Mithilesh singh

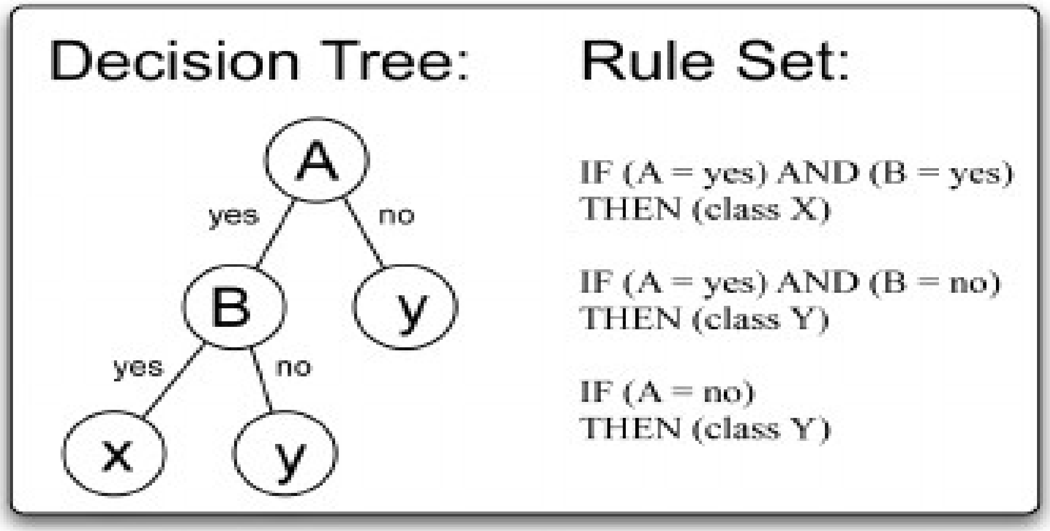
Decision Tree Algorithm

Decision Tree is one of the most popular algorithms in machine learning. It is relatively simple, yet able to produce good accuracy. But the important is the interpretability of decision.

→ DT is **Non-Parametric** Supervised Machine Learning Algorithm

→It uses a set of rules to make decisions, similarly to how humans make decisions.

Nested if-else condition .



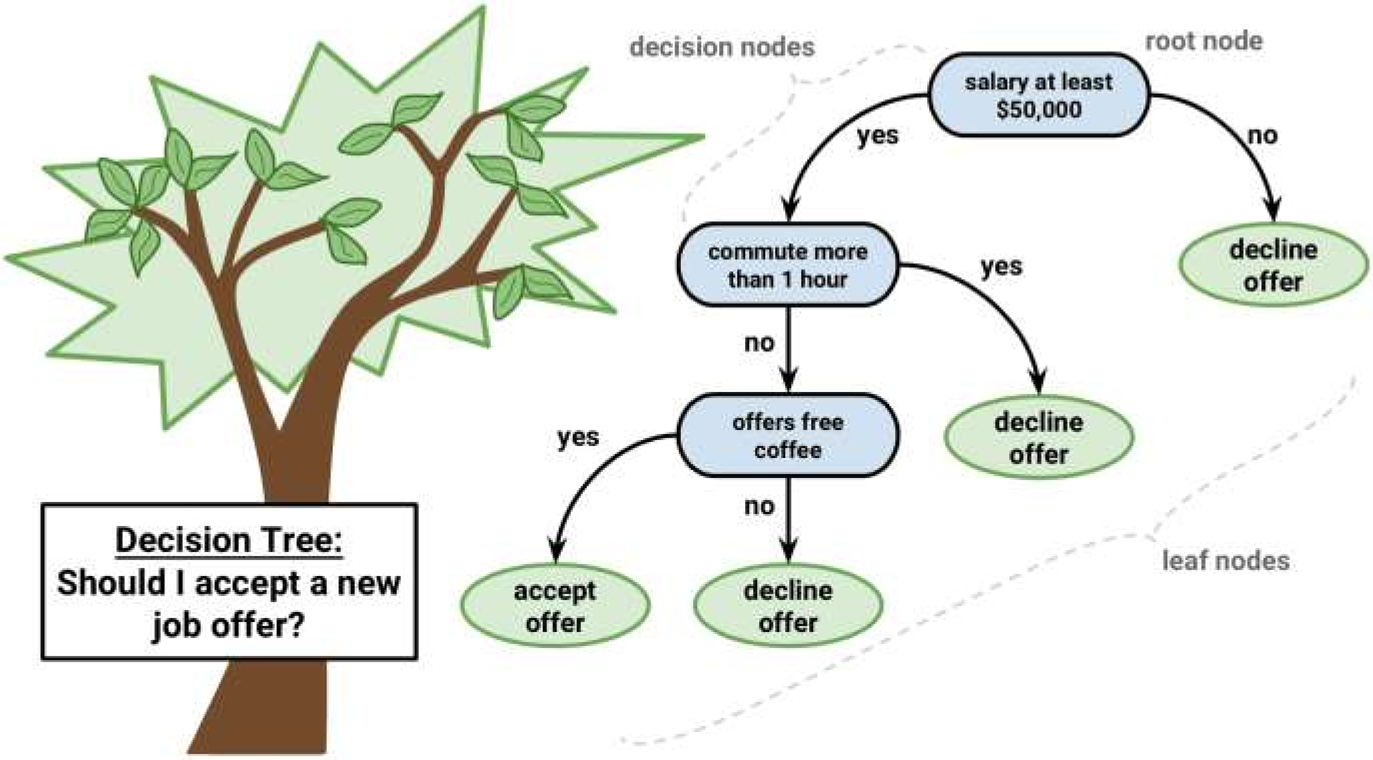
IF (A = yes) AND (B = yes)
THEN (class X)

