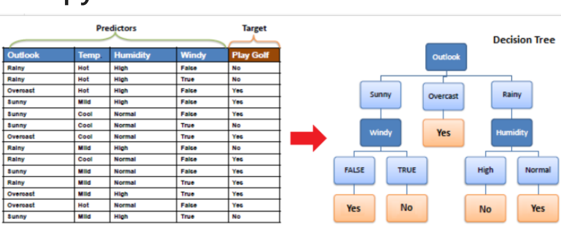
Entropy In Decision Tree

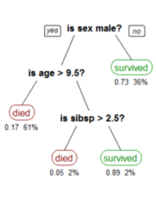


- In the decision tree classification the the classification is done like tree structure.Where the top feature is called the root and and then it will start spliting on the basis of information and convert into the leaf node at the end or final class label.
- But the Problem is that how to decide the feature for the root node for the spliting the node in recurssive manner in downword direction so that we will get good accuracy or good classification.

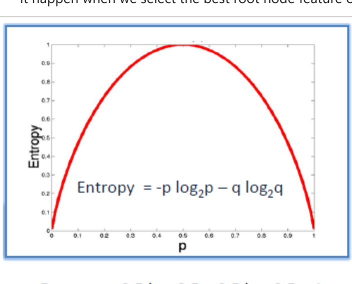
ID3 Algorithm

- In this algorithm we use the `entropy as measure parameter` in order to get the best root attribute or feature that so will get the high informative feature at the root and and then we are getting leaf attributes as spliting goes on increasing.
- It is very important to get the best attribuite at the root node.If the root node doses not conatin the high information that's means the spliting will not going to long.
- If the root node doses conatains the high information that's means the spliting will going to long and increae the depth of decision tree may lead to overfit the model.

Entropy help us to measure the purity of the split.



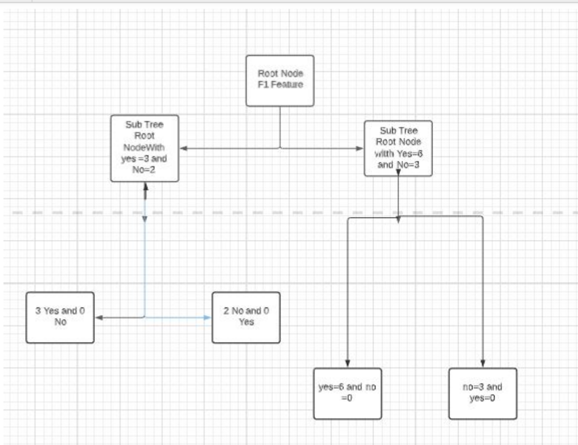
- Decision tree stop spliting itself when we got the at the leaf node with only one type of class instead of more than class probablity and it happen when we select the best root node feature out of the other.



$$\text{Entropy} = -0.5 \log_2 0.5 - 0.5 \log_2 0.5 = 1$$

- Entropy is measure of purity in the classification or measure of randomness inside the attribute.
- From the above chart we are able to see the we are consider the binary classification in that classification `p` is the one class perobablity and `q` is the second class probanlity.
- **Higher the entropy of any attribute that time the draw the conclusion from that infomation very harder.**
- Entropy is max at the 0.5 probablity of both attribute that time decision tree is showing the highest entropy i.e.1 that's indicate the it is hard to classify or not diffrentiable.
- Entropy ranges from the 0 to 1.
- If the entropy Closer to zero and 1 bit that is indicates the good and away from the zero bit that means the probablity of two distinct class are getting complex and harder to classify.

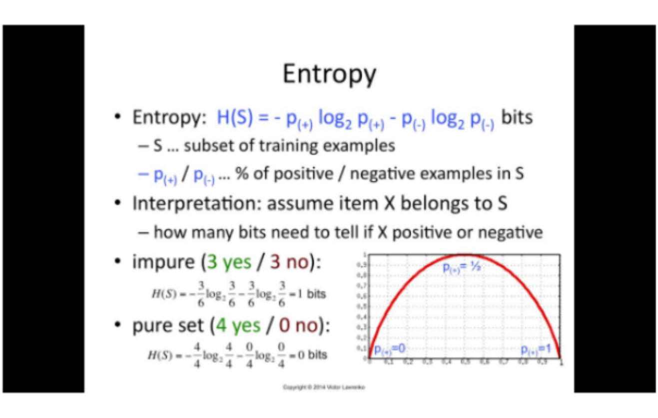
Entropy Formulae



- For getting best paramter in decision tree we always need to calculate the entropy at each node.
- In the above Decision Tree while spliting the information from root node to sub tree root node at that time we have two classes are avalible as an output that is `yes=3` and `no=2.`
- Whereas,The second right hand side we have one more sub-tree root node in that `yes decision are 6 times` and `no decision 3 times avalible.`
- Now decision tree need to take the decision of after the main root node which brach is suitable to split further that is always been calculated by entropy measure or purity of splitness.
- In order to take the decision we have the formulae to calculate entropy of each node for that :-

