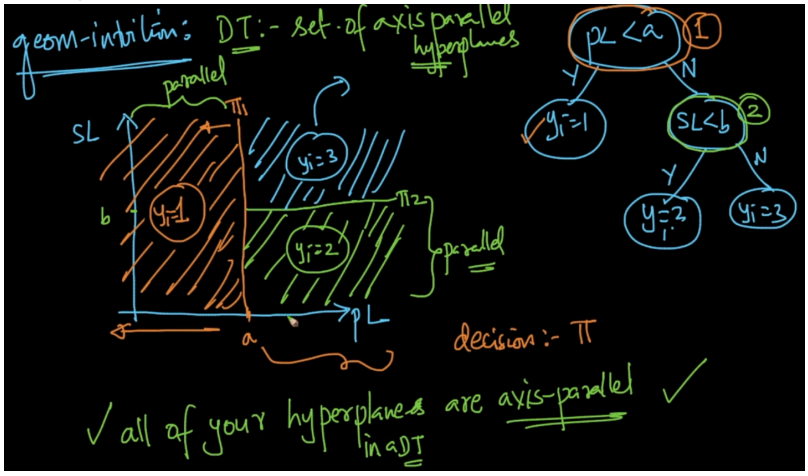
Decision Trees -

Decision tree is a type of supervised learning algorithm (having a predefined target variable) that is mostly used in classification problems.

Decision Tree starts with root node with a condition and further divided into branches.

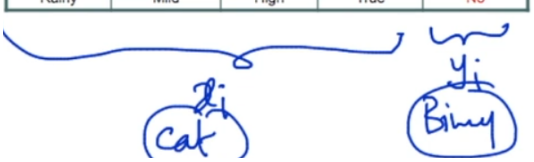
- DT have root node, leaf node and internal node.
- DT have set of access parallel hyperplanes.
- It works for both categorical and continuous input and output variables.

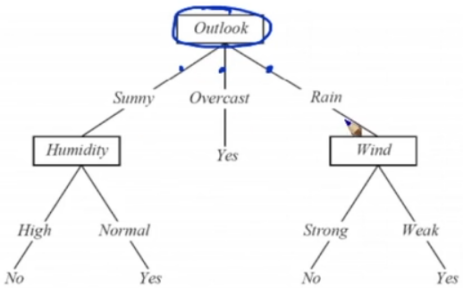


- Decision Trees are a non-parametric supervised learning method used for **both classification and regression tasks.**
- Tree models where the target variable can take a discrete set of values are called **classification trees.**
- Decision trees where the target variable can take continuous values (typically real numbers) are called **regression trees.**

Real Life Example

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





Types of Decision Trees

Types of decision tree is based on the type of target variable we have. It can be of two types:

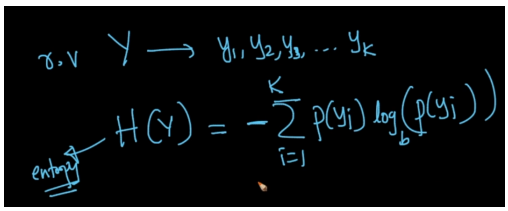
1. **Categorical Variable Decision Tree-** Decision Tree which has categorical target variable then it called as categorical variable decision tree. Example:- In above scenario of student problem, where the target variable was “Student will play Tennis or not” i.e. YES or NO.
2. **Continuous Variable Decision Tree-** Decision Tree has continuous target variable then it is called as Continuous Variable Decision Tree.

Example:- Let’s say we have a problem to predict whether a customer will pay his renewal premium with an insurance company (yes/ no). Here we know that income of customer is a significant variable but insurance company does not have income details for all customers. Now, as we know this is an important variable, then we can build a decision tree to predict customer income based on occupation, product and various other variables. In this case, we are predicting values for continuous variable.

