BHARATHIDASAN UNIVERSITY

Tiruchirappalli- 620024

Tamil Nadu, India.

Programme: M.Sc. Statistics

Course Title: Introduction to Machine learning

Course Code: 23ST01VAC

Unit-III

Decision Tree

Dr. T. Jai Sankar
Associate Professor and Head
Department of Statistics

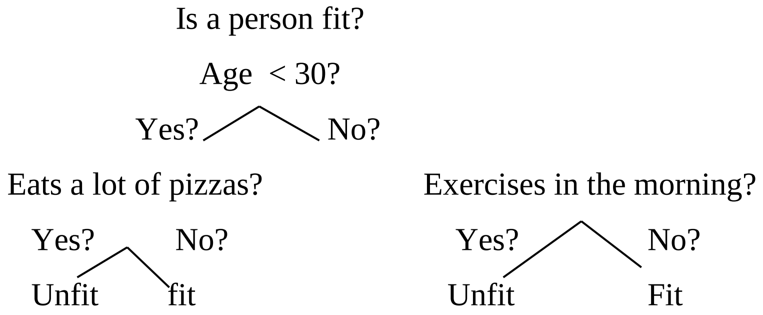
Ms. E. Devi
Guest Faculty
Department of Statistics

UNIT-III

Decision Tree

Decision tree

Decision trees are a type of supervised machine learning where the data is continuously spilt according to a certain parameter. The tree can be explained by two entities, namely decision nodes and leaves. The leaves are the decisions or the final outcomes. And the decision nodes are where the data is split.



Example:

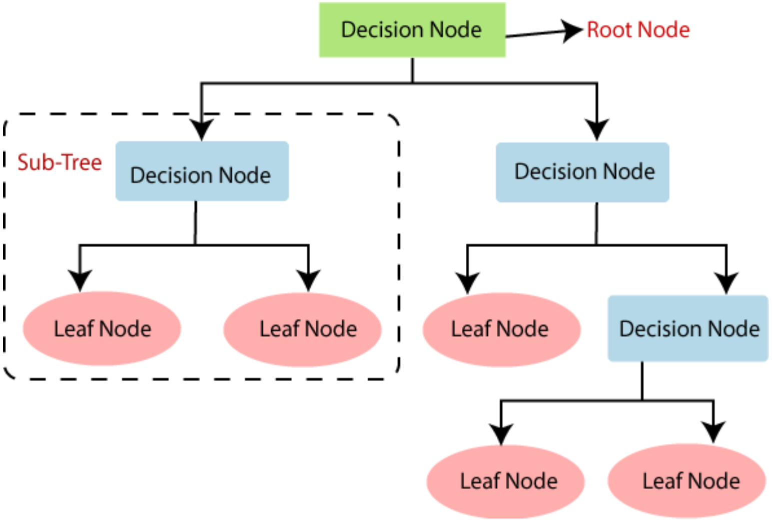
An example of a decision tree can be explained using above binary tree. Let's say you want to predict whether a person is fit given their information like age, eating habit, and physical activity etc. Here, the decision nodes are question like 'What's the age?', 'Does he exercise?', and 'Does he eat a lot of pizzas?' and the leaves, which are outcomes like either 'fit', or 'unfit'. In this case this was a binary classification problem (a yes/no type problem).

Decision Tree Classification Algorithm:

- Decision Tree is a **Supervised learning technique** that can be used for both classification and Regression problems, but mostly it is preferred for solving Classification problems. It is a tree-structured classifier, where **internal nodes represent the features of a dataset, branches represent the decision rules** and **each leaf node represents the outcome.**
- In a Decision tree, there are two nodes, which are the **Decision Node** and **Leaf Node.** Decision nodes are used to make any decision and have multiple branches, whereas Leaf nodes are the output of those decisions and do not contain any further branches.
- The decisions or the test are performed on the basis of features of the given dataset.
- **It is a graphical representation for getting all the possible solutions to a problem/decision based on given conditions.**
- It is called a decision tree because, similar to a tree, it starts with the root node, which

expands on further branches and constructs a tree-like structure.

- In order to build a tree, we use the **CART algorithm,** which stands for **Classification and Regression Tree algorithm.**
- A decision tree simply asks a question, and based on the answer (Yes/No), it further split the tree into subtrees.
- Below diagram explains the general structure of a decision tree:



Uses of Decision Trees

There are various algorithms in Machine learning, so choosing the best algorithm for the given dataset and problem is the main point to remember while creating a machine learning model. Below are the two reasons for using the Decision tree:

- Decision Trees usually mimic human thinking ability while making a decision, so it is easy to understand.
- The logic behind the decision tree can be easily understood because it shows a tree-like structure.

Decision Tree Terminologies

- **Root Node:** Root node is from where the decision tree starts. It represents the entire dataset, which further gets divided into two or more homogeneous sets.
- **Leaf Node:** Leaf nodes are the final output node, and the tree cannot be segregated further after getting a leaf node.
- **Splitting:** Splitting is the process of dividing the decision node/root node into sub-nodes according to the given conditions.
- **Branch/Sub Tree:** A tree formed by splitting the tree.

- **Pruning:** Pruning is the process of removing the unwanted branches from the tree.
- **Parent/Child node:** The root node of the tree is called the parent node, and other nodes are called the child nodes.

