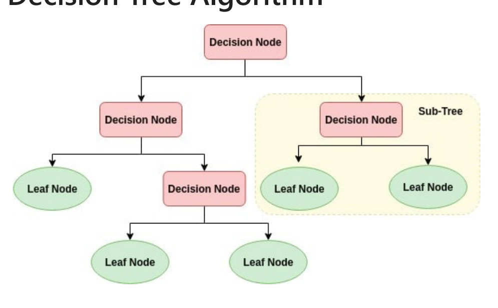
Decision Tree Algorithm



Introduction about the Decision Tree Algorithm.

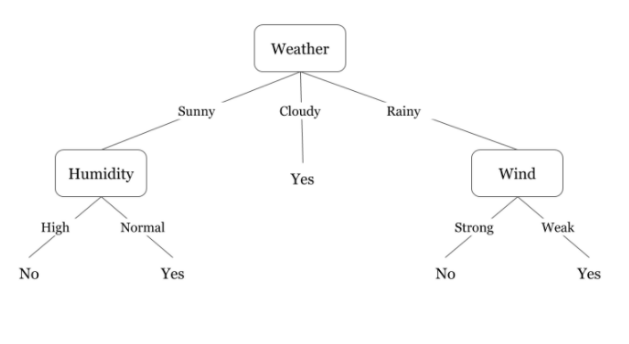
- Decision Trees (DTs) are probably one of the most useful **Non-parametric supervised learning method use for both Regression and Classification task.**
- DTs algorithms are perfect to solve classification (where machines sort data into classes, like whether an email is spam or not) and regression (where machines predict values, like a property price) problems.
- **Regression Trees** are used when the dependent variable is **continuous or quantitative (e.g. if we want to estimate the probability that a customer will default on a loan)**, and **Classification Trees** are used when the dependent variable is **categorical or qualitative (e.g. if we want to estimate the blood type of a person).**
- Our goal is to create a model to predict the value of target variable by using the Regression as well as Classification.
- The importance of DTs relies on the fact that they have lots of applications in the real world. Being one of the mostly used algorithms in ML, they are applied to different functionalities in several industries:

