Decision Tree

Bias: Bias describes how well a model matches the training set. A model with high bias won't match the data set closely, while a model with low bias will match the data set very closely. Bias comes from models that are overly simple and fail to capture the trends present in the data set

In his 1980 paper entitled “The need for bias in learning generalizations”, Tom Mitchell introduced the first use of the word “bias” in machine learning. He defined it to mean that a learning algorithm will not generalize unless it introduces some form of preference or restriction over the space of possible functions. Without any limitation or preference, the learning algorithm can memorize any data set without generalizing. This was later formalized in terms of the VC dimension (for a fixed-complexity function space), the No Free Lunch theorem, and structural risk minimization (for nested families of function spaces of increasing complexity.

This use of “bias” is closely related to the bias-variance tradeoff, because a learning algorithm with no bias (in the Mitchell sense) will have low bias and high variance in the bias-variance sense.

The most common interpretation of bias is with regards to the [bias–variance tradeoff](https://en.wikipedia.org/wiki/Bias%E2%80%93variance_tradeoff). Essentially, bias here is a source of error in your model that causes it to over-generalize and underfit your data. In contrast, variance is sensitivity to noise in the data that causes your model to overfit. We call it a tradeoff because improving one will often make the other metric worse.

# They are supervised learning algorithm which has a pre-defined target variable & they are mostly used in non-linear decision making with simple linear decision surface. In other words, they are adaptable for solving any kind of problem at hand (classification or regression).

## Why use decision trees?

*One of the best and mostly used supervised learning methods* are tree-based algorithms. **They empower predictive modeling with higher accuracy, better stability and provide ease of interpretation**. Unlike linear modeling techniques, they map non-linear relationships quite well. Methods like decision trees, random forest, gradient boosting are being popularly used in all kinds of data science problems. Hence, for every analyst, it’s important to learn these algorithms and apply them at the time of modeling.

# <https://towardsdatascience.com/decision-tree-algorithm-explained-83beb6e78ef4>

# Entropy : ID3 algorithm uses entropy to calculate the homogeneity of a sample.

# Measure how pure split is and Entropy value ranges from 0 – 1 .

# 0 mean highly pure and 1 is highly impure

# Entropy value gives measure of quality of split at a time one node eg: f1 node or f2 or f3 at a time .

# It will not give overall entropy of tree having different structural hierarchy of nodes to choice best hierarchy tree having least entropy .

# This is done by information gain .

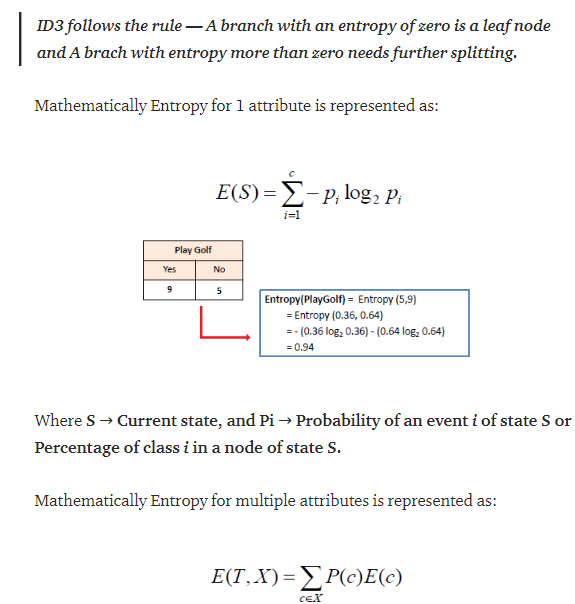
# 

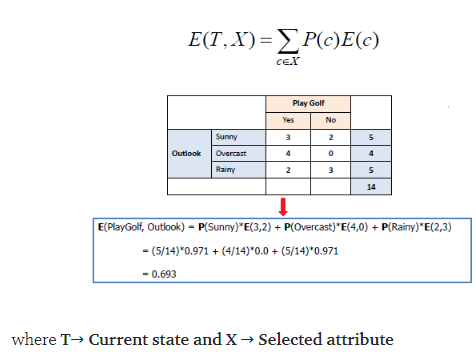
# Entropy is a measure of the randomness in the information being processed. The higher the entropy, the harder it is to draw any conclusions from that information. Flipping a coin is an example of an action that provides information that is random.