Decision Tree algorithm

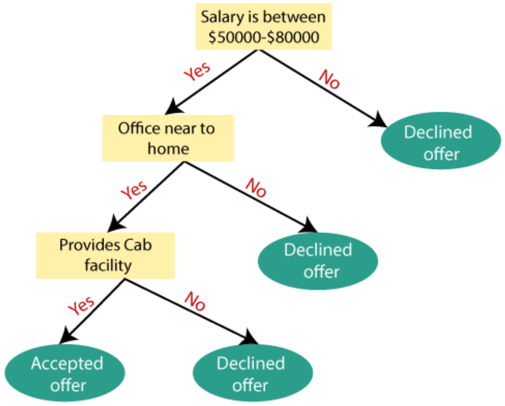
In a decision tree, for predicting the class of the given dataset, the algorithm starts from the root node of the tree. This algorithm compares the values of root attribute with the record (real dataset) attribute and, based on the comparison, follows the branch and jumps to the next node.

For the next node, the algorithm again compares the attribute value with the other sub-nodes and move further. It continues the process until it reaches the leaf node of the tree. The complete process can be better understood using the below algorithm:

- **Step-1:** Begin the tree with the root node, says S, which contains the complete dataset.
- **Step-2:** Find the best attribute in the dataset using **Attribute Selection Measure (ASM).**
- **Step-3:** Divide the S into subsets that contains possible values for the best attributes.
- **Step-4:** Generate the decision tree node, which contains the best attribute.

  Step-5: Recursively make new decision trees using the subsets of the dataset created in step -3. Continue this process until a stage is reached where you cannot further classify the nodes and called the final node as a leaf node.

Example: Suppose there is a candidate who has a job offer and wants to decide whether he should accept the offer or Not. So, to solve this problem, the decision tree starts with the root node (Salary attribute by ASM). The root node splits further into the next decision node (distance from the office) and one leaf node based on the corresponding labels. The next decision node further gets split into one decision node (Cab facility) and one leaf node. Finally, the decision node splits into two leaf nodes (Accepted offers and Declined offer). Consider the below diagram:



Pruning: Getting an Optimal Decision tree

Pruning is a process of deleting the unnecessary nodes from a tree in order to get the optimaldecision tree.

A too-large tree increases the risk of over fitting, and a small tree may not capture all the important features of the dataset. Therefore, a technique that decreases the size of the learningtree without reducing accuracy is known as Pruning. There are mainly two types of tree **pruning** technology used:

- **Cost Complexity Pruning**
- **Reduced Error Pruning.**

Advantages of the Decision Tree

- It is simple to understand as it follows the same process which a human followwhile making any decision in real-life.
- It can be very useful for solving decision-related problems.
- It helps to think about all the possible outcomes for a problem.
- There is less requirement of data cleaning compared to other algorithms.

Disadvantages of the Decision Tree

- The decision tree contains lots of layers, which makes it complex.
- It may have an over fitting issue, which can be resolved using the **Random Forestalgorithm.**
- For more class labels, the computational complexity of the decision tree mayincrease.

Python Implementation of Decision Tree

Now we will implement the Decision tree using Python. For this, we will use the dataset "**user_data.csv**," which we have used in previous classification models. By using the same dataset, we can compare the Decision tree classifier with other classification models suchas KNN SVM, Logistic Regression, etc.

Steps will also remain the same, which are given below:

- ✓ **Data Pre-processing step**
- ✓ **Fitting a Decision-Tree algorithm to the Training set**
- ✓ **Predicting the test result**
- ✓ **Test accuracy of the result(Creation of Confusion matrix)**
- ✓ **Visualizing the test set result.**

Decision Trees work

- The decision of making strategic splits heavily affects a tree’s accuracy. The decision criteria are different for classification and regression trees.
- Decision trees use multiple algorithms to decide to split a node into two or more sub-nodes. The creation of sub-nodes increases the homogeneity of resultant sub-nodes. In other words, we can say that the purity of the node increases with respect to the target variable. The decision tree splits the nodes on all available variables and then selects the split which results in most homogeneous sub-nodes.

Algorithms used in Decision Trees:

- ➢ **ID3** → (extension of D3)
- ➢ **C4.5** → (successor of ID3)
- ➢ **CART** → (Classification And Regression Tree)
- ➢ **CHAID** → (Chi-square automatic interaction detection Performs multi-level splitswhen computing classification trees)
- ➢ **MARS** → (multivariate adaptive regression spines)

The ID3 algorithm builds decision trees using a top-down greedy search approach through the space of possible branches with no backtracking. A greedy algorithm, as the name suggests, always makes the choice that seems to be the best at that moment.

Types of Decision Trees:

There are two main types of Decision trees:

1. classification trees (yes/no types):

What we have seen above is an example of classification tree, where the outcome was a variable like ‘fit’ or ‘unfit’. Here the decision variable is categorical.

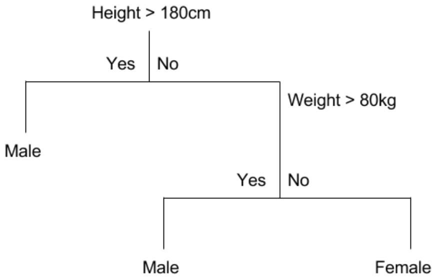
2. Regression trees (continuous data types):

Here the decision or the outcome variable is continuous, eg.: a number like 123.