Formulae For The Entropy

For the Binary Classification ¶



- Here,we have just the `(+)` and `(-)` for diffrentiating two class probablity.
- Here `p` is the `percentage` for one class that is `Yes`
- `q` is the `percentage` for second class that is `NO`.

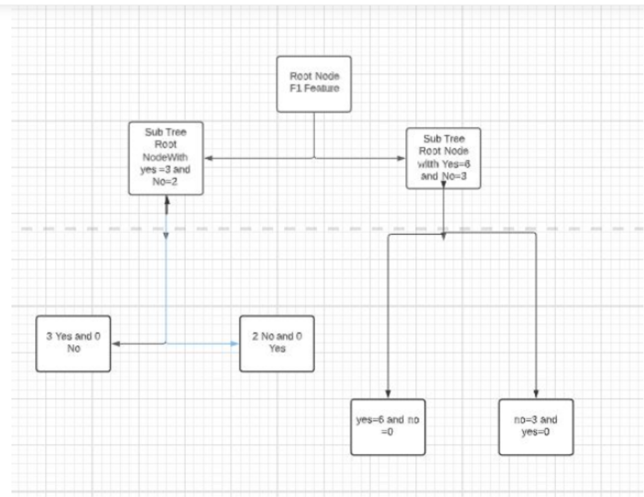
Entropy: Formulas

- Formulas for computing entropy:

$$\text{entropy}(p_1, p_2, \ldots, p_n) = -p_1 \log p_1 - p_2 \log p_2 \ldots - p_n \log p_n$$

$$H(X) = -\sum_x p(x) \log p(x)$$

Entropy Calculation



Entropy: Formulas

- Formulas for computing entropy:

$$\text{entropy}(p_1, p_2, \ldots, p_n) = -p_1 \log p_1 - p_2 \log p_2 \ldots - p_n \log p_n$$

$$H(X) = -\sum_x p(x) \log p(x)$$

- `X` stands for the subset of training .
- `Entropy of the left hand sub-node = -(3/5)*log2(3/5)-(2/5)*log2(2/5)`
- `Entropy of the left hand sub-node = 0.78 bits`
- `Entropy of the right hand sub-node = -(6/9)*log2(6/9)-(3/9)*log2(3/9)`
- `Entropy of the right hand sub-node = 0.57 bits`
- Since From the above the entropy of the right hand side is lesser as compare to the left hand side sub tree.
- From that we decision tree will be prefer the right hand side sub node for the further spliting because It contains the lesser entropy.

Information Gain

Information gain is the reduction in entropy or surprise by transforming a dataset and is often used in training decision trees. Information gain is calculated by comparing the entropy of the dataset before and after a transformation.

The information gain is based on the decrease in entropy after a dataset is split on an attribute. Constructing a decision tree is all about finding attribute that returns the highest information gain (i.e., the most homogeneous branches). ... The result is the Information Gain, or decrease in entropy.

Information gain is the amount of information that's gained by knowing the value of the attribute, which is the entropy of the distribution before the split minus the entropy of the distribution after it.

Formulae For Information Gain.

Information Gain

$$IG(D_p, f) = I(D_p) - \frac{N_{left}}{N} I(D_{left}) - \frac{N_{right}}{N} I(D_{right})$$

f: feature split on
D_p: dataset of the parent node
D_{left}: dataset of the left child node
D_{right}: dataset of the right child node
I: impurity criterion (Gini Index or Entropy)
N: total number of samples
N_{left}: number of samples at left child node
N_{right}: number of samples at right child node

