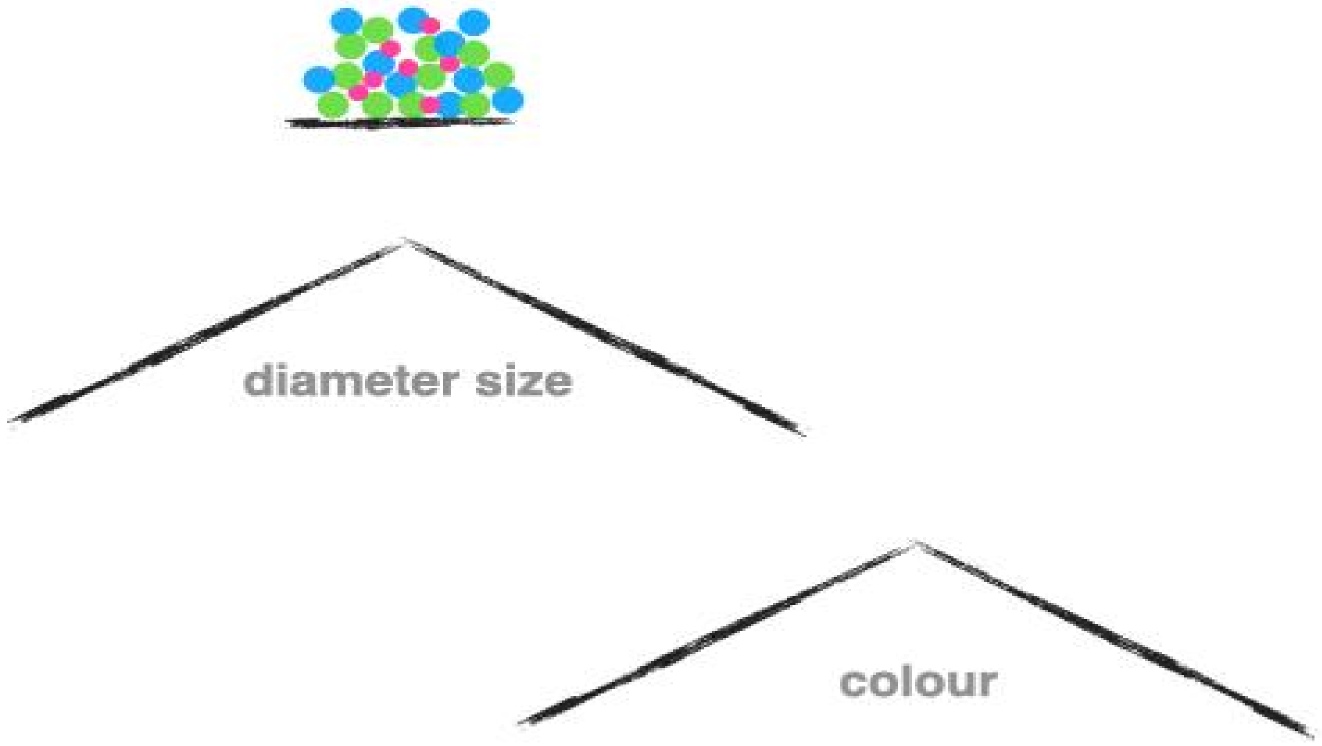
Decision Trees

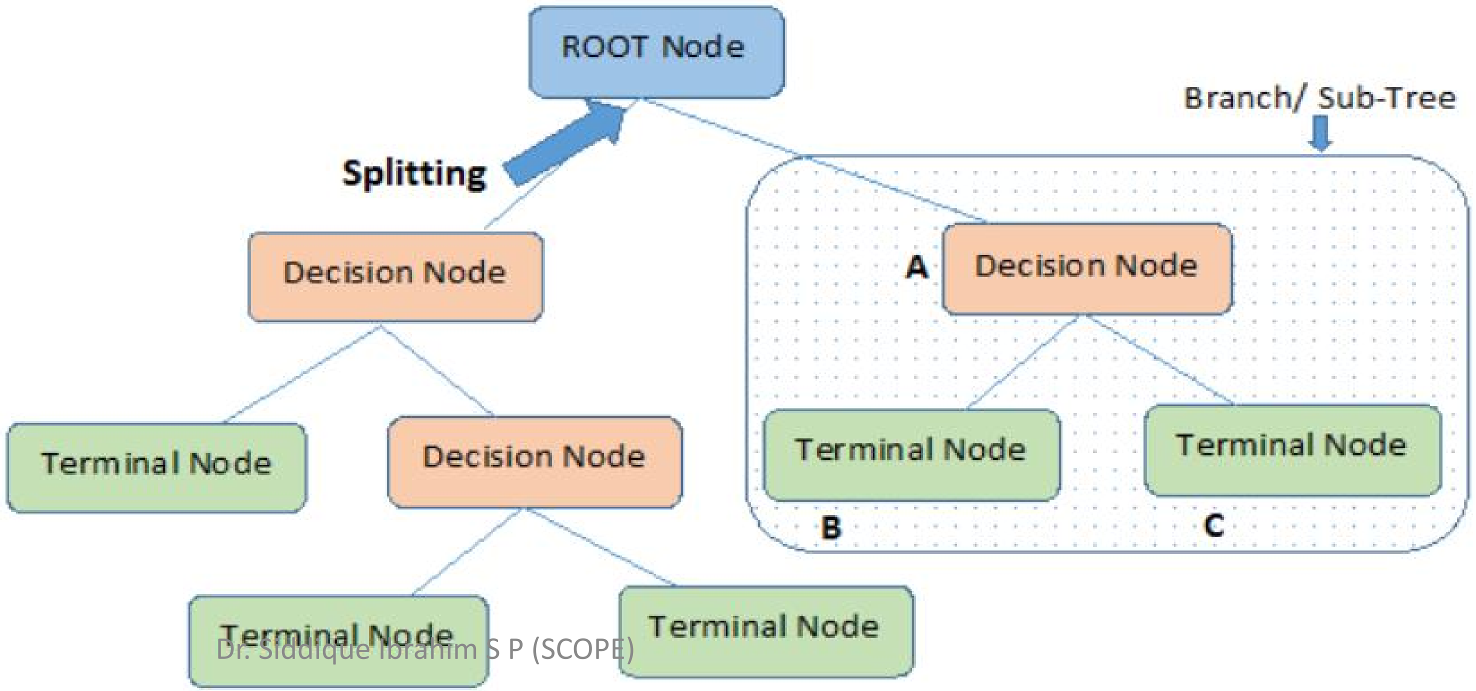


Decision Trees

- Decision Trees is a non-parametric Supervised learning technique that can be used for both classification and Regression 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.
- Leaf nodes are the output of those decisions and do not contain any further branches.

Decision Tree

- 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.
- A decision tree simply asks a question, and based on the answer (Yes/No), it further split the tree into subtrees.



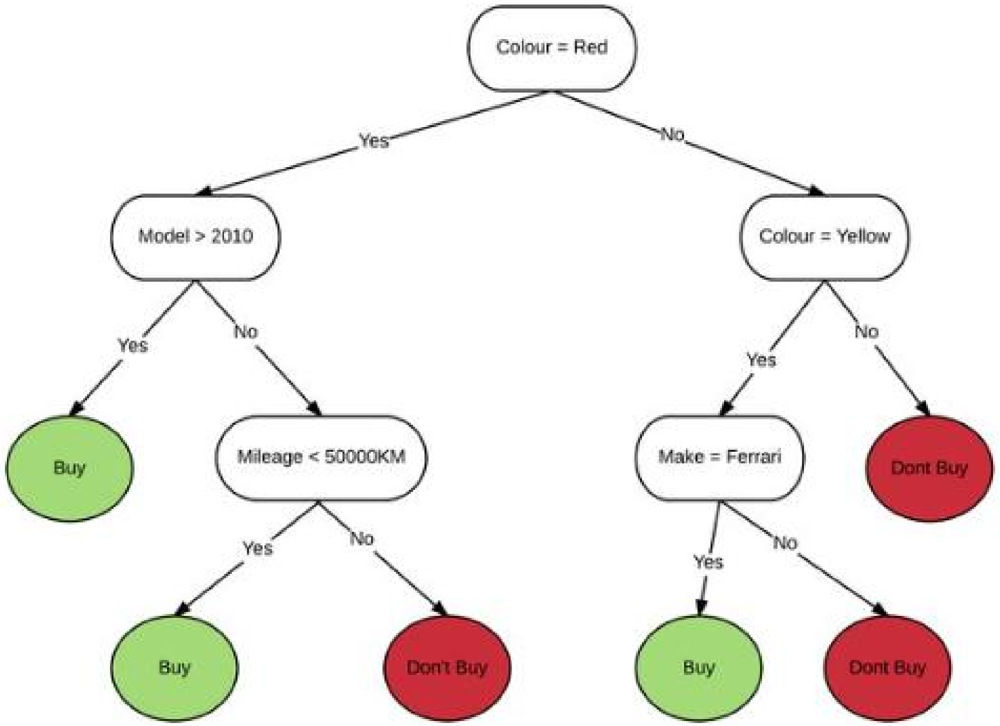
- It breaks down a dataset into smaller and smaller subsets while at the same time an associated decision tree is incrementally developed.
- Decision trees can handle both categorical and numerical data.

Decision Tree Terminologies

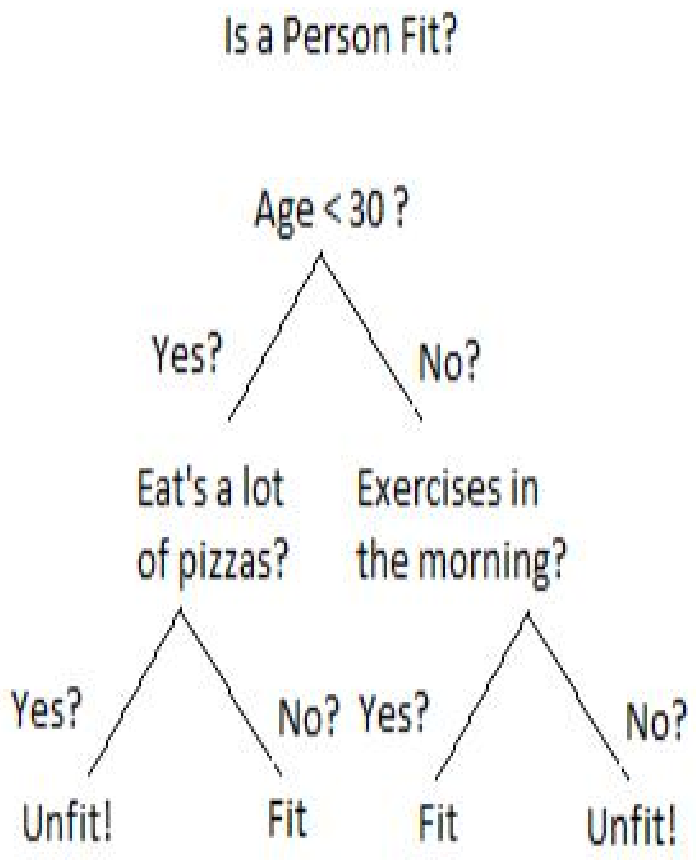
- Root Node: Root node is from where the decision tree starts. It represents the entire dataset, which further gets divided into two or more homogeneous sets.
- Leaf Node: Leaf nodes are the final output node, and the tree cannot be segregated further after getting a leaf node.
- Splitting: Splitting 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