IF (A = yes) AND (B = no)
THEN (class Y)

IF (A = no)
THEN (class Y)

→ A rule is a conditional statement that can easily be understood by humans and used within a database to identify a set of records.

Decision trees can perform both classification and regression tasks.



Decision tree based on the nested if-else classifier.

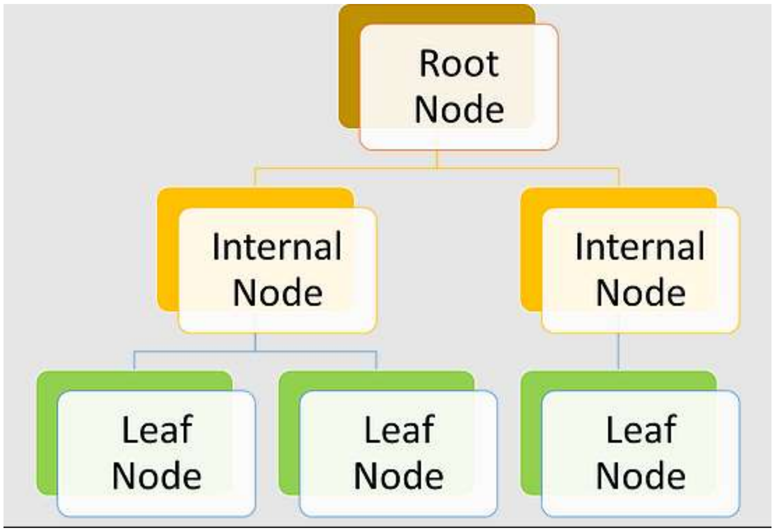
In general, a decision tree takes a statement or hypothesis or condition and then makes a decision on whether the condition holds or does not.

The conditions are shown along the branches and the outcome of the condition, as applied to the target variable.

Terminology:

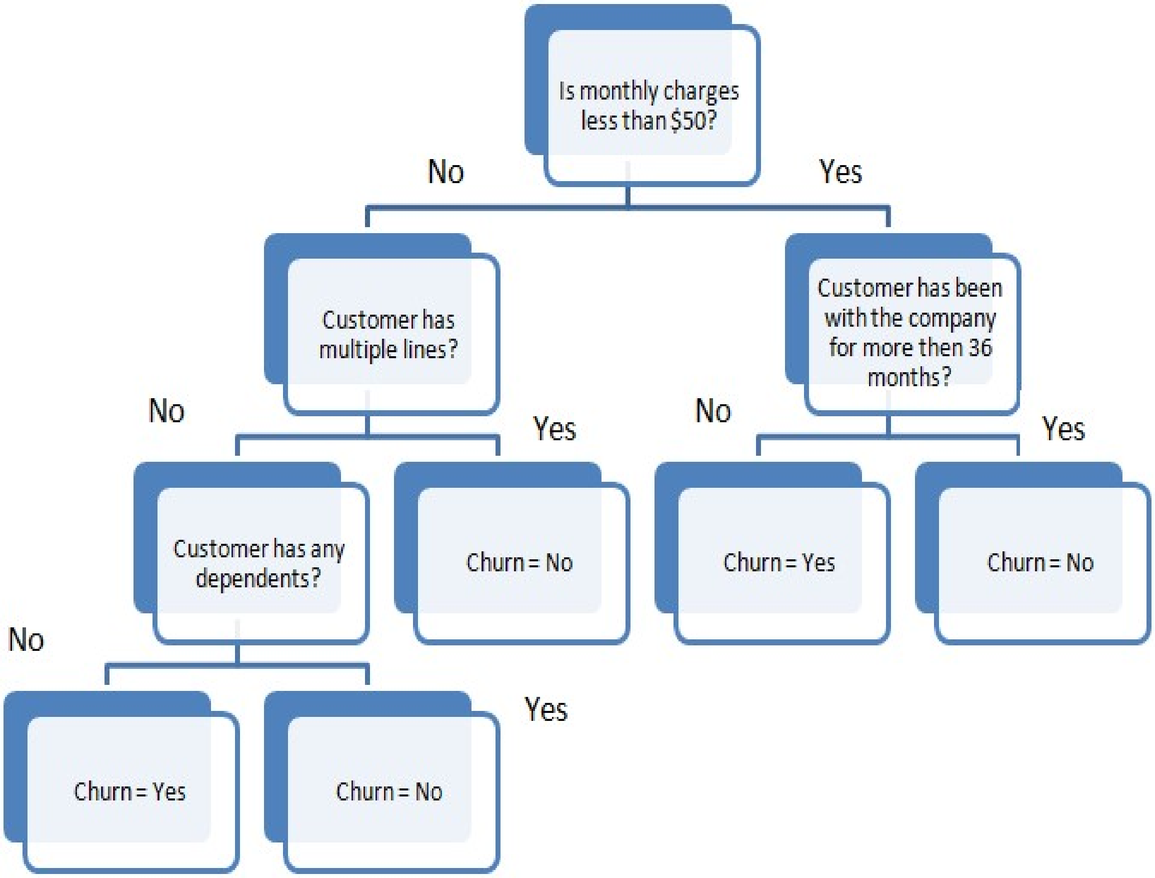
1. **Root Node:** It represents entire population or sample and this further gets divided into two or more homogeneous sets.
2. **Splitting:** It is a process of dividing a node into two or more sub-nodes.
3. **Decision Node:** When a sub-node splits into further sub-nodes, then it is called decision node. This ultimately leads to a prediction.
4. **Leaf/ Terminal Node:** Nodes (no further split) is called Leaf or Terminal node.
5. **Pruning:** When we reduce the size of decision trees by removing nodes (opposite of Splitting), the process is called pruning.(a)Post Pruning (b)Pre Pruning.
6. **Branch / Sub-Tree:** A sub section of decision tree is called branch or sub-tree.