Applications:-

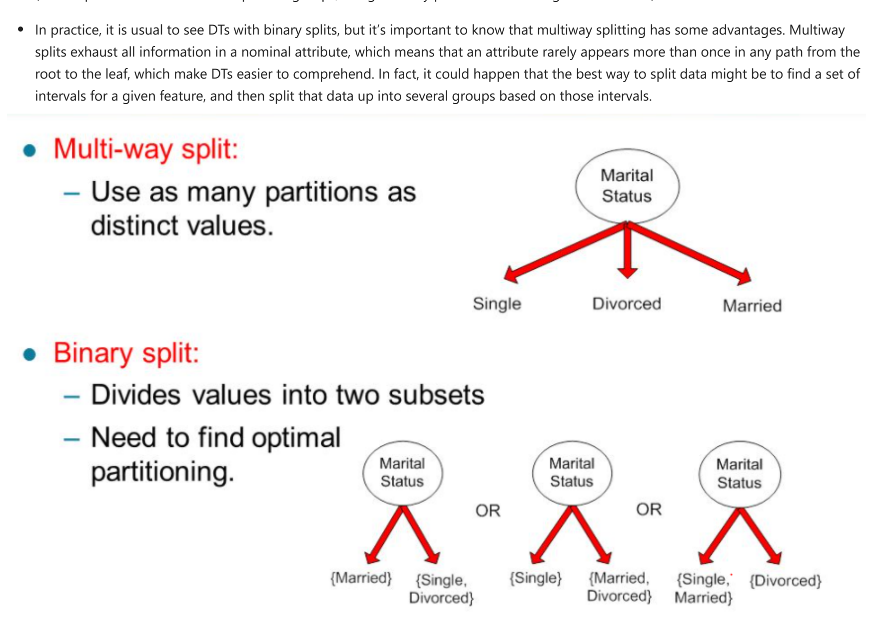
- DTs are being used in the healthcare industry to improve the screening of positive cases in the early detection of cognitive impairment(Perceptional losses), and also to identify the main risk factors of developing some type of dementia in the future.
- https://www.news-medical.net/news/20190117/Scientists-design-two-AI-algorithms-to-improve-early-detection-of-cognitive-impairment.aspx
- Sophia, the robot that was made a citizen of Saudi Arabia, uses DTs algorithms to chat with humans.
- https://www.theverge.com/2017/11/10/16617092/sophia-the-robot-citizen-ai-hanson-robotics-ben-goertzel
- In fact, chatbots that use these algorithms are already bringing benefits in industries like health insurance by gathering data from customers through the application of innovative surveys and friendly chats.
- https://yourstory.com/2018/11/age-alexa-siri-chatbots-next-health-assistants/
- Google recently acquired Onward, a company that uses DTs to develop chatbots that are exceptionally functional in delivering world-class customer care, and Amazon is investing in the same direction to guide customers quickly to a path of resolution.
- https://venturebeat.com/2019/01/10/why-25-of-companies-are-copying-amazons-customer-support-model-using-bots-and-ai-vb-live/
- It is possible to predict the most likely causes of forest disturbances, like wildfire, logging of tree plantations, large or small scale agriculture, and urbanization by training DTs to recognize different causes of forest loss from satellite imagery.
- https://www.europeanscientist.com/en/agriculture/new-analysis-reveals-causes-of-global-forest-loss/
- DTs and satellite imagery are also used in agriculture to classify different crop types and identify their phenological stages.
- DTs are great tools to perform **sentiment analysis of texts**, and identify the emotions behind them. Sentiment analysis is a powerful technique that can help organizations to learn about customers choices and their decision drivers.
- DTs are also used to improve **financial fraud detection.** The MIT showed that it could significantly improve the performance of alternative ML models by using DTs that were trained with several sources of raw data to find patterns of transactions and credit cards that match cases of fraud.
- http://news.mit.edu/2018/machine-learning-financial-credit-card-fraud-0920
- The firm Sesame Credit (a company affiliated with Alibaba) uses DTs and other algorithms to engine a system of social evaluation, taking into consideration various factors such as the punctuality with which bills are paid and other online activities.
- Actually, after the Chinese government announced it will apply its so-called social credit system to flights and trains and stop people who have committed misdeeds (misdoing or sin) from taking such transport for up to a year.
- https://bluenotes.anz.com/posts/2016/11/china-credit-and-seeing-unforseen-risk
- https://aleteia.org/2018/06/28/the-chinese-social-credit-system-the-real-big-brother-of-the-future/

Structure & Working Principle of the Decision Tree.

- DTs are composed of nodes, branches and leafs. Each node represents an attribute (or feature), each branch represents a rule (or decision), and each leaf represents an outcome. The depth of a Tree is made by the number of levels, not including the root node because the root node contains the more information than the other that why it will be on the top.
- Moreover the descision tree is like flowchart tree structure where an internal node represents the feature and branch represents the decision rule.
- Each node in the DT acts as a test case for some condition, and each branch descending from that node corresponds to one of the possible answers to that test case.
- It learn the partition on the basis of the top most feature value which is widely known as atrribute value or root value.It partition the tree in recursive manner call recursive partitioning.
- The flowchart like structure help you in decision making kind of human decision making.So,it visualization like flowchart diagram which easily mimics the human thinking.
- That is why Decision tree is easy to understand and interpret.
- Morever,We can say the decision tree classify the example by sorting them down tree from the root to the leaf nodes.
- Leaf node providing the classification to the example that is final lable to the any rows of the data.
- Each edges or line decending from the node to one of the possible answer to the test cases.
- This process recursive in nature.
- It is repeated for every sub-tree root to the new node.



- DTs apply a top-down approach to data, so that given a data set, they try to group and label observations that are similar between them, and look for the best rules that split the observations that are dissimilar between them until they reach certain degree of similarity.
- [illegible]



Why The Decision Tree ?

- [illegible]

How The Decision Tree Works ?