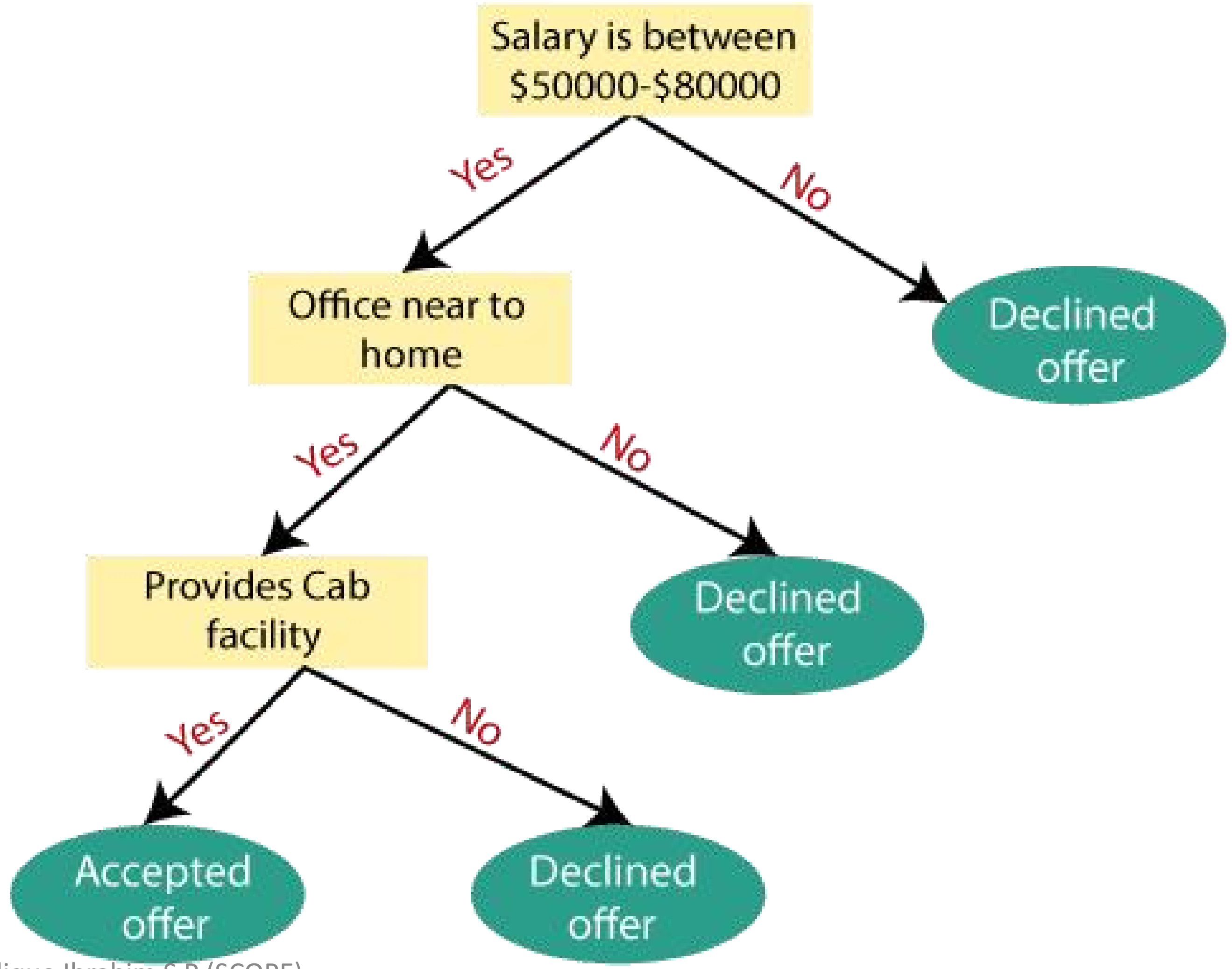
- In order to classify an unknown sample, the attribute values of the sample are tested against the decision Tree.
- The path is traced from the root to a leaf node the holds the class prediction for that sample.
- Decision Tree can easily be converted to classification rules.



Examples

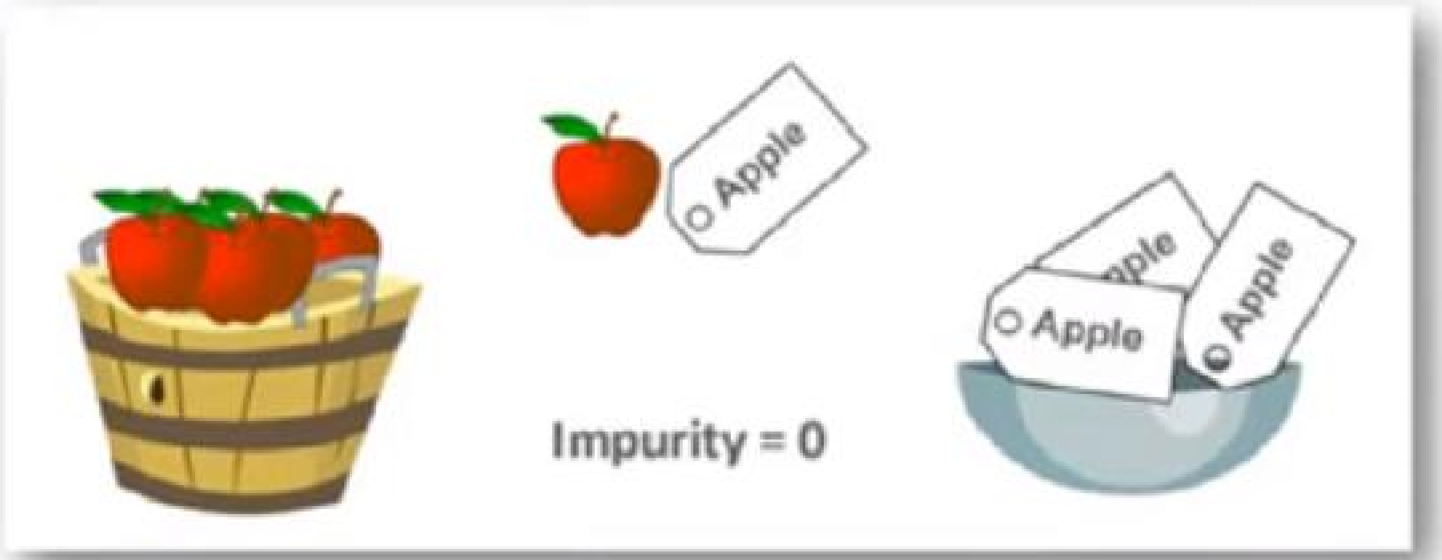


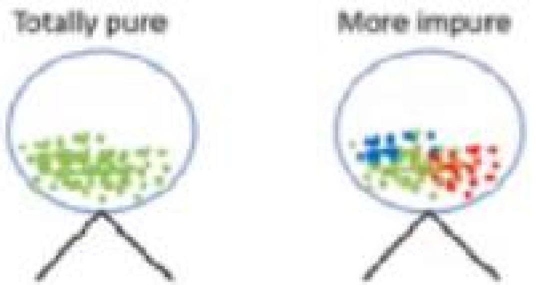
Is a Person Fit?
Age < 30 ?
Yes?
No?
Eat's a lot of pizzas?
Exercises in the morning?
Yes?
No?
Yes?
No?
Unfit!
Fit
Fit
Unfit!

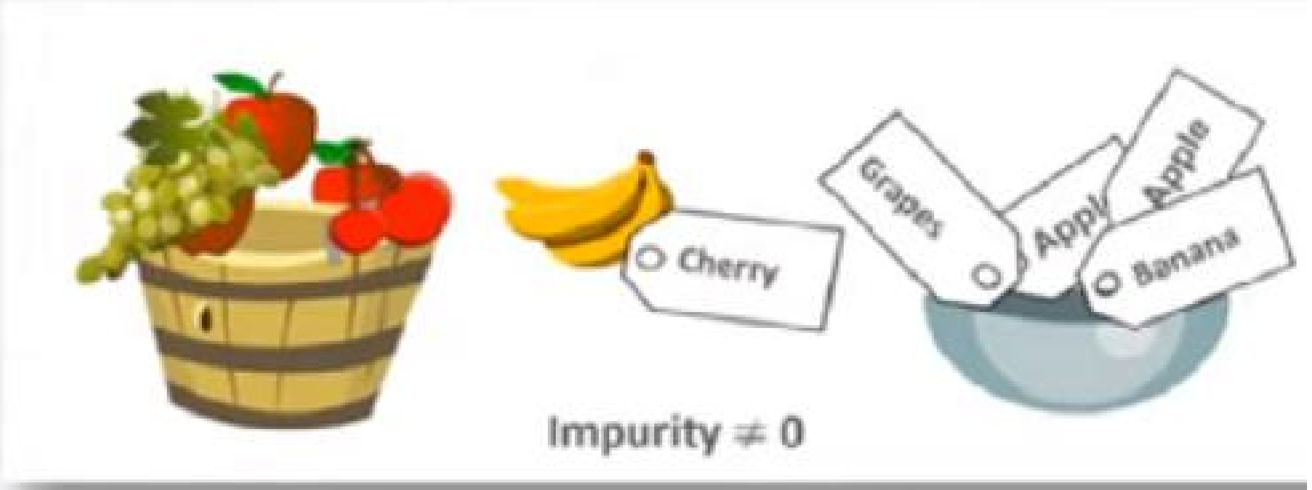


Salary is between $50000-$80000
Yes
No
Office near to home
Declined offer
Yes
No
Provides Cab facility
Declined offer
Yes
No
Accepted offer
Declined offer

What is Impurity in Decision Tree?







Types of Decision Trees

- There are two main types of Decision Trees:
  - Classification trees
  - Regression trees
- Classification trees (Yes/No types)
- Regression trees (Continuous data types)
  - Here the decision or the outcome variable is **Continuous**, Ex, a number like 1 2 3 4 5...
    - **Iterative Dichotomiser 3** (ID3 Algorithm)

Why use Decision Trees?

- 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.

uppose the following training dataset is given. We need to determine the class label sing the four features age, income, student, credit_rating.

RID	*age*	*income*	*student*	*credit_rating*	*Class: buys_computer*
1	youth	high	no	fair	no
2	youth	high	no	excellent	no
3	middle_aged	high	no	fair	yes
4	senior	medium	no	fair	yes
5	senior	low	yes	fair	yes
6	senior	low	yes	excellent	no
7	middle_aged	low	yes	excellent	yes
8	youth	medium	no	fair	no
9	youth	low	yes	fair	yes
10	senior	medium	yes	fair	yes
11	youth	medium	yes	excellent	yes
12	middle_aged	medium	no	excellent	yes
13	middle_aged	high	yes	fair	yes
14	senior	medium	no	excellent	no

Measuring Impurity

Given a data table that contains attributes and class of the attributes, we can measure homogeneity (or heterogeneity) of the table based on the classes. We say a table is pure or homogenous if it contains only a single class. If a data table contains several classes, then we say that the table is impure or heterogeneous. There are several indices to measure degree of impurity quantitatively. Most well known indices to measure degree of impurity are entropy, gini index, and classification error. The formulas are given below

$$Entropy = \sum_j -p_j log_2 p_j$$

$$Gini\ Index = 1 - \sum_j p_j^2$$

$$Classification\ Error = 1 - max\{p_j\}$$

There are three different ways to make a split in the decision tree.

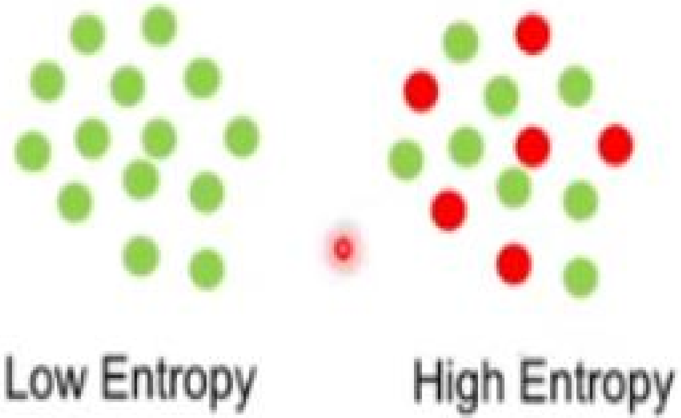
1. Information Gain and Entropy
2. Gini index
3. Gain ratio

Entropy

Entropy:

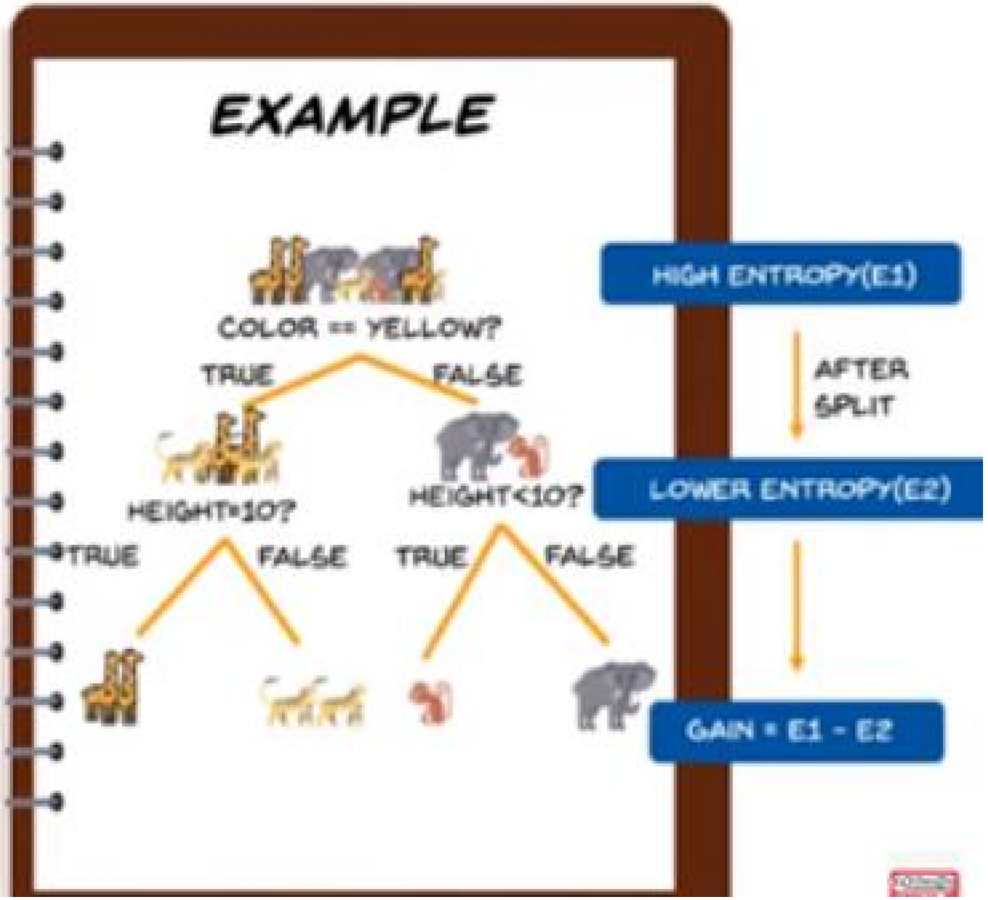
In Machine Learning, **Entropy** is the quantitative measure of the **randomness** of the information being processed.

A **high value of Entropy** means that the **randomness** in the system is **high** and thus making accurate predictions is tough.
A **low value of Entropy** means that the **randomness** in the system is **low** and thus making accurate predictions is easier.



What is Entropy?

- **Entropy:** Entropy is a metric to measure the impurity in a given attribute.
- It is the degree of randomness in data.



Entropy can be calculated as:

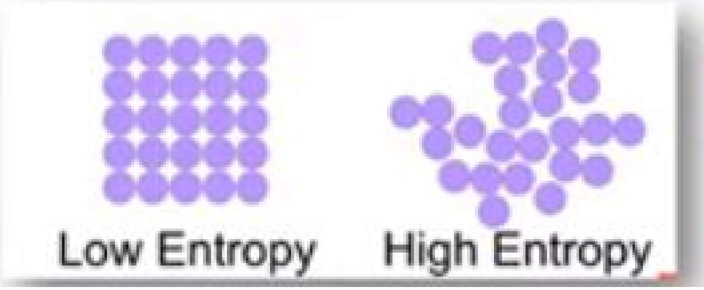
Entropy(s)= -P(yes)log2 P(yes)- P(no) log2 P(no)

Where,

S= Total number of samples

P(yes)= probability of yes

P(no)= probability of no



Building a Decision Tree

1. First test all attributes and select the **one attribute** that would function as the **best** root.
2. Break-up the training set into **subsets** based on the branches of the root node.
3. Test the **remaining attributes** to check which one fit best underneath the **branches** of the root node;
4. Continue this process for all other branches until
   a. all examples of a subset are of one type
   b. there are no examples left (return majority classification of the parent)
   c. there are **no more** attributes left (default value should be majority classification)

Working principles of Decision Tree 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.

Attribute (Feature) Selection Measures

1. Information Gain   2. Gini Index

Information gain:

- ***Information gain*** measures how well a given attribute separates the training examples according to their target classification.
- Information Gain refers to the **decline** (changes) in entropy after the dataset is called split (**Entropy Reduction**). It calculates how much information gained from a feature about a class.
- Split the node based on information gain value, and build the decision tree.
- Decision tree algorithm always attempts to maximize the information gain value.
- Node/attribute contains the **highest information gain** is splitted first.
- Calculate Gain by find the difference between the *entropy* before and the *entropy* after the split

Information Gain = Entropy(S) - [(Weighted Avg) * Entropy(each feature)]

$$\text{Gain}(S, A) = \text{Entropy}(S) - I(\text{attribute})$$

Entropy

- Entropy, also called as Shannon Entropy is denoted by H(S) for a finite set S, is the measure of the amount of uncertainty or randomness in data.

$$H(S) = \sum_{x \in X} p(x) \log_2 \frac{1}{p(x)}$$

Entropy H(s)= -P(Yes)log2 P(Yes)- P(No) log2 P(No)

Binary or boolean classification

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

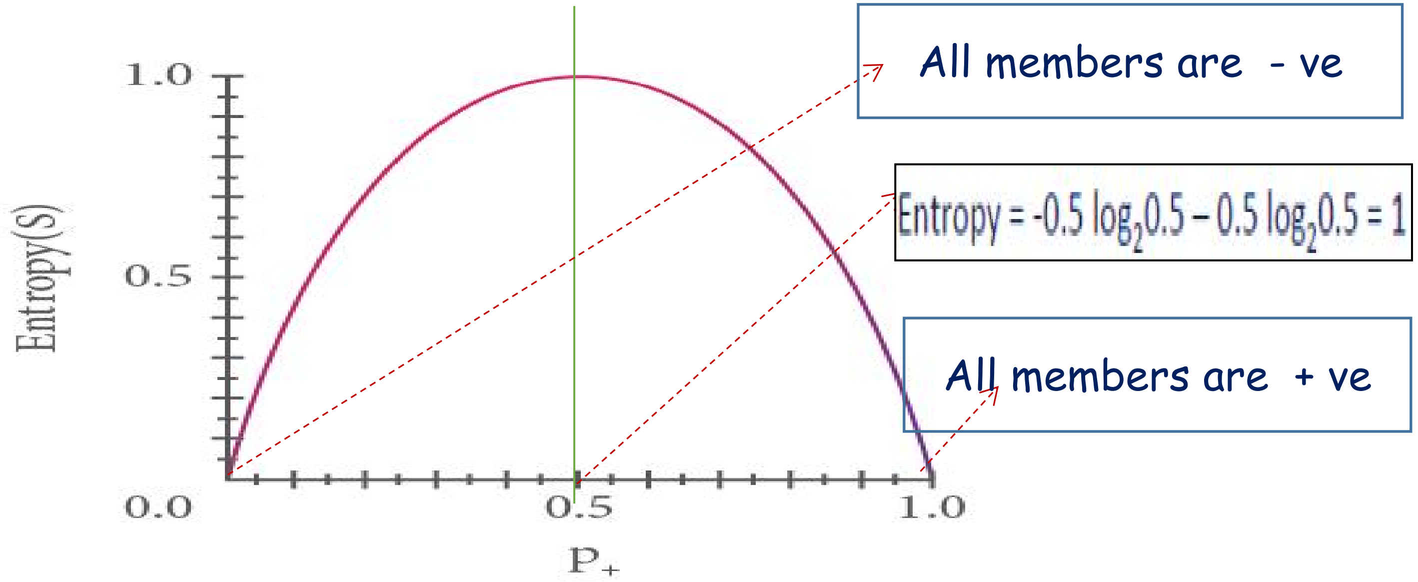
Multi-class classification

Where,

- S= Total number of samples
- P(Yes)= Probability of Yes
- P(No)= Probability of No

Example

- For the set S = {Y,Y,Y,N,N,N,N,N}
- Total instances: 8
- Instances of N: 5
- Instances of Y: 3



Entropy H(S)= -P(yes)log2 P(yes)+ P(no) log2 P(no)

$$EntropyH(s) = -\left[\left(\frac{3}{8}\right) log_2 \frac{3}{8} + \left(\frac{5}{8}\right) log_2 \frac{5}{8}\right]$$

= -[0.375 * (-1.415) + 0.625 * (-0.678)]

=-(-0.53-0.424)

= 0.954

- If number of yes = number of no, Then P(s)=0.5 and Entropy(s) = 1
- If it contains either all yes or all no, Then P(s) = 1 or 0 and Entropy(s) = 0

Attribute Selection Measures

- Attribute selection measure or ASM used to select the best attribute for the nodes of the tree.

    - Information Gain
    - Gini Index

- Information Gain
    - Information gain is the measurement of changes in entropy after the segmentation of a dataset based on an attribute.
    - It is the measure of how good an attribute is for predicting the class of each of the training data..
    - According to the value of information gain, we split the node and build the decision tree.

Information Gain

- Information Gain= Entropy(S)-[(Weighted Avg) *Entropy(each feature) (or)

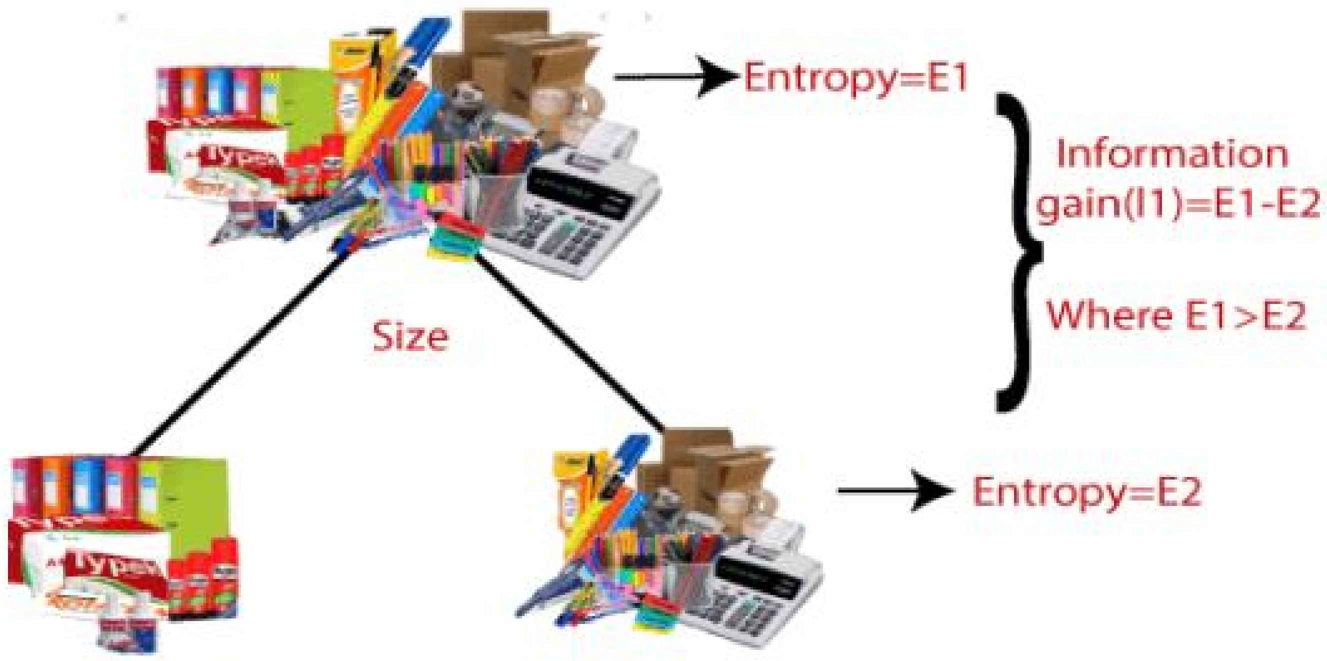
$$IG(S, A) = H(S) - H(S, A)$$

(or)

$$\text{Information Gain} = \mathbf{H(S) - H(S|X)}$$

Decision tree - ID3 algorithm

- **Iterative Dichotomiser 3** (ID3) algorithm uses this *information gain* measure to select among the candidate attributes at each step that return the **highest** data gain while growing the tree.



