For taking steps to know about Data Science and Machine Learning, till now in my blogs, I have covered briefly an introduction to Data Science, Python, Statistics, Machine Learning, Regression, Linear and Logistic Regression. In this fifth of the series, I shall cover Decision Trees.

Introduction to Decision Trees :

A **decision tree** is a decision support tool that uses a tree-like graph or model of decisions and their possible consequences, including chance event outcomes, resource costs, and utility. It is one way to display an algorithm that only contains conditional control statements.

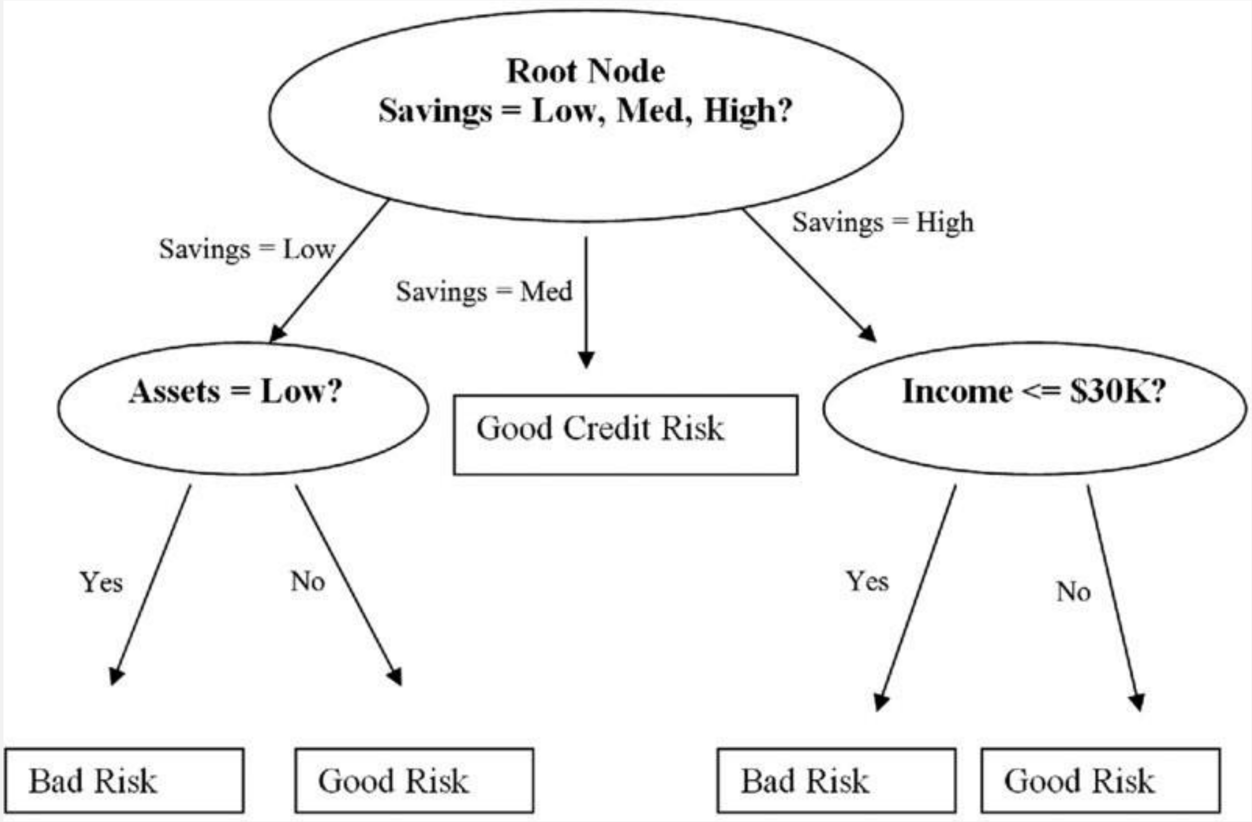
A decision tree is a flowchart-like structure in which each internal node represents a “test” on an attribute (e.g. whether a coin flip comes up heads or tails), each branch represents the outcome of the test, and each leaf node represents a class label (decision taken after computing all attributes). The paths from root to leaf represent classification rules.

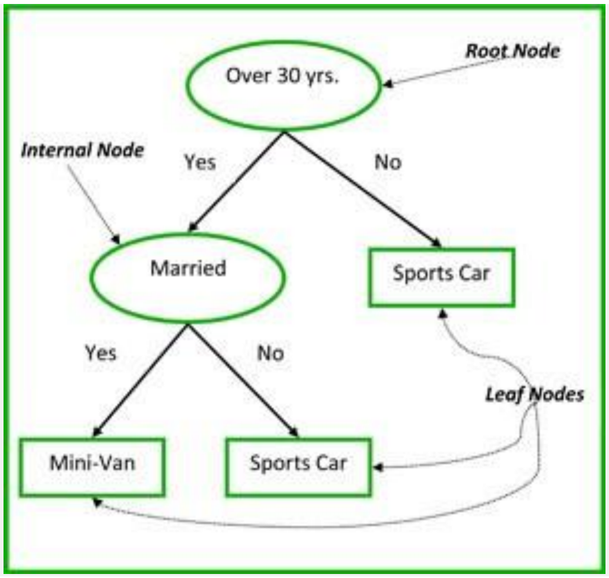
Tree based learning algorithms are considered to be one of the best and mostly used supervised learning methods. Tree based methods empower predictive models with high accuracy, stability and ease of interpretation. Unlike linear models, they map non-linear relationships quite well. They are adaptable at solving any kind of problem at hand (classification or regression). Decision Tree algorithms are referred to as **CART (Classification and Regression Trees)**.

"The possible solutions to a given problem emerge as the leaves of a tree, each node representing a point of deliberation and decision."

- Niklaus Wirth (1934 —), Programming language designer

Methods like decision trees, random forest, gradient boosting are being popularly used in all kinds of data science problems.

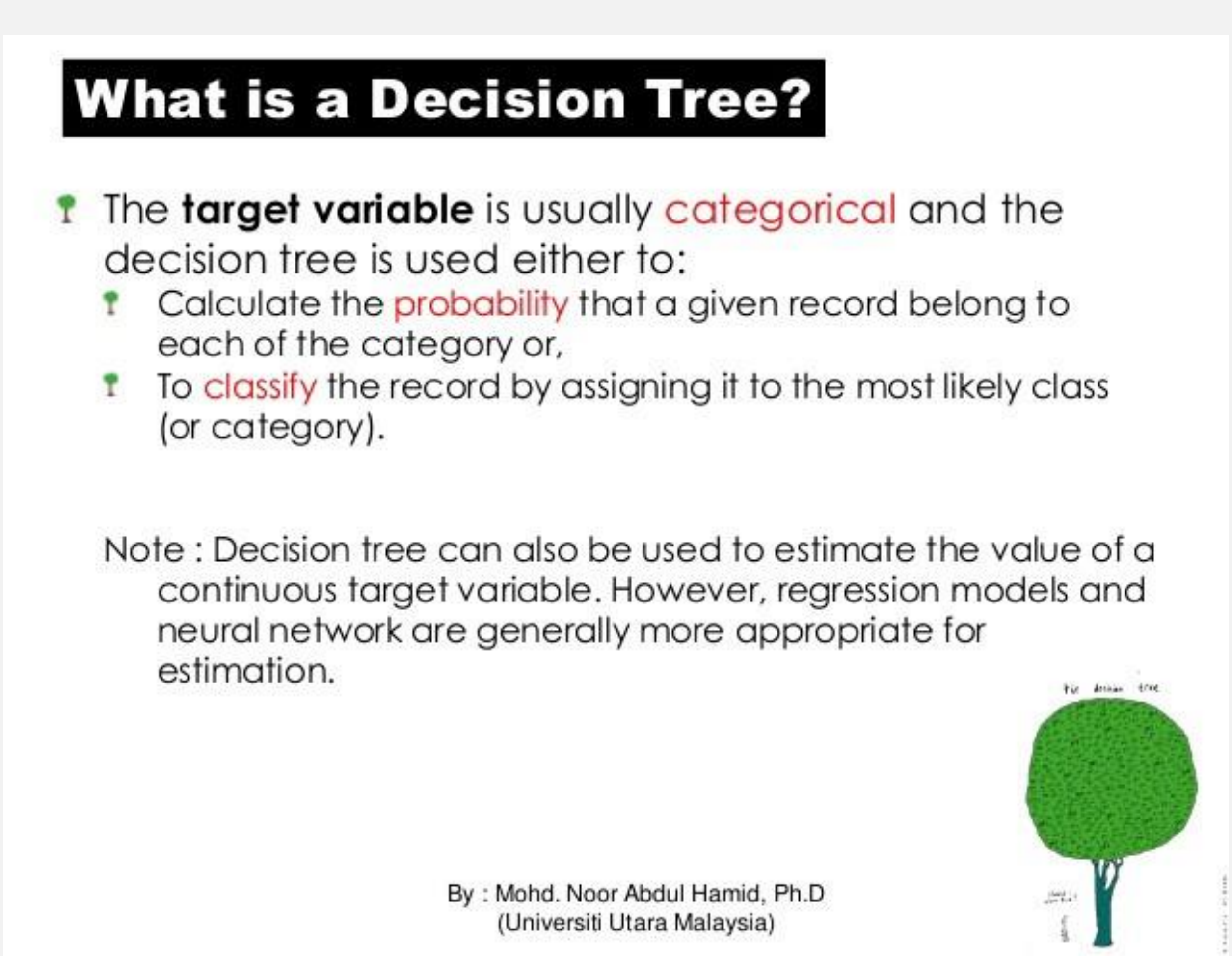




Common terms used with Decision trees:

1. **Root Node:** It represents entire population or sample and this further gets divided into two or more homogeneous sets.
2. **Splitting:** It is a process of dividing a node into two or more sub-nodes.
3. **Decision Node:** When a sub-node splits into further sub-nodes, then it is called decision node.
4. **Leaf/ Terminal Node:** Nodes do not split is called Leaf or Terminal node.
5. **Pruning:** When we remove sub-nodes of a decision node, this process is called pruning. You can say opposite process of splitting.