1. Classification trees:

Definition: A classification tree is an algorithm where the largest variable is fixed or categorical. The algorithm is then used to identify the “class” within which a target variable would most likely fall.

An example of classification – type problem would be determining who will or will not subscribe to a digital platform; or who will or will not graduate from high school. These are examples of simple binary classifications where the categorical dependent variable can assume only one or two, mutually exclusive values. In other cases, you might have to predict among a number of different variables.



2. Regression trees:

A regression tree refers to an algorithm where the target variable is and the algorithm is used to predict it’s value.

As an example of a regression type problem, you may want to predict the selling prices of a residential house, which is a continuous dependent variable.

This will depend on both continuous factors like square footage as well as categorical factors like the style of home, area in which the property is located and so on.

Uses of classification and Regression trees:

Classification trees	Regression trees
A classification tree splits the dataset based on the homogeneity of data. For instance, there are two variable; income and age; which determine whether or not a consumer will buy a particular kind of phone.	A regression model is fit to the target variable using each of the independent variables. After this, the data split at several points for each independent variable. At each such point, the error b/w the predicted values and actual values is squared to get a sum of squared errors (SSE). The SSE is compared across the variables and variable or point which has the lowest SSE is chosen as the split point. The process is continued.

Advantages of classification and regression trees:

I. The results are simplistic: the interpretation of results summarized in classification or regression tree is usually fairly simple

II. Classification and regression trees are non parametric & non linear; the results from classification and regression trees can be simplistic if

   a. the predictor variables and the dependent variable are linear.

   b. the predictor variables and the dependent variable follow some specific non linear link function.

   c. the predictor variables and the dependent variables are monotonic. As a result, classification and regression trees can actually reveal relationship b/w these variables that would not have been possible using other techniques.

III. Classification and regression trees implicitly perform feature selection: feature selection or variable screening is an important part of analytics. When we use decision trees, the top few nodes on which the tree is split are the most important variables within the set. As a result, feature selection gets performed automatically and we don't need to do it again.

Limitations of classification and regression trees:

There are many classification and regression trees examples where the use of a decision tree has not led to the optimal result. Some of the limitations of classification and regression trees

i. **Over fitting:** Over fitting occurs when the tree takes into account a lot of noise that exists in the data and comes up with an inaccurate result.

ii. **High variance:** In this case, a small variance in the data can lead to a very high variance in the prediction, there by affecting the stability of the outcome.

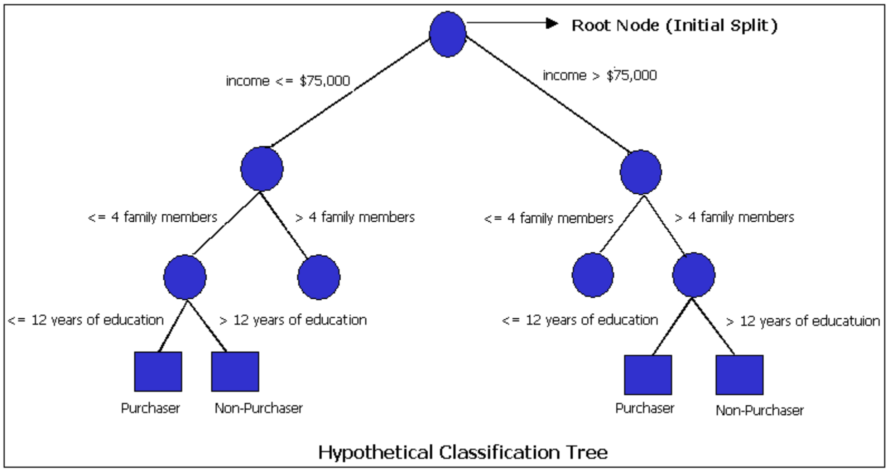
iii. **Low bias:** A decision tree that is very complex usually has a low bias. This makes it very difficult for the model to incorporate any new data.

Construction of classification tree:

- A classification tree is built through a process known as binary recursive partitioning. This is an iterative process of splitting the data into partitions, and then splitting it up further on each of the branches.
- The process starts with a training set consisting of pre classified records (target field or dependent variable with a known class or label such a purchaser on non-purchaser).
- The goal is to build a tree that distinguishes among the classes, and that there are only two target classes, and that each split is a binary partition. The partition (splitting) criterion generalizes to multiple classes, and any multi-way partitioning can be achieved through repeated binary splits.
- To choose the best splitter at a node, the algorithm considers each input field in turn. In essence, each field is sorted every possible split is tried and considered, the best split is the one that produces the largest decrease in diversity of the classification label within each partition (i.e., increase in homogeneity this is

splitter for the node. The process in continued at subsequent nodes until a full tree is generated.

Example:



Inductive Decision tree:

- Decision tree induction is the method of learning the decision trees from the training set.
- The training set consists of attributes and class labels. Applications of decision tree induction include astronomy, financial analysis, medical diagnosis, manufacturing and production.