- [illegible]

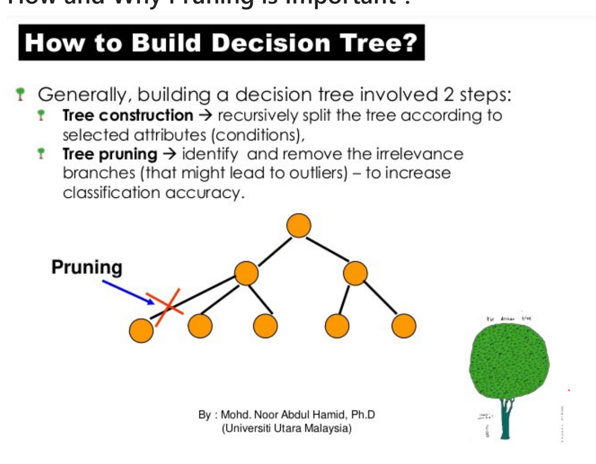
Steps follow by Decision tree while operation:-

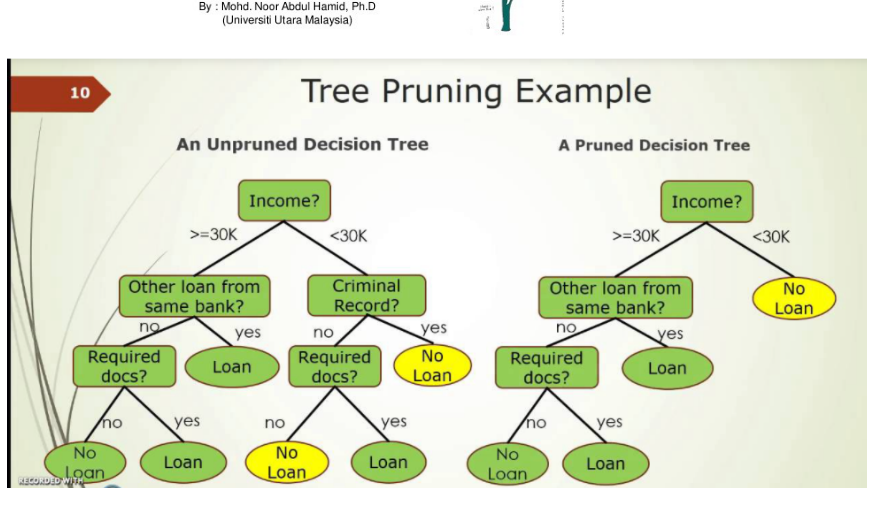
- [illegible]

Pruning Of The Decision Tree.

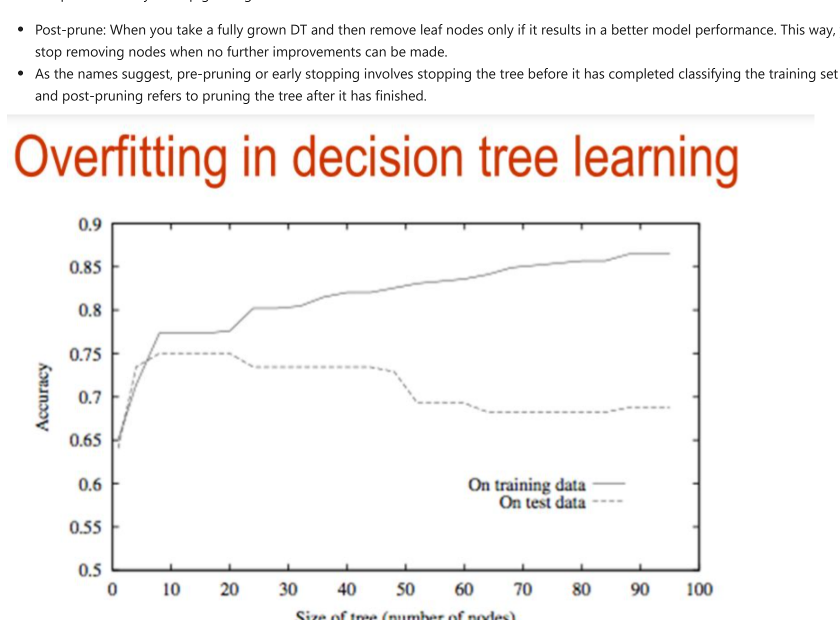
- [illegible]
- It help to improve the accuracy and speed of the decision tree.

How and Why Pruning is Important ?