# 

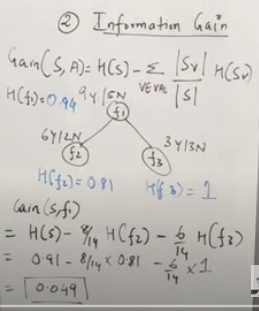
From the above graph, it is quite evident that the entropy H(X) is zero when the probability is either 0 or 1. The Entropy is maximum when the probability is 0.5 because it projects perfect randomness in the data and there is no chance if perfectly determining the outcome

***ID3 follows the rule — A branch with an entropy of zero is a leaf node and A brach with entropy more than zero needs further splitting.***





**Information gain**



It’s a measure which gives tree with better split based on structural hierarchy of the tree .

Sv = Number pf splits in sub-node , and S : Total number in split in root node

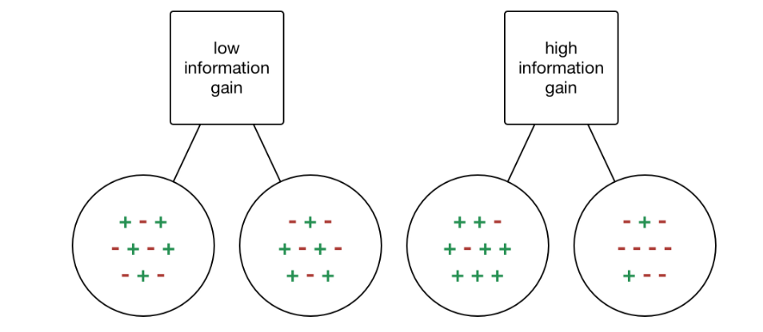
H(S) : Entropy of root node , H(f2) : Entropy of root node , H(f3): Entropy of root node

IG values gives the measure of the tree with best structural hierarchy with good split .

Decision calculates IG values for verity of decision tree structures and chooses tree with highest IG values.

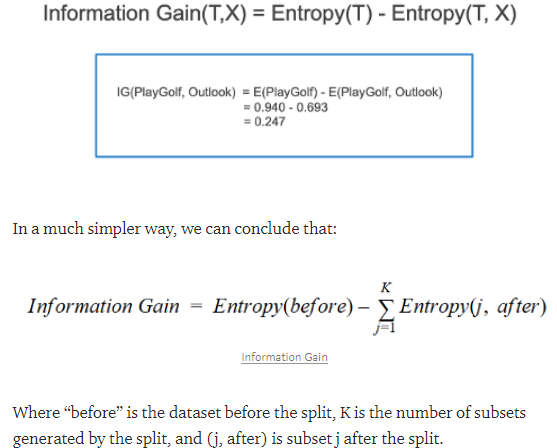
 The **information gain** is based on the decrease in entropy after a data-set is split on an attribute. Constructing a **decision tree** is all about finding attribute that returns the highest **information gain** (i.e., the most homogeneous branches).

Information gain or **IG**is a statistical property that measures how well a given attribute separates the training examples according to their target classification. Constructing a decision tree is all about finding an attribute that returns the highest information gain and the smallest entropy.



Information gain is a decrease in entropy. It computes the difference between entropy before split and average entropy after split of the dataset based on given attribute values. ID3 (Iterative Dichotomiser) decision tree algorithm uses information gain.

Mathematically, IG is represented as:



# Gini index

# Both Entropy and Gini Index measures impurity of split

# 

# Entropy value range is 0 - 1

# Gini index value range is 0 – 0.5

# You can see the graph in above picture that at .50 probability entropy is 1 and Gini index is 0.5.

# The main difference is computation , you can see the formula of Entropy and Gini Index.

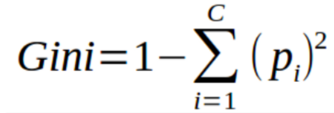
# Entropy calculation takes more time due to presence of logarithm and Gini index takes less time .