Attribute selection measure:

- While implementing a decision tree, the main issue axis that how to select the best attribute for the root node and for sub – nodes. So, to solve such problems there is a technique which is called as attribute selection or measure or ASM.
- If the dataset consists of **N** attributes then deciding which attribute to place at the root or at different levels of the tree as internal nodes is a complicated step. By just randomly selecting any node to be the root can’t solve the issue. If we follow a random approach, it may give us bad results with low accuracy.

For solving this attribute selection problem, researchers worked and devised some solutions.

They suggested using some criteria like :

1. **Entropy**,
2. **Information gain,**
3. **Gini index,**
4. **Gain Ratio,**
5. **Reduction in Variance**
6. **Chi-Square**

These criteria will calculate values for every attribute. The values are sorted, and attributes are placed in the tree by following the order i.e, the attribute with a high value(in case of information gain) is placed at the root. While using Information Gain as a criterion, we assume attributes to be categorical, and for the Gini index, attributes are assumed to be continuous.

Attribute Selection Measures

There are 3 prominent attribute selection measures for decision tree induction, each pairedwith one of the 3 prominent decision tree classifiers.

- Information gain - used in the ID3 algorithm
- Gain ratio - used in the C4.5 algorithm
- Gini index - used in the CART algorithm

Steps in ID3 algorithm:

1. It begins with the original set S as the root node.
2. On each iteration of the algorithm, it iterates through the very unused attribute of theset S and calculates **Entropy (H)** and **Information gain (IG)** of this attribute.
3. It then selects the attribute which has the smallest Entropy or Largest Information gain.
4. The set S is then split by the selected attribute to produce a subset of the data.
5. The algorithm continues to recur on each subset, considering only attributes neverselected before.

C4.5 → (successor of ID3):

- **C4.5** is the successor to ID3 and removed the restriction that features must be categorical by dynamically defining a discrete attribute (based on numerical variables) that partitions the continuous attribute value into a discrete set of intervals.
- C4.5 converts the trained trees (i.e. the output of the ID3 algorithm) into sets of if-then rules. This accuracy of each rule is then evaluated to determine the order in which they should be applied. Pruning is done by removing a rule's precondition if the accuracy of the rule improves without it.

There are two popular technique for ASM, Which are;

1) Information gain, 2) Entropy

Entropy

In machine learning, 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.

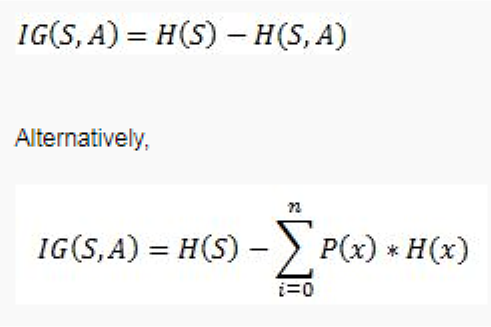
$$\mathrm{H}(X) = -\sum_{i=1}^{n} \mathrm{P}(x_i) \log_b \mathrm{P}(x_i)$$

Example: consider a coin toss whose probability of heads is 0.5 and probability of tails is 0.5. Here the entropy is the highest possible, since there's no way of determining what the outcome might be.

When there is no randomness, entropy will be zero. In particular, lower imply less uncertainty while higher values imply high uncertainty.

Information Gain

Information gain can be defined as the amount of information gained about a random variable or signal from observing another random variable. It can be considered as the difference between the entropy of parent node and weighted average entropy of child nodes.



$$IG(S,A) = H(S) - H(S,A)$$

Alternatively,

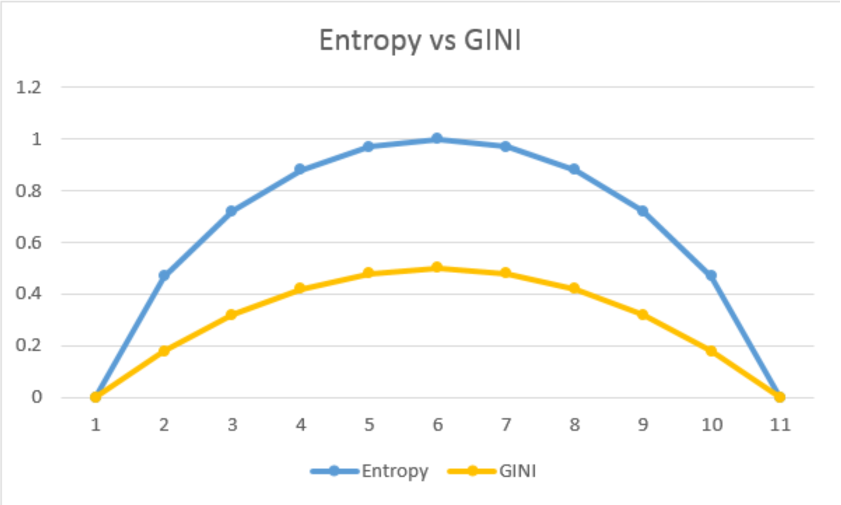
$$IG(S,A) = H(S) - \sum_{i=0}^{n} P(x) * H(x)$$

Gini Impurity

Gini impurity is a measure of how often a randomly chosen element from the set would be incorrectly labeled if it was randomly labeled according to the distribution of labels in the subset.

$$Gini(E) = 1 - \sum_{j=1}^{c} p_j^2$$

Gini impurity is **lower bounded by 0**, with 0 occurring if the data set contains only one class.