Training Dataset

	Age	Income	Student	Credit	Buys_computer
P1	<=30	high	no	fair	no
P2	<=30	high	no	excellent	no
P3	31...40	high	no	fair	yes
P4	>40	medium	no	fair	yes
P5	>40	low	yes	fair	yes
P6	>40	low	yes	excellent	no
P7	31...40	low	yes	excellent	yes
P8	<=30	medium	no	fair	no
P9	<=30	low	yes	fair	yes
P10	>40	medium	yes	fair	yes
P11	<=30	medium	yes	excellent	yes
P12	31...40	medium	no	excellent	yes
P13	31...40	high	yes	fair	yes
P14	>40	medium	no	excellent	no

- Entropy in D: We now put calculate the Entropy by putting probability values in the formula stated above.

$$Info(D) = I(9,5) = -\frac{9}{14}\log_2\left(\frac{9}{14}\right) - \frac{5}{14}\log_2\left(\frac{5}{14}\right) = 0.940$$

Information Gain (Age) =0.246

Information Gain (Income) =0.029

Information Gain (Student) = 0.151

Information Gain (credit_rating) =0.048

Gini Index

- Gini index is a measure of impurity or purity used while creating a decision tree in the CART(Classification and Regression Tree) algorithm.
- An attribute with the low Gini index should be preferred as compared to the high Gini index.
- It only creates binary splits, and the CART algorithm uses the Gini index to create binary splits.
- Gini index can be calculated using the below formula:

$$\text{Gini Index} = 1 - \sum (P(x=k))^2$$

Decision Tree classifier (ID3)

Decision tree generation consists of two phases:

Tree construction

- Initially all the training examples are at the root
- Attributes are categorical(if continuous-valued, they are discretized in advance)
- Partition examples based on selected attributes
- Attributes are selected on the basis of a heuristic or statistical measure (e.g.,information gain)

Tree pruning

- Identify and remove branches that reflect noise or outliers.

Decision Tree classifier (ID3)

ID3 Steps

- Calculate the Information Gain of each feature.
- Considering that all rows don't belong to the same class, split the dataset S into subsets using the feature for which the Information Gain is maximum.
- Make a decision tree node using the feature with the maximum Information gain.
- If all rows belong to the same class, make the current node as a leaf node with the class as its label.
- Repeat for the remaining features until we run out of all features, or the decision tree has all leaf nodes.

Example :dataset of COVID-19 infection

Binary or Boolean classification

ID	Fever	Cough	Breathing issues	Infected
1	NO	NO	NO	NO
2	YES	YES	YES	YES
3	YES	YES	NO	NO
4	YES	NO	YES	YES
5	YES	YES	YES	YES
6	NO	YES	NO	NO
7	YES	NO	YES	YES
8	YES	NO	YES	YES
9	NO	YES	YES	YES
10	YES	YES	NO	YES
11	NO	YES	NO	NO
12	NO	YES	YES	YES
13	NO	YES	YES	NO
14	YES	YES	NO	NO

Example Cont'd

- From the total of 14 rows in our dataset S, there are 8 rows with the target value YES and 6 rows with the target value NO.
- The entropy of **S** is calculated as:

$$\text{Entropy } H(S) = -(8/14) * \log_2(8/14) - (6/14) * \log_2(6/14)$$

$$= 0.99$$

- Next step calculate the Information Gain for each feature.

Example Cont'd

- Information Gain Calculation for Fever:

Fever	Cough	Breathing issues	Infected
YES	YES	YES	YES
YES	YES	NO	NO
YES	NO	YES	YES
YES	YES	YES	YES
YES	NO	YES	YES
YES	NO	YES	YES
YES	YES	NO	YES
YES	YES	NO	NO

In our Data set **8** rows with **YES for** Fever, there are **6** rows having target value **YES** and **2** rows having target value **NO.**

Total rows

$|S| = 14$

For v = YES, $|S_v| = 8$

$\text{Entropy}(S_v) = - (6/8) * \log_2(6/8) - (2/8) * \log_2(2/8) = 0.81$

For v = NO, $|S_v| = 6$

$\text{Entropy}(S_v) = - (2/6) * \log_2(2/6) - (4/6) * \log_2(4/6) = 0.91$

Expanding the summation in the IG formula:

H(S, Fever) = Entropy(S) - (|S_{YES}| / |S|) * Entropy(S_{YES}) - (|S_{NO}| / |S|) * Entropy(S_{NO})

$H(S, \text{Fever}) = 0.99 - (8/14) * 0.81 - (6/14) * 0.91 = 0.13$

Example Cont'd

- **Information Gain Calculation for Cough:**

Fever	Cough	Breathing issues	Infected
YES	YES	YES	YES
YES	YES	NO	NO
YES	YES	YES	YES
NO	YES	NO	NO
NO	YES	YES	YES
YES	YES	NO	YES
NO	YES	NO	NO
NO	YES	YES	YES
NO	YES	YES	NO
YES	YES	NO	NO

In our Data set **10** rows with **YES for** Fever, there are **5** rows having target value **YES** and **5** rows having target value **NO.**

Total rows

$|S| = 14$

For v = YES, $|S_v| = 8$

$\text{Entropy}(S_v) = -(5/8) * \log_2(5/8) - (5/8) * \log_2(5/8) = 0.84$

For v = NO, $|S_v| = 6$

$\text{Entropy}(S_v) = -(5/6) * \log_2(5/6) - (1/6) * \log_2(1/6) = 0.68$

Expanding the summation in the IG formula:

$H(S, \text{Cough}) = \text{Entropy}(S) - (|S_{YES}| / |S|) * \text{Entropy}(S_{YES}) - (|S_{NO}| / |S|) * \text{Entropy}(S_{NO})$

$H(S, \text{Cough}) = 0.99 - (8/14) * 0.84 - (6/14) * 0.68 = 0.21$

Example Cont'd

- **Information Gain Calculation for Breathing Issues:**

ID	Fever	Cough	Breathing issues	Infected
2	YES	YES	YES	YES
4	YES	NO	YES	YES
5	YES	YES	YES	YES
7	YES	NO	YES	YES
8	YES	NO	YES	YES
9	NO	YES	YES	YES
12	NO	YES	YES	YES
13	NO	YES	YES	NO

In our Data set **8** rows with **YES for** Fever, there are **7** rows having target value **YES** and **1** row having target value **NO.**

Total rows
$|S| = 14$
For v = YES, $|S_v| = 8$
$Entropy(S_v) = - (7/8) * \log_2(7/8) - (1/8) * \log_2(1/8) = 0.54$

For v = NO, $|S_v| = 6$
$Entropy(S_v) = - (1/6) * \log_2(1/6) - (5/6) * \log_2(5/6) = 0.65$

Expanding the summation in the IG formula:

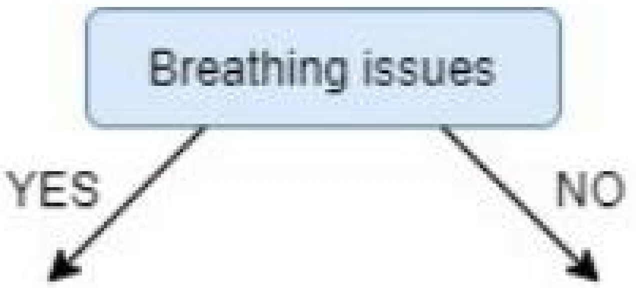
H(S, Breathing Issues) = Entropy(S) - ($|S_{YES}|$ / $|S|$) * Entropy(S_{YES}) - ($|S_{NO}|$ / $|S|$) * Entropy(S_{NO})

H(S, Breathing Issues) = 0.99 - (8/14) * 0.54 - (6/14) * 0.65 = .40

Example Cont'd

- Since the feature **Breathing issues** have the highest Information Gain it is used to create the root node. Hence, after this initial step our tree looks like this:

H(S, Fever) = 0.13
H(S, Cough)= 0.21
H(S, Breathing Issues) =**0.40**



- Next, from the remaining two unused features, namely, Fever and Cough, we decide which one is the best for the left branch of Breathing Issues.

Example Cont'd

- Since the left branch of **Breathing Issues** denotes **YES**, we will work with the subset of the original data i.e the set of rows having **YES** as the value in the Breathing Issues column. These **8 rows** are shown below:

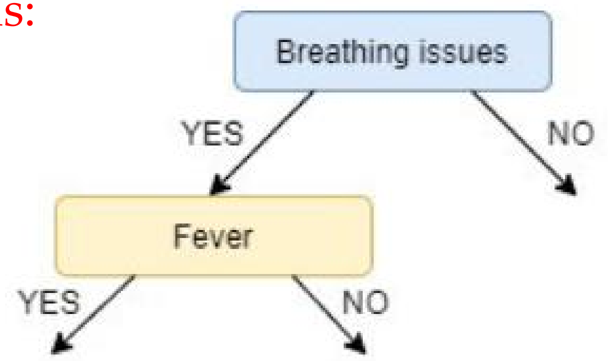
Fever	Cough	Breathing issues	Infected
YES	YES	YES	YES
YES	NO	YES	YES
YES	YES	YES	YES
YES	NO	YES	YES
YES	NO	YES	YES
NO	YES	YES	YES
NO	YES	YES	YES
NO	YES	YES	NO

Information Gain(S_{BY}, Fever) = 0.20
Information Gain(S_{BY}, Cough) = 0.09

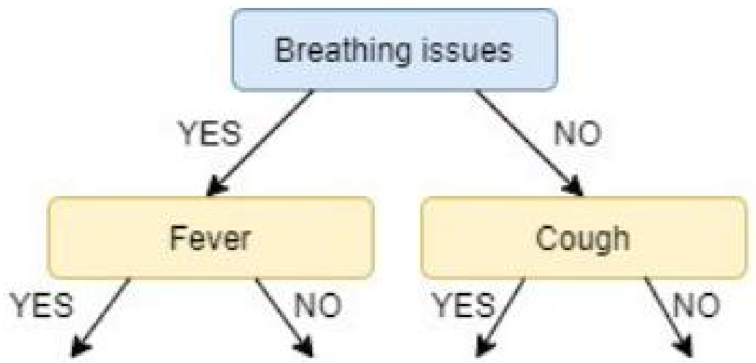
- IG of Fever is greater than that of Cough, so we select **Fever** as the left branch of Breathing Issues.

Example Cont'd

Our tree now looks like this:



But, since there is only one unused feature left we have no other choice but to make it the right branch of the root node. So our tree now looks like this:



Example Cont'd

- There are no more unused features, so we stop here and jump to the final step of creating the leaf nodes. For the left leaf node of Fever, we see the subset of rows from the original data set that has **Breathing Issues** and **Fever** both values as **YES**.

Fever	Cough	Breathing issues	Infected
YES	YES	YES	YES
YES	NO	YES	YES
YES	YES	YES	YES
YES	NO	YES	YES
YES	NO	YES	YES

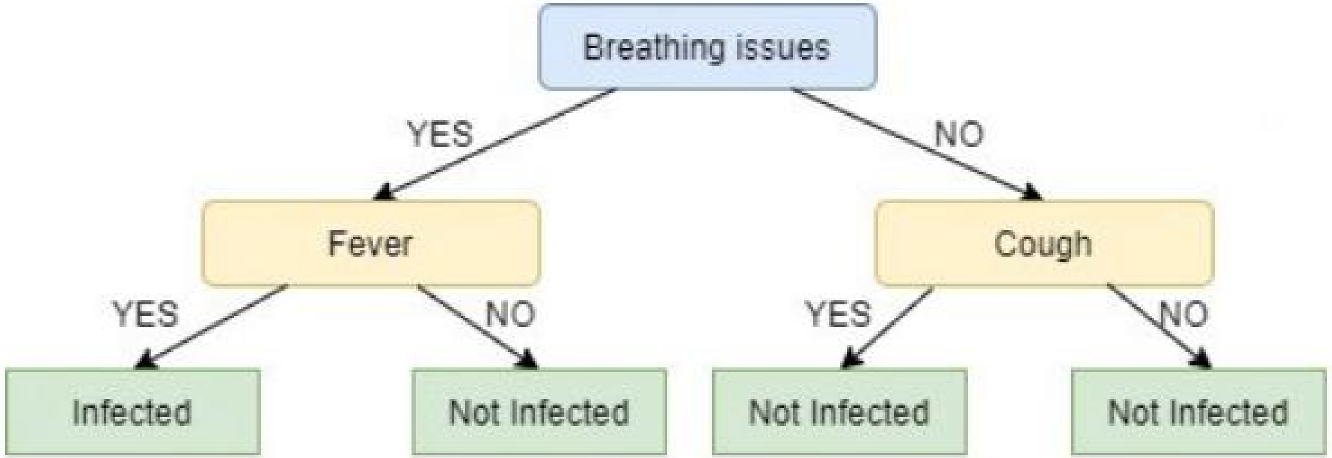
- Since all the values in the target column are **YES,** we label the left leaf node as **YES**, but to make it more logical we label it **Infected.**

- Similarly, for the right node of Fever we see the subset of rows from the original data set that have **Breathing Issues** value as **YES** and **Fever** as **NO**.