6. **Branch / Sub-Tree:** A sub section of entire tree is called branch or sub-tree.
7. **Parent and Child Node:** A node, which is divided into sub-nodes is called parent node of sub-nodes whereas sub-nodes are the child of parent node.



What is a Decision Tree?

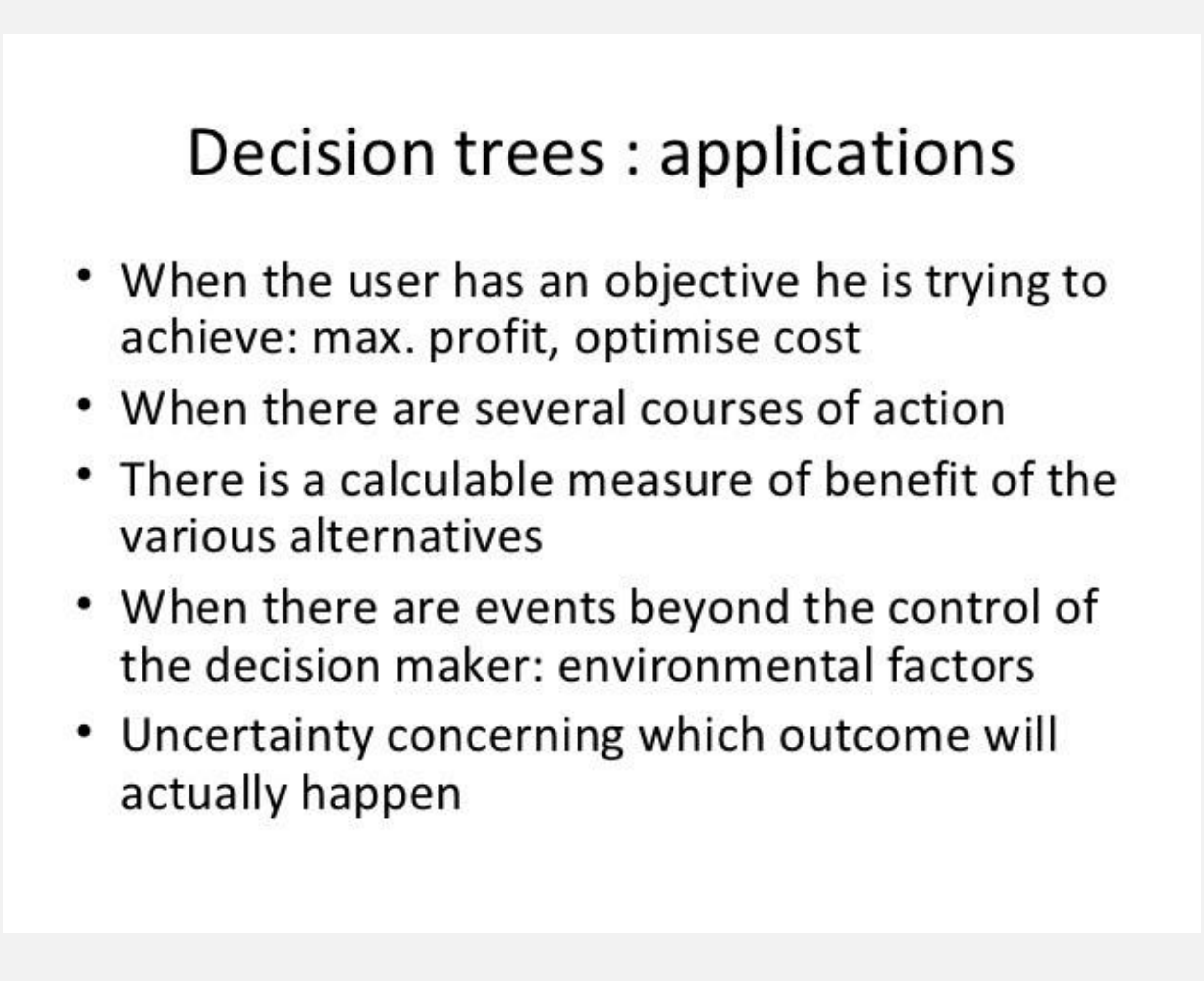
- The **target variable** is usually categorical and the decision tree is used either to:
  - Calculate the probability that a given record belong to each of the category or,
  - To classify the record by assigning it to the most likely class (or category).

Note : Decision tree can also be used to estimate the value of a continuous target variable. However, regression models and neural network are generally more appropriate for estimation.

By : Mohd. Noor Abdul Hamid, Ph.D
(Universiti Utara Malaysia)

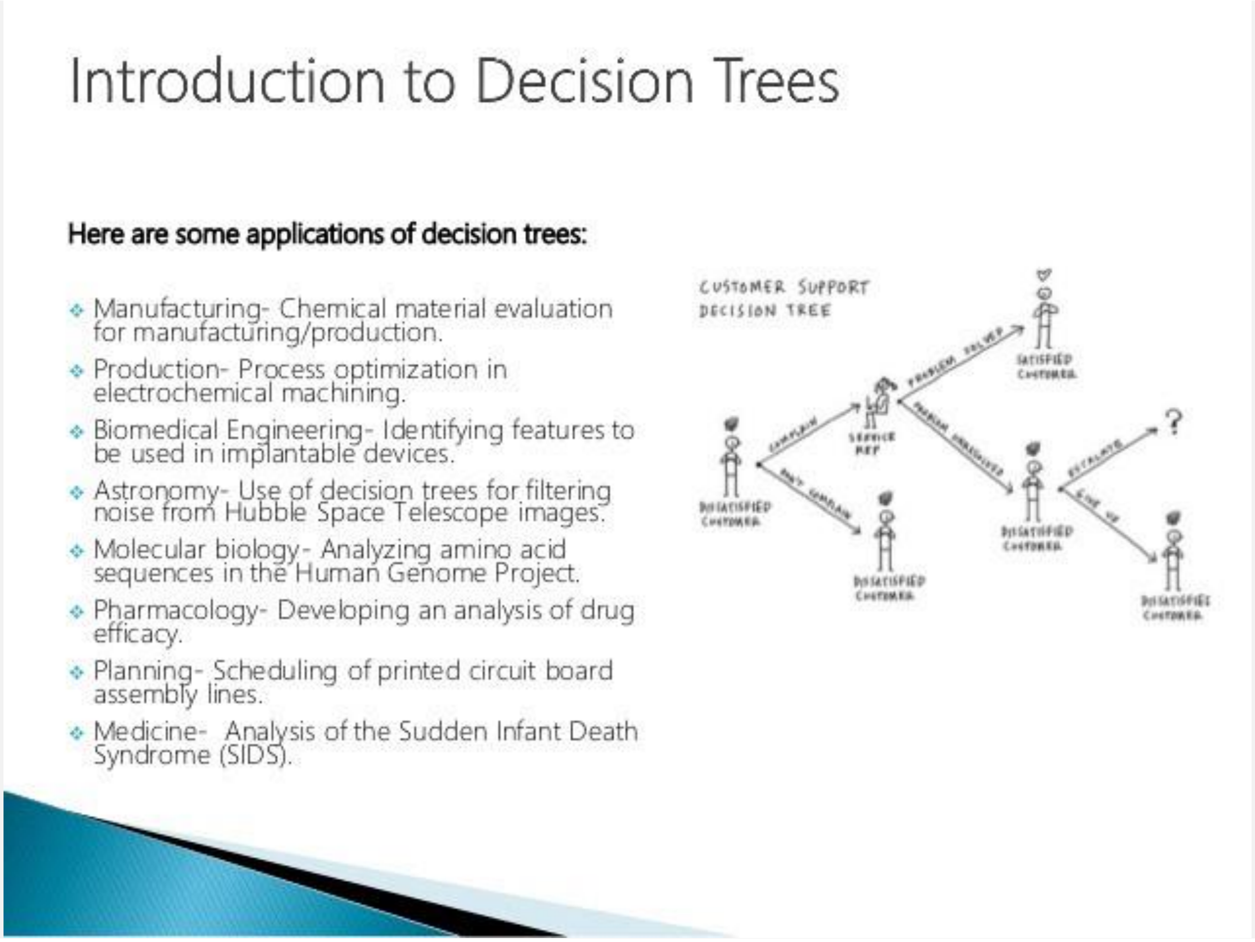
Applications for Decision Tree :

Decision trees have a natural “if … then … else …” construction that makes it fit easily into a programmatic structure. They also are well suited to categorization problems where attributes or features are systematically checked to determine a final category. For example, a decision tree could be used effectively to determine the species of an animal.



Decision trees : applications

- When the user has an objective he is trying to achieve: max. profit, optimise cost
- When there are several courses of action
- There is a calculable measure of benefit of the various alternatives
- When there are events beyond the control of the decision maker: environmental factors
- Uncertainty concerning which outcome will actually happen



Introduction to Decision Trees

Here are some applications of decision trees:

- Manufacturing- Chemical material evaluation for manufacturing/production.
- Production- Process optimization in electrochemical machining.
- Biomedical Engineering- Identifying features to be used in implantable devices.
- Astronomy- Use of decision trees for filtering noise from Hubble Space Telescope images.
- Molecular biology- Analyzing amino acid sequences in the Human Genome Project.
- Pharmacology- Developing an analysis of drug efficacy.
- Planning- Scheduling of printed circuit board assembly lines.
- Medicine- Analysis of the Sudden Infant Death Syndrome (SIDS).