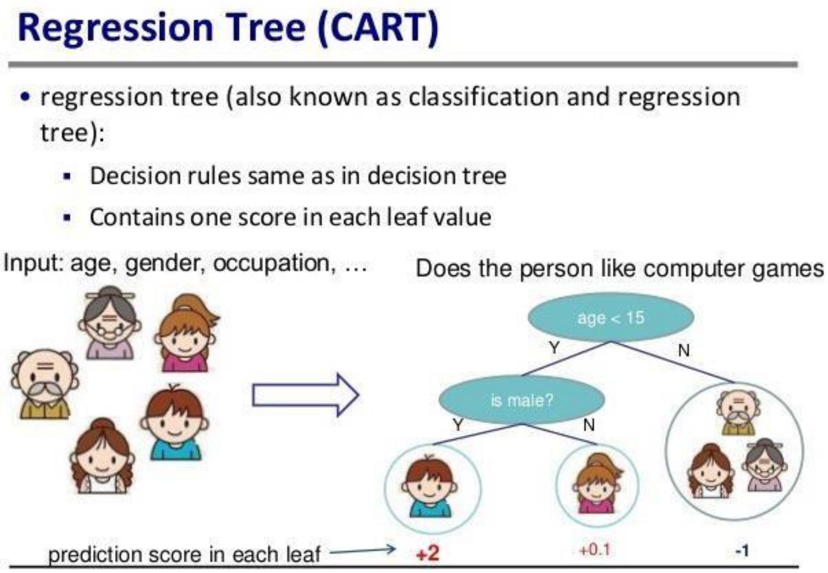


There are many algorithms there to build a decision tree. They are

1. **CART** (Classification and Regression Trees) — This makes use of Gini impurity as the metric.
2. **ID3** (Iterative Dichotomiser 3) — This uses entropy and information gain as metric.

In this article, I will go through ID3. Once you got it it is easy to implement the same using CART.

CART: (Classification and Regression Trees) is very similar to C4.5, but it differs in that it supports numerical target variables (regression) and does not compute rule sets. CART constructs binary trees using the feature and threshold that yields the largest information gain at each node.



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

prediction score in each leaf

CHAID:

- ✓ Chi-square Automatic Interaction Detector (CHAID) was a technique created by Gordon V. Kass in 1980. CHAID is a tool used to discover the relationship between variables.
- ✓ CHAID analysis builds a predictive medal, or tree, to help determine how variables best merge to explain the outcome in the given dependent variable.
- ✓ In CHAID analysis, nominal, ordinal, and continuous data can be used, where continuous predictors are split into categories with approximately equal number of observations.
- ✓ CHAID creates all possible cross tabulations for each categorical predictor until the best outcome is achieved and no further splitting canbe performed.
- ✓ In the CHAID technique, we can visually see the relationships between the split variables and the associated related factor within the tree.
- ✓ The development of the decision, or classification tree, starts with identifying the target variable or dependent variable; which would be considered the root.
- ✓ CHAID analysis splits the target into two or more categories that are called the

initial, or parent nodes, and then the nodes are split using statistical algorithms into child nodes. Unlike in regression analysis, the CHAID technique does not require the data to be normally distributed.