DT start question from root node then move through the tree branches according to which groups it belong to until it reach a leaf node.



Variable selection criterion

Example of telecommunication customer churning: -


Is monthly charges less than $50?
No
Yes
Customer has multiple lines?
Customer has been with the company for more then 36 months?
No
Yes
No
Yes
Customer has any dependents?
Churn = No
Churn = Yes
Churn = No
No
Churn = Yes
Churn = No
Yes

Example of a decision tree

Mathematics behind the selection criteria of Root node (stamp process where the split will happen). node and finally Leaf Node.

If the decision tree build is appropriate, then the depth of the tree will be less or else the depth will be more. To build the decision tree in an efficient way we use the concept of Gini or **Entropy**.

variable selection criterion in Decision Trees can be done via two popular attribute selection approaches: -

1. Gini Index or Gini Impurity
2. Entropy or Information Gain

Entropy:- Purity and impurity in a junction are the primary focus of the Entropy and Information Gain framework.

Entropy is the degree of uncertainty, impurity or disorder of a random variable, or a measure of purity. It characterizes the impurity of an arbitrary class of examples.

Entropy is the measurement of impurities or randomness in the data points.
Here, if all elements belong to a single class, then it is termed as “Pure”, and if not then the distribution is named as “Impurity”.

It is computed between 0 and 1, however, heavily relying on the number of groups or classes present in the data set it can be more than 1 while depicting the same significance i.e. extreme level of disorder.

In more simple terms, If a dataset contains homogeneous subsets of observations, then no impurity or randomness is there in the dataset, and if all the observations belong to one class, the entropy of that dataset becomes zero.

Information Gain?

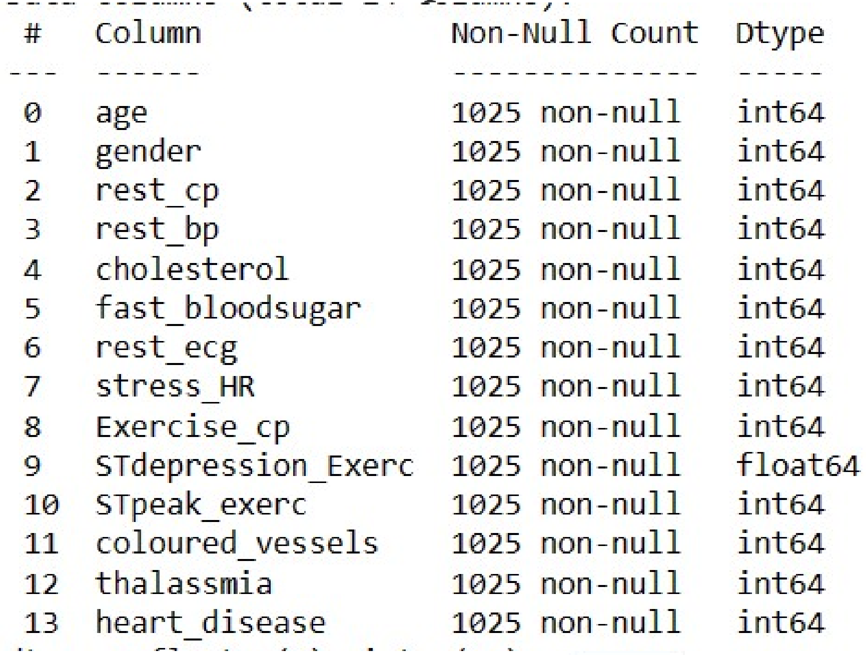
The concept of entropy plays an important role in measuring the information gain. However, “Information gain is based on the information theory”.
Information gain is used for determining the best features/attributes that render maximum information about a class. It follows the concept of entropy while aiming at decreasing the level of entropy, beginning from the root node to the leaf nodes.