Example Cont'd

Fever	Cough	Breathing issues	Infected
NO	YES	YES	YES
NO	YES	YES	NO
NO	YES	YES	NO

- Here not all but **most** of the **values** are **NO,** hence **NO** or **Not Infected** becomes our **right leaf node.** We repeat the same process for the node **Cough**, however here both left and right leaves turn out to be the same i.e. **NO** or **Not Infected** as shown below :



Example

Day	Outlook	Temprature	Humidity	Wind	Play_Tennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

- S is a collection of 14 examples of a Boolean concept, including 9 positive and 5 negative examples [9+, 5].

Then the entropy of S relative to this Boolean classification is:

$$Entropy([9+, 5-]) = -(9/14)\log_2(9/14) - (5/14)\log_2(5/14)$$
$$= 0.940$$

Example Cont'd

The next step calculates the Information Gain for each feature.

Information Gain Calculation for Outlook

Day	Outlook	Temperature	Humidity	Wind	Play_Tennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes

1. Sunny (5 Times) : In the given data, 5 days were sunny. Among those 5 days, tennis was played on 2 days and tennis was not played on 3 days.

Probability of playing tennis = 2/5 = 0.4
Probability of not playing tennis = 3/5 = 0.6
Entropy when sunny = -0.4 * log2(0.4) – 0.6 * log2(0.6)
= 0.97

2. Overcast: In the given data, 4 days were overcast and tennis was played on all four days.

Day	Outlook	Temperature	Humidity	Wind	Play_Tennis
D3	Overcast	Hot	High	Weak	Yes
D7	Overcast	Cool	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes

Probability of playing tennis = 4/4 = 1
Probability of not playing tennis = 0/4 = 0
Entropy when overcast = 0.0

Example Cont'd

Information Gain Calculation for Outlook Cont'd

Day	Outlook	Temperature	Humidity	Wind	Play_Tennis
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D10	Rain	Mild	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

3. Rain: In the given data, 5 days were rainy. Among those 5 days, tennis was played on 3 days and tennis was not played on 2 days

Probability of not playing tennis = 2/5 = 0.4
Probability of playing tennis = 3/5 = 0.6
Entropy when rainy = -0.6 * log2(0.6) – 0.4 * log2(0.4)
= 0.97

Entropy among the three branches

Entropy among three branches = ((number of sunny days)/(total days) * (entropy when sunny)) + ((number of overcast days)/(total days) * (entropy when overcast)) + ((number of rainy days)/(total days) * (entropy when rainy))

= ((5/14) * 0.97) + ((4/14) * 0) + ((5/14) * 0.97)
= 0.69

Information Gain = **H(S) - H(S|X)**

Reduction in randomness = entropy source – entropy of branches
= 0.940 – 0.69
= 0.246

Example Cont'd

The next step calculates the Information Gain for each feature.

Day	Outlook	Temperature	Humidity	Wind	Play_Tennis
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D9	Sunny	Cool	Normal	Weak	Yes

Information Gain Calculation for Temperature

1. Cool (4Times) : In the given data, 4 days were sunny. Among those 4 days, tennis was played on 3days and tennis was not played on 1 day.

Probability of playing tennis = 3/4= 0.75
Probability of not playing tennis = 1/4 = 0.25
Entropy when Cool = -0.75* log2(0.75) – 0.25 * log2(0.25)
= 0.81

2. Hot: In the given data, 4 days were Hot Among those 4 days, tennis was played on 2 days and tennis was not played on 2 days.

Day	Outlook	Temperature	Humidity	Wind	Play_Tennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D13	Overcast	Hot	Normal	Weak	Yes

Probability of playing tennis = 2/4 = 0.5
Probability of not playing tennis = 2/4 = 0.5
Entropy when Hot = 1

Example Cont'd

Information Gain Calculation for Temperature Cont'd

3. Mild: In the given data, 6 days were rainy. Among those 6 days, tennis was played on 5 days and tennis was not played on 1 day

Day	Outlook	Temprature	Humidity	Wind	Play_Tennis
D4	Rain	Mild	High	Weak	Yes
D8	Sunny	Mild	High	Weak	No
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D14	Rain	Mild	High	Strong	No

Probablity of playing tennis = 4/6 = 0.67
Probablity of not playing tennis = 2/6 = 0.33
Entropy when Mild = -0.67 * log2(0.67) – 0.33 * log2(0.33)
= 0.9179

Entropy among the three branches

Entropy among three branches = ((number of Cool days)/(total days) * (entropy when Cool)) + ((number of Hot days)/(total days) * (entropy when Hot)) + ((number of Mild days)/(total days) * (entropy when Mild))

= ((4/14) * 0.81) + ((4/14) * 1) + ((6/14) * 0.917)
= 0.9108

Information Gain = **H(S) - H(S|X)**

Reduction in randomness = entropy source – entropy of branches
= 0.940 – .9108
= 0.0292

Example Cont'd

Third Attribute - Humidity

H(Humidity=high) = -(3/7)*log(3/7)-(4/7)*log(4/7)
= 0.983

H(Humidity=normal) = -(6/7)*log(6/7)-(1/7)*log(1/7)
= 0.591

Day	Outlook	Temperature	Humidity	Wind	Play_Tennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D8	Sunny	Mild	High	Weak	No
D12	Overcast	Mild	High	Strong	Yes
D14	Rain	Mild	High	Strong	No

Day	Outlook	Temprature	Humidity	Wind	Play_Tennis
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes

Average Entropy Information for Humidity - I(Humidity) =
p(high)*H(Humidity=high) + p(normal)*H(Humidity=normal)
= (7/14)*0.983 + (7/14)*0.591
= 0.787

Information Gain = H(S) - I(Humidity)
= 0.94 - 0.787
= 0.153

Example Cont'd

Fourth Attribute - Wind

Categorical values - weak, strong

H(Wind=weak) = -(6/8)*log(6/8)-(2/8)*log(2/8) = 0.811

H(Wind=strong) = -(3/6)*log(3/6)-(3/6)*log(3/6) = 1

Average Entropy Information for Wind -

I(Wind) = p(weak)*H(Wind=weak) + p(strong)*H(Wind=strong)

= (8/14)*0.811 + (6/14)*1

= 0.892

Information Gain = H(S) - I(Wind)

= 0.94 - 0.892

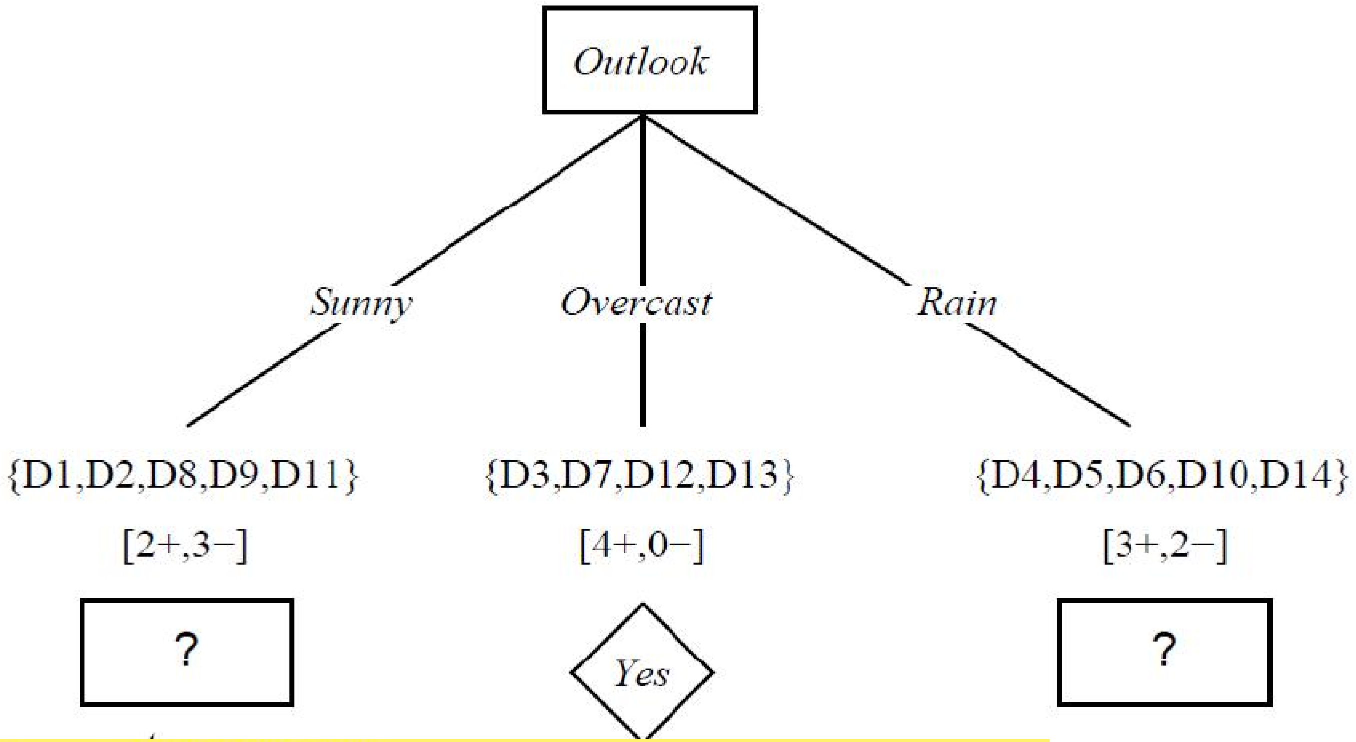
= 0.048

The information gain values for all four attributes are

- Gain(S, Outlook) = 0.246
- Gain(S, Humidity) = 0.151
- Gain(S, Wind) = 0.048
- Gain(S, Temperature) = 0.029

- According to the information gain measure, the Outlook attribute provides the best prediction of the target attribute, PlayTennis, over the training examples. Therefore, Outlook is selected as the decision attribute for the root node, and branches are created below the root for each of its possible values i.e., Sunny, Overcast, and Rain.

Example Cont'd



S_{sunny} = {D1,D2,D8,D9,D11}

$Gain(S_{sunny}, Humidity) = .970 - (3/5)\,0.0 - (2/5)\,0.0 = .970$

$Gain(S_{sunny}, Temperature) = .970 - (2/5)\,0.0 - (2/5)\,1.0 - (1/5)\,0.0 = .570$