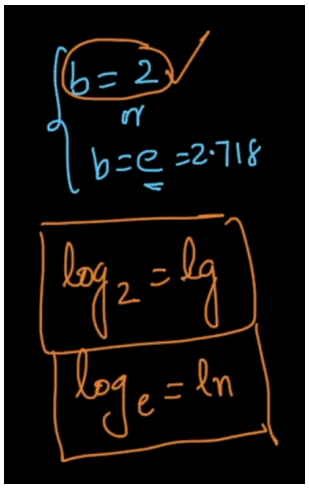
Building Decision Tree Entropy

- At any stage of branch when we break node or data set into two parts using any feature we are trying to reduce error effectively in classification its called **Entropy.** In regression its called **error.**

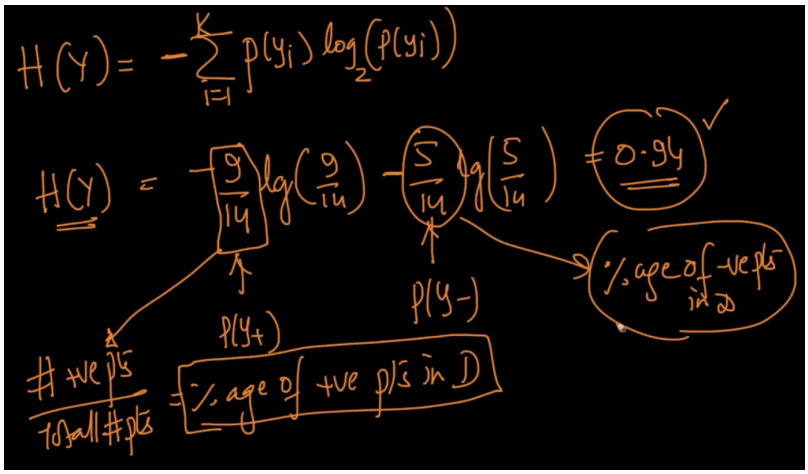
Here Y is random variable



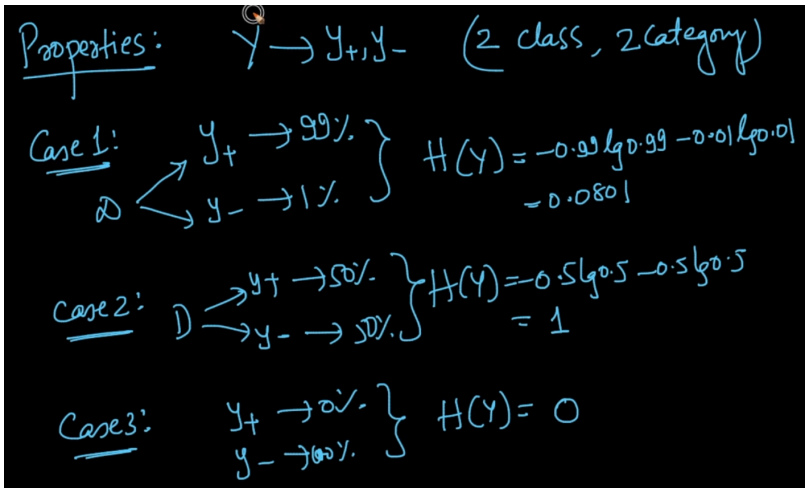
Here b is the base of log



How to calculate Entropy H(Y)-
Here entropy is calculated for above Tenis example.
k = 2, means PlayTennis is either true or false.



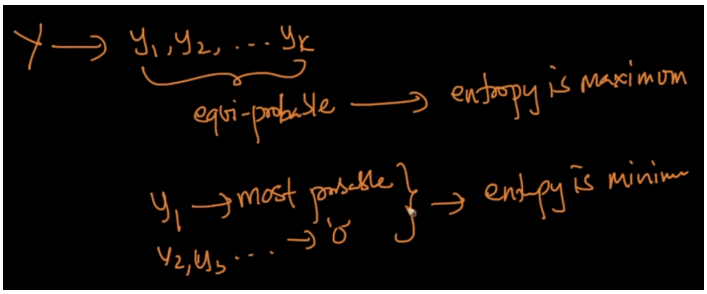
Cases-



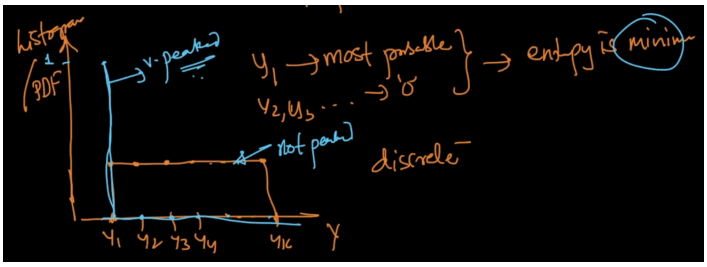
Conclusion of Cases- All this holds for two class classification

- If both the point are equally probable ie both are 50% then we get maximum entropy 1.
- If one class fully domainate i.e one having 0% other having 100% then we get entropy 0.