As a result, the decision making tree is one of the more popular classification algorithms being used in Data Mining and Machine Learning. Example applications include:

· Evaluation of brand expansion opportunities for a business using historical sales data

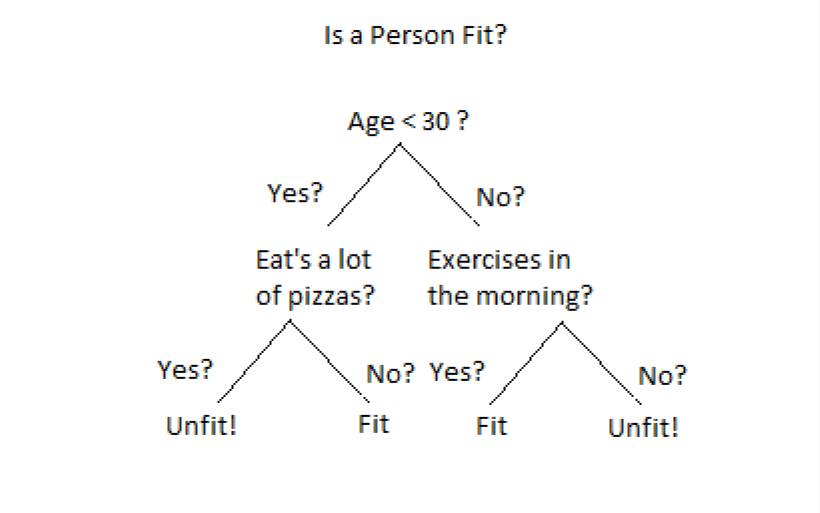
· Determination of likely buyers of a product using demographic data to enable targeting of limited advertisement budget

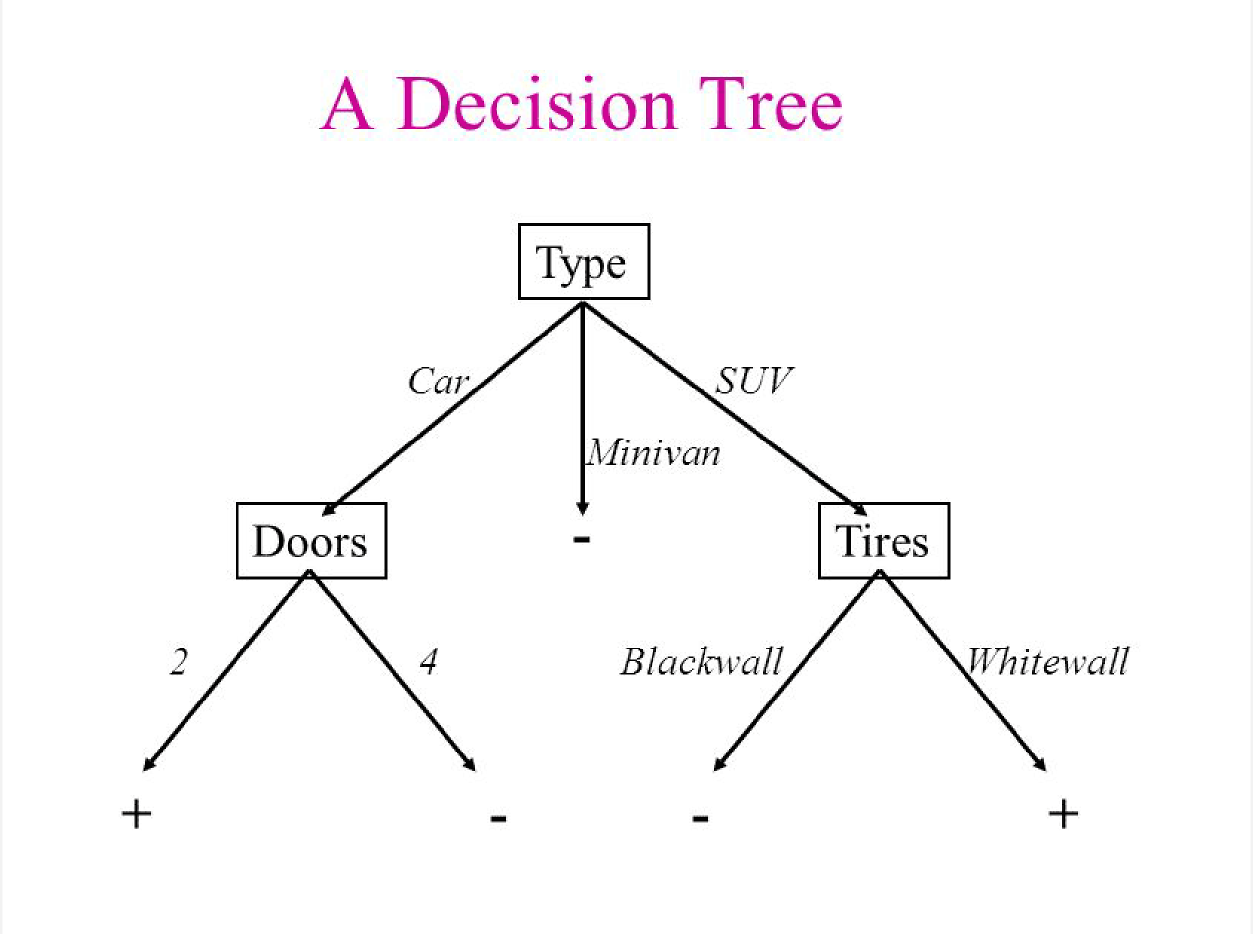
· Prediction of likelihood of default for applicant borrowers using predictive models generated from historical data

· Help with prioritization of emergency room patient treatment using a predictive model based on factors such as age, blood pressure, gender, location and severity of pain, and other measurements

· Decision trees are commonly used in operations research, specifically in decision analysis, to help identify a strategy most likely to reach a goal.

Because of their simplicity, tree diagrams have been used in a broad range of industries and disciplines including civil planning, energy, financial, engineering, healthcare, pharmaceutical, education, law, and business.





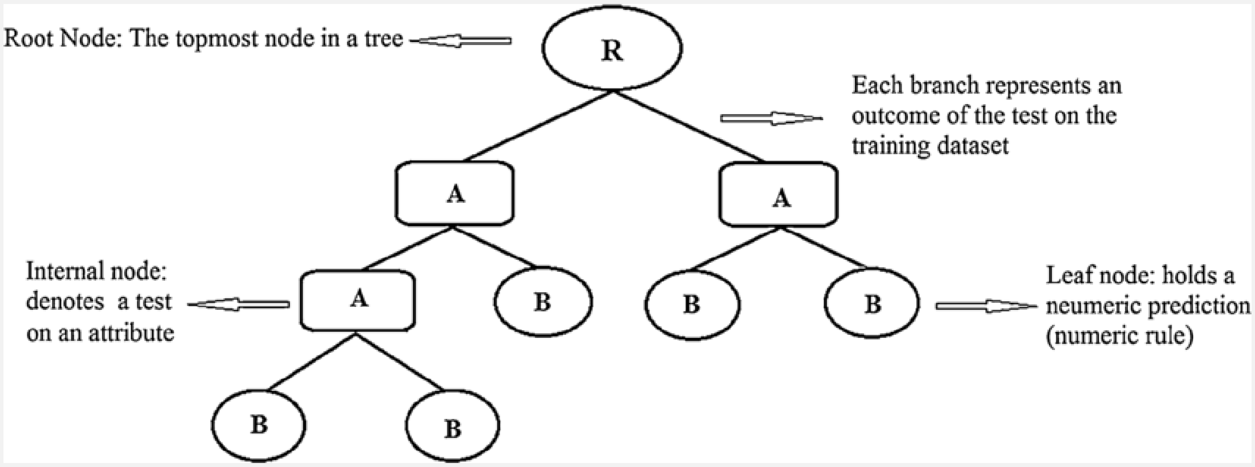
How does Decision Tree works ?

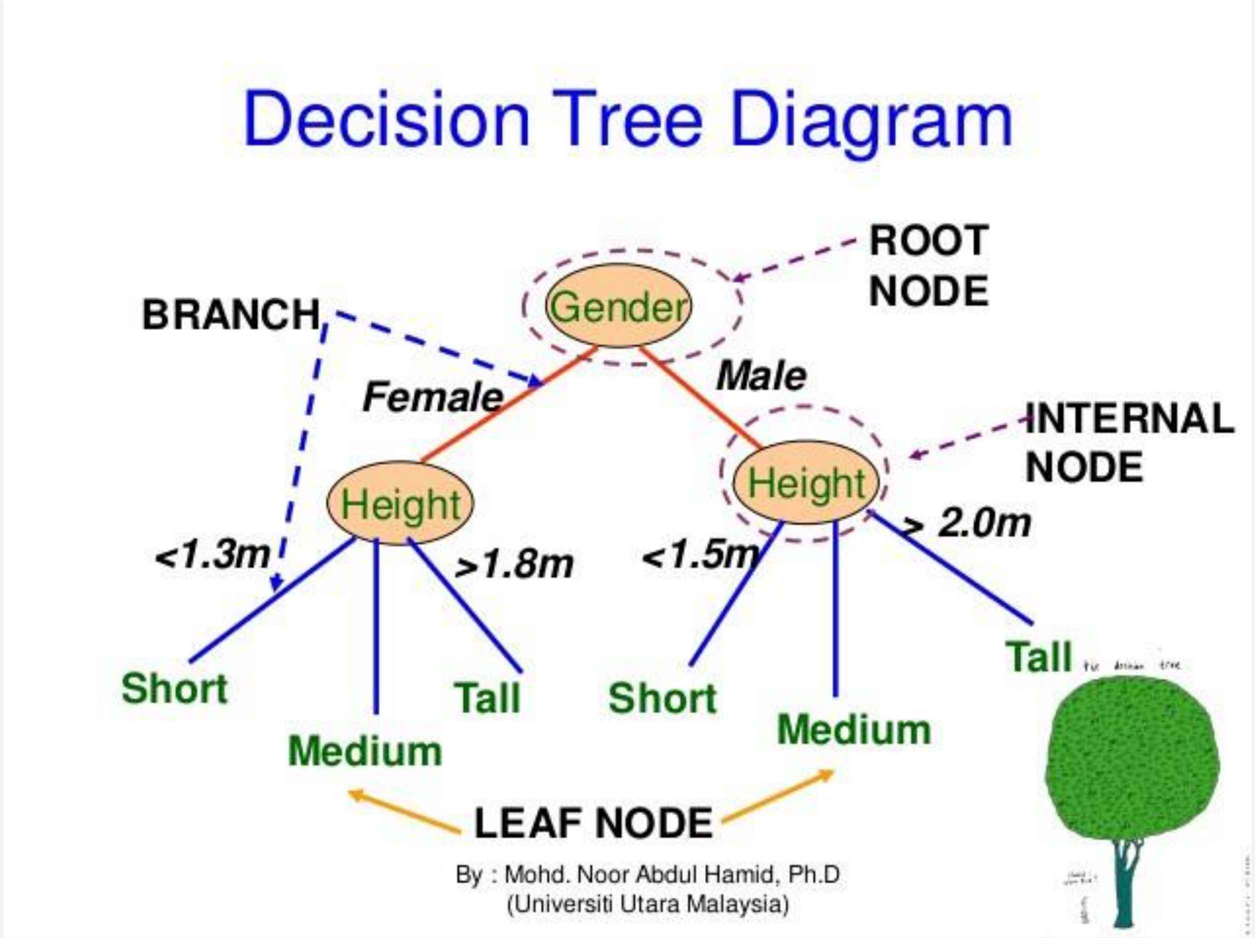
Decision tree is a type of supervised learning algorithm (having a pre-defined target variable) that is mostly used in classification problems. It works for both categorical and continuous input and output variables. In this technique, we split the population or sample into two or more homogeneous sets (or sub-populations) based on most significant splitter / differentiator in input variables.

