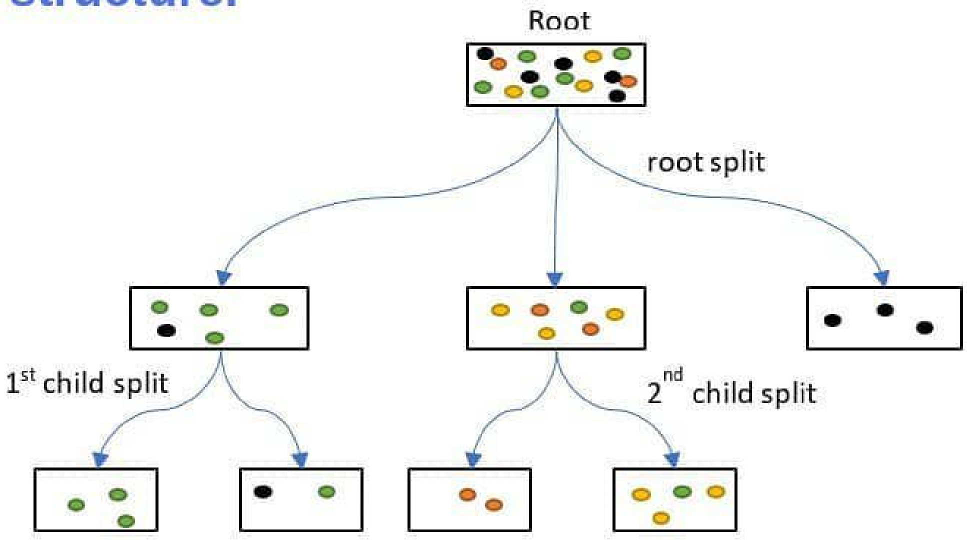
Decision trees

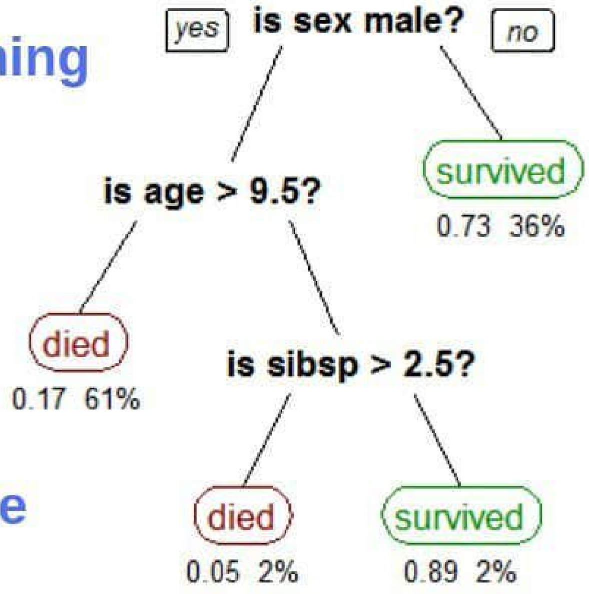
- Decision Trees (DTs) are a non-parametric supervised learning method used for classification and regression.
- They are capable of fitting complex data sets while allowing the user to see how a decision was taken.
- Decision trees learn from data to approximate a sine curve with a set of if-then-else decision rules.
- The deeper the tree, the more complex the decision rules and the fitter the model.
- In this technique, we split the population or sample into two or more homogeneous sets (or sub-populations) based on most significant splitter/differentiator in input variables and it looks like a tree structure.



@learn.machinelearning

Decision trees

- A decision node has two or more branches. Leaf node represents a classification or decision.
- Decision trees can handle both categorical and numerical data.
- Let's see a simple example of how the decision tree looks with both numerical and categorical data.
- The most important steps in Decision trees are Spitting, Pruning and tree selection.
- **Splitting:** The process of partitioning the data set into subsets.
- **Pruning:** Pruning is the process of reducing the size of the tree by turning some branch nodes into leaf nodes, and removing the leaf nodes under the original branch.



- **Tree Selection:** This is where we do parameter tuning to select the best tree.