For n-class classification-



- Entropy is max for equi-probable.
- Entropy is min for class having one side 0 and other side most probable.



- **The more peaked is a distribution, less is its entropy.**

How does a tree based algorithms decide where to split -

The decision of making strategic splits heavily affects a tree's accuracy. The decision criteria is different for classification and regression trees.

Decision trees use multiple algorithms to decide to split a node in two or more sub-nodes. The creation of sub-nodes increases the homogeneity of resultant sub-nodes. In other words, we can say that purity of the node increases with respect to the target variable. Decision tree splits the nodes on all available variables and then selects the split which results in most homogeneous sub-nodes.

The algorithm selection is also based on type of target variables. Let's look at the four most commonly used algorithms in decision tree:

1. Gini
2. Information Gain
3. Chi-Square
4. Reduction in Variance

1. Information Gain

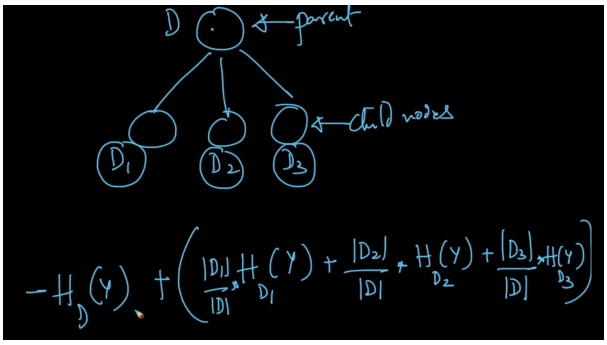
- Information gain (IG) measures how much "information" a feature gives us about the class.
- Used to construct a Decision Tree.
- Decision Trees algorithm will always tries to maximize Information gain.
- An attribute with highest Information gain will tested/split first.

$$\text{Information gain} = \text{entropy (parent)} - [\text{weightes average}] * \text{entropy (children)}$$

Formulae-

$$IG(Y, var) = \sum_{i=1}^{k} \frac{|D_i|}{|D|} * H_{D_i}(Y) - H_D(Y)$$

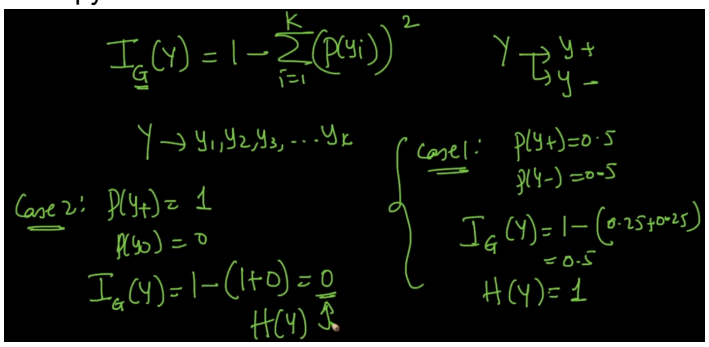
Explainated formulae-



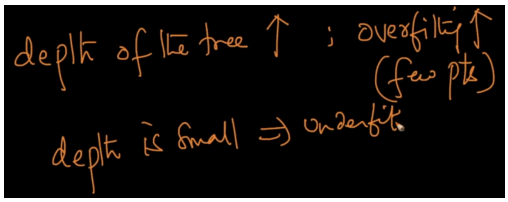
2. Gini Impurity

Gini says, if we select two items from a population at random then they must be of same class and probability for this is 1 if population is pure.

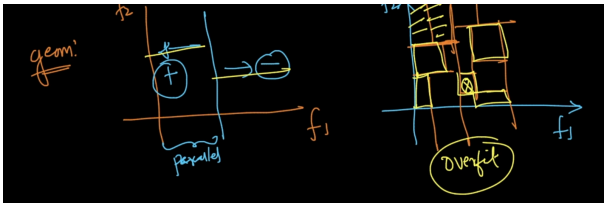
- It works with categorical target variable “Success” or “Failure”.
- It performs only Binary splits
- Higher the value of Gini higher the homogeneity.
- CART (Classification and Regression Tree) uses Gini method to create binary splits.
- Gini Impurity tells us the probability of misclassifying an observation.
- Used to decide the optimal split from a root node, and subsequent splits.
- Gini impurity is similar to Entropy, now a days we use GI bcz its faster to compute and consider as an alternative to Entropy.