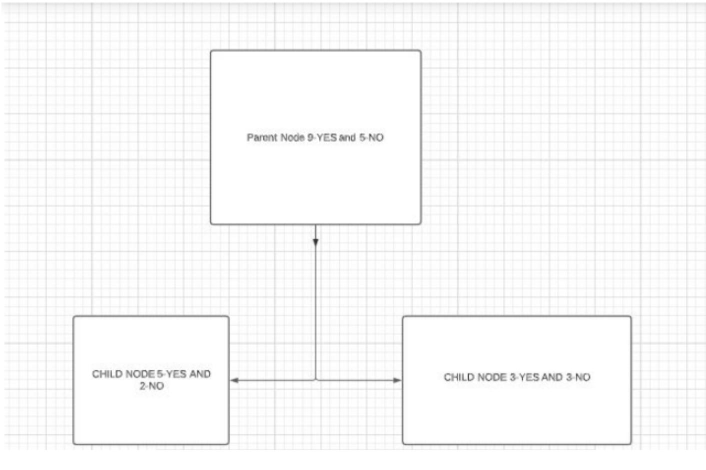
- `IG(Datasets,particular_feature) =` Entropy of that particular or parrent feature - Σtotal entropy of dataset after the spliting.
- `Avg_IG(Datasets,particular_feature) =` Entropy of that particular feature or parrent feature - Σtotal entropy in dataset./N`
- `Here the N =` Number of sample in atttribute in parrent class like 5 yes and 2 no that is 7
- `Here the Nleft =` Number of sample in atttribute in left child class class like 2 yes and 1 no that is 3
- `Here the Nright=` Number of sample in atttribute in right child class class like 1 yes and 1 no that is 2

$$Gain(S, A) = \underbrace{Entropy(S)}_{\text{original entropy of S}} - \underbrace{\sum_{v \in values(A)} \frac{|S_v|}{|S|} \cdot Entropy(S_v)}_{\text{relative entropy of S}}$$

- S = Sub set of data.
- A = Parrent attribute
- S = Entropy of root node subset
- After summation of taken the entropy after the spliting.

Why do We required IG `(Information Gain)` ?

Example



- In the every decision tree we need highest informative attribute as root node.But how decide that root attroibute we required a measure that is called as `Information Gain.`
- It takes the all feature as root node check the information gain then only decision Tree select the required feature at the root position.

Let's calculate the infomation gain in the respect of parent node.

`Entropy_of_root_node =` `-(9/14)*log2(9/14)-(5/14)*log2(5/14) = 0.94 bit`

`summation Entropy_of_all node expect the parrent node.=` `-(7/14)*entropy(leaf root) - (6/14)*entropy(right root)= - .495 -.4285   = -0.9235`

`Information_gain=` `0.91- .495-.4285  = 0.0135 bit`

- This is how decision Tree always check the information gain for each attribute and consider the one attribute or which is giving the highest Information gain and algorithm consider that as root attribute because it contains the infomation aftre the spliting.

GiNi-Impurity In Decision Tree.

- Gini Impurity is a measurement of the likelihood of an incorrect classification of a new instance of a random variable.
- If that new instance were randomly classified according to the distribution of class labels from the data set.
- Gini impurity is lower bounded by 0, with 0 occurring if the data set contains only one class.

Gini impurity [edit]

Not to be confused with Gini coefficient.

Used by the CART (classification and regression tree) algorithm, Gini impurity is a measure of how often a randomly chosen element from the set would be incorrectly labeled if it were randomly labeled according to the distribution of labels in the subset. Gini impurity can be computed by summing the probability of each item being chosen times the probability of a mistake in categorizing that item. It reaches its minimum (zero) when all cases in the node fall into a single target category.

Decision Trees

How is Gini Impurity Calculated?

The formula for finding the gini impurity score of a particular level is :

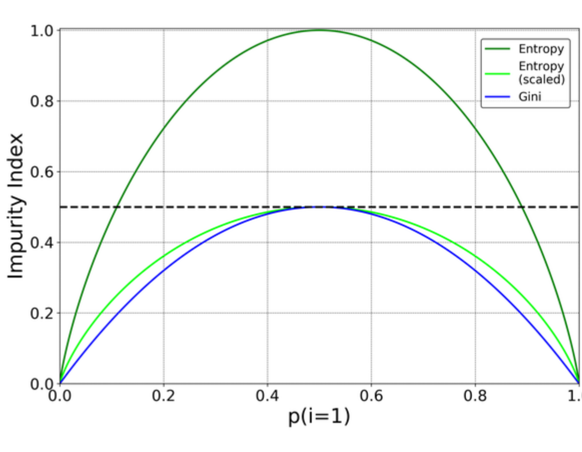
$$Gini = 1 - \sum_{i=1}^{C} (p_i)^2$$

Here p_i is the ratio of class i in the node whose gini index is being calculated.