@learn.machinelearning

Decision trees

- So the first step is to calculate the entropy of dataset.
- **Entropy:-** A decision tree is built top-down from a root node and involves partitioning the data into subsets that contain instances with similar values (homogeneous). the algorithm uses entropy to calculate the homogeneity of a sample.
- **Information Gain:-** 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 an attribute that returns the highest information gain (i.e., the most homogeneous branches).
- IG is given as entropy - maximum weighted entropy

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

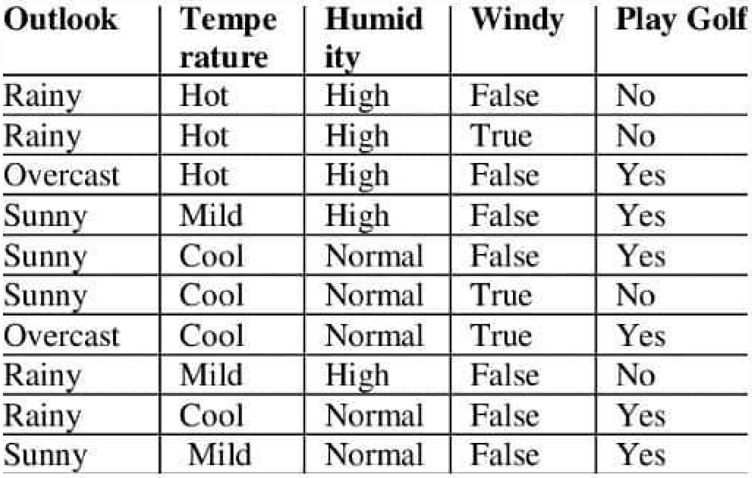
$$Gain(S, A) = Entropy(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} Entropy(S_{v)}$$

And we split the feature which has maximum IG.

@learn.machinelearning

Decision trees

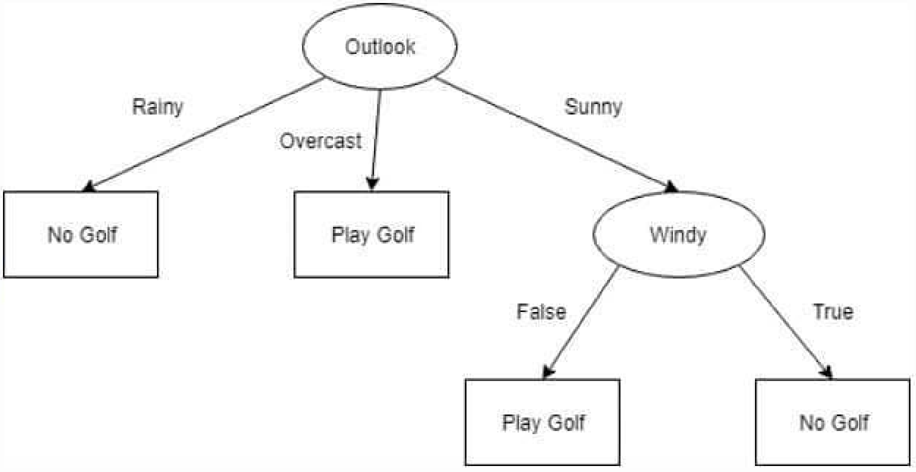
- So how does a Decision tree works?
- The important things you need to know before learning how decision tree works are Entropy and Information gain.
- Let's take a common example to lean how DT works.



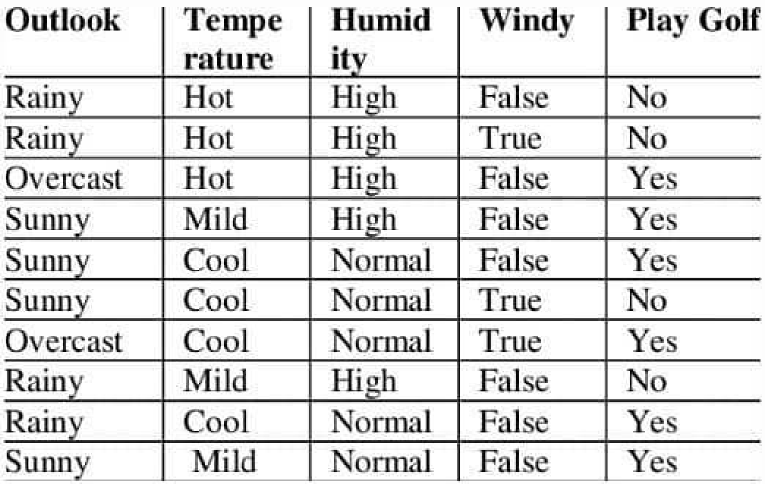
Outlook	Temperature	Humidity	Windy	Play Golf
Rainy	Hot	High	False	No
Rainy	Hot	High	True	No
Overcast	Hot	High	False	Yes
Sunny	Mild	High	False	Yes
Sunny	Cool	Normal	False	Yes
Sunny	Cool	Normal	True	No
Overcast	Cool	Normal	True	Yes
Rainy	Mild	High	False	No
Rainy	Cool	Normal	False	Yes
Sunny	Mild	Normal	False	Yes

Say we have the above dataset, and would like to predict if we would play golf or not given the weather.

