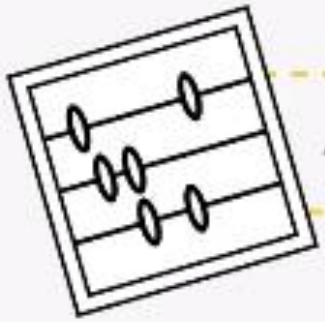
CLASSIFICATION

Classification: Classification is Machine Learning task of predicting the value of a categorical variable(target or class).

This is done by building a classifier.

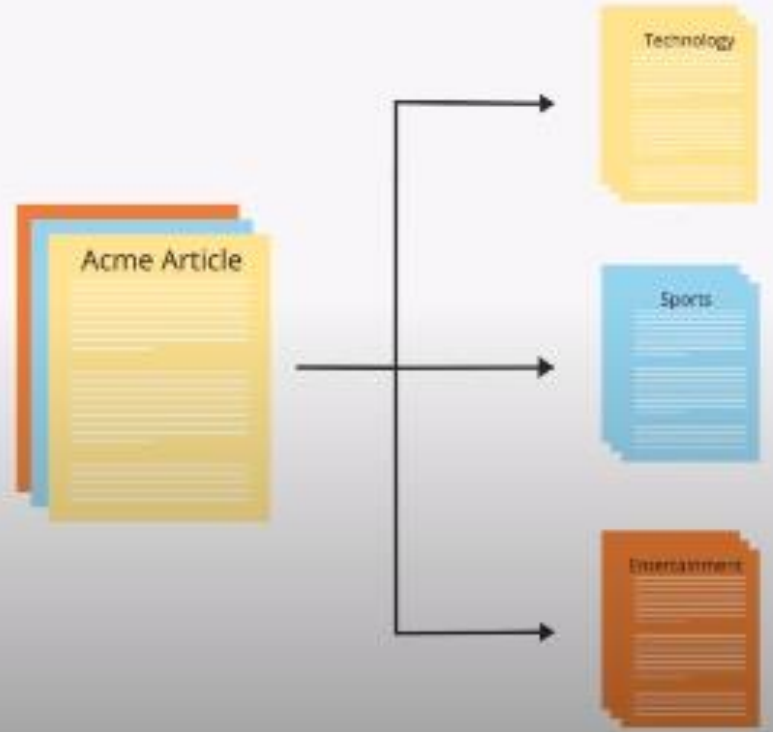
Classifier: A classifier has a set of variables need to be trained. Different classifiers have different algorithms to optimize the process.

Classification



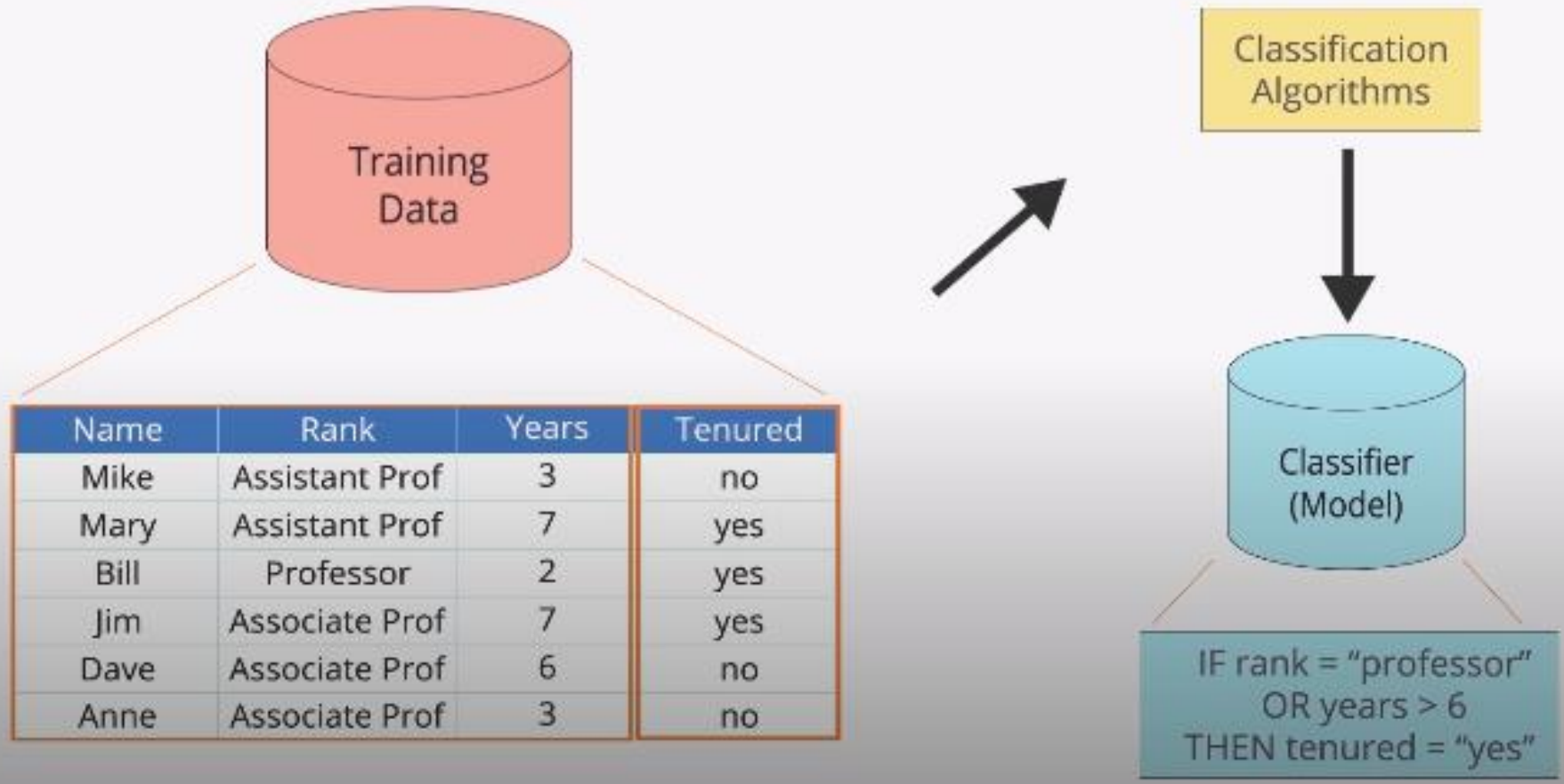
A machine learning task that identifies the class to which an instance belongs.