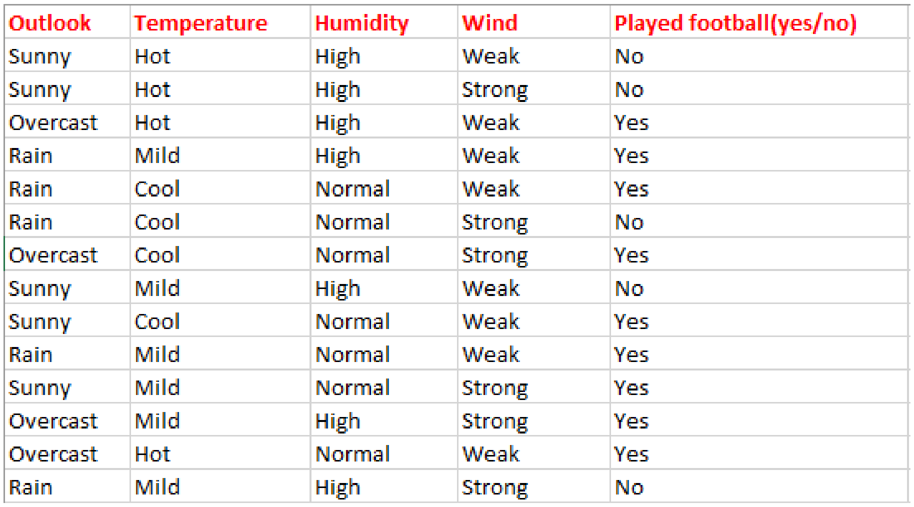
Decision Tree Raising

- ✓ A completed decision tree model can be overly-complex, contain unnecessary structure, and be difficult to interpret. Tree pruning is the process of removing the unnecessary structure from a decision tree in order to make it more efficient, more easily-readable for humans, and more accurate as well.
- ✓ This increased accuracy is due to pruning’s ability to reduce over fitting. **Over fitting** refers to the idea that models built from algorithms are too specifically-tailored to the particular training dataset that was used to generate them. While these models may do very well at categorizing said training data, over fitted models would perform poorly on another set of unseen testing data. Over fitting is not the sole concern of decision trees; the potential to over fit applies to nearly all machine learning classification algorithms.
- ✓ There are different approaches to pruning. The C4.5 pruning method is post-pruning, with sub tree raising. This means that pruning occurs after the tree has been constructed, as opposed to pre-pruning, and involves sub tree raising, as opposed to sub tree replacement, an alternative form of pruning. Sub tree raising entails raising entire sub trees to replace nodes closer to the root, which also means reclassifying leaves of sub trees closer to the root which may have been replaced during this process.

PROBLEM:

Classification using the ID3 algorithm

Consider whether a dataset based on which we will determine whether to play football or not.



Outlook	Temperature	Humidity	Wind	Played football(yes/no)
Sunny	Hot	High	Weak	No
Sunny	Hot	High	Strong	No
Overcast	Hot	High	Weak	Yes
Rain	Mild	High	Weak	Yes
Rain	Cool	Normal	Weak	Yes
Rain	Cool	Normal	Strong	No
Overcast	Cool	Normal	Strong	Yes
Sunny	Mild	High	Weak	No
Sunny	Cool	Normal	Weak	Yes
Rain	Mild	Normal	Weak	Yes
Sunny	Mild	Normal	Strong	Yes
Overcast	Mild	High	Strong	Yes
Overcast	Hot	Normal	Weak	Yes
Rain	Mild	High	Strong	No

Here There are for independent variables to determine the dependent variable. The independent variables are Outlook, Temperature, Humidity, and Wind. The dependent variable is whether to play football or not.

As the first step, we have to find the parent node for our decision tree. For that follow the steps:

Find the entropy of the class variable.

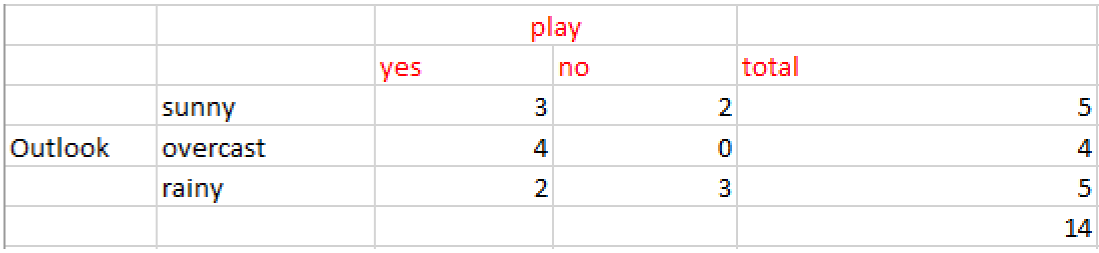
$$E(S) = -[(9/14)\log(9/14) + (5/14)\log(5/14)] = 0.94$$

note:

Here typically we will take log to base 2. Here total there are 14 yes/no.

Out of which 9 yes and 5 no. Based on it we calculated probability above.

From the above data for outlook we can arrive at the following table easily



		play		
		yes	no	total
	sunny	3	2	5
Outlook	overcast	4	0	4
	rainy	2	3	5
				14

Now we have to calculate average weighted entropy. i.e., we have found the total of weights of each feature multiplied by probabilities.

E(S, outlook) = (5/14)*E(3,2) + (4/14)*E(4,0) + (5/14)*E(2,3)

= (5/14)(-(3/5)log(3/5)-(2/5)log(2/5))+ (4/14)(0) + (5/14)((2/5)log(2/5)(3/5)log(3/5))

= 0.693

The next step is to find the information gain. It is the difference between parent entropy and average weighted entropy we found above.

IG(S, outlook) = 0.94 - 0.693 = 0.247

Similarly find Information gain for Temperature, Humidity, and Windy.

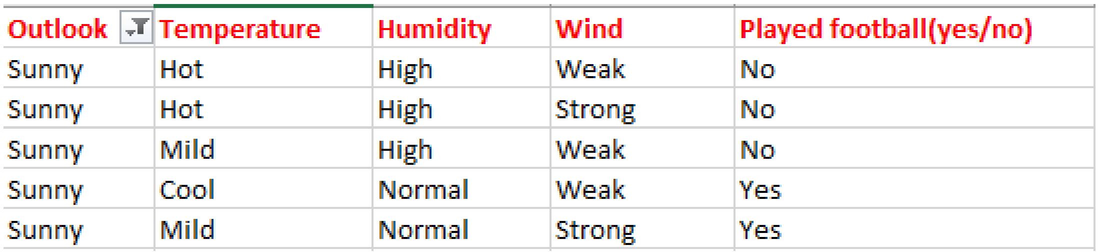
IG(S, Temperature) = 0.940 - 0.911 = 0.029