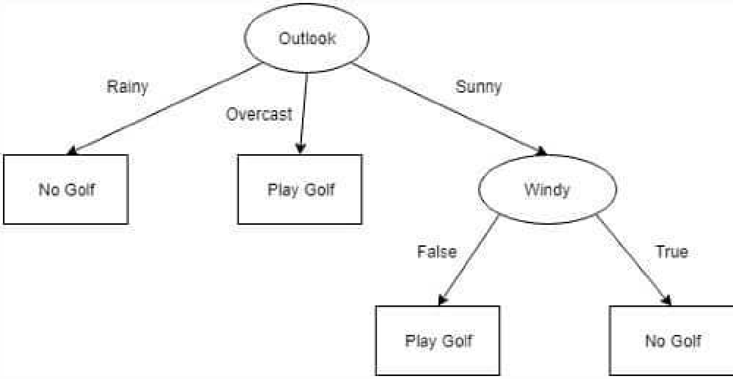
The decision tree for this problem might look like the one below.



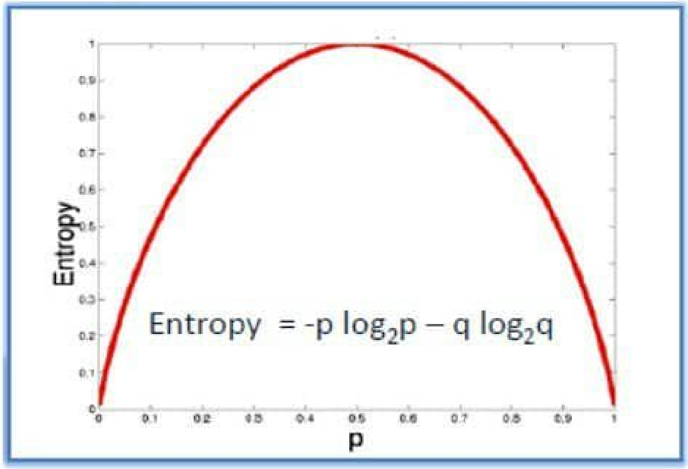
Decision trees



Outlook	Temperature	Humidity	Windy	Play Golf
Rainy	Hot	High	False	No
Rainy	Hot	High	True	No
Overcast	Hot	High	False	Yes
Sunny	Mild	High	False	Yes
Sunny	Cool	Normal	False	Yes
Sunny	Cool	Normal	True	No
Overcast	Cool	Normal	True	Yes
Rainy	Mild	High	False	No
Rainy	Cool	Normal	False	Yes
Sunny	Mild	Normal	False	Yes

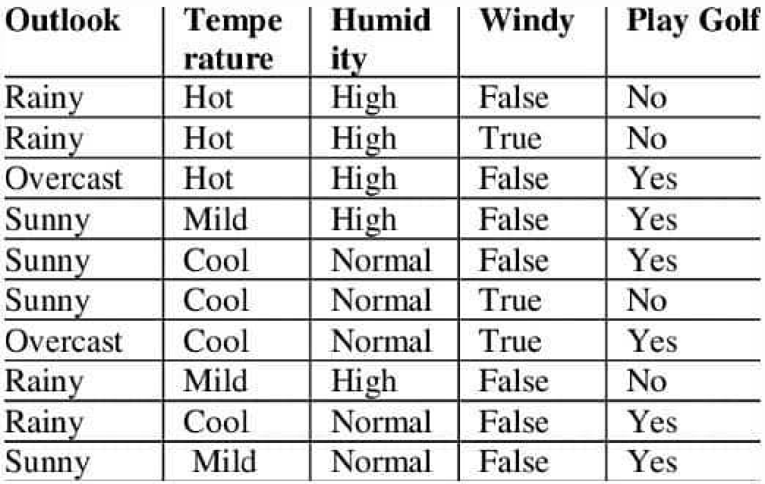


- So the first step is to calculate the entropy of dataset.
- Let's say we have binary classification problem with 6 samples as "YES" and 4 samples as "NO".
- the entropy of dataset will be - (6/10)*log2(6/10) - (4/10)*log2(6/10) = 0.73
- The entropy of dataset is 0.73
- if we have equal samples let say we have 5 as YES and 5 as NO. The entropy will be 1.
- Entropy controls how a Decision Tree decides to split the data. It actually effects how a Decision Tree draws its boundaries.