Classification involves training a model to predict qualitative target.



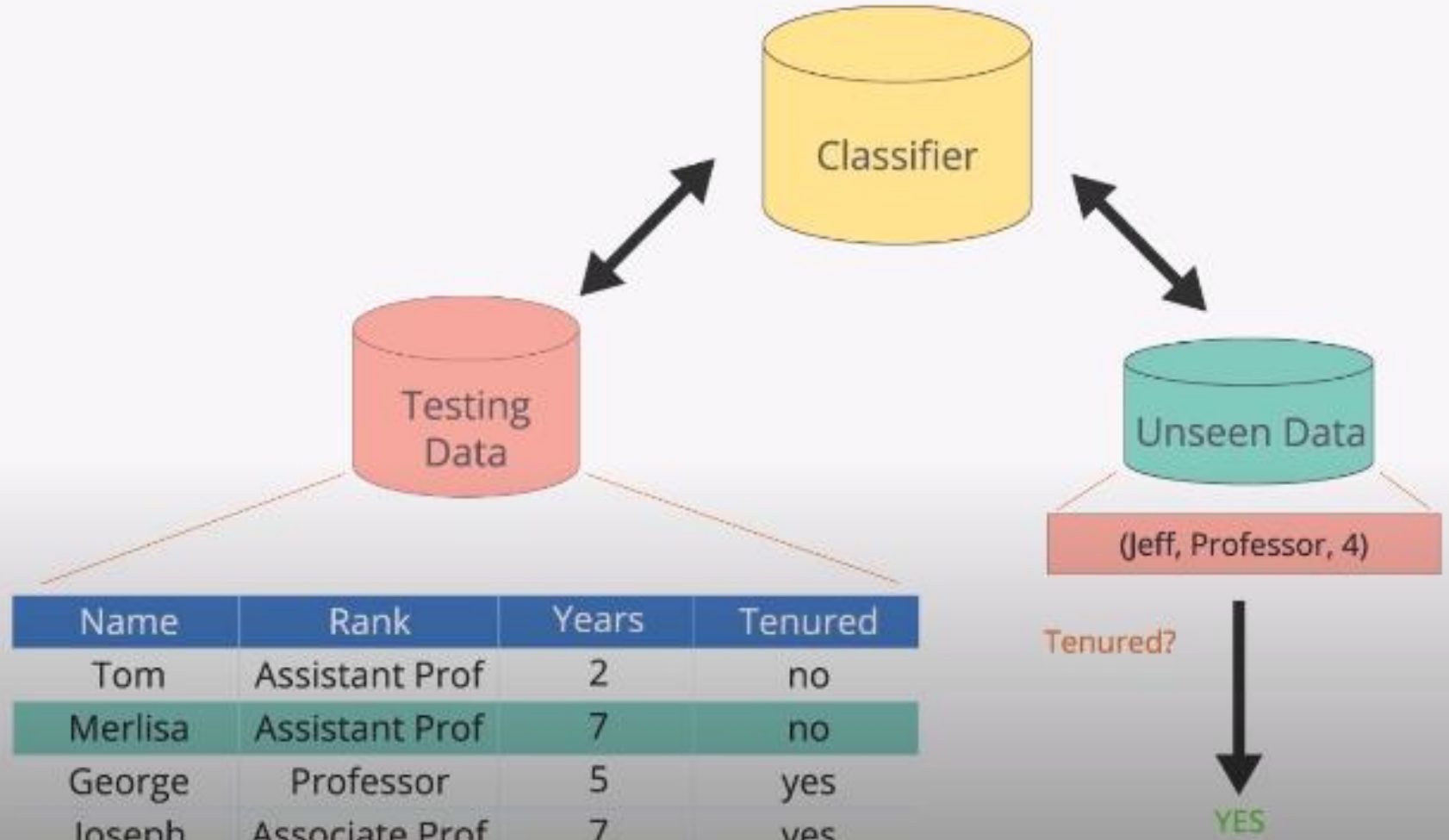
Classification: Example

Training a classifier model with respect to the available data



Name	Rank	Years	Tenured
Mike	Assistant Prof	3	no
Mary	Assistant Prof	7	yes
Bill	Professor	2	yes
Jim	Associate Prof	7	yes
Dave	Associate Prof	6	no
Anne	Associate Prof	3	no

Classification: Example



Name	Rank	Years	Tenured
Tom	Assistant Prof	2	no
Merlisa	Assistant Prof	7	no
George	Professor	5	yes
Joseph	Associate Prof	7	yes

Fraud Detection

Detects data streams of transactions and learns the fraudulent patterns

Fraud Detection