Information gain computes the difference between entropy before and after split and specifies the impurity in class elements.

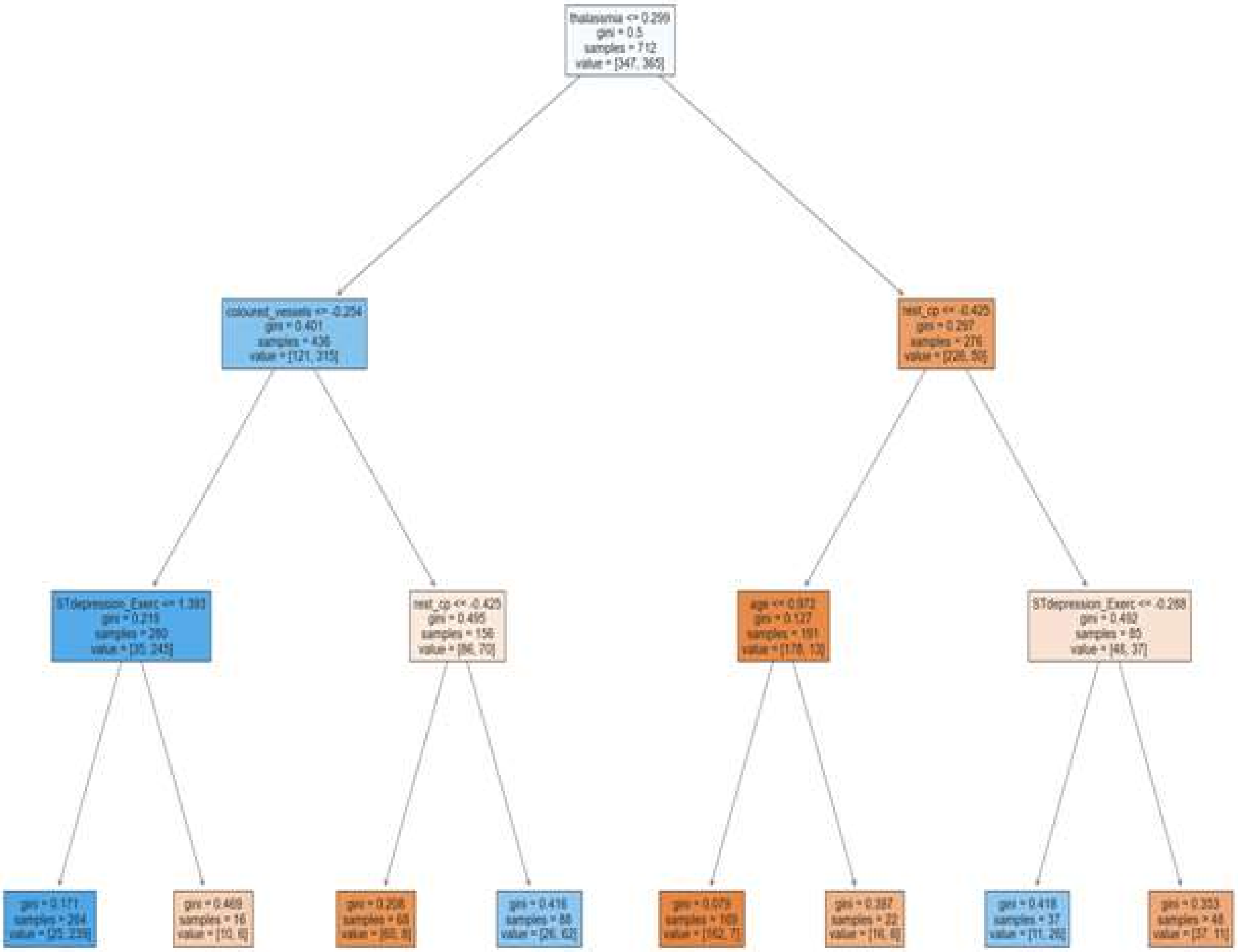
Information Gain = Entropy before splitting - Entropy after splitting

