guideline on the assessment of cardiovascular risk: A report of the American College of Cardiology/American Heart Association Task Force on Practice Guidelines. *J. Am. Coll. Cardiol.* **63**, 2935–2959 (2014).
- <https://github.com/topics/heart-disease-prediction>
- <https://www.javatpoint.com/machine-learning-decision-tree-classification-algorithm>
- <https://www.javatpoint.com/machine-learning-random-forest-algorithm>
- <https://www.kdnuggets.com/2020/01/decision-tree-algorithm-explained.html>
- *<https://images.app.goo.gl/a2AoPU9KYmyN2PtP9>*
- *<https://images.app.goo.gl/a2AoPU9KYmyN2PtP9>*
- *<https://www.news-medical.net/health/Structure-and-Function-of-the-Heart.aspx>*