Fraud detection is a classification algorithm.

- Income
- Purchase information
- Occupation

Classification systems that include optimization techniques show a greater fluctuation in the implementation of many different technnologies.

Fraud Detection

A biometric technique used mainly for surveillance purposes

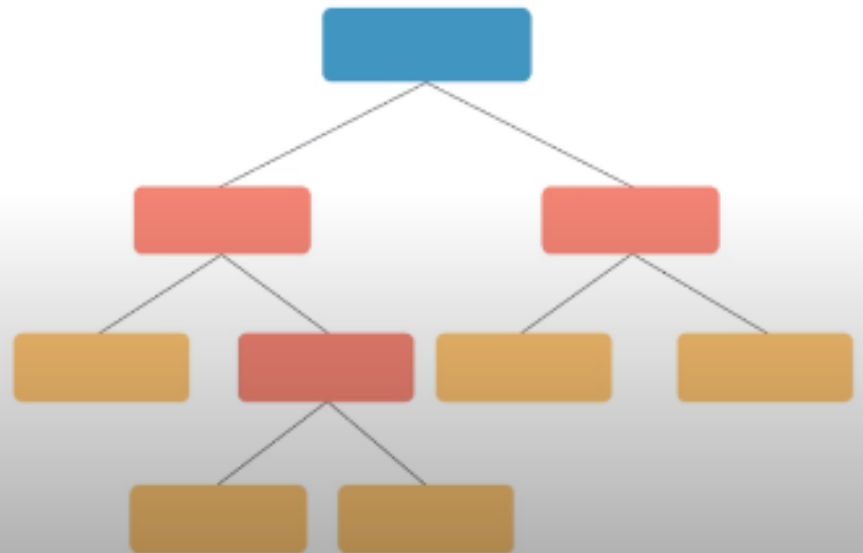
Refers to identifying an unknown face image using computational algorithms

Successful face recognition methodology depends heavily on the particular choice of the features

DECISION TREE

Decision Tree Classifier

Decision tree: A flow chart shaped like a tree, where every internal node denotes a check on an attribute.