Example:-

Let’s say we have a sample of 30 students with three variables *Gender* (Boy/ Girl), *Class* (IX/ X) and *Height* (5 to 6 ft). 15 out of these 30 play cricket in leisure time. Now, we want to create a model to predict who will play cricket during leisure period? In this problem, we need to segregate students who play cricket in their leisure time based on highly significant input variable among all three.

This is where decision tree helps, it will segregate the students based on all values of three variable and identify the variable, which creates the best homogeneous sets of students (which are heterogeneous to each other). In the snapshot below, you can see that variable Gender is able to identify best homogeneous sets compared to the other two variables.


Root Node: The topmost node in a tree
R
Each branch represents an outcome of the test on the training dataset
A
A
Internal node: denotes a test on an attribute
A
B
B
B
Leaf node: holds a neumeric prediction (numeric rule)
B
B



Decision tree identifies the most significant variable and its value that gives best homogeneous sets of population. To identify the variable and the split, decision tree uses various algorithms.

Types of Decision Trees

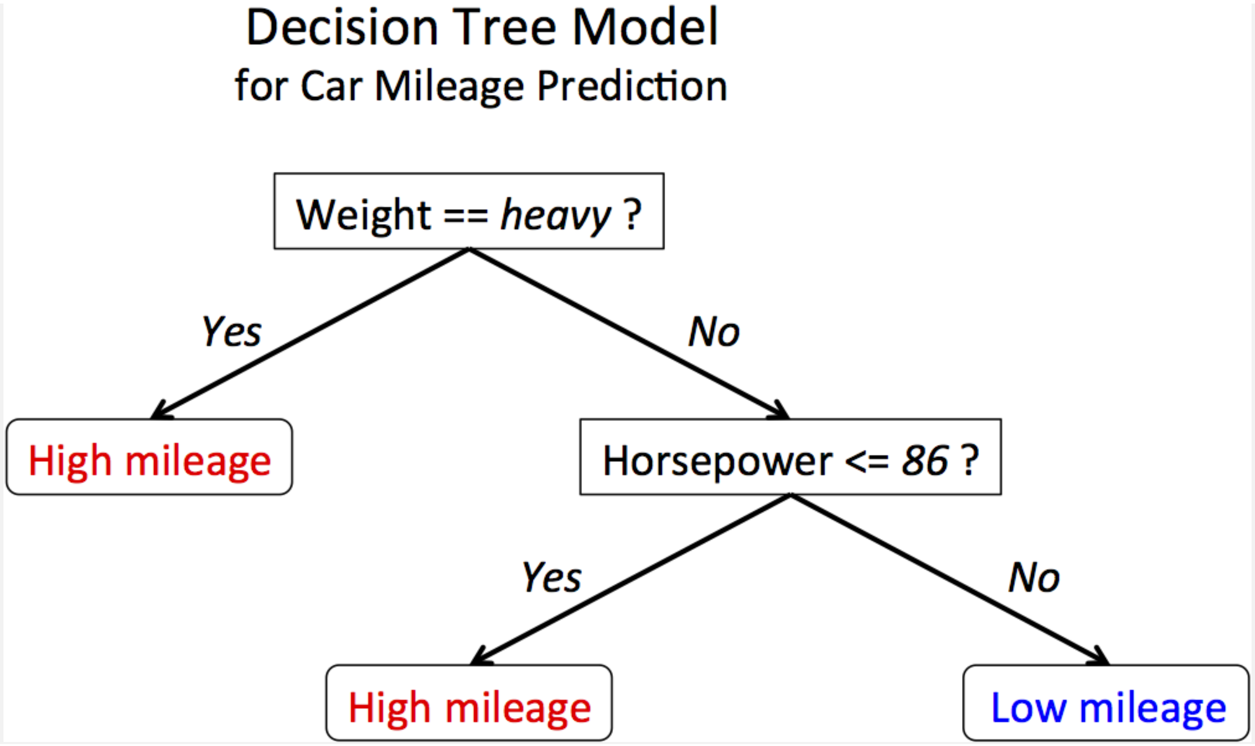
Types of decision tree is based on the type of target variable we have. It can be of two types:

1. **Categorical Variable Decision Tree:** Decision Tree which has categorical target variable then it called as

categorical variable decision tree. E.g.:- In above scenario of student problem, where the target variable was “Student will play cricket 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.

E.g.:- 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.



Decision Tree Algorithm Pseudocode

The decision tree algorithm tries to solve the problem, by using tree representation. Each internal node of the tree corresponds to an attribute, and each leaf node corresponds to a class label.

1. Place the best attribute of the dataset at the **root** of the tree.
2. Split the training set into **subsets**. Subsets should be made in such a way that each subset contains data with the same value for an attribute.

3. Repeat step 1 and step 2 on each subset until you find **leaf nodes** in all the branches of the tree.

In decision trees, for predicting a class label for a record we start from the **root** of the tree. We compare the values of the root attribute with record's attribute. On the basis of comparison, we follow the branch corresponding to that value and jump to the next node.

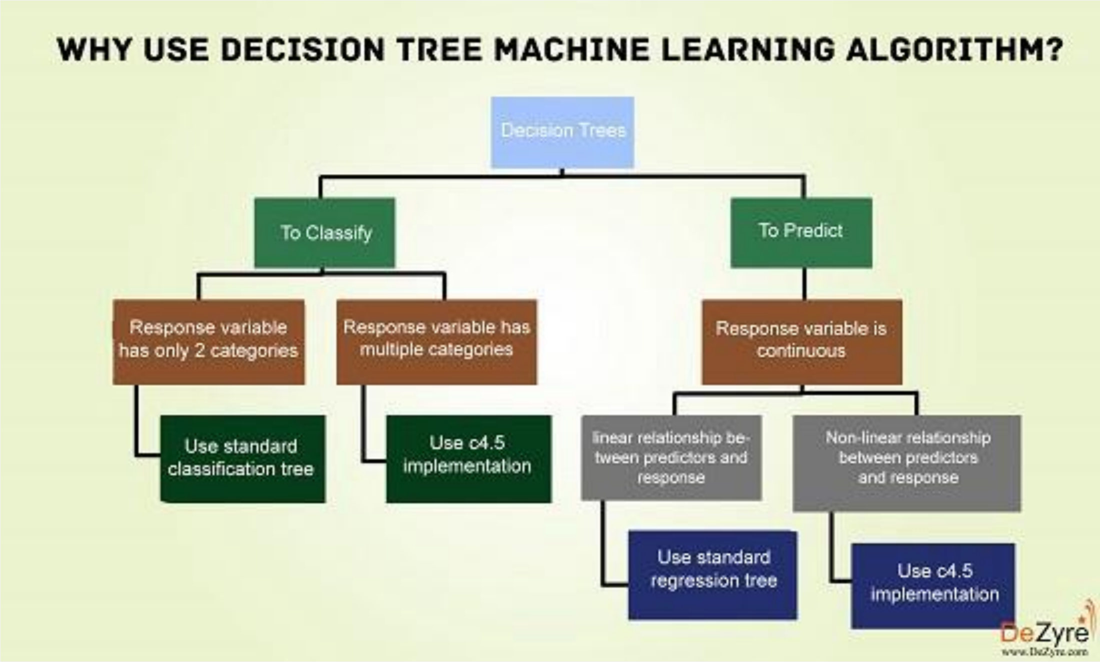
We continue comparing our record's attribute values with other **internal nodes** of the tree until we reach **a leaf node** with predicted class value. The modeled decision tree can be used to predict the target class or the value.