There are total of c classes.

Why We required GINI-Index In Decision Tree.

- Whenever the we found the yes and no are same attribute avalible during the training in one particular node that node will give us the High entropy as 1.That will not split further.
- Gini Index: It is calculated by subtracting the sum of squared probabilities of each class from one. It favors larger partitions and easy to implement whereas information gain favors smaller partitions with distinct values. ... The classic CART algorithm uses the Gini Index for constructing the decision tree.
- Gini index or Gini impurity measures the degree or probability of a particular variable being wrongly classified when it is randomly chosen. But what is actually meant by 'impurity'? If all the elements belong to a single class, then it can be called pure.
- In other side the gini impurity won't give us the 1.It will gives us the 0.5 as highest gini index.
- `Gini-Index = 1 -  Σ (pi)2`
- If we have one particular node in that yes=3 and no =3 that time,here the p is the percentile
- `GINI-INDEX = 1-Σ([P_yes2 + P_No2])`
- `Gini_idex  = 1- [(3/6)2 + (3/6)2]`
- `Gini-index  = 1-[0.25+0.25]`
- `GINI-index = 1-0.5`
- `gini_index = 0.5`
- That is making the gini index maximum value.



GINI inpurity is computationaly is not much expensive that much the entropy because in the entropy we need calculate the IG and other things also.

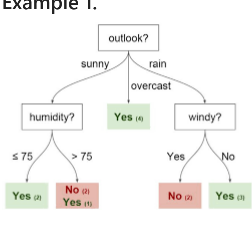
Decision Tree Split For The Numerical Feature.

In the dependent faetures all are avaliable in the continous at this condition how decision tree work !!

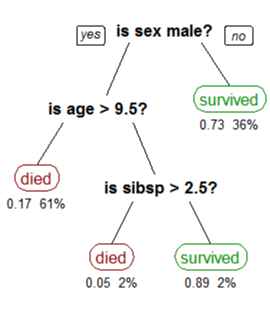
- First decision tree sort all independent variables.
- Then it will consider the threshould values for each faeture.
- It will create the branch on that basis .like age >20 one branch and age<20 this is kind of another branch.
- It will create the branch by using diffrent diffrent threshold values for the each feature simultaeously it calculate the information gain by using entropy criteria and finaly determine which threshold values gives us the best information gain on that basis of threshold values our split of decision tree will be happen.
- It takes the small threshold because it is sorted and try to improve the threshold value whenever algorithm will not fit properly so that we will get the higher information gain.

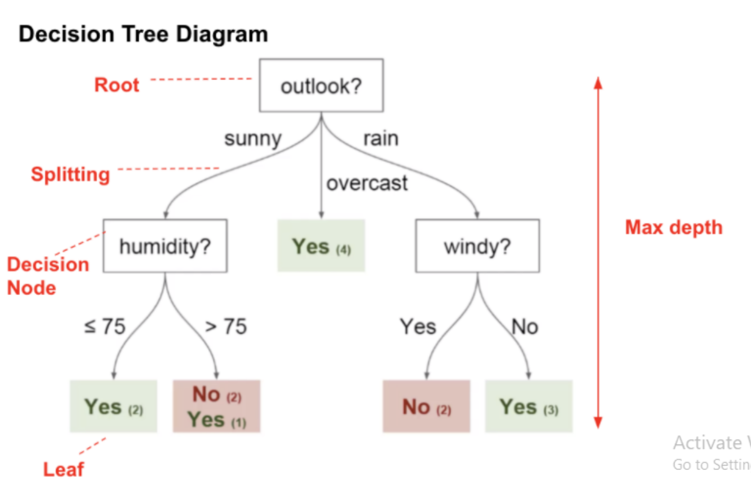
Whenever the dimensionality of the data will increase that performace of decision tree will be getting deteriorate as well as It will take more time for training.

Example 1.



Example 2.





Interview questions on Decision Tree.

How the decision tree algorithm works ?

- This is can explain with the help of working principle of Entropy,Information Gain,GiNi Impurity.

How the decision tree impacted by an Outliers ?