Given a probability distribution such that

Cardio_vascular disease prediction:-



```
 #   Column               Non-Null Count  Dtype  
---  ------               --------------  -----  
 0   age                  1025 non-null   int64  
 1   gender               1025 non-null   int64  
 2   rest_cp              1025 non-null   int64  
 3   rest_bp              1025 non-null   int64  
 4   cholesterol          1025 non-null   int64  
 5   fast_bloodsugar      1025 non-null   int64  
 6   rest_ecg             1025 non-null   int64  
 7   stress_HR            1025 non-null   int64  
 8   Exercise_cp          1025 non-null   int64  
 9   STdepression_Exerc   1025 non-null   float64
 10  STpeak_exerc         1025 non-null   int64  
 11  coloured_vessels     1025 non-null   int64  
 12  thalassmia           1025 non-null   int64  
 13  heart_disease        1025 non-null   int64  
```



Gini index / Gini impurity:- The Gini Index, also known as Impurity, calculates the likelihood that somehow a randomly picked instance would be erroneously cataloged.

Gini index is measure of inequality in sample. It’s value between 0 and 1. GI= 0 means sample are perfectly homogeneous and all element are similar, whereas, GI= 1 means maximal inequality among elements.

GI is sum of the square of the probabilities of each class.

$$Gini\,index = 1 - \sum_{i=1}^{n} p_i^2$$

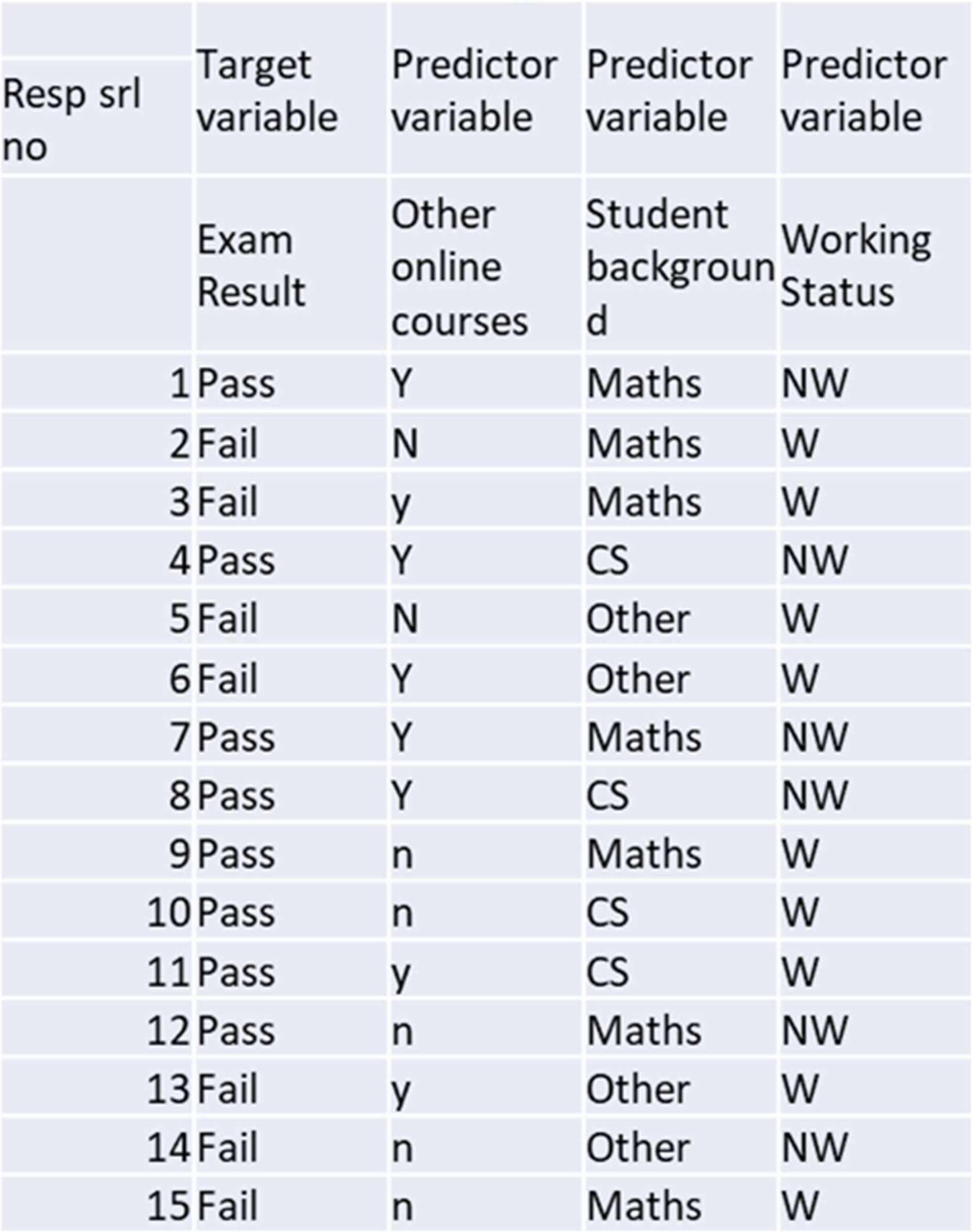
i is number of classes

The Gini Index or Impurity measures the probability for a random instance being misclassified when chosen randomly.