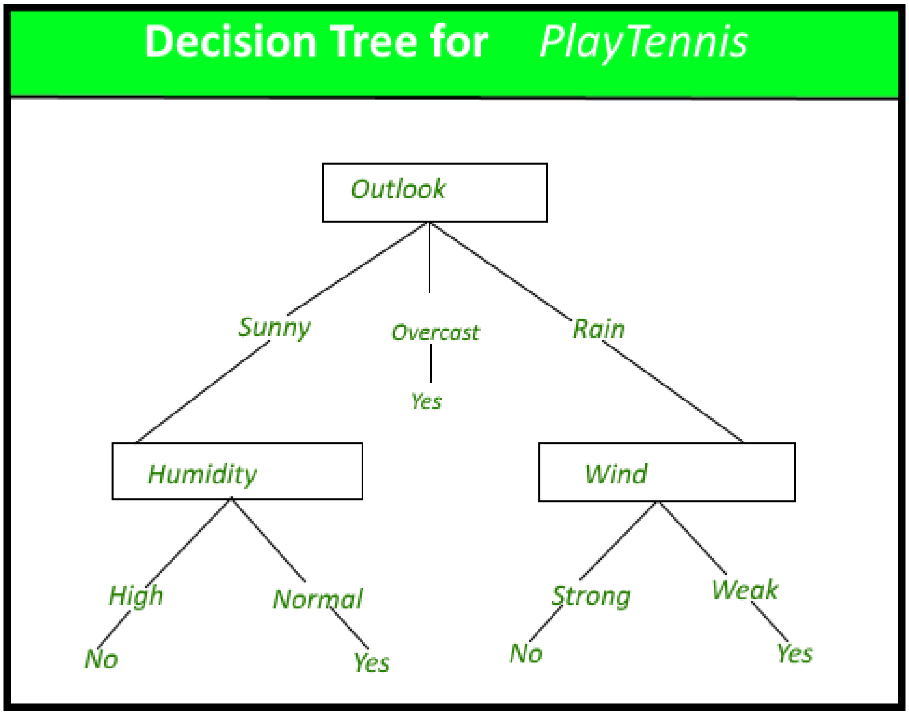
Assumptions while creating Decision Tree

Some of the assumptions we make while using Decision tree:

- At the beginning, the whole training set is considered as the **root.**
- Feature values are preferred to be categorical. If the values are continuous then they are discretized prior to building the model.
- Records are **distributed recursively** on the basis of attribute values.

- Order to placing attributes as root or internal node of the tree is done by using some statistical approach.


WHY USE DECISION TREE MACHINE LEARNING ALGORITHM?
Decision Trees
To Classify
To Predict
Response variable has only 2 categories
Response variable has multiple categories
Response variable is continuous
Use standard classification tree
Use c4.5 implementation
linear relationship between predictors and response
Non-linear relationship between predictors and response
Use standard regression tree
Use c4.5 implementation
DeZyre
www.DeZyre.com



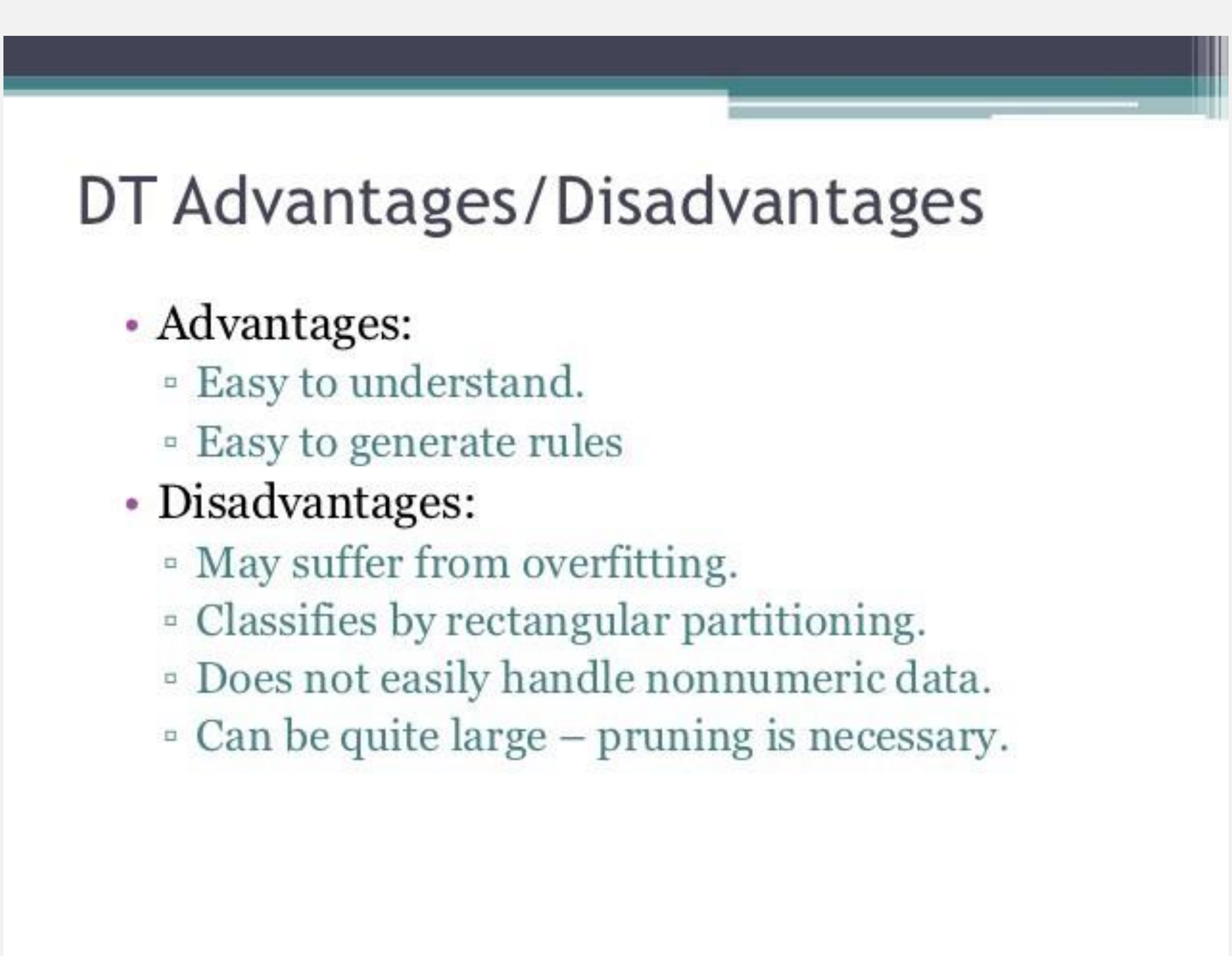
Advantages of Decision Tree:

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 e.g., 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. Decision trees implicitly perform variable screening or feature selection.
4. Decision trees require relatively **little effort from users for data preparation**.
5. **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.
6. **Data type is not a constraint:** It can handle both numerical and categorical variables. Can also *handle multi-output problems*.
7. **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.

8. Non-linear relationships between parameters do not affect tree performance.
9. The number of hyper-parameters to be tuned is almost null.



DT Advantages/Disadvantages

- Advantages:
  - Easy to understand.
  - Easy to generate rules
- Disadvantages:
  - May suffer from overfitting.
  - Classifies by rectangular partitioning.
  - Does not easily handle nonnumeric data.
  - Can be quite large – pruning is necessary.

Disadvantages of Decision Tree:

1. **Over fitting:** Decision-tree learners can create over-complex trees that do not generalize the data well. This is called overfitting. 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.

2. **Not fit for continuous variables**: While working with continuous numerical variables, decision tree loses information, when it categorizes variables in different categories.
3. Decision trees can be unstable because small variations in the data might result in a completely different tree being generated. This is called **variance**, which needs to be lowered by methods like **bagging** and **boosting**.
4. *Greedy* algorithms cannot guarantee to return the globally optimal decision tree. This can be mitigated by training multiple trees, where the features and samples are randomly sampled with replacement.
5. Decision tree learners create *biased* trees if some classes dominate. It is therefore recommended to balance the data set prior to fitting with the decision tree.
6. Information gain in a decision tree with categorical variables gives a biased response for attributes with greater no. of categories.
7. Generally, it gives low prediction accuracy for a dataset as compared to other machine learning algorithms.

8. Calculations can become complex when there are many class label.

Regression Trees vs Classification Trees

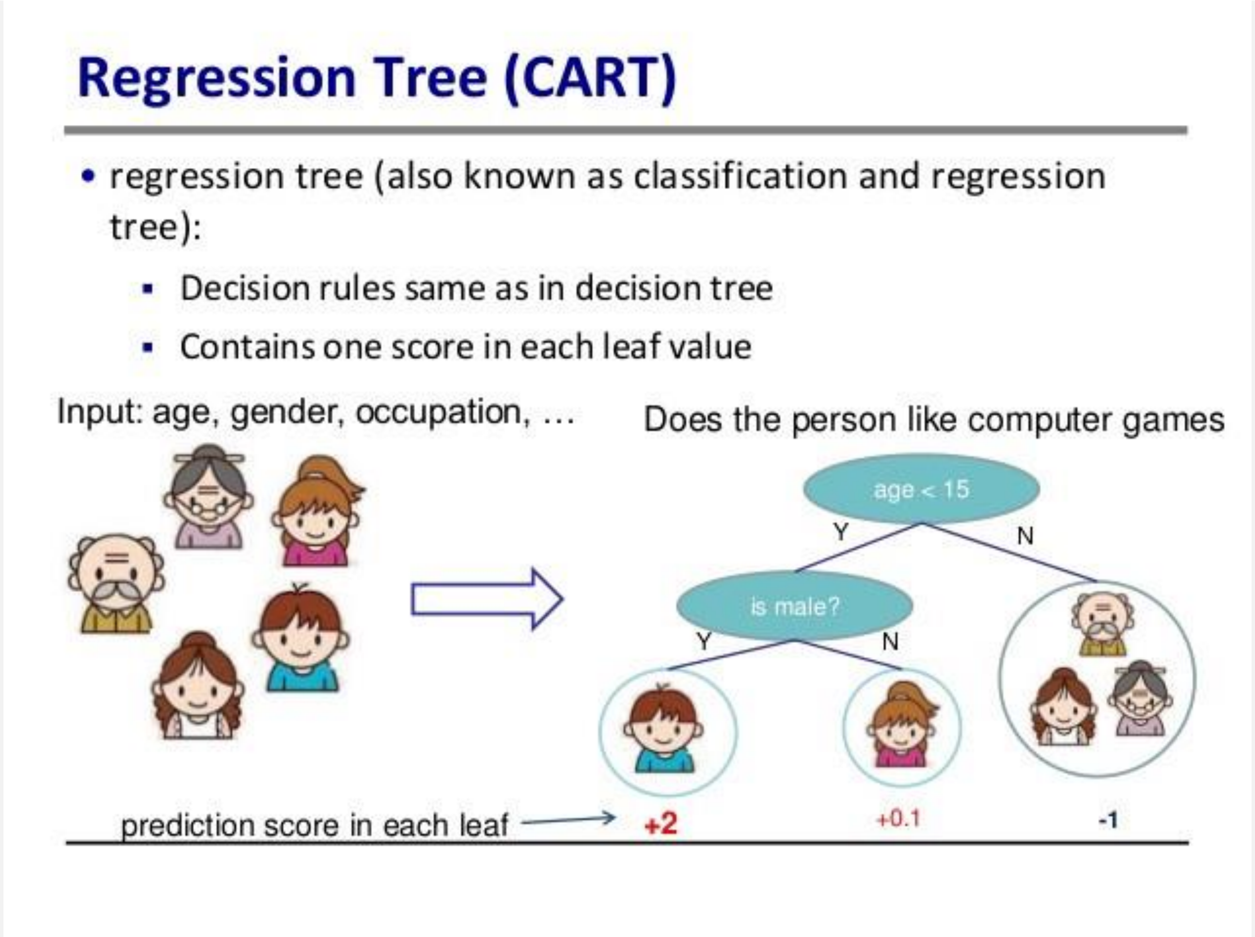
The terminal nodes (or leaves) lies at the bottom of the decision tree. This means that decision trees are typically drawn upside down such that leaves are the bottom & roots are the tops.