- As the number of splits in DTs increase, their complexity rises. In general, simpler DTs are preferred over super complex ones, since they are easier to understand and they are less likely to fall into overfitting.
- Overfitting refers to a model that learns the training data (the data it uses to learn) so well that it has problems to generalize to new (unseen) data.
- In other words, the model learns the detail and noise (irrelevant information or randomness in a dataset) in the training data to the extent that it negatively impacts the performance of the model on new data. This means that the noise or random fluctuations in the training data is picked up and learned as concepts by the model.
- Under this condition, your model works perfectly well with the data you provide upfront, but when you expose that same model to new data, it breaks down. It's unable to repeat its highly detailed performance.
- So, how do you avoid overfitting in DTs? You need to exclude branches that fit data too specifically. You want a DT that can generalize and work well on new data, even though this may imply losing precision on the training data. It's always better to avoid a DT model that learns and repeats specific details like a parrot, and try to develop one that has the power and flexibility to have a decent performance on new data you provide to it.
- Pruning is a technique used to deal with overfitting, that reduces the size of DTs by removing sections of the Tree that provide little predictive or classification power.
- The goal of this procedure is to reduce complexity and gain better accuracy by reducing the effects of overfitting and removing sections of the DT that may be based on noisy or erroneous data. There are two different strategies to perform pruning on DTs:
- Pre-prune: When you stop growing DT branches when information becomes unreliable.
- Post-prune: When you take a fully grown DT and then remove leaf nodes only if it results in a better model performance. This way, you stop removing nodes when no further improvements can be made.
- As the names suggest, pre-pruning or early stopping involves stopping the tree before it has completed classifying the training set and post-pruning refers to pruning the tree after it has finished.



- In summary, a big DT that correctly classifies or predicts every example of the training data might not be as good as a smaller one that does not fit all the training data perfectly.

Main DTs algorithms.

- Now you may ask yourself: how do DTs know which features to select and how to split the data? To understand that, we need to get into some details.
- All DTs perform basically the same task: they examine all the attributes of the dataset to find the ones that give the best possible result by splitting the data into subgroups. They perform this task recursively by splitting subgroups into smaller and smaller units until the Tree is finished (stopped by certain criteria).
- This decision of making splits heavily affects the Tree's accuracy and performance, and for that decision, DTs can use different algorithms that differ in the possible structure of the Tree (e.g. the number of splits per node), the criteria on how to perform the splits, and when to stop splitting.
- So, how can we define which attributes to split, when and how to split them? To answer this question, we must review the main DTs algorithm:

Types of the Decision Tree Algorithm.

- Here couple of algorithms to build a decision tree.
- CART = CART is the Classification and Regression tree and it uses Gini Index as Attribute Selection Measures.
- ID3 = ID3 is use as entropy function and information gain as attribute selection measures.
- **CART(Classification and Regression Trees) :-** uses the Gini Index(Classification) as metric.
- **ID3(Iterative Dichotomiser):-** Uses the entropy function and information gain as metric.

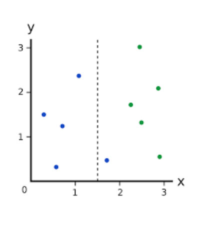
CHAID

- The Chi-squared Automatic Interaction Detection (CHAID) is one of the oldest DT algorithms methods that produces **multiway** DTs (splits can have more than two branches) suitable for classification and regression tasks.
- When building **Classification Trees** (where the dependent variable is categorical in nature), CHAID relies on the **Chi-square independence tests** to determine the best split at each step. Chi-square tests check if there is a relationship between two variables, and are applied at each stage of the DT to ensure that each branch is significantly associated with a statistically significant predictor of the response variable.
- In other words, it chooses the independent variable that has the strongest interaction with the dependent variable.
- Additionally, categories of each predictor are merged if they are not significantly different between each other, with respect to the dependent variable. In the case of **Regression Trees** (where the dependent variable is continuous), CHAID relies on **F-tests** (instead of Chi-square tests) to calculate the difference between two population means. If the F-test is significant, a new partition (child node) is created (which means that the partition is statistically different from the parent node). On the other hand, if the result of the F-test between target means is not significant, the categories are merged into a single node.
- CHAID does not replace missing values and handles them as a single class which may merge with another class if appropriate. It also produces DTs that tend to be wider rather than deeper (multiway characteristic), which may be unrealistically short and hard to relate to real business conditions. Additionally, it has no pruning function.
- Although not the most powerful (in terms of detecting the smallest possible differences) or fastest DT algorithm out there, CHAID is easy to manage, flexible and can be very useful.

CART.