- Outliers does not impact Decision tree as they DOES NOT USE any distance metrics to calculate .They uses probilities by entropy,information gain or gini index to create the tree.So basically no impact.
- Decision tree is baisically with low bias and high variance if we do not appply any hyperparamter tunning.
- In decision tree the tree break itself into the form of braches until the leaf node will not get the one result like either yes or no.
- In order to convert the high variance to low variance we will prefer the hyperparameter tunning that is `Pruining.`

Which Library you will use to Construct the decision Tree?

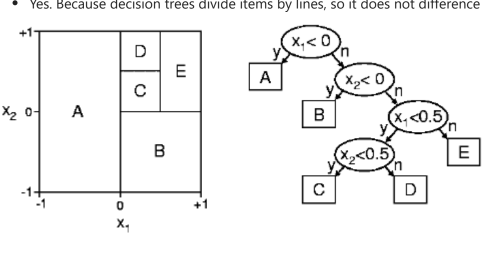
- We will see below

What is criteria use in DecisionTreeRegressor ?

- Mean Squared error.

Is outliers does not effect on performace of decision tree ?

- Yes. Because decision trees divide items by lines, so it does not difference how far is a point from lines.



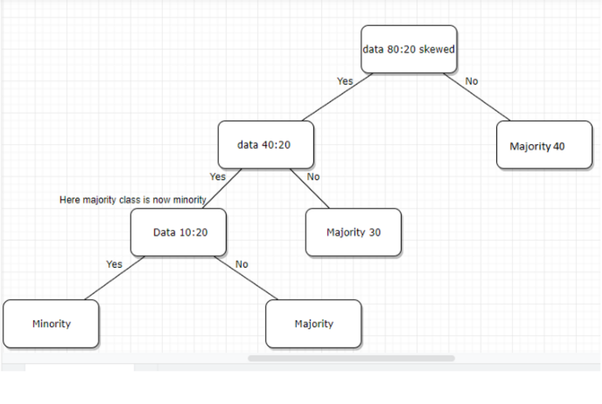
- Yes all tree algorithms are robust to outliers. Tree algorithms split the data points on the basis of same value and so value of outlier won't affect that much to the split.
- For example: Want to determine the buying behavior of customers depending upon their house size. House size is numeric continuous variable ranging from 1-1000 sq ft.
- So , now consider majority of my customers house size in range of 100-500. If I got some customers with house size of 1000 then what it does is simply split the data on the basis of some value where entropy at next level is less than that of the current level.
- Split gets decided on the value of house size such that i will get more homogeneous group of customers.

Advantages ans Disadvantages

Advantages

- **Good for categorical data:**
- For categorical data splitting is easier compared to continue data. That's why the decision tree is good with categorical data where else struggle with continuous data.
- **Won't be affected by outliers:**
- Decision tree will first split signal data points. After a while when DT can't extract any information from the signal point that is when DT can't split signal data point further it will switch to outliers. At this point, we need to prune the Decision tree so that outliers are not processed.
- **Redundant features avoided By Pruining:**
- Here also Decision tree first treat original features and when it is about to treat redundant feature, we should cut decision tree so that redundant features are not processed. Note: There is a trade-off between class skew and outlier/redundant feature
- **No Need to impute missing data:**
- Decision tree will handle missing values. It will keep all missing values in one node during the split.
- **Interpretable:**
- Decision tree is interpretable. We can say why it did? What it did.
- **The tendency to underfit:**
- NO, when we allow decision tree to allow properly **it will never underfit.**
- **Assumptions:**
- Decision tree doesn't have any underlying data assumptions.
- Good for class skewed data: Decision tree performs well for class skewed data provided we should not do pruning (cutting of decision tree).

This is how It handel the Skewed data.



Disadvantages

- **Overfit:**
- Decision Tree will overfit if we allow to grow it i.e., each leaf node will represent one data point. In order to overcome this issue of overfitting, we should prune the tree. **We can prune decision Tree by setting Max-depth of the tree or by setting minimum data points in each node.**
- **Speed is less:**
- Since decision tree split the data according to columns its speed reduces when the number of columns increases.
- **Greedy Approach:**
- Splitting data according to the first best split and it will split whole data by that path only. But it is possible that different paths of the split are more informative so that split may not be the best split.
- **Not good for regression:**
- Decision Tree can be used for regression but decision tree won't perform well.
- **Train/Test time:**
- Comparing to logistics regression decision Tree takes more time for training and it is slower for testing.

Thank You !!