$Gain(S_{sunny}, Wind) = .970 - (2/5)\,1.0 - (3/5)\,.918 = .019$

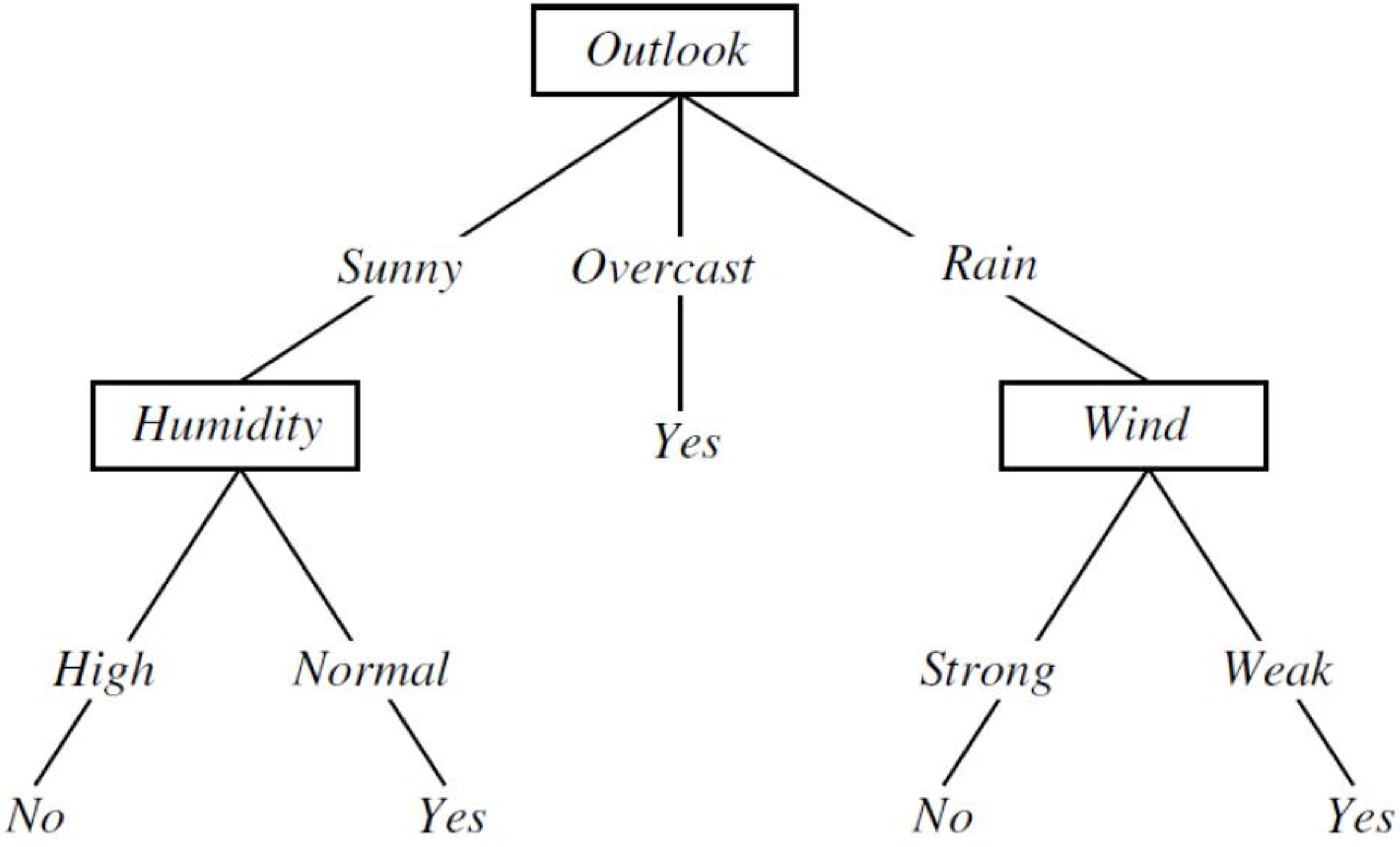
S_{Rain} = { D4, D5, D6, D10, D14}

$Gain(S_{Rain}, Humidity) = 0.970 - (2/5)1.0 - (3/5)0.917 = 0.019$

$Gain(S_{Rain}, Temperature) = 0.970 - (0/5)0.0 - (3/5)0.918 - (2/5)1.0 = 0.019$

$Gain(S_{Rain}, Wind) = 0.970 - (3/5)0.0 - (2/5)0.0 = 0.970$

Example Cont'd



CART: Classification and Regression Tree

Classification and Regression Tree

- The CART algorithm is a type of classification algorithm that is required to build a decision tree on the basis of Gini's impurity index.
- It is a basic machine learning algorithm and provides a wide variety of use cases.
- It is a dynamic learning algorithm that can produce a regression tree as well as a classification tree depending upon the dependent variable.
- CART can be applied to both regression and classification problems

Steps in Classification and Regression Tree:

- The algorithm works repeatedly in three steps:
  - Find each feature's best split. For each feature with K different values there exist K-1 possible splits. Find the split, which maximizes the splitting criterion. The resulting set of splits contains the best splits (one for each feature).
  - Find the node's best split. Among the best splits from Step, I find the one, which maximizes the splitting criterion.
  - Split the node using the best node split from Step ii and repeat from Step I until the stopping criterion is satisfied

Example

Outlook	Temp	Humidity	Windy	Play
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

- For the given Play Tennis Data set apply the Decision Tree algorithm and find the optimal decision tree.
- Also predict class label for the following example...?

Outlook	Temp	Humidity	Windy	Play
Sunny	Hot	Normal	TRUE	?

Outlook	Temp	Humidity	Windy	Play
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

Outlook

Overcast	4	Yes	4
		No	0
Sunny	5	Yes	2
		No	3
Rainy	5	Yes	3
		No	2

Attributes	Rules	Error	Total Error
Outlook	Overcast	0/4	4/14
	Sunny	2/5	
	Rainy	2/5	

Outlook	Temp	Humidity	Windy	Play
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

Temp

Hot	4	Yes	2
		No	2
Mild	6	Yes	4
		No	2
Cold	4	Yes	3
		No	1

Attributes	Rules	Error	Total Error
Temp	Hot	2/4	5/14
	Mild	2/6	
	Cool	1/4	

Outlook	Temp	Humidity	Windy	Play
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

Humidity

High	7	Yes	3
		No	4
Normal	7	Yes	6
		No	1

Attributes	Rules	Error	Total Error
Humidity	High	3/7	4/14
	Normal	1/7	

Outlook	Temp	Humidity	Windy	Play
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

Windy

False	8	Yes	6
		No	2
True	6	Yes	3
		No	3

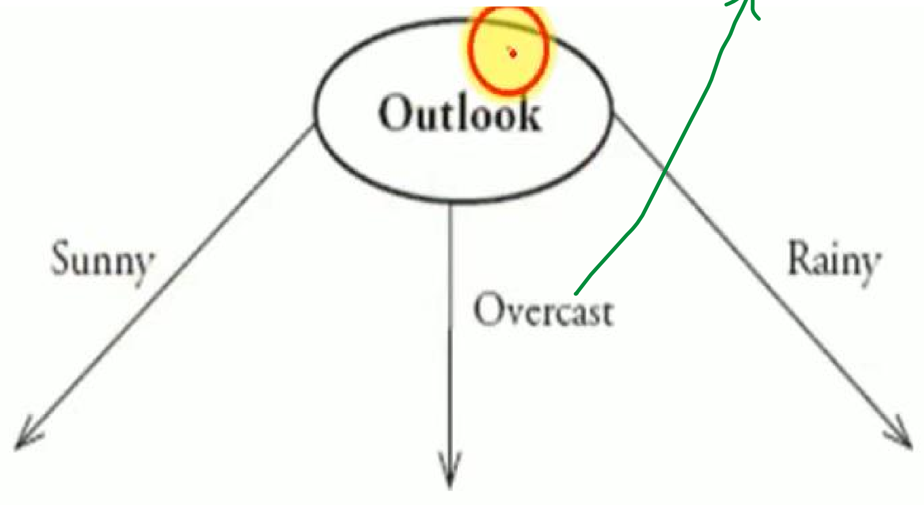
Attributes	Rules	Error	Total Error
Windy	FALSE	2/8	5/14
	TRUE	3/6	

Outlook	Temp	Humidity	Windy	Play
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

Attributes	Rules	Error	Total Error
Outlook	Overcast	0/4	4/14
	Sunny	2/5	
	Rainy	2/5	
Temp	Hot	2/4	5/14
	Mild	2/6	
Humidity	High	3/7	4/14
	Normal	1/7	
Windy	FALSE	2/8	5/14
	TRUE	3/6	

Outlook	Temp	Humidity	Windy	Play
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

Day	Outlook	Temp.	Humidity	Wind	Decision
3	Overcast	Hot	High	Weak	Yes
7	Overcast	Cool	Normal	Strong	Yes
12	Overcast	Mild	High	Strong	Yes
13	Overcast	Hot	Normal	Weak	Yes



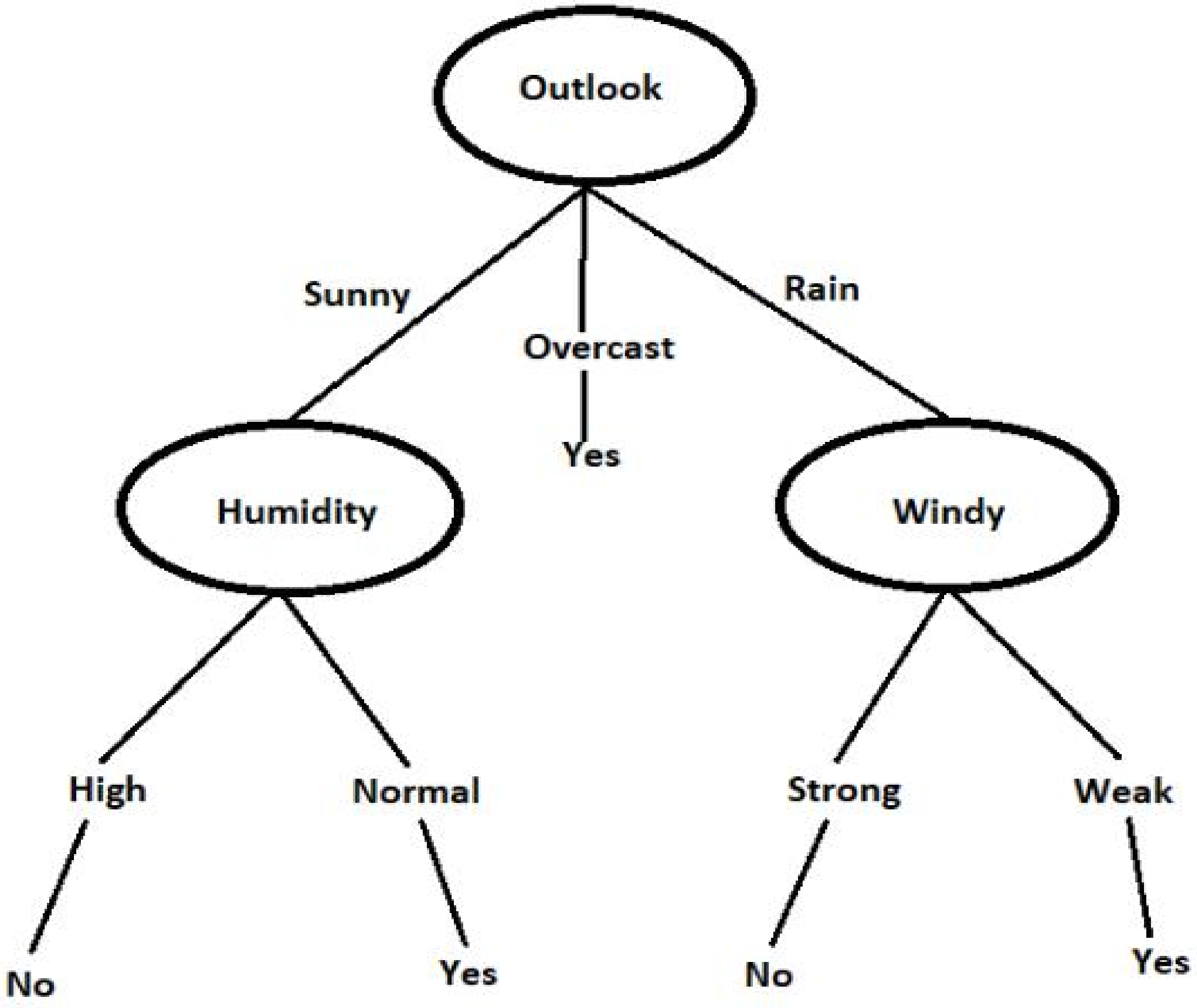
Temp	Humidity	Windy	Play
Hot	High	False	No
Hot	High	True	No
Mild	High	False	No
Cool	Normal	False	Yes
Mild	Normal	True	Yes

$\boxed{\text{YES}}$

Temp	Humidity	Windy	Play
Mild	High	False	Yes
Cool	Normal	False	Yes
Cool	Normal	True	No
Mild	Normal	False	Yes
Mild	High	True	No

Outlook	Temp	Humidity	Windy	Play
Sunny	Hot	High	FALSE	No
Sunny	Hot	High	TRUE	No
Sunny	Mild	High	FALSE	No
Sunny	Cool	Normal	FALSE	Yes
Sunny	Mild	Normal	TRUE	Yes

Attributes	Rules	Error	Total Error
Temp	Hot	0/2	1/5
	Mild	1/2	
	Cool	0/1	
Humidity	High	0/3	**0/5**
	Normal	0/2	
Windy	FALSE	1/3	2/5
	TRUE	1/2	


Outlook
Sunny
Overcast
Rain
Yes
Humidity
Windy
High
Normal
Strong
Weak
No
Yes
No
Yes

CART - By using Gini index

Outlook	Temp	Humidity	Windy	Play
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

CART - By using Gini index

- Gini index can be calculated using the below formula:

$$I_{Gini} = 1 - \sum_{i=1}^{j} p_i^2$$

$$I_{Gini} = 1 - (\text{the probability of target "No"})^2 - (\text{the probability of target "Yes"})^2$$

Outlook

Outlook is a nominal feature. It can be sunny, overcast or rain. I will summarize the final decisions for outlook feature.