# All state of the art model such as Ensembled Random forest uses Gini index and later tress structure is chosen using Information Gain value .

# Gini index or Gini impurity measures the degree or probability of a particular variable being wrongly classified when it is randomly chosen. ... A Gini Index of 0.5 denotes equally distributed elements into some classes

“measure how often a randomly chosen element from the set would be incorrectly labeled”

You can understand the Gini index as a cost function used to evaluate splits in the dataset. It is calculated by subtracting the sum of the squared probabilities of each class from one. It favors larger partitions and easy to implement whereas information gain favors smaller partitions with distinct values.



Gini Index works with the categorical target variable “Success” or “Failure”. It performs only Binary splits.

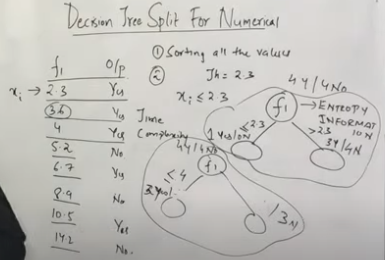
Higher the value of Gini index higher the homogeneity.

Steps to Calculate Gini index for a split

1. Calculate Gini for sub-nodes, using the above formula for success(p) and failure(q) (p²+q²).
2. Calculate the Gini index for split using the weighted Gini score of each node of that split.

CART (Classification and Regression Tree) uses the Gini index method to create split points.

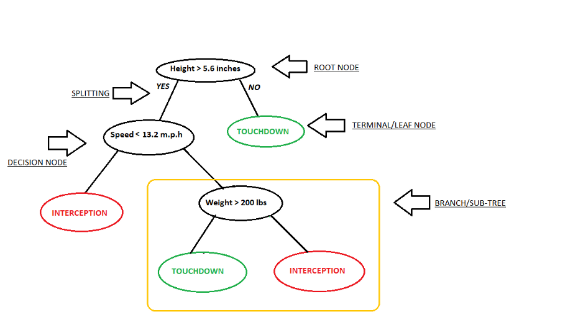
**Decision Tree Split For Numerical Feature**



*The***splitting of numerical features:**

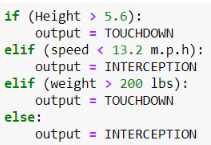
* Can be performed by sorting the features in the ascending order .
* Take first value as threshold value split and calculate Entropy and information gain for each value of the threshold
* Value of highest information gain with respect to threshold value considered to be tree.
* We need to repeat step 2 for all values of feature . This is reason why training of decision tree for numeric variable takes more time compared to categorical variable .

NFL data set for predicting whether the players will score a touchdown or not. Below model uses 3 features/attributes/columns from the data set, namely height, speed, and strength.



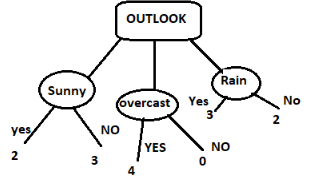
In this problem, we need to segregate players who will score during the match time based on highly significant input variable among all three. This is where decision tree helps, it will segregate the players based on all values of three variable and identify the variable, which creates the best homogeneous sets of players (which are heterogeneous to each other).

**Intuition Development:**



# Concept of pure node:

Pure node is a node wherein all the datapoints belong to the same class and thus it is very easy to make the prediction at such node.



**Overfitting & Underfitting in CART:**

The consequences of overfitting and underfitting and its causes will be discussed at a later point(Stop criterion).

Larger the depth of the tree more are the chances of variance(overfitting).

Whereas smaller the depth of the tree more are the chances of bias tree(underfitting).