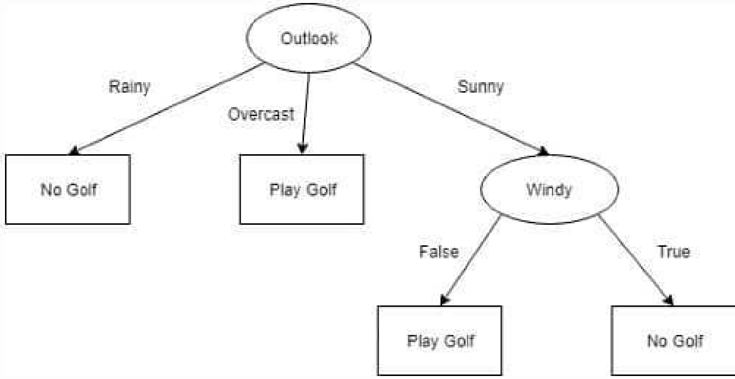


Entropy = $-0.5 \log_2 0.5 - 0.5 \log_2 0.5 = 1$

Decision trees



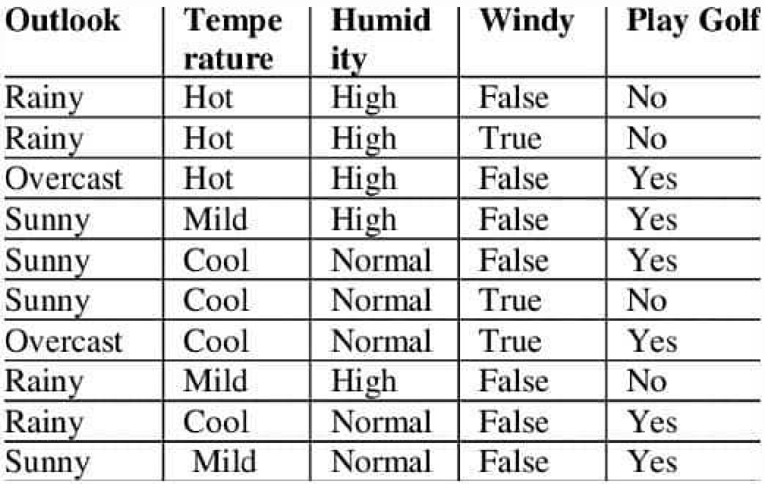
Outlook	Temperature	Humidity	Windy	Play Golf
Rainy	Hot	High	False	No
Rainy	Hot	High	True	No
Overcast	Hot	High	False	Yes
Sunny	Mild	High	False	Yes
Sunny	Cool	Normal	False	Yes
Sunny	Cool	Normal	True	No
Overcast	Cool	Normal	True	Yes
Rainy	Mild	High	False	No
Rainy	Cool	Normal	False	Yes
Sunny	Mild	Normal	False	Yes



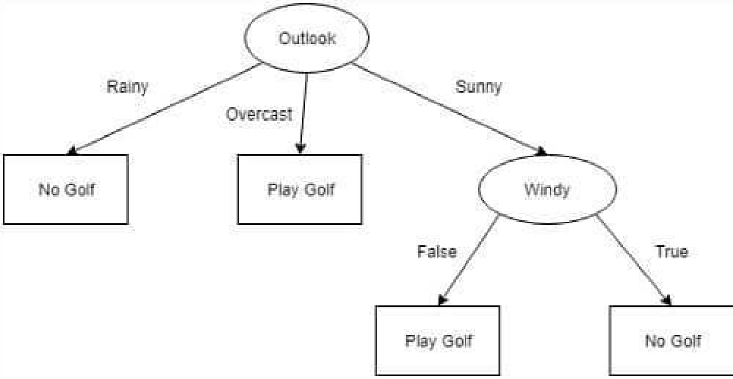
- **The next step is to calculate the entropy for each feature. Then it is added proportionally, to get total entropy for the split (weighted entropy).**
- **The resulting entropy is subtracted from the entropy before the split(parent). The result is the Information Gain and we try to maximize this at every point.**
- **First, we calculate the IG when outlook feature is used to split the dataset. If the feature is categorical we split using the categories. If it is continuous there are many methods to split, we discuss this later.**
- **Entropy for the category(Rainy) -(3/4)*log2(3/4)-(1/4)*log2(1/4) is 0.81 as we have 3 NO samples for rainy and 1 YES sample for rainy**