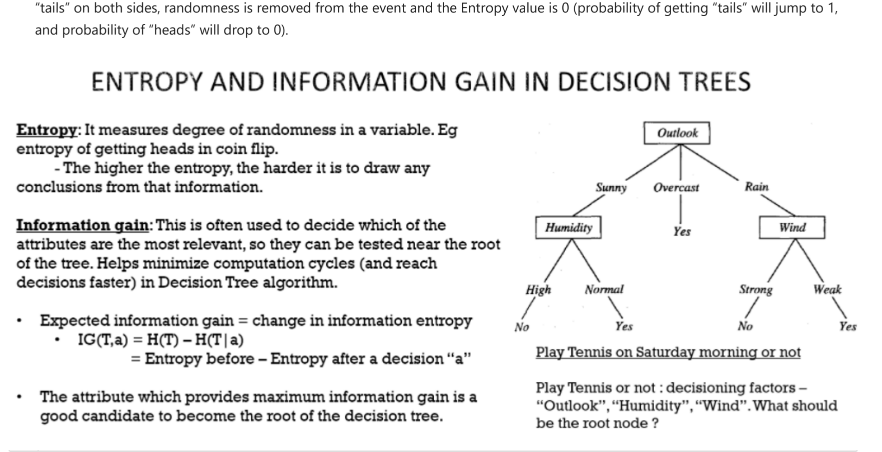
- CART is a DT algorithm that produces binary Classification or Regression Trees, depending on whether the dependent (or target) variable is categorical or numeric, respectively. It handles data in its raw form (no preprocessing needed), and can use the same variables more than once in different parts of the same DT, which may uncover complex interdependencies between sets of variables.
- In the case of Classification Trees, CART algorithm uses a metric called **Gini Impurity** to create decision points for classification tasks. Gini Impurity gives an idea of how fine a split is (a measure of a node's "purity"), by how mixed the classes are in the two groups created by the split. When all observations belong to the same label, there's a perfect classification and a Gini Impurity value of 0 (minimum value). On the other hand, when all observations are equally distributed among different labels, we face the worst case split result and a Gini Impurity value of 1 (maximum value).
- In the case of Regression Trees, CART algorithm looks for splits that minimize the Least Square Deviation (LSD), choosing the partitions that minimize the result over all possible options. The LSD (sometimes referred as "variance reduction") metric minimizes the sum of the squared distances (or deviations) between the observed values and the predicted values. The difference between the predicted and observed values is called "residual", which means that LSD chooses the parameter estimates so that the sum of the squared residuals is minimized.
- LSD is well suited for metric data and has the ability to correctly capture more information about the quality of the split than other algorithms.
- The idea behind CART algorithm is to produce a sequence of DTs, each of which is a candidate to be the "optimal Tree". This optimal Tree is identified by evaluating the performance of every Tree through testing (using new data, which the DT has never seen before) or performing cross-validation (dividing the dataset into "k" number of folds, and perform testings on each fold).
- CART doesn't use an internal performance measure for Tree selection. Instead, DTs performances are always measured through testing or via cross-validation, and the Tree selection proceeds only after this evaluation has been done.

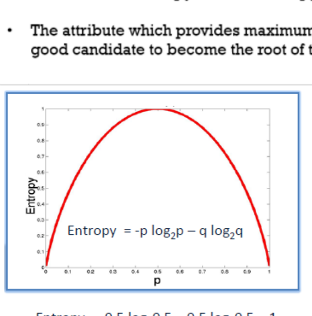
Not Purely Classified means there is Gini Impurity