IG(S, Humidity) = 0.940 - 0.788 = 0.152

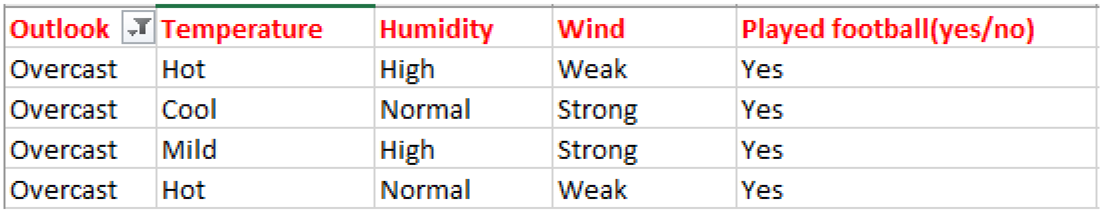
IG(S, Windy) = 0.940 - 0.8932 = 0.048

Now select the feature having the largest entropy gain. Here it is Outlook. So it forms the first node(root node) of our decision tree.

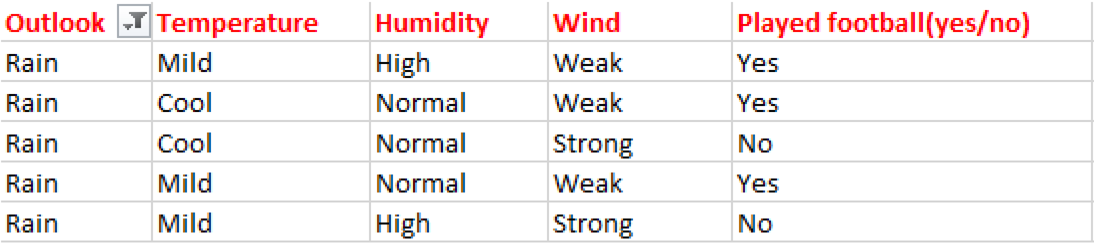
Now our data look as follows



Outlook	Temperature	Humidity	Wind	Played football(yes/no)
Sunny	Hot	High	Weak	No
Sunny	Hot	High	Strong	No
Sunny	Mild	High	Weak	No
Sunny	Cool	Normal	Weak	Yes
Sunny	Mild	Normal	Strong	Yes

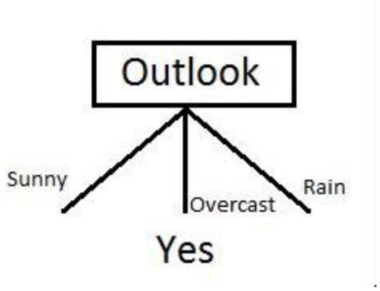


Outlook	Temperature	Humidity	Wind	Played football(yes/no)
Overcast	Hot	High	Weak	Yes
Overcast	Cool	Normal	Strong	Yes
Overcast	Mild	High	Strong	Yes
Overcast	Hot	Normal	Weak	Yes

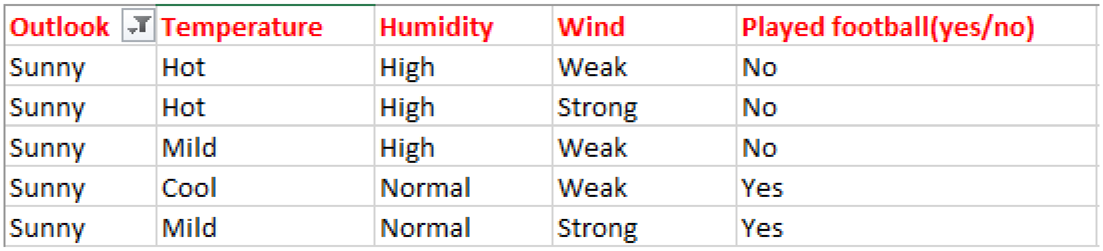


Outlook	Temperature	Humidity	Wind	Played football(yes/no)
Rain	Mild	High	Weak	Yes
Rain	Cool	Normal	Weak	Yes
Rain	Cool	Normal	Strong	No
Rain	Mild	Normal	Weak	Yes
Rain	Mild	High	Strong	No

Since overcast contains only examples of class ‘Yes’ we can set it as yes. That means If outlook is overcast football will be played. Now our decision tree looks as follows.



The next step is to find the next node in our decision tree. Now we will find one under sunny. We have to determine which of the following Temperature, Humidity or Wind has higher information gain.

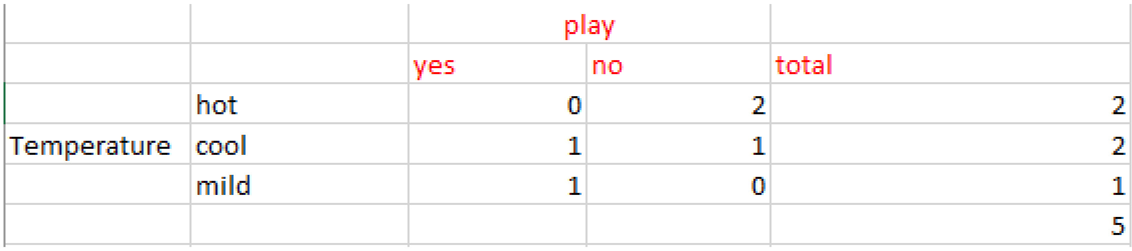


Outlook	Temperature	Humidity	Wind	Played football(yes/no)
Sunny	Hot	High	Weak	No
Sunny	Hot	High	Strong	No
Sunny	Mild	High	Weak	No
Sunny	Cool	Normal	Weak	Yes
Sunny	Mild	Normal	Strong	Yes

Calculate parent entropy E(sunny)

E(sunny) = (-(3/5)log(3/5)-(2/5)log(2/5)) = 0.971.