Both the trees work almost similar to each other. The primary differences and similarities between Classification and Regression Trees are:

1. Regression trees are used when dependent variable is continuous. Classification Trees are used when dependent variable is categorical.
2. In case of Regression Tree, the value obtained by terminal nodes in the training data is the mean response of observation falling in that region. Thus, if an unseen data observation falls in that region, we'll make its prediction with mean value.
3. In case of Classification Tree, the value (class) obtained by terminal node in the training data is the mode of observations falling in that region. Thus, if an unseen

data observation falls in that region, we'll make its prediction with mode value.

4. Both the trees divide the predictor space (independent variables) into distinct and non-overlapping regions.
5. Both the trees follow a top-down greedy approach known as recursive binary splitting. We call it as 'top-down' because it begins from the top of tree when all the observations are available in a single region and successively splits the predictor space into two new branches down the tree. It is known as '*greedy*' because, the algorithm cares (looks for best variable available) about only the current split, and not about future splits which will lead to a better tree.
6. This splitting process is continued until a user defined stopping criteria is reached. For e.g.: we can tell the algorithm to stop once the number of observations per node becomes less than 50.
7. In both the cases, the splitting process results in fully grown trees until the stopping criteria is reached. But, the fully grown tree is likely to over fit data, leading to poor accuracy on unseen data. This bring 'pruning'. Pruning is one of the technique used tackle overfitting.



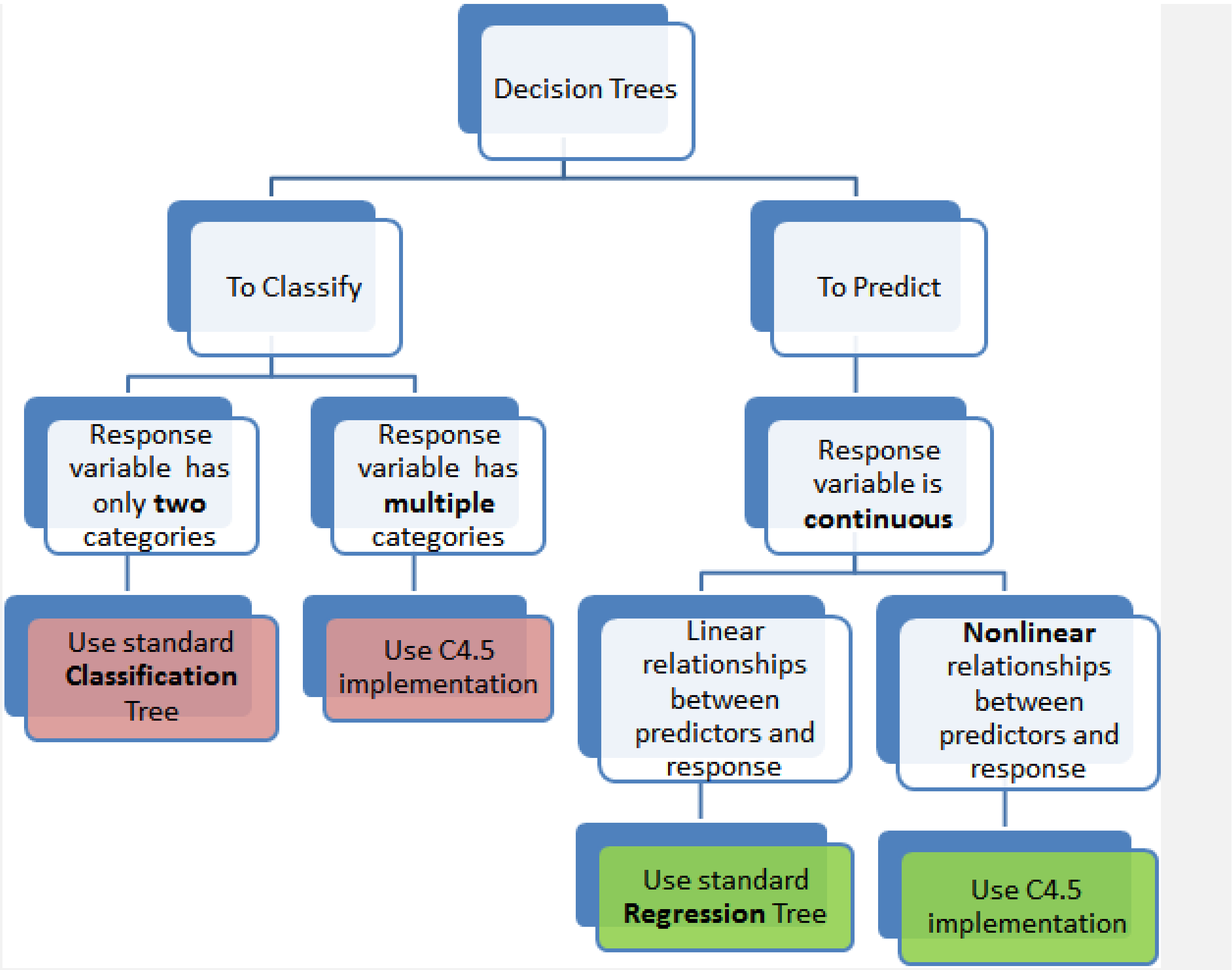
Regression Tree (CART)

- regression tree (also known as classification and regression tree):
  - Decision rules same as in decision tree
  - Contains one score in each leaf value

Input: age, gender, occupation, …

Does the person like computer games

This tree below summarizes at a high level the types of decision trees available.


Decision Trees
To Classify
To Predict
Response variable has only **two** categories
Response variable has **multiple** categories
Response variable is **continuous**
Use standard **Classification** Tree
Use C4.5 implementation
Linear relationships between predictors and response
Nonlinear relationships between predictors and response
Use standard **Regression** Tree
Use C4.5 implementation

How does a tree 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. The four most commonly used algorithms in decision tree are:

Gini Index

Gini index 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.

1. It works with categorical target variable “Success” or “Failure”.
2. It performs only Binary splits
3. Higher the value of Gini higher the homogeneity.
4. CART (Classification and Regression Tree) uses Gini method to create binary splits.

Steps to Calculate Gini for a split

1. Calculate Gini for sub-nodes, using formula sum of square of probability for success and failure (p^2+q^2).

2. Calculate Gini for split using weighted Gini score of each node of that split

Chi-Square

It is an algorithm to find out the statistical significance between the differences between sub-nodes and parent node. We measure it by sum of squares of standardized differences between observed and expected frequencies of target variable.

1. It works with categorical target variable “Success” or “Failure”.
2. It can perform two or more splits.
3. Higher the value of Chi-Square higher the statistical significance of differences between sub-node and Parent node.
4. Chi-Square of each node is calculated using formula,
5. Chi-square = $((\text{Actual} — \text{Expected})^2 / \text{Expected})^{1/2}$
6. It generates tree called CHAID (Chi-square Automatic Interaction Detector)

Steps to Calculate Chi-square for a split:

1. Calculate Chi-square for individual node by calculating the deviation for Success and Failure both
2. Calculated Chi-square of Split using Sum of all Chi-square of success and Failure of each node of the split

Information Gain:

Less impure node requires less information to describe it. And, more impure node requires more information. Information theory is a measure to define this degree of disorganization in a system known as Entropy. If the sample is completely homogeneous, then the entropy is zero and if the sample is an equally divided (50% — 50%), it has entropy of one.

Entropy can be calculated using formula:- Entropy = -p log2 p — q log2q

Here p and q is probability of success and failure respectively in that node. Entropy is also used with categorical target variable. It chooses the split which has lowest entropy compared to parent node and other splits. The lesser the entropy, the better it is.

Steps to calculate entropy for a split:

1. Calculate entropy of parent node

2. Calculate entropy of each individual node of split and calculate weighted average of all sub-nodes available in split.

We can derive information gain from entropy as **1- Entropy.**

Reduction in Variance

Reduction in variance is an algorithm used for continuous target variables (regression problems). This algorithm uses the standard formula of variance to choose the best split. The split with lower variance is selected as the criteria to split the population:

$$\text{Variance} = \frac{\Sigma(X - \overline{X})^2}{n}$$

Above X-bar is mean of the values, X is actual and n is number of values.