Attributes	Rules	Yes	NO	Number of instances
Outlook	Overcast	4	0	4
	Sunny	2	3	5
	Rainy	3	2	5

Gini(Outlook=Sunny) = $1 - (2/5)^2 - (3/5)^2 = 1 - 0.16 - 0.36 =$ **0.48**

Gini(Outlook=Overcast) = $1 - (4/4)^2 - (0/4)^2 =$ **0**

Gini(Outlook=Rain) = $1 - (3/5)^2 - (2/5)^2 = 1 - 0.36 - 0.16 =$ **0.48**

Then, we will calculate the weighted sum of gini indexes for outlook feature.

Gini(Outlook) = (5/14) x 0.48 + (4/14) x 0 + (5/14) x 0.48 = 0.171 + 0 + 0.171

= **0.342**

By using Gini index

- Temperature

Attributes	Rules	Yes	NO	Number of instances
Temp	Hot	2	2	4
	Cool	3	1	4
	Mild	4	2	6

Gini(Temp=Hot) = 1 – (2/4)2 – (2/4)2 = **0.5**

Gini(Temp=Cool) = 1 – (3/4)2 – (1/4)2 = 1 – 0.5625 – 0.0625 = **0.375**

Gini(Temp=Mild) = 1 – (4/6)2 – (2/6)2 = 1 – 0.444 – 0.111 = **0.445**

We'll calculate weighted sum of gini index for temperature feature

Gini(Temp) = (4/14) x 0.5 + (4/14) x 0.375 + (6/14) x 0.445 = 0.142 + 0.107 + 0.190 = **0.439**

By using Gini index

Humidity

- Humidity is a binary class feature. It can be high or normal.

Attributes	Rules	Yes	NO	Number of instances
Humidity	High	3	4	7
	Normal	6	1	7

Gini(Humidity=High) = 1 – (3/7)2 – (4/7)2 = 1 – 0.183 – 0.326 = 0.489

Gini(Humidity=Normal) = 1 – (6/7)2 – (1/7)2 = 1 – 0.734 – 0.02 = 0.244

The weighted sum the for humidity feature will be calculated next

Gini(Humidity) = (7/14) x 0.489 + (7/14) x 0.244 = 0.367

By using Gini index

Wind

Wind is a binary class similar to humidity. It can be weak and strong..

Attributes	Rules	Yes	NO	Number of instances
Wind	Weak	6	2	8
	Strong	3	3	6

Gini(Wind=Weak) = 1 – (6/8)2 – (2/8)2 = 1 – 0.5625 – 0.062 = 0.375

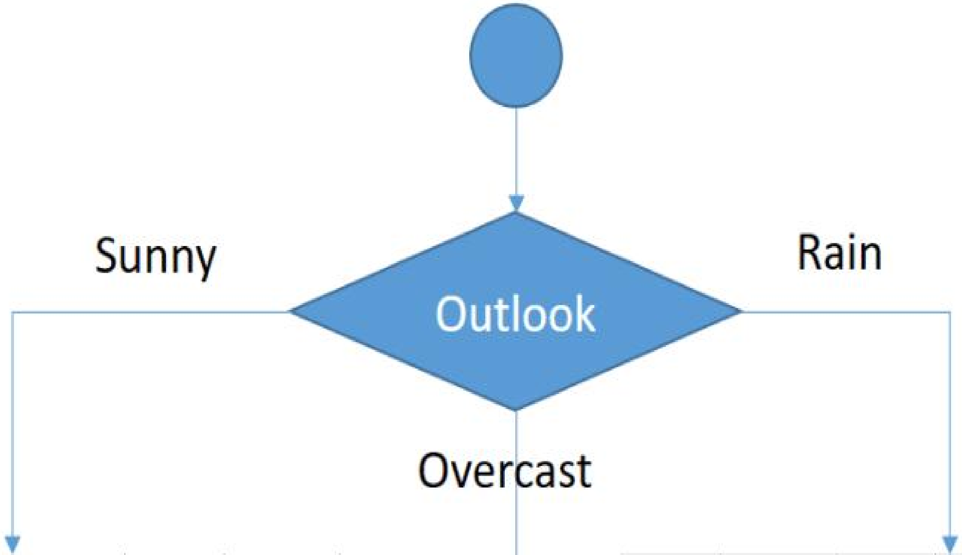
Gini(Wind=Strong) = 1 – (3/6)2 – (3/6)2 = 1 – 0.25 – 0.25 = 0.5

Gini(Wind) = (8/14) x 0.375 + (6/14) x 0.5 = 0.428

By using Gini index

- After calculated gini index values for each feature.

Feature	Gini index
Outlook	**0.342**
Temperature	0.439
Humidity	0.367
Wind	0.428

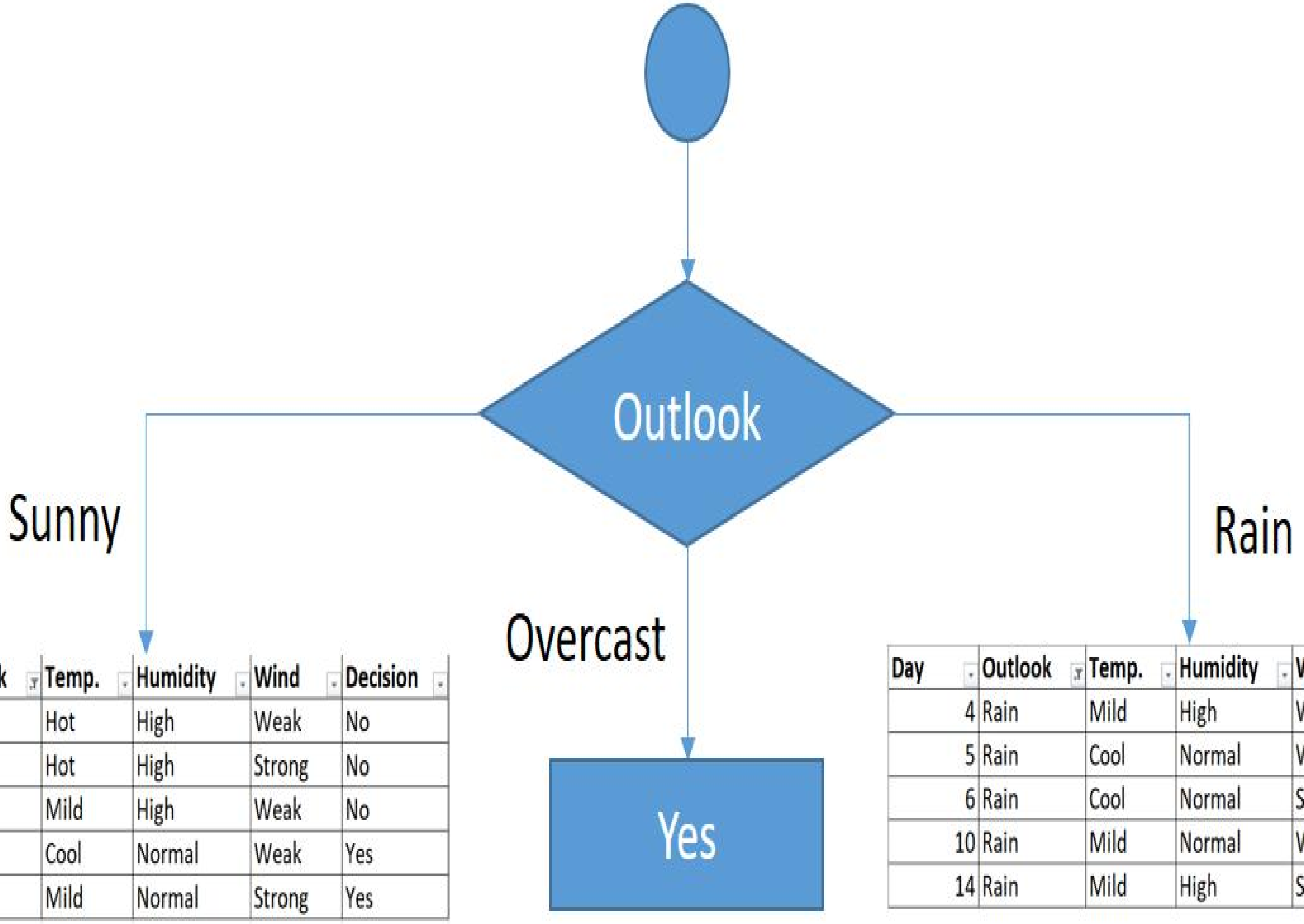


Day	Outlook	Temp.	Humidity	Wind	Decision
1	Sunny	Hot	High	Weak	No
2	Sunny	Hot	High	Strong	No
8	Sunny	Mild	High	Weak	No
9	Sunny	Cool	Normal	Weak	Yes
11	Sunny	Mild	Normal	Strong	Yes

Day	Outlook	Temp.	Humidity	Wind	Decision
4	Rain	Mild	High	Weak	Yes
5	Rain	Cool	Normal	Weak	Yes
6	Rain	Cool	Normal	Strong	No
10	Rain	Mild	Normal	Weak	Yes
14	Rain	Mild	High	Strong	No

Day	Outlook	Temp.	Humidity	Wind	Decision
3	Overcast	Hot	High	Weak	Yes
7	Overcast	Cool	Normal	Strong	Yes
12	Overcast	Mild	High	Strong	Yes
13	Overcast	Hot	Normal	Weak	Yes

- You might realize that the sub dataset in the overcast leaf has only yes decisions. This means that the overcast leaf is over.



Day	Outlook	Temp.	Humidity	Wind	Decision
1	Sunny	Hot	High	Weak	No
2	Sunny	Hot	High	Strong	No
8	Sunny	Mild	High	Weak	No
9	Sunny	Cool	Normal	Weak	Yes
11	Sunny	Mild	Normal	Strong	Yes

Day	Outlook	Temp.	Humidity	Wind	Decision
4	Rain	Mild	High	Weak	Yes
5	Rain	Cool	Normal	Weak	Yes
6	Rain	Cool	Normal	Strong	No
10	Rain	Mild	Normal	Weak	Yes
14	Rain	Mild	High	Strong	No

- Focus on the sub dataset for sunny outlook. We need to find the Gini index scores for temperature, humidity, and wind features respectively.

Day	Outlook	Temp.	Humidity	Wind	Decision
1	Sunny	Hot	High	Weak	No
2	Sunny	Hot	High	Strong	No
8	Sunny	Mild	High	Weak	No
9	Sunny	Cool	Normal	Weak	Yes
11	Sunny	Mild	Normal	Strong	Yes

Gini of temperature for a sunny outlook:

Temperature	Yes	No	Number of instances
Hot	0	2	2
Cool	1	0	1
Mild	1	1	2

Gini(Outlook=Sunny and Temp.=Hot) = 1 – (0/2)2 – (2/2)2 = **0**

Gini(Outlook=Sunny and Temp.=Cool) = 1 – (1/1)2 – (0/1)2 = **0**

Gini(Outlook=Sunny and Temp.=Mild) = 1 – (1/2)2 – (1/2)2 = 1 – 0.25 – 0.25 = **0.5**