@learn.machinelearning

Decision trees

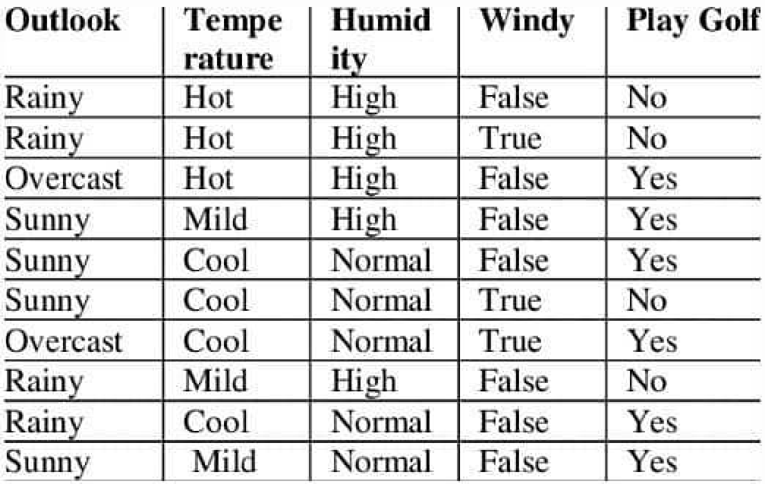


Outlook	Temperature	Humidity	Windy	Play Golf
Rainy	Hot	High	False	No
Rainy	Hot	High	True	No
Overcast	Hot	High	False	Yes
Sunny	Mild	High	False	Yes
Sunny	Cool	Normal	False	Yes
Sunny	Cool	Normal	True	No
Overcast	Cool	Normal	True	Yes
Rainy	Mild	High	False	No
Rainy	Cool	Normal	False	Yes
Sunny	Mild	Normal	False	Yes

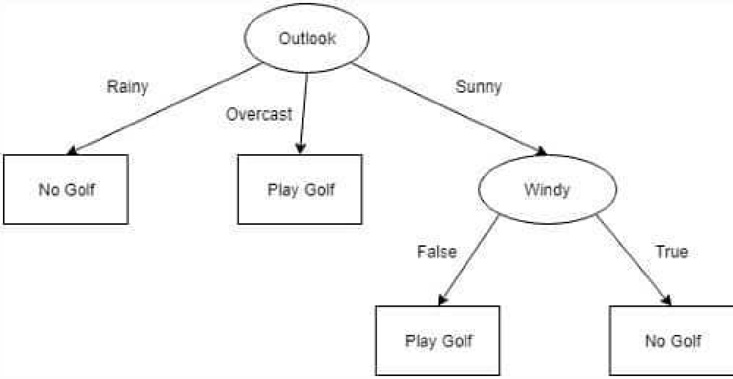


- Entropy for the category(Overcast) -(2/2)*log2(2/2)-(0/2)*log2(0/2) is 0 as we have 2 YES samples and 0 NO samples.
- Entropy for the category(sunny) -(3/4)*log2(3/4)-(1/4)*log2(1/4) is 0.81 as we have 3 YES samples and 1 NO sample.
- final entropy of outlook will be (4/10)*0.81 + 2(10)*0+(4/10)*0.81 = 0.648.
- let's calculate the entropy of other features.
- The entropy of Temperature (3/10)*0.91+ (3/10)*0.91+ (4/10)*0.81 = 0.87
- The entropy of Humidity (5/10)*0.97 + (5/10)*0.72 = 0.845
- The entropy of windy (7/10)*0.86 + (3/10)* = 0.91

Decision trees



Outlook	Temperature	Humidity	Windy	Play Golf
Rainy	Hot	High	False	No
Rainy	Hot	High	True	No
Overcast	Hot	High	False	Yes
Sunny	Mild	High	False	Yes
Sunny	Cool	Normal	False	Yes
Sunny	Cool	Normal	True	No
Overcast	Cool	Normal	True	Yes
Rainy	Mild	High	False	No
Rainy	Cool	Normal	False	Yes
Sunny	Mild	Normal	False	Yes