The Entropy and Information Gain method focuses on purity and impurity in a node.

Solving problem prediction of student pass fail in exam Calculation of Gini Index: -

The dataset of student pass fail in exam:



Resp srl no	Target variable	Predictor variable	Predictor variable	Predictor variable
	Exam Result	Other online courses	Student backgroun d	Working Status
1	Pass	Y	Maths	NW
2	Fail	N	Maths	W
3	Fail	y	Maths	W
4	Pass	Y	CS	NW
5	Fail	N	Other	W
6	Fail	Y	Other	W
7	Pass	Y	Maths	NW
8	Pass	Y	CS	NW
9	Pass	n	Maths	W
10	Pass	n	CS	W
11	Pass	y	CS	W
12	Pass	n	Maths	NW
13	Fail	y	Other	W
14	Fail	n	Other	NW
15	Fail	n	Maths	W

Gini Index-

The Gini Index or Impurity measures the probability for a random instance being misclassified when chosen randomly.

The lower the Gini Index, the better the lower the likelihood of misclassification.

Gini Impurity of a dataset is a number **between 0-0.5**, which indicates the likelihood of new, random data being misclassified.

The Gini Index or Gini Impurity is calculated by subtracting the sum of the squared probabilities of each class from one.

In simple terms, it calculates the probability of a certain randomly selected feature that was classified incorrectly.

The formula for Gini Index/impurity

$$Gini = 1 - \sum_{i=1}^{C} (p_i)^2$$

P(i) represents the ratio of Pass/Total no. of observations in node.

Calculate the Gini index for Student Background attribute.

There are three Sub nodes :- Maths, CS, Others.

Gini formula requires us to calculate each sub node.

Then do a weighted average to calculate the overall Gini Index for the Background node.

Maths sub node: 4 Pass, 3 Fail

$$Gini_{maths} = 1 - \left(\frac{4}{7}\right)^2 - \left(\frac{3}{7}\right)^2 = .4897$$

CS sub node: 4Pass, 0 Fail

$$Gini_{CS} = 1 - \left(\frac{4}{4}\right)^2 - \left(\frac{0}{5}\right)^2 = 0$$

Others sub node: 0Pass, 4 Fail

$$Gini_{others} = 1 - \left(\frac{0}{4}\right)^2 - \left(\frac{4}{4}\right)^2 = 0$$

The overall Gini Index for Background:-

$$Gini_{bkgrd} = \frac{7}{15} * .4897 + \frac{4}{15} * 0 + \frac{4}{15} * 0 = \ .2286$$

Gini Index for Working Status :-

$$Gini_{working} = 1 - \left(\frac{6}{9}\right)^2 - \left(\frac{3}{9}\right)^2 = .44$$

$$Gini_{notworking} = 1 - \left(\frac{5}{6}\right)^2 - (6)^2 = .278$$

$$Gini_{workstatus} = \frac{9}{15} * .44 + \frac{6}{15}.278 = \ .378$$

Gini Index for Other Online Courses.

$$Gini_{online} = 1 - \left(\frac{5}{8}\right)^2 - \left(\frac{3}{8}\right)^2 = .4688$$

$$Gini_{notonline} = 1 - \left(\frac{3}{7}\right)^2 - \left(\frac{4}{7}\right)^2 = .4898$$

$$Gini_{\textbf{Other Online}} = \frac{8}{15} * .4688 + \frac{7}{15}.4898 = \ .479$$