Overfitting and Underfitting case-



- Depth is calculated using Cross-Validation.
- Decision stump is under-fitting the data with less depth.



Left graph is Underfit and right one is Overfit.

Geometrically-

- Graph having low hyperplane are generally Underfit.
- Graph having high no. of hyperplane are Overfit.

How can we avoid over-fitting in decision trees -

Overfitting is one of the key challenges faced while using tree based algorithms. If there is no limit set of a decision tree, it will give you 100% accuracy on training set because in the worse case it will end up making 1 leaf for each observation. Thus, preventing overfitting is pivotal while modeling a decision tree and it can be done in 2 ways:

1. Setting constraints on tree size
2. Tree pruning

Are tree based algorithms better than linear models?

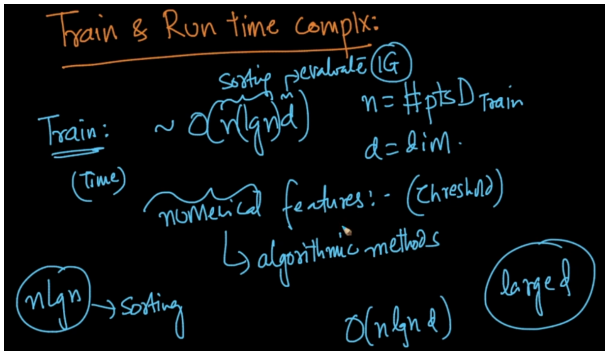
Let's look at some key factors which will help you to decide which algorithm to use:

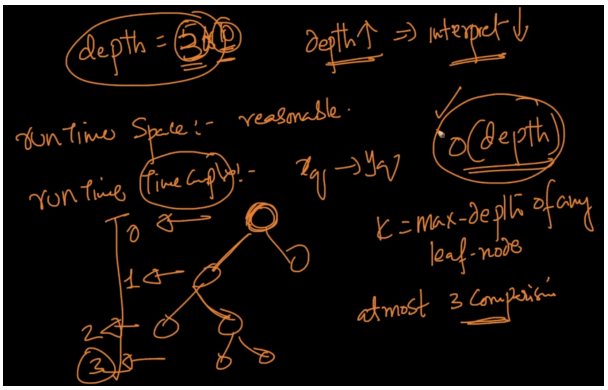
1. If the relationship between dependent & independent variable is well approximated by a linear model, linear regression will perform better than tree based model.
2. If there is a high non-linearity & complex relationship between dependent & independent variables, a tree model will perform better than a classical regression method.
3. If you need to build a model which is easy to explain to people, a decision tree model will always do better than a linear model. Decision tree models are even simpler to interpret than linear regression.

Feature Standardization

As Decision Tree is a distance based method, i.e why we do **not** need to perform Feature Standardization.