Now Calculate the information gain of Temperature. IG(sunny, Temperature)



		play		
		yes	no	total
	hot	0	2	2
Temperature	cool	1	1	2
	mild	1	0	1
				5

E(sunny, Temperature) = (2/5)*E(0,2) + (2/5)*E(1,1) + (1/5)*E(1,0) =2/5 =0.4

Now calculate information gain.

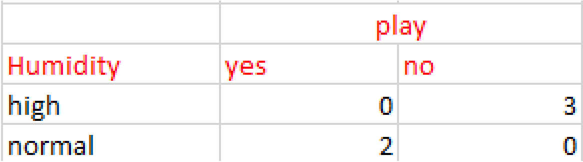
IG(sunny, Temperature) = 0.971–0.4 =0.571

Similarly we get

IG(sunny, Humidity) = 0.971

IG(sunny, Windy) = 0.020

Here IG(sunny, Humidity) is the largest value. So Humidity is the node that comes under sunny.

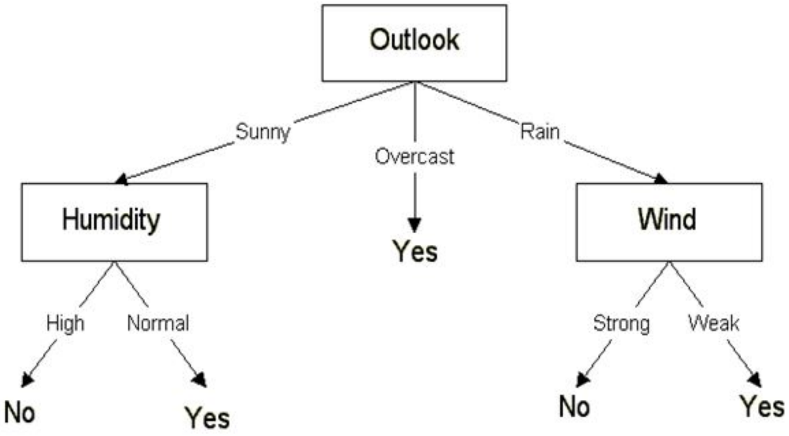


	play	
Humidity	yes	no
high	0	3
normal	2	0

For humidity from the above table, we can say that play will occur if humidity is normal and will not occur if it is high. Similarly, find the nodes under rainy.

Note: A branch with entropy more than 0 needs further splitting.

Finally, our decision tree will look as below:



Classification using CART algorithm:

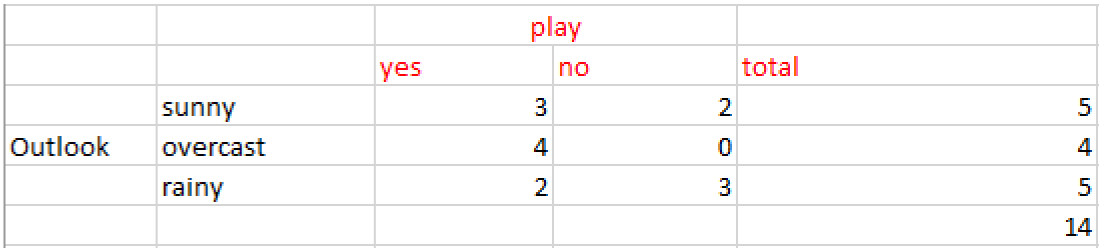
Classification using CART is similar to it. But instead of entropy, we use Gini impurity.

So as the first step we will find the root node of our decision tree. For that Calculate the Gini index of the class variable

$$\text{Gini}(S) = 1 - [(9/14)^2 + (5/14)^2] = 0.4591$$

As the next step, we will calculate the Gini gain. For that first, we will find the average weighted Gini impurity of Outlook, Temperature, Humidity, and Windy.

First, consider case of Outlook



		play		
		yes	no	total
	sunny	3	2	5
Outlook	overcast	4	0	4
	rainy	2	3	5
				14

Gini(S, outlook) = (5/14)gini(3,2) + (4/14)*gini(4,0)+ (5/14)*gini(2,3)

= $(5/14)(1 - (3/5)^2 - (2/5)^2) + (4/14)*0 + (5/14)(1 - (2/5)^2 - (3/5)^2)$

= 0.171+0+0.171 = 0.342

Gini gain (S, outlook) = 0.459 - 0.342 = 0.117

Gini gain(S, Temperature) = 0.459 - 0.4405 = 0.0185

Gini gain(S, Humidity) = 0.459 - 0.3674 = 0.0916

Gini gain(S, windy) = 0.459 - 0.4286 = 0.0304

Choose one that has a higher Gini gain. Gini gain is higher for outlook. So we can choose it as our root node.

Now you have got an idea of how to proceed further. Repeat the same steps we used in the ID3 algorithm.

*****