**Stop criterion:**

If we continue to grow the tree fully until each leaf node corresponds to the lowest impurity, then the data have typically been overfitted. If splitting is stopped too early, error on training data is not sufficiently high and performance will suffer due to bais. Thus, preventing overfitting & underfitting are pivotal while modeling a decision tree and it can be done in 2 ways:

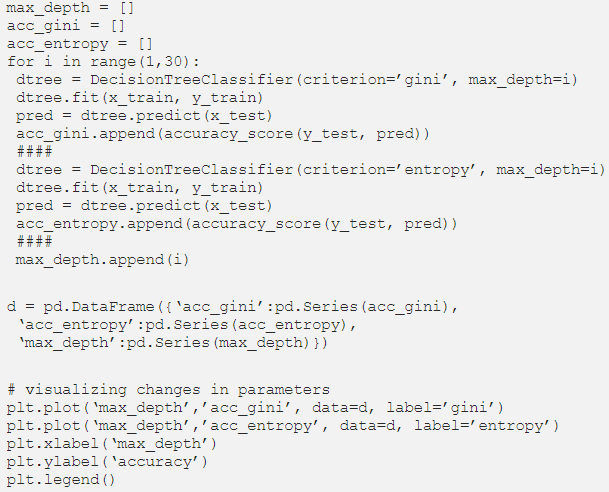
1. **Setting constraints on tree size:**

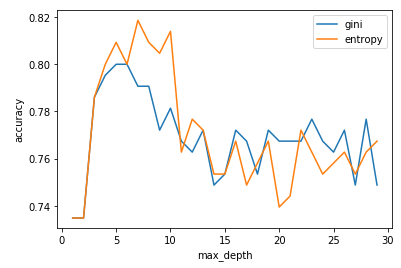
* Providing minimum number of samples for a node split.
* Deploying the minimum number of samples for a terminal node (leaf).
* Allowing maximum depth of tree (vertical depth).
* Maximum number of terminal nodes.
* Maximum features to consider for the split.

**Tree pruning:**pruningis a technique in machine learning that*reduces the size of decision trees* by removing sections of the tree. It also reduces the complexity of the final classifier, and hence improves predictive accuracy by the reduction of overfitting. Tree prunning can be done in two ways by pre-prunning or by post-prunning.

**Pre-prunning:**

* Stop splitting the current node if it does not improve the entropy by at least some pre-set(threshold) value.
* Stop partitioning if the number of datapoints are less than some preset(Threshold) values.
* Restricting the depth of the tree to some pre-set(Threshold) value.





***-*Post-prunning:**

* It can be done by first allowing the tree to grow to its full potential and then prunning the tree at each level after calculating the cross-validation accuracy at each level.

**Disadvantages of CART:**

1. A small change in the dataset can make the tree structure unstable which can cause variance.
2. Decision tree learners create **underfit trees** if some classes are imbalanced. It is therefore recommended to balance the data set prior to fitting with the decision tree.

**Preparing data for CART:**

* The splitting of numerical features can be performed by sorting the features in the ascending order and trying each value as the threshold point and calculating the information gain for each value as the threshold. Finally, if that value obtained is equal to the threshold which gives the maximum I.G value then hurray..!!
* Feature scaling(column standardization) not necessary to perform in decision trees. However, it helps with data visualization/manipulation and might be useful if you intend to compare performance with other data or other methods like SVM.
* In order to handle categorical features in Decision trees, we must never perform one hot encoding on a categorical variable even if the categorical variables are nominal since most of the libraries can handle categorical variables automatically. we can still assign a number for each variable if desired.
* If height or depth of the tree is exactly one then such a tree is called as a decision stump.
* Imbalanced class does have a detrimental impact on the tree’s structure so it can be avoided by either using upsampling or by using downsampling depending upon the dataset.
* Apart from skewed classes, high dimensionality can also have an adverse effect on the structure of the tree if dimensionality is very high that means we have a lot of features which means that to find the splitting criterion on each node it will consume a lot of time.
* Outliers also impact the tree’s structure as the depth increases the chance of outliers in the tree increases.
* Feature importance can be determined by calculating the normalized sum at every level as we have t reduce the entropy and we then select the feature that helps to reduce the entropy by the large margin. so for whichever feature the normalized sum is highest, we can then think of it as the most important feature. similarly, feature which has the second highest normalized sum can be thought of as a second important feature.