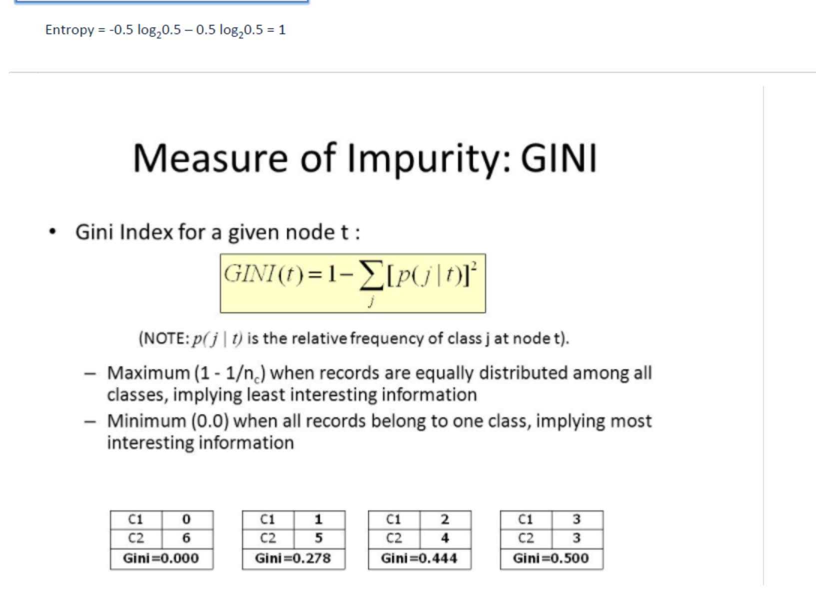
ID3.

- The Iterative Dichotomiser 3 (ID3) is a DT algorithm that is mainly used to produce Classification Trees. Since it hasn't proved to be so effective building Regression Trees in its raw data, ID3 is mostly used for classification tasks (although some techniques such as building numerical intervals can improve its performance on Regression Trees).
- ID3 splits data attributes (dichotomizes) to find the most dominant features, performing this process iteratively to select the DT nodes in a top-down approach.
- For the splitting process, ID3 uses the **Information Gain metric** to select the most useful attributes for classification. Information Gain is a concept extracted from Information Theory, that refers to the decrease in the level of randomness in a set of data: basically it measures how much "information" a feature gives us about a class. ID3 will always try to maximize this metric, which means that the **attribute with the highest Information Gain will split first.**
- Information Gain is directly linked to the concept of Entropy, which is the measure of the amount of uncertainty or randomness in the data. Entropy values range from 0 (when all members belong to the same class or the sample is completely homogeneous) to 1 (when there is perfect randomness or unpredictability, or the sample is equally divided).
- You can think it this way: if you want to make an unbiased coin toss, there is complete randomness or an Entropy value of 1 ("heads" and "tails" are equally like, with a probability of 0.5 each). On the other hand, if you make a coin toss, with for example a coin that has "tails" on both sides, randomness is removed from the event and the Entropy value is 0 (probability of getting "tails" will jump to 1, and probability of "heads" will drop to 0).





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



- In this graph you can see the relationship between Entropy and the probability of different coin tosses. At the highest level of Entropy, the probability of getting "tails" is equal to the one of getting "heads" (0.5 each), and we face complete uncertainty. Entropy is directly linked to the probability of an event.
- This is important because Information Gain is the decrease in Entropy, and the attribute that yields the largest Information Gain is chosen for the DT node.
- But ID3 has some disadvantages: it can't handle numeric attributes nor missing values, which can represent serious limitations.

C4.5

- C4.5 is the successor of ID3 and represents an improvement in several aspects. C4.5 can handle both continuous and categorical data, making it suitable to generate Regression and Classification Trees. Additionally, it can deal with missing values by ignoring instances that include non-existing data.
- Unlike ID3 (which uses Information Gain as splitting criteria), C4.5 uses **Gain Ratio** for its splitting process.
- Gain Ratio is a modification of the Information Gain concept that reduces the bias on DTs with huge amount of branches, by taking into account the number and size of the branches when choosing an attribute. Since Information Gain shows an unfair favoritism towards attributes with many outcomes, Gain Ratio corrects this trend by considering the intrinsic information of each split (it basically "normalizes" the Information Gain by using a split information value). This way, the attribute with the maximum Gain Ratio is selected as the splitting attribute.
- Additionally, C4.5 includes a technique called windowing, which was originally developed to overcome the memory limitations of earlier computers. Windowing means that the algorithm randomly selects a subset of the training data (called a "window") and builds a DT from that selection. This DT is then used to classify the remaining training data, and if it performs a correct classification, the DT is finished. Otherwise, all the misclassified data points are added to the windows, and the cycle repeats until every instance in the training set is correctly classified by the current DT. This technique generally results in DTs that are more accurate than those produced by the standard process due to the use of randomization, since it captures all the "rare" instances together with sufficient "ordinary" cases.
- Another capability of C4.5 is that it can prune DTs.
- C4.5's pruning method is based on estimating the error rate of every internal node, and replacing it with a leaf node if the estimated error of the leaf is lower. In simple terms, if the algorithm estimates that the DT will be more accurate if the "children" of a node are deleted and that node is made a leaf node, then C4.5 will delete those children.