Train & Runtime Complexity

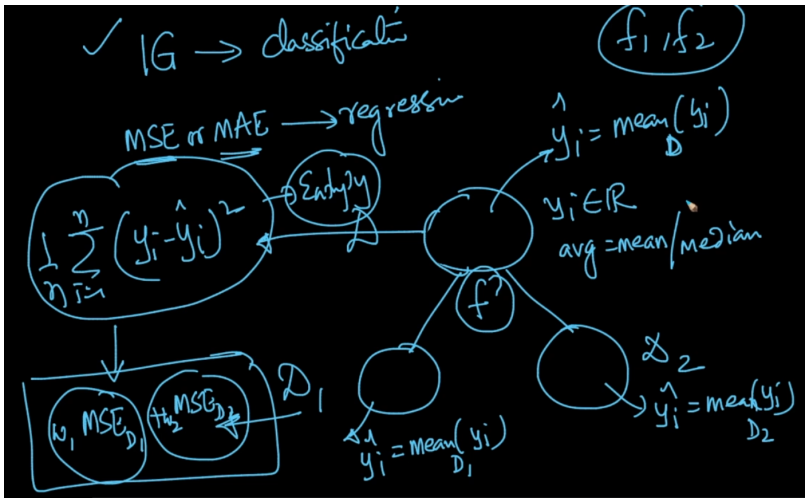


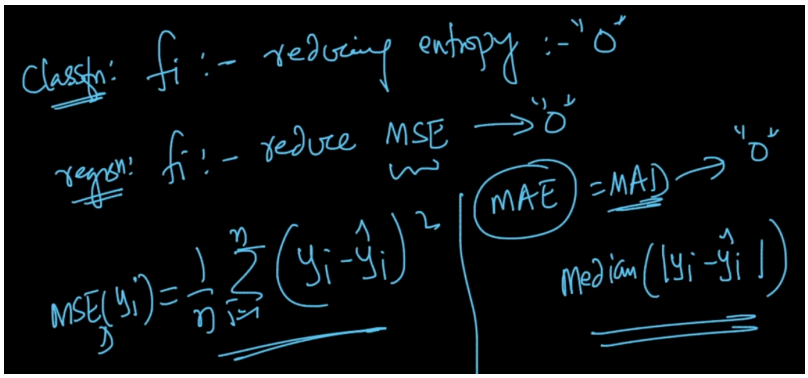


- At max height of decision tree is trained to be 5-10 level of depth.

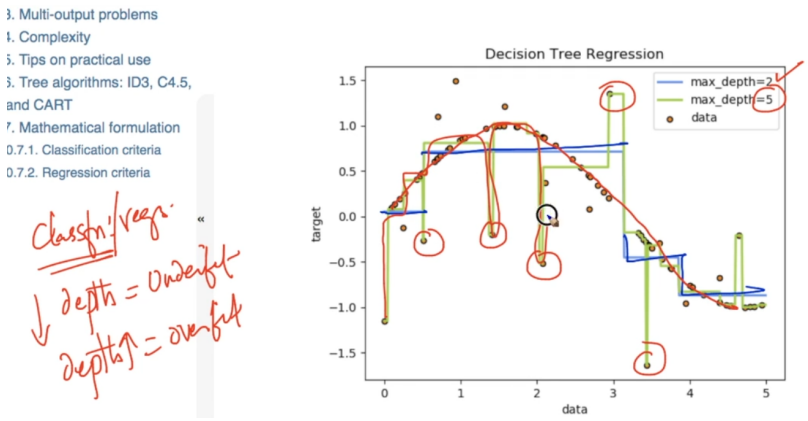
Regression using Decision Tree

- Instead of using Information Gain(IG) we use Mean Square Error(MSE) or Mean Absolute Deviation(MAD) is used to make regression tree.





- At any stage of branch when we break node or data set into two parts using any feature we are trying to reduce error effectively in regression its called error.

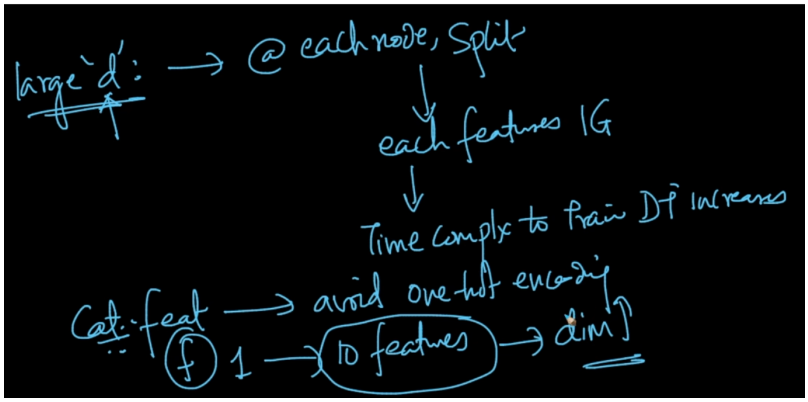


- All lines are axis parallel in decision tree model for regression/classification.