Gini(Outlook=Sunny and Temp.) = (2/5)x0 + (1/5)x0 + (2/5)x0.5 = **0.2**

Gini of humidity for sunny outlook

Humidity	Yes	No	Number of instances
High	0	3	3
Normal	2	0	2

Gini(Outlook=Sunny and Humidity=High) = 1 – (0/3)2 – (3/3)2 = **0**

Gini(**Outlook=Sunny and Humidity=Normal**) = 1 – (2/2)2 – (0/2)2 = **0**

Gini(**Outlook=Sunny and Humidity**) = (3/5)x0 + (2/5)x0 = **0**

Gini of wind for sunny outlook

Wind	Yes	No	Number of instances
Weak	1	2	3
Strong	1	1	2

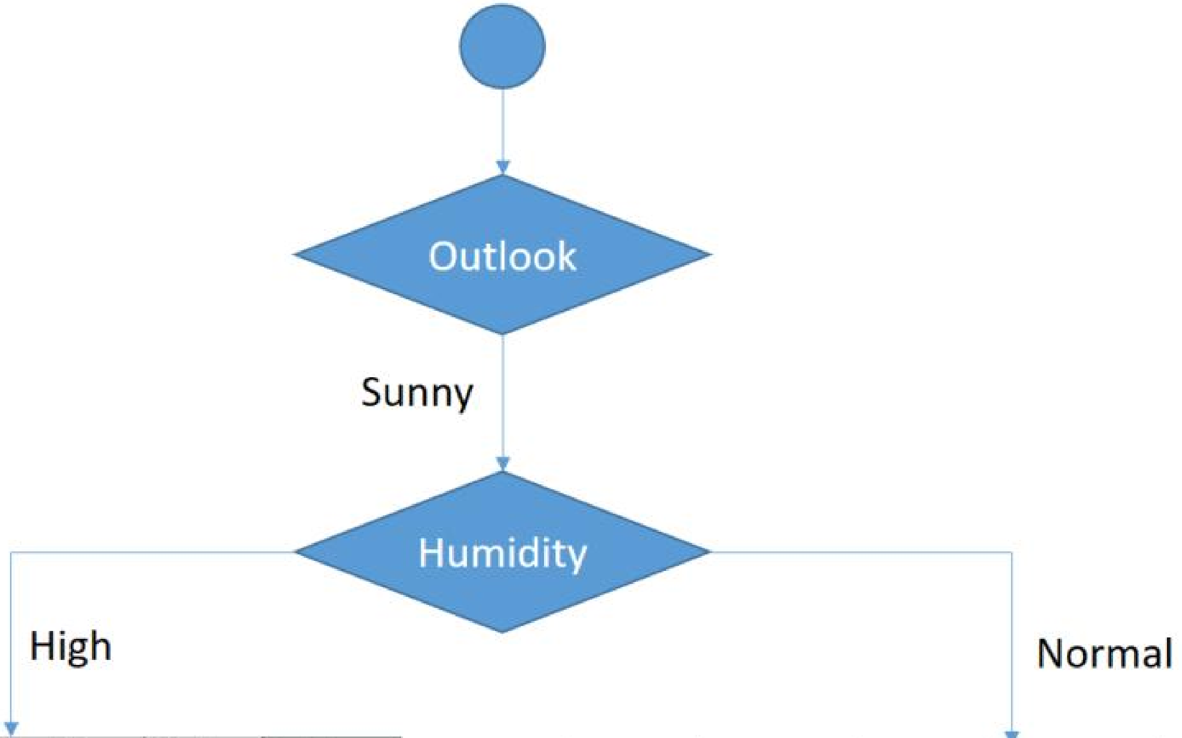
Gini(Outlook=Sunny and Wind=Weak) = 1 – (1/3)2 – (2/3)2 = **0.266**

Gini(Outlook=Sunny and Wind=Strong) = 1- (1/2)2 – (1/2)2 = **0.2**

Gini(Outlook=Sunny and Wind) = (3/5)x0.266 + (2/5)x0.2 = **0.466**

Decision for a sunny outlook

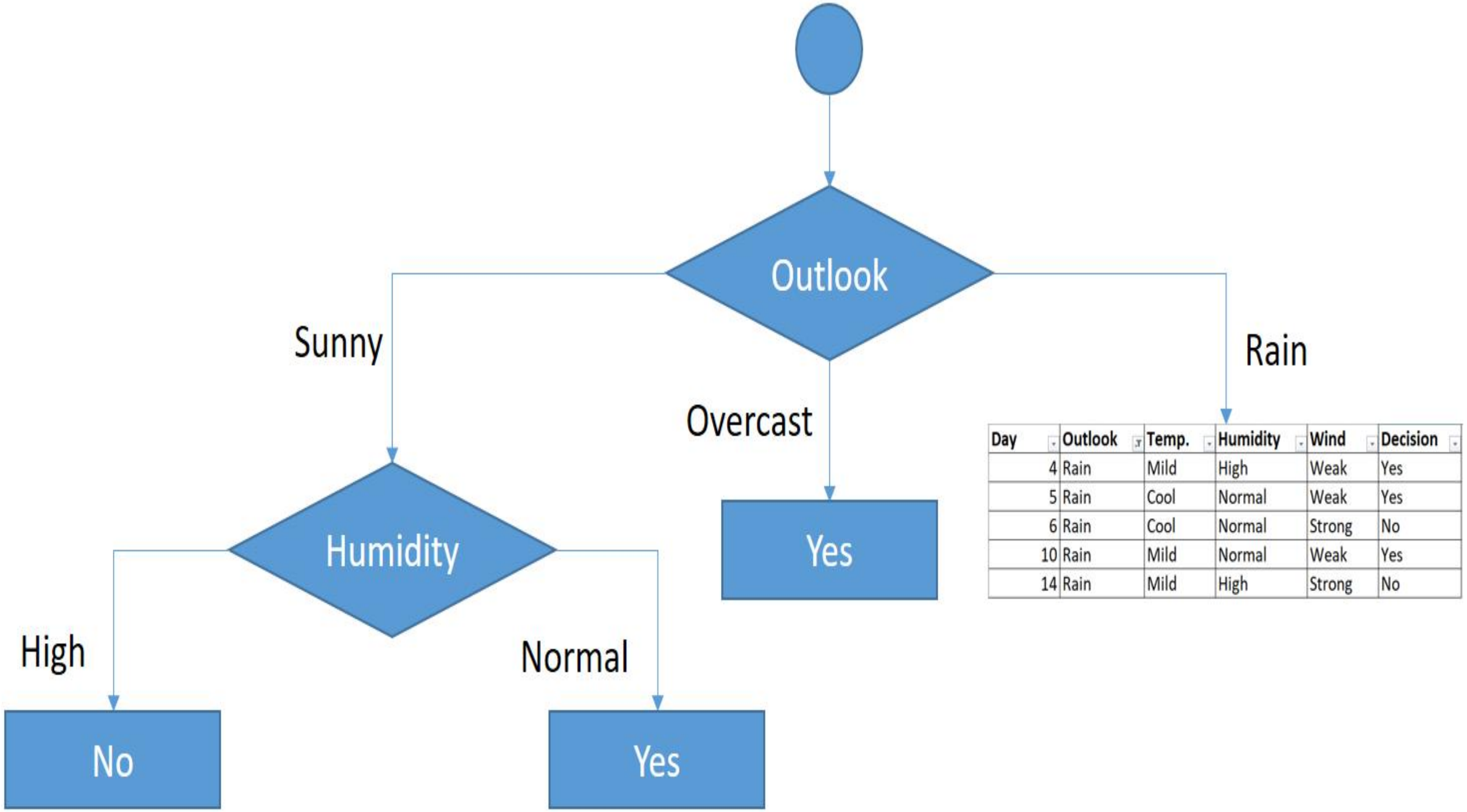
Feature	Gini index
Temperature	0.2
Humidity	**0**
Wind	0.466



Day	Outlook	Temp.	Humidity	Wind	Decision
1	Sunny	Hot	High	Weak	No
2	Sunny	Hot	High	Strong	No
8	Sunny	Mild	High	Weak	No

Day	Outlook	Temp.	Humidity	Wind	Decision
9	Sunny	Cool	Normal	Weak	Yes
11	Sunny	Mild	Normal	Strong	Yes

- As seen, the decision is always no for high humidity and sunny outlook. On the other hand, the decision will always be yes for normal humidity and a sunny outlook. This branch is over.



Day	Outlook	Temp.	Humidity	Wind	Decision
4	Rain	Mild	High	Weak	Yes
5	Rain	Cool	Normal	Weak	Yes
6	Rain	Cool	Normal	Strong	No
10	Rain	Mild	Normal	Weak	Yes
14	Rain	Mild	High	Strong	No

Rain outlook

Day	Outlook	Temp.	Humidity	Wind	Decision
4	Rain	Mild	High	Weak	Yes
5	Rain	Cool	Normal	Weak	Yes
6	Rain	Cool	Normal	Strong	No
10	Rain	Mild	Normal	Weak	Yes
14	Rain	Mild	High	Strong	No

Gini of temperature for rain outlook

Temperature	Yes	No	Number of instances
Cool	1	1	**2**
Mild	2	1	**3**

Gini(Outlook=Rain and Temp.=Cool) = 1 – (1/2)2 – (1/2)2 = **0.5**

Gini(Outlook=Rain and Temp.=Mild) = 1 – (2/3)2 – (1/3)2 = **0.444**

Gini(Outlook=Rain and Temp.) = (2/5)x0.5 + (3/5)x0.444 = **0.466**

Gini of humidity for rain outlook

Humidity	Yes	No	Number of instances
High	1	1	2
Normal	2	1	3

Gini(Outlook=Rain and Humidity=High) = 1 – (1/2)2 – (1/2)2 = **0.5**

Gini(Outlook=Rain and Humidity=Normal) = 1 – (2/3)2 – (1/3)2 = **0.444**

Gini(Outlook=Rain and Humidity) = (2/5)x0.5 + (3/5)x0.444 = **0.466**

Gini of wind for rain outlook

Wind	Yes	No	Number of instances
Weak	3	0	3
Strong	0	2	2

Gini(Outlook=Rain and Wind=Weak) = 1 – (3/3)2 – (0/3)2 = 0

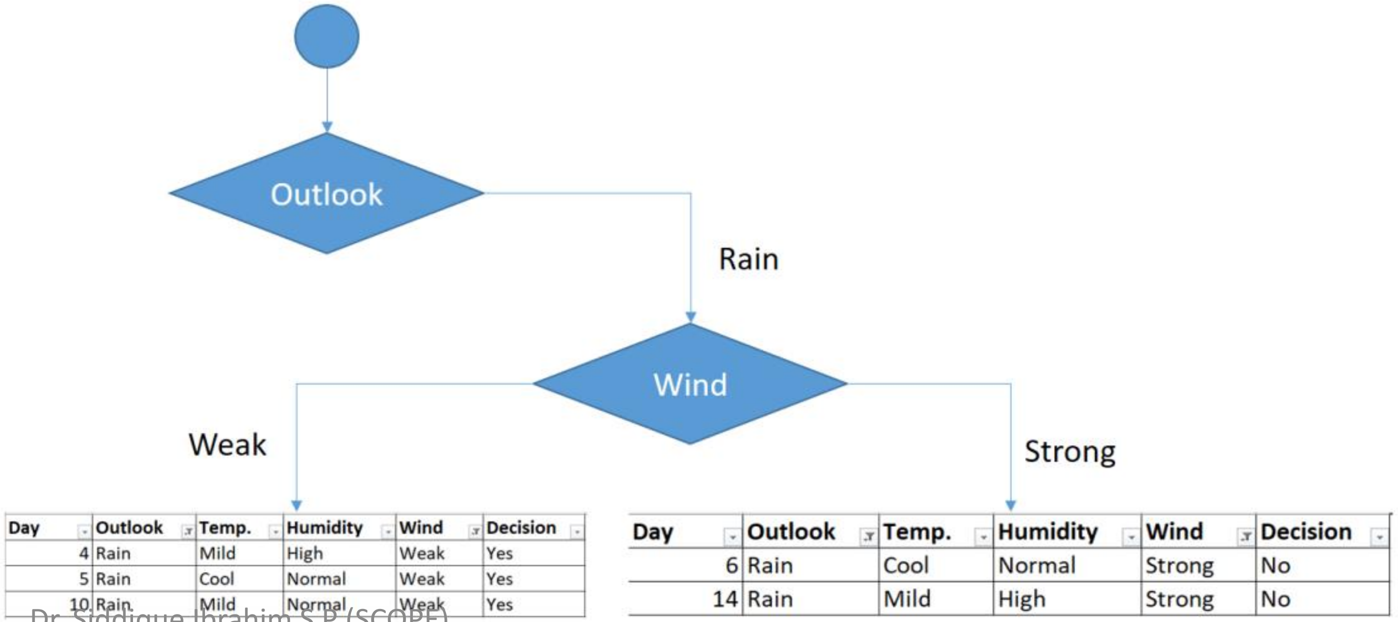
Gini(Outlook=Rain and Wind=Strong) = 1 – (0/2)2 – (2/2)2 = 0

Gini(Outlook=Rain and Wind) = (3/5)x0 + (2/5)x0 = 0

The decision for rain outlook

Feature	Gini index
Temperature	0.466
Humidity	0.466
Wind	**0**

- Put the wind feature for the rain outlook branch and monitor the new sub-data sets.



- The decision is always yes when the wind is weak. On the other hand, the decision is always no if the wind is strong. This means that this branch is over.