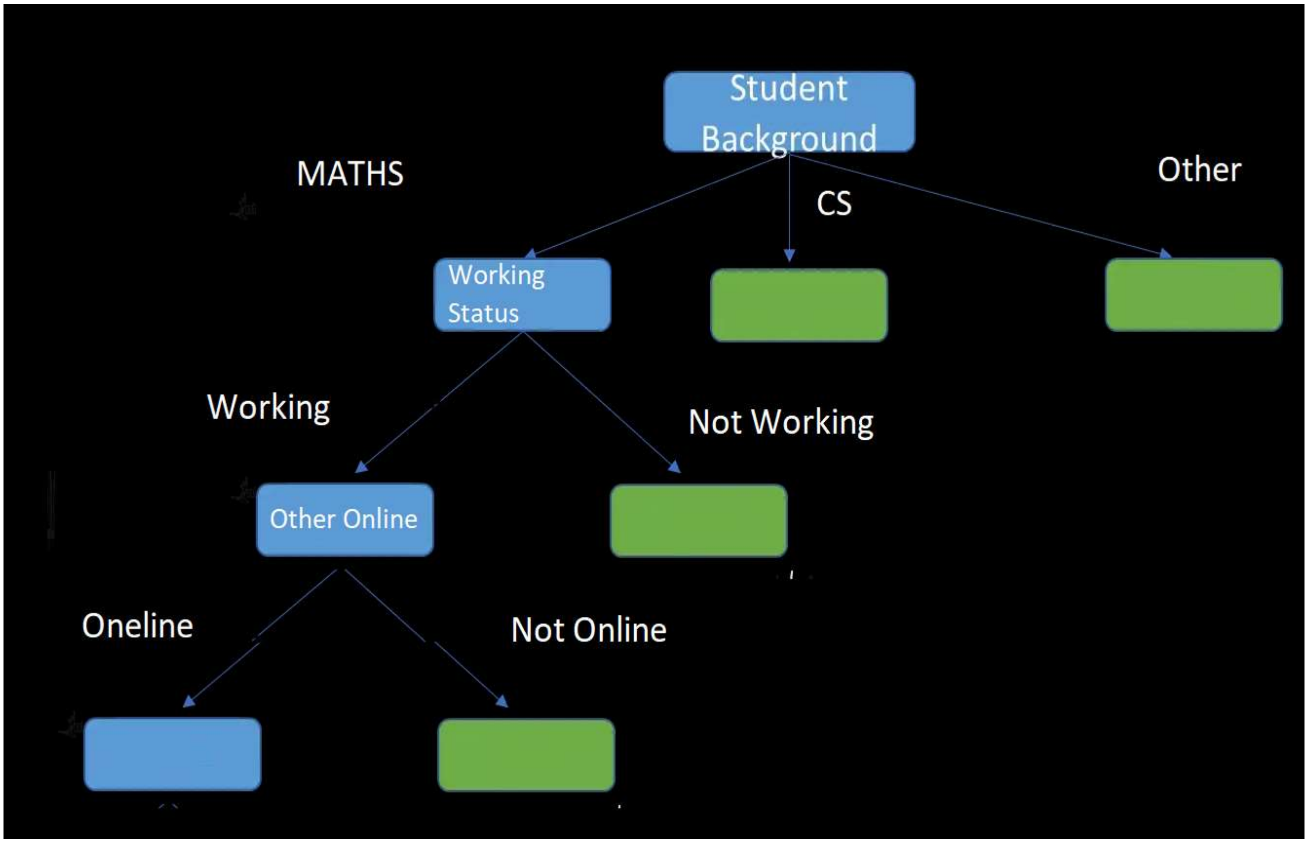
Final Gini Values of different Nodes:-

$$Gini_{bkgrd} = \frac{7}{15} * .4897 + \frac{4}{15} * 0 + \frac{4}{15} * 0 = .2286$$

$$Gini_{workstatus} = \frac{9}{15} * .44 + \frac{6}{15} . 278 = .378$$

$$Gini_{\textbf{Other Online}} = \frac{8}{15} * .4688 + \frac{7}{15} . 4898 = .479$$

The Gini Index is lowest for the Student Background variable.



Both criteria are broadly similar and seek to determine which variable would split the data to lead to the underlying child nodes being most homogenous or pure.