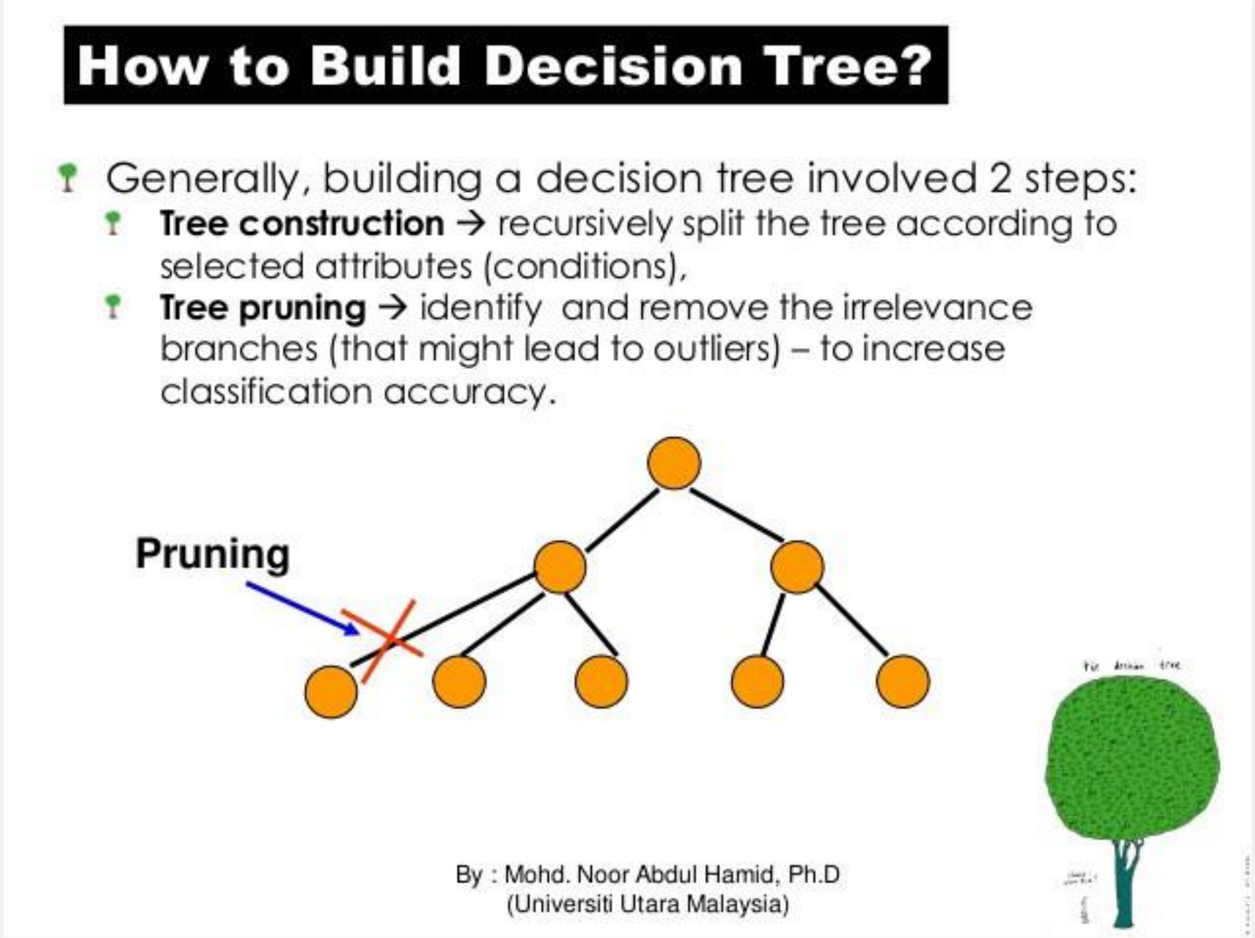
Steps to calculate Variance:

1. Calculate variance for each node.
2. Calculate variance for each split as weighted average of each node variance.

Key parameters of tree modelling and how can we avoid over–fitting in decision trees:

Overfitting is one of the key practical challenges faced while modeling decision trees. If there is no limit set of a decision tree, it will give you 100% accuracy on training set because in the worst case, it will end up making 1 leaf for each observation. The model is having an issue of overfitting, is considered when the algorithm continues to go deeper and deeper to reduce the training set error but results with an increased test set error i.e. accuracy of prediction for our model goes down. It generally happens when it builds many branches due to outliers and irregularities in data. 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



How to Build Decision Tree?

- Generally, building a decision tree involved 2 steps:
  - **Tree construction** → recursively split the tree according to selected attributes (conditions),
  - **Tree pruning** → identify and remove the irrelevance branches (that might lead to outliers) – to increase classification accuracy.

By : Mohd. Noor Abdul Hamid, Ph.D
(Universiti Utara Malaysia)

Setting Constraints on Tree Size

This can be done by using various parameters which are used to define a tree. The parameters used for defining a tree are:

1. **Minimum samples for a node split**

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

- 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.
- Too high values can lead to under-fitting hence, it should be tuned using CV.

1. **Minimum samples for a terminal node (leaf)**

- Defines the minimum samples (or observations) required in a terminal node or leaf.
- Used to control over-fitting similar to min_samples_split.
- 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.

1. **Maximum depth of tree (vertical depth)**

- The maximum depth of a tree.
- Used to control over-fitting as higher depth will allow model to learn relations very specific to a particular sample.
- Should be tuned using CV.

1. **Maximum number of terminal nodes**

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

1. **Maximum features to consider for split**

- The number of features to consider while searching for a best split. These will be randomly selected.
- As a thumb-rule, square root of the total number of features works great but we should check upto 30–40% of the total number of features.
- Higher values can lead to over-fitting but depends on case to case.

Tree Pruning

The technique of setting constraint is a greedy-approach. In other words, it will check for the best split instantaneously and move forward until one of the specified stopping condition is reached. For e.g.: consider the following case when you're driving:

There are 2 lanes:

1. A lane with cars moving at 80 km/h
2. A lane with trucks moving at 30 km/h

At this instant, you are the yellow car and you have 2 choices:

1. Take a left and overtake the other 2 cars quickly
2. Keep moving in the present lane

Analyzing these choice: In the former choice, you'll immediately overtake the car ahead and reach behind the truck and start moving at 30 km/h, looking for an opportunity to move back right. All cars originally behind you move ahead in the meanwhile. This would be the optimum choice if your objective is to maximize the distance covered in next say 10 seconds. In the later choice, you drive through at same speed, cross trucks and then overtake maybe depending on situation ahead.

This is exactly the difference between normal decision tree & pruning. A decision tree with constraints won't see the truck ahead and adopt a greedy approach by taking a left. On the other hand if we use pruning, we in effect look at a few steps ahead and make a choice.

So we know, pruning is better. To implement it in decision tree:

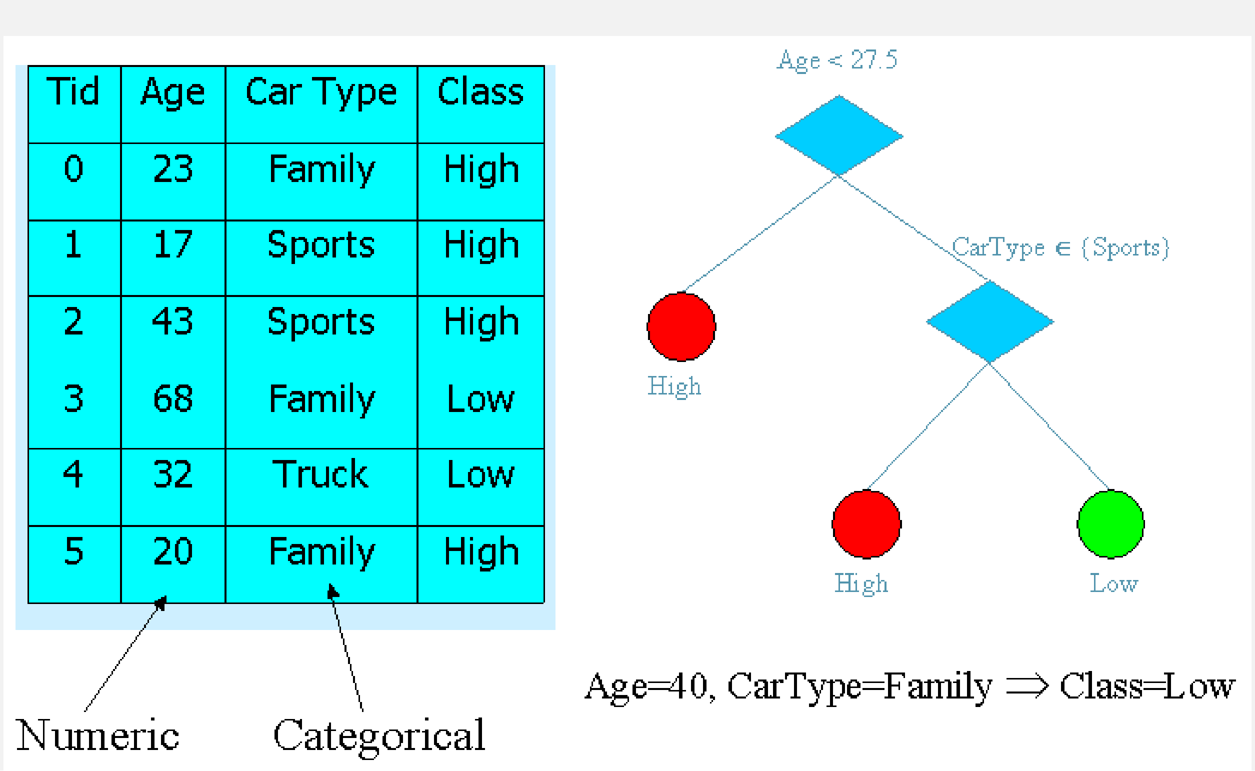
1. We first make the decision tree to a large depth.
2. Then we start at the bottom and start removing leaves which are giving us negative returns when compared from the top.
3. Suppose a split is giving us a gain of say -10 (loss of 10) and then the next split on that gives us a gain of 20. A simple decision tree will stop at step 1 but in pruning, we will see that the overall gain is +10 and keep both leaves.

Are tree based models better than linear models?