DARKNESS OF THE DTs:-

- DTs tend to overfit on their training data, making them perform badly if data previously shown to them doesn't match to what they are shown later
- They also suffer from high variance, which means that a small change in the data can result in a very different set of splits, making interpretation somewhat complex. They suffer from an inherent instability, since due to their hierarchical nature, the effect of an error in the top splits propagate down to all of the splits below.
- In Classification Trees, the consequences of misclassifying observations are more serious in some classes than others. For example, it is probably worse to predict that a person will not have a heart attack when he/she actually will, than vice versa. This problem is mitigated in algorithms like C5.0, but remains as a serious issue in others.
- DTs can also create biased Trees if some classes dominate over others. This is a problem in unbalanced datasets (where different classes in the dataset have different number of observations), in which case it is recommended to balance dataset prior to building the DT.
- In the case of Regression Trees, DTs can only predict within the range of values they created based on the data they saw before, which means that they have boundaries on the values they can produce.
- At each level, DTs look for the best possible split so that they optimize the corresponding splitting criteria.
- But DTs splitting algorithms can't see far beyond the current level in which they are operating (they are "greedy"), which means that they look for a locally optimal and not a globally optimal at each step.
- DTs algorithms grow Trees one node at a time according to some splitting criteria and don't implement any backtracking technique. .

Decision Making in DT with Attribute Selection Measures(ASM)

- Information Gain.
- Gain Ratio.
- Gini Index..

Information Gain

- The information gain is based on the entropy therefore we first need to understand about the entropy.
- Entropy- it is amout of uncartainty avalible in the dataset.More uncartainty more entropy in datasets.
- Based on the entropy we compute the information gain.
- `Information gain = entropy gain = (entropy before the split attribute - entropy after the split on attribute )`
- `Total IG= (entropy of the root node - sum of entropy of the leaf node)`
- Compute the entropy for datasets for every feature.:
  - 1.Calculate entropy for all categorical values.
  - 2.Take avarage information entropy for a current attribute.
  - 3.Calculate the gain for the current attribute and pick the highest gain attribute.
  - 4.Reapeat till we get the tree we desired.
- Based on the imformation gain ALGORITHMS will take decision either or .

Gain Ratio

- Treats all variable the same regardless of their distribution and their important.
- This is problem when we comes to the continuous variable and descreate variable with many posssible values because its training examples.
- So,Alternative measure to information gain is the gain ratio.so,The gain ratio tries to correct the information gain baised towards the attribute with many possible values by adding denominator to information gain called split information.
- This split information tries to measure how broadly and uniformly the attribute split the data.

Gini Index

- Gini Index is a measurement of the likehood incorrect classification of new instance of random variable .If that new instance were randomly classified according to the distribution class label from the datasets.
- If our model is pure then likehood of incorrect classification is 0.If our sample of mixture of diffrent classes then likehood of incorret classification will be high.

Opmization parameter For Decision Tree.

- **Criterion :-** optional (default = gini) or choose the attribute selection measure : This is allow us to use the diffrent-diffrent attribute selection measure.

  support criteria 'gini' for the Gini index and entropy for the information gain.

Splitter-string, optional (default = best) or split strategy :This parameter also use the split strategy.Supported strategies are best to choose the best split and random best split.

- **Max-depth-** int or None , optional (default=None) or maximum depth of the tree : If None ,the n nodes are expanded until all leaves contain less min_sample_split sample.The higher value of maximum depth cause overfitting , and a lower causes overfitting (source).

When to stop splitting ?

- You might ask when to stop growing a tree ? As a problem usually large set of features,it results in large number of split,which in turn gives us huge tree.
- Such tree are complex and can lead to overfitting.So,we need to know when to stop ?
- As problem usually large set of feature it result in large number of split which return the huge tree and such trees are complex and can lead to the overfitting that is where we pass the max_depth
- Sometimes we can set max_depth in particular let say 10,15 and 100 then then decision will not go beyond 10,50 or 100 and then it will stop.
  - set max_depth

Another way dealing with this is called Prouning.

Thanks !!!