Entropy

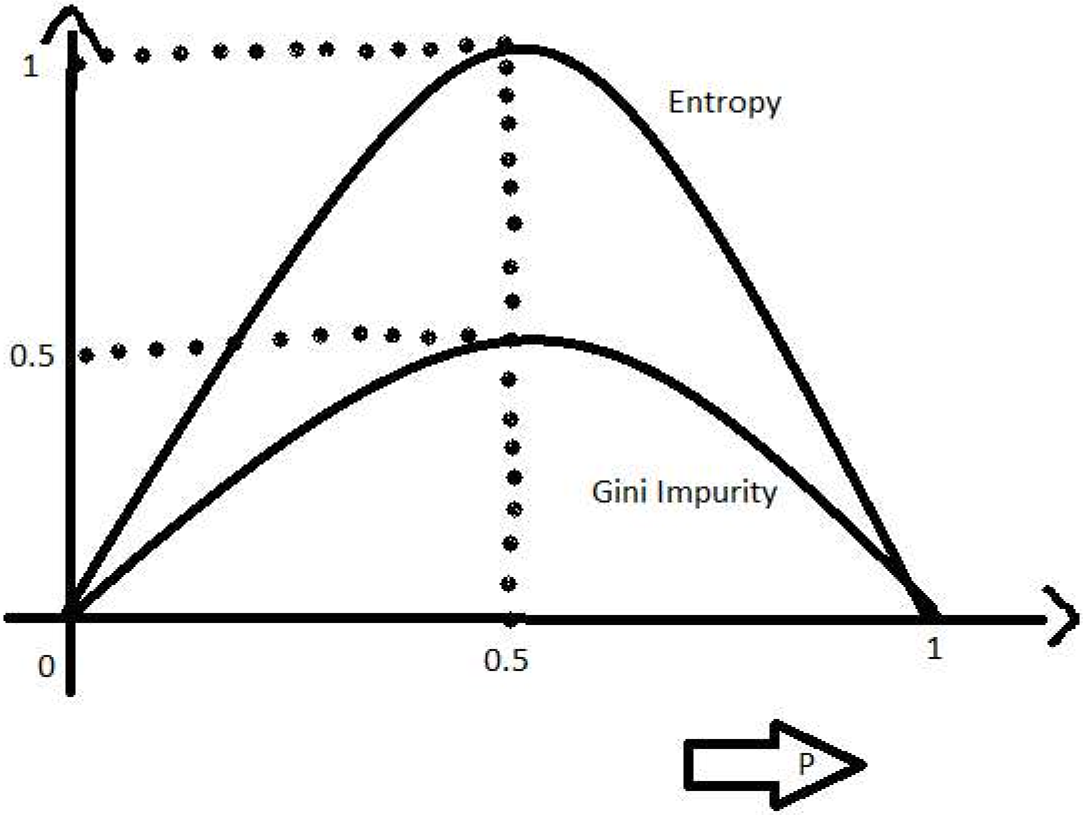
Entropy is amount of information is needed to accurately describe some sample. So if sample is homogeneous, means all the element are similar than Entropy is 0, else if sample is equally divided than entropy is maximum 1.

Mathematically it is written as ,

$$E = -\sum_{i=1}^{n} p_i \, log_2(p_i)$$

Where the pi is the probability of randomly selecting an example in class i.

Entropy v/s Gini Impurity:



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

$$Gini(E) = 1 - \sum_{j=1}^{c} p_j^2$$

The internal working of both methods is very similar and both are used for computing the feature/split after every new splitting.

Gini Impurity is more efficient than entropy in terms of computing power.

Pl refer the graph for Gini and entropy,

Entropy first increases up to 1 and then starts decreasing, but in the case of Gini impurity it only goes up to 0.5 and then it starts decreasing, hence Gini requires less computational power.

The range of Entropy lies in between 0 to 1 and the range of Gini Impurity lies between 0 to 0.5.

```
from sklearn.tree import DecisionTreeClassifier# classification
#from sklearn.tree import DecisionTreeRegressor# Regression
dt=DecisionTreeClassifier(criterion="entropy")
# criterion=entropy/gini
dt.fit(X_train1,Y_train1)
Y_pred_dt_train=dt.predict(X_train1)
Y_pred_dt_test=dt.predict(X_test1)
Print(accuracy_score(Y_train1,Y_pred_dt_train))
print("######"*20)
print(accuracy_score(Y_test1,Y_pred_dt_test))
```

Overfitting and Decision Trees

Overfitting can be a big challenge with Decision Trees.

There are several ways we can prevent the decision tree from becoming too unwieldy: -

1. Pre pruning or Early stopping: Preventing the tree from growing too big or deep
2. Post Pruning: Allowing a tree to grow to its full depth and then getting rid of various branches based on various criteria

3. Ensembling or using averages of multiple models such as Random Forest

Pruning

Pre pruning

The pre-pruning technique refers to the early stopping of the growth of the decision tree. The pre-pruning technique involves tuning the hyperparameters of the decision tree model prior to the training.

The hyperparameters

max_depth, min_samples_leaf, min_samples_split

We can set max_depth or min_samples parameters in order not to overcomplicate the tree.

which can be tuned to early stop the growth of the tree and prevent the model from overfitting.

The best way is to use the RandomSearchCV or GridSearchCV technique to find the best set of hyperparameters.

A challenge with the early stopping may prevent some more fruitful splits down the line.

Post Pruning

In this technique, we allow the tree to grow to its maximum depth. Then we remove parts of the tree to prevent overfitting.

A good method to avoid overfitting is pruning — it removes insignificant nodes.

Advantages and Disadvantages of Trees Decision trees

1. Trees give a visual schema of the relationship More explainable. The hierarchy of the tree provides insight into variable importance.

2. White box model which is explainable. This is in contrast to black box models such as neural networks.

Disadvantages

1. Prone to overfitting and hence lower predictive accuracy

2. Can be non-robust, i.e., a small change in the data can cause a large change in the final estimated tree

3. Decision tree learners create biased trees if some classes dominate. It is required to balance the dataset prior to fitting with the decision tree.