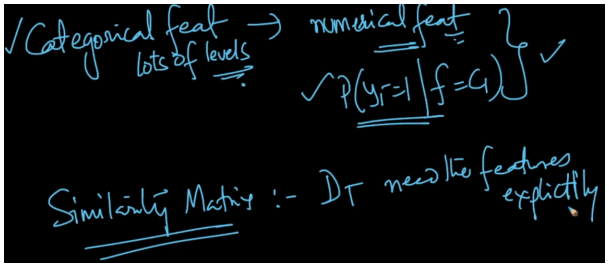
Real World Decision Tree Cases-

- For every imbalance data we first balance it by Upsampling or Downsampling otherwise it will impact on Entropy calc.

- For large data time complexity to train data increases.
- For categorical feature we avoid hot encoding.



- For categorical feature with lots of category/level is useful to convert them into Numerical feature. To avoid Overfitting and sparsity issue.
- Decision tree can read feature explicitly not in case of Similarity Matrix.



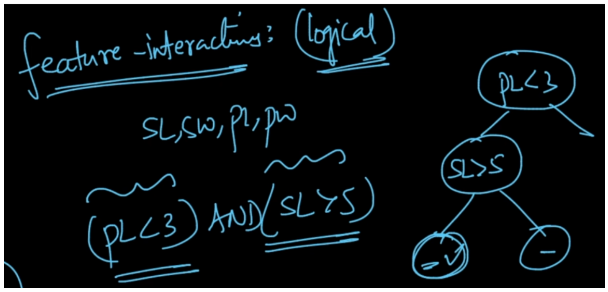
- Decision Tree naturally extended to Multi-class classification.

Decision Surface-

- Decision surface that we get is non-linear.
- It basically divides the data into axis - parallel plane/hyper-plane/hyper-space.

Feature Interaction-

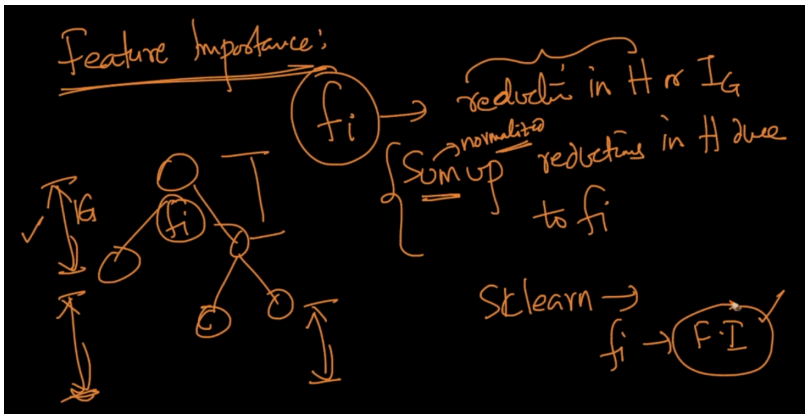
- Comparing two feature is called Feature Interaction.
- FI in Dt to decide the class of query point.



- **Outlier-** When depth is large then model is prone to outlier.
- **Interpretability-** Its very easy in DT.

Feature Importance-

- We can sum up the reduction in entropy of each feature based in the importance of the feature .
- If one feature occur more than once then we conclude that feature is more important.



Advantages -

1. Easy to Understand- Decision tree output is very easy to understand even for people from 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.

2. Useful in Data exploration- Decision tree is one of the fastest way to identify most significant variables and relation between two or more variables. With the help of decision trees, we can create new variables / features that has better power to predict target variable. It can also be used in data

exploration stage. For example, we are working on a problem where we have information available in hundreds of variables, there decision tree will help to identify most significant variable.

3. Less data cleaning required- It requires less data cleaning compared to some other modeling techniques. It is not influenced by outliers and missing values to a fair degree.

4. 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 the space distribution and the classifier structure.

Disadvantages -

1. Over fitting- Over fitting is one of the most practical difficulty for decision tree models. This problem gets solved by setting constraints on model parameters and pruning (discussed in detailed below).

2. Not fit for continuous variables- While working with continuous numerical variables, decision tree looses information when it categorizes variables in different categories.

Refer -

1. https://www.analyticsvidhya.com/blog/2016/04/tree-based-algorithms-complete-tutorial-scratch-in-python/ (https://www.analyticsvidhya.com/blog/2016/04/tree-based-algorithms-complete-tutorial-scratch-in-python/)
2. https://towardsdatascience.com/decision-tree-in-machine-learning-e380942a4c96 (https://towardsdatascience.com/decision-tree-in-machine-learning-e380942a4c96)