- Information gain of the features will be root entropy - weighted entropy.
- IG for outlook is 0.73 - 0.648 = 0.082
- IG for temperature is 0.73 - 0.87 = -0.14
- IG for humidity is 0.73 - 0.845 = -0.115
- IG for windy is 0.73 - 0.91 = -0.18
- So the highest IG is for Outlook feature and we use this feature to split the dataset as you can see in the above decision tree because we are gaining more information through this split.
- A branch with the entropy of 0 is a leaf node(overcast).
- And in the next level we stopped splitting the rainy and overcast this is because of pruning.
- And we do the same for the next iterations. As depth increases, it leads to overfitting as we are creating too complex rules.

Decision trees

- As depth decreases, it leads to underfitting as we are not learning anything.
- Let's learn more about decision tree pruning.
- **Pruning:-** Pruning is a method of limiting tree depth to reduce overfitting in decision trees.
- There are two types of pruning: pre-pruning, and post-pruning.
- **Pre-pruning:-** Pre-pruning a decision tree involves setting the parameters of a decision tree before building it. There a few ways to do this:

  A. Set maximum tree depth

  B. Set the maximum number of terminal nodes

  C. Set minimum samples for a node split:

  D. Controls the size of the resultant terminal nodes

  E. Set the maximum number of features
- **Post-pruning:-** To post-prune, validate the performance of the model on a test set. Afterwards, cut back splits that seem to result from overfitting noise in the training set. Pruning these splits dampens the noise in the data set.

Decision trees

- Pre-pruning:-
- Minimum samples for a node split

  A. Defines the minimum number of samples (or observations) which are required in a node to be considered for splitting.

  B. Used to control over-fitting. Higher values prevent a model from learning relations which might be highly specific to the particular sample selected for a tree.

  C. Too high values can lead to under-fitting hence, it should be tuned using CV.
- Minimum samples for a terminal node (leaf)

  A. Defines the minimum samples (or observations) required in a terminal node or leaf.

  B. Used to control over-fitting similar to min_samples_split.

  C. Generally lower values should be chosen for imbalanced class problems because the regions in which the minority class will be in majority will be very small.

Decision trees

- **Pre-pruning:-**
- **Maximum depth of the tree (vertical depth)**

  A. The maximum depth of a tree.

  B. Used to control over-fitting as higher depth will allow the model to learn relations very specific to a particular sample.

  C. Should be tuned using CV.
- **Maximum number of terminal nodes**

  A. The maximum number of terminal nodes or leaves in a tree.

  B. Can be defined in place of max_depth. Since binary trees are created, a depth of ‘n’ would produce a maximum of 2^n leaves.
- **Maximum features to consider for a split**

  A. The number of features to consider while searching for the best split. These will be randomly selected.

  B. As a thumb-rule, square root of the total number of features works great but we should check up to 30-40% of the total number of features.

  C. Higher values can lead to over-fitting but depend on case to case.

Decision trees

- **Advantages**
- **Easy to Understand:** Decision tree output is very easy to understand even for people from the non-analytical background. It does not require any statistical knowledge to read and interpret them. Its graphical representation is very intuitive and users can easily relate their hypothesis.
- **Useful in Data exploration:** Decision tree is one of the fastest ways to identify the most significant variables and the relation between two or more variables. With the help of decision trees, we can create new variables/features that have a better power to predict the target variable. You can refer article (Trick to enhance the power of regression model) for one such trick. It can also be used in the data exploration stage. For example, we are working on a problem where we have information available in hundreds of variables, their decision tree will help to identify the most significant variable.
- **Less data cleaning required:** It requires less data cleaning compared to some other modelling techniques. It is not influenced by outliers and missing values to a fair degree.

Decision trees

- **Advantages**
- **data type is not a constraint:** It can handle both numerical and categorical variables.
- **Non-Parametric Method:** Decision tree is considered to be a non-parametric method. This means that decision trees have no assumptions about space distribution and the classifier structure.
- **DisAdvantages**
- **Overfitting:** Overfitting is one of the most practical difficulties for decision tree models. This problem gets solved by setting constraints on model parameters and pruning.
- **Not fit for continuous variables:** While working with continuous numerical variables, decision tree loses information when it categorizes variables in different categories.