If one can use logistic regression for classification problems and linear regression for regression problems, why is there a need to use trees? Actually, we can use any algorithm. It is dependent on the type of problem we are solving. Some key factors which will help us to decide which algorithm to use:

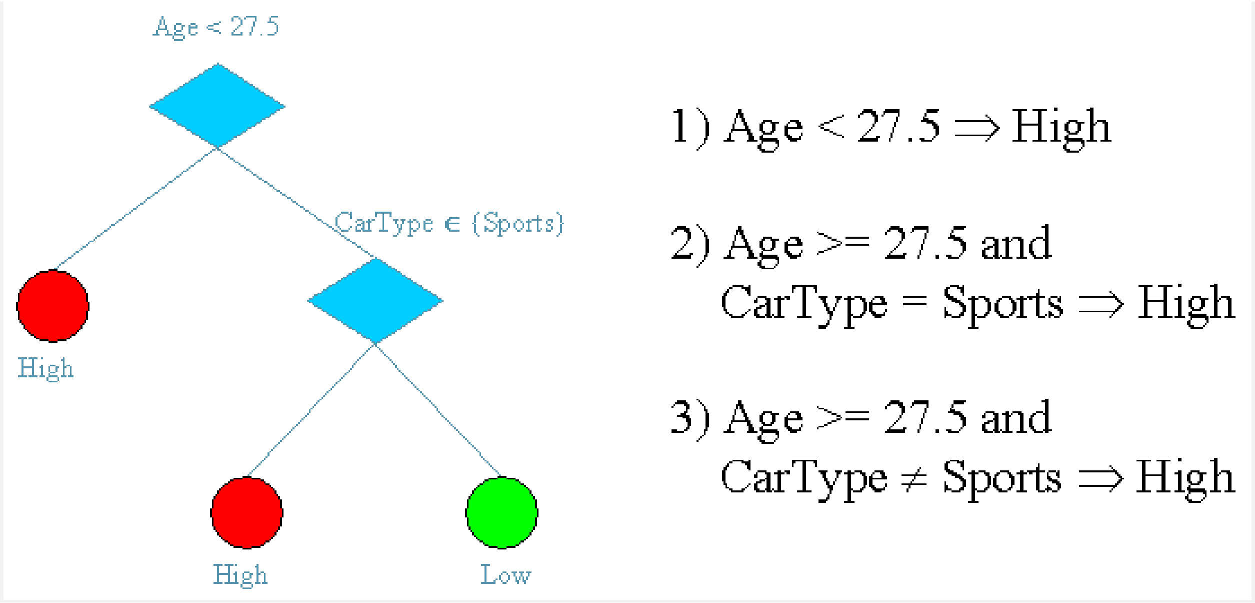
1. If the relationship between dependent & independent variable is well approximated by a linear model, linear regression will outperform tree based model.
2. If there is a high non-linearity and complex relationship between dependent & independent variables, a tree model will outperform a classical regression method.

3. 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.**e.g. Decison Tree Classification**



Tid	Age	Car Type	Class
0	23	Family	High
1	17	Sports	High
2	43	Sports	High
3	68	Family	Low
4	32	Truck	Low
5	20	Family	High

From Tree to Rules :



1) Age < 27.5 ⇒ High

2) Age >= 27.5 and
CarType = Sports ⇒ High

3) Age >= 27.5 and
CarType ≠ Sports ⇒ High

How Decision Trees work: Algorithm

· Build tree

· Start with data at root node

· Select an attribute and formulate a logical test on attribute

· Branch on each outcome of the test, and move subset of examples satisfying that outcome to corresponding child node

· Recurse on each child node

· Repeat until leaves are “pure”, i.e., have example from a single class, or “nearly pure”, i.e., majority of examples are from the same class

· Prune tree

· Remove subtrees that do not improve classification accuracy

· Avoid over-fitting, i.e., training set specific artifacts

· Build tree

· Evaluate split-points for all attributes

· Select the “best” point and the “winning” attribute

· Split the data into two

· Breadth/depth-first construction

· CRITICAL STEPS:

· Formulation of good split tests

· Selection measure for attributes

- **How to capture good splits?**

· Prefer the simplest hypothesis that fits the data

· Minimum message/description length

· Dataset D

· Hypotheses H1, H2, ..., Hx describing D

· MML(Hi) = Mlength(Hi)+Mlength(D|Hi)

· Pick Hk with minimum MML

· Mlength given by Gini index, Gain, etc.

Tree pruning

- Data encoding: sum classification errors
- Model encoding:

· Encode the tree structure

· Encode the split points

- Pruning: choose smallest length option

· Convert to leaf

· Prune left or right child

· Do nothing

- Hunt’s Method

· Attributes: Refund (Yes, No), Marital Status (Single, Married, Divorced), Taxable Income

· Class: Cheat, Don’t Cheat

Finding good split points

· Use Gini index for partition purity

· If S is pure, Gini(S) = 0, Gini is a kind of entropy calculation

· Find split-point with minimum Gini

· Only need class distributions

How informative is an attribute ? :

· Statistic measure of informativity, measuring how well an attribute distinguishes between examples of different classes.

· Informativity is measured as the decrease in entropy of the training set of examples.

· Entropy is the measure of impurity of the sample set: E(S) = -p+log2p+ — p-log2p-

Working with Decision Trees in Python:

```
#Import Library

# Import other necessary libraries like pandas, numpy...

from sklearn import tree

# Assumed you have, X (predictor) and Y (target) for training data set and x_test (predictor) of test_dataset

# Create tree object

model = tree.DecisionTreeClassifier(criterion=’gini’) # for classification, here you can change the algorithm as gini or entropy (information gain) by default it is gini

# model = tree.DecisionTreeRegressor() for regression

# Train the model using the training sets and check score

model.fit(X, y)

model.score(X, y)

#Predict Output
```

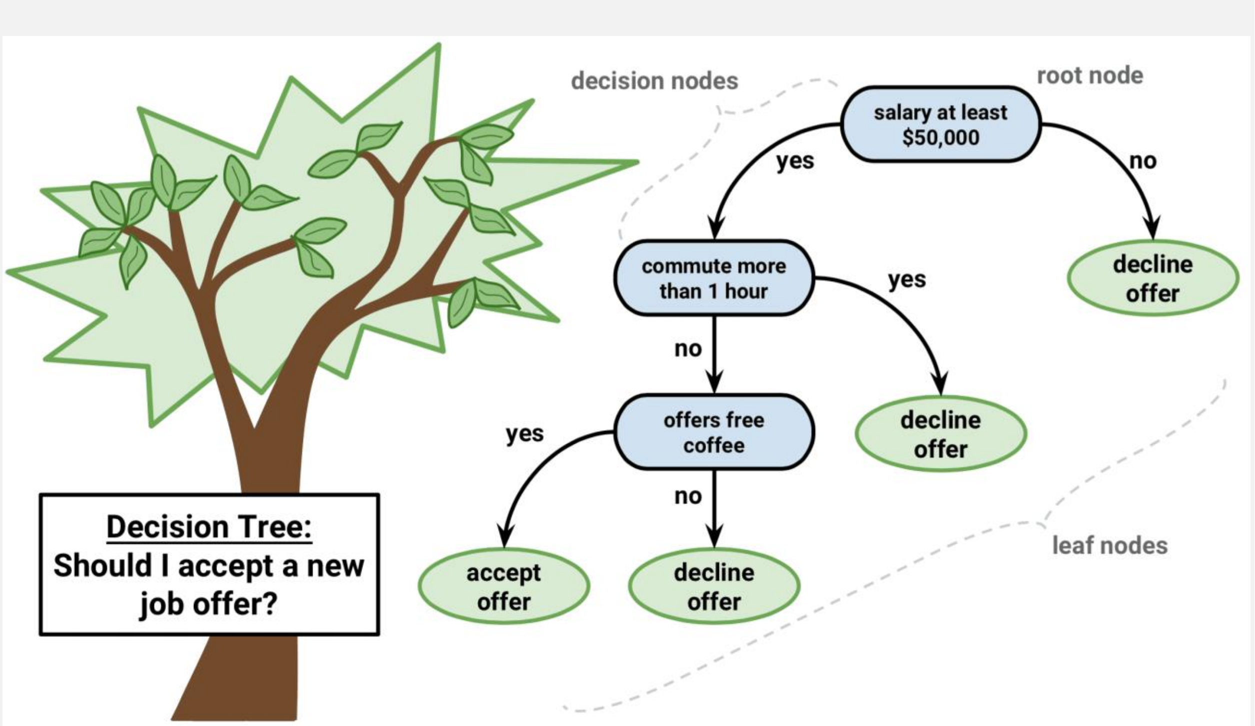
```
predicted = model.predict(x_test)
```

Summary :

Not all problems can be solved with linear methods. The world is non-linear. It has been observed that tree based models have been able to map non-linearity effectively. Methods like decision trees, random forest, gradient boosting are being popularly used in all kinds of data science problems.

Decision Tree algorithm belongs to the family of supervised learning algorithms. Unlike other supervised learning algorithms, decision tree algorithm can be used for solving **regression and classification problems** too. The general motive of using Decision Tree is to create a training model which can use to predict class or value of target variables by **learning decision rules** inferred from prior data (training data). The primary challenge in the decision tree implementation is to identify which attributes do we need to consider as the root node and each level. Decision tress often mimic the human level thinking so it’s simple to understand the data and make some good interpretations.

Dividing efficiently based on maximum information gain is key to decision tree classifier. However, in real world with millions of data dividing into pure class in practically not feasible (it may take longer training time) and so we stop at points in nodes of tree when fulfilled with certain parameters (for example impurity percentage). Decision tree is classification strategy as opposed to the algorithm for classification. It takes top down approach and uses divide and conquer method to arrive at decision. We can have multiple leaf classes with this approach.



"When your values are clear to you, making decisions become easier." — Roy E. Disney

GreyAtom

GreyAtom is committed to building an educational ecosystem…
Follow
1.4K

- Machine Learning

1.4K claps