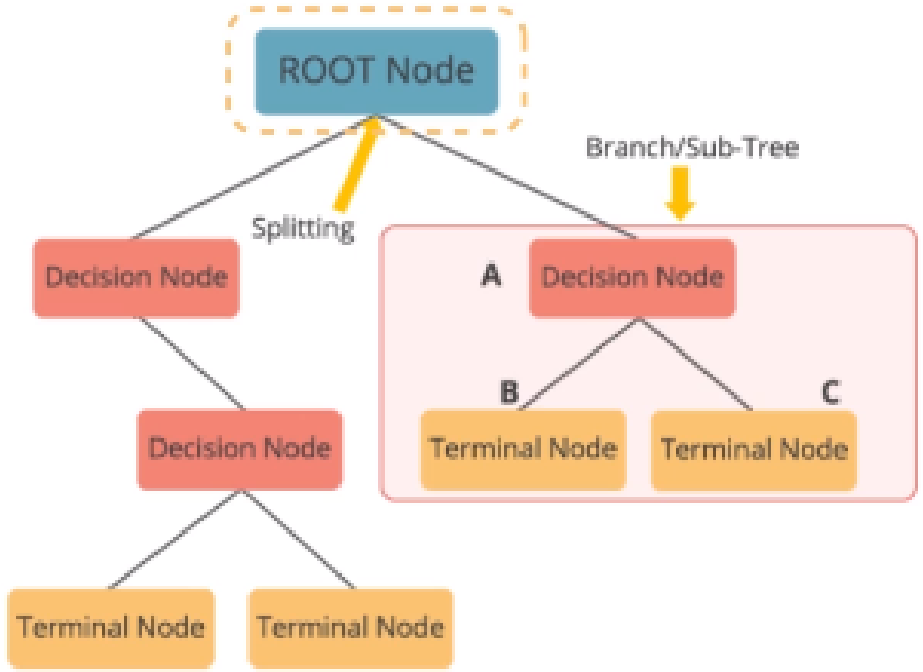


Each branch represents the outcome of the test, and each leaf node represents the class label

A path from root to leaf represents classification rules

Decision Tree Classifier

- Root Node: The entire population or sample that further gets divided
- Splitting: Division of nodes into two or more sub-nodes
- Decision Node: A sub-node splits into further sub-nodes
- Leaf/Terminal Node: Node that does not split
- Pruning: Process of removing sub-nodes
- Branch/Sub-Tree: A subsection of the entire tree
- Parent Node: A node which is divided into sub-nodes and the sub-nodes are the child node



ID3(Iterative Dichotomiser)

- The algorithm iteratively divides attributes into two groups which are the most dominant attribute and others to construct a tree on the basis of Information Gain(IG).
- Then, it calculates the entropy and information gains of each attribute.
- In this way, the most dominant attribute can be founded.
- After then, the most dominant one is put on the tree as decision node.
- Thereafter, entropy and gain scores would be calculated again among the other attributes.
- Thus, the next most dominant attribute is found.
- Finally, this procedure continues until reaching a decision for that branch. That’s why, it is called Iterative Dichotomiser.

ID3(Iterative Dichotomiser)

- ID3 starts with a dataset as the root node.
- It iterates every unused attribute of the set and calculates the entropy or information gain of that attribute.
- It then selects the attribute which has the smallest entropy, or largest information gain.
- The set is then split in regards of the selected attribute, which creates a subset of the data. This continues recursively on each subset, until one of the following cases occurs:
  - ❖ Every element in the subset belongs to the same class.
  - ❖ The are no more attributes to be selected, but the examples do not belong to the same class.
  - ❖ There are no examples in the subset.

Formulas for ID3

$$\textbf{Information Gain} = \frac{-p}{p+n}\log_2\frac{p}{p+n} - \frac{n}{p+n}\log_2\frac{n}{p+n}$$

$$E(A) = \sum \frac{p_i + n_i}{p+n}\left(I(p, n)\right)$$

Gain = IG – E(A)

Example

Name	Hair	Height	Weight	Location	Class
Sam	Blonde	Average	Light	No	Yes
Jacob	Blonde	Tall	Average	Yes	No
Alfred	Brown	Short	Average	Yes	No
Jennifer	Blonde	Short	Average	No	Yes
Ross	Red	Average	Heavy	No	Yes
Smith	Brown	Tall	Heavy	No	No
Roselin	Brown	Average	Heavy	No	No
Cavin	Blonde	Short	Light	Yes	No

Sol^n:- Overall gain

Yes = 3 No = 5

$$I.G = -\frac{3}{8}\log_2\left(\frac{3}{8}\right) - \frac{5}{8}\log_2\left(\frac{5}{8}\right)$$

$$= 0.9544$$

Name	Hair	Height	Weight	Location	Class
Sam	Blonde	Average	Light	No	Yes
Jacob	Blonde	Tall	Average	Yes	No
Alfred	Brown	Short	Average	Yes	No
Jennifer	Blonde	Short	Average	No	Yes
Ross	Red	Average	Heavy	No	Yes
Smith	Brown	Tall	Heavy	No	No
Roselin	Brown	Average	Heavy	No	No
Cavin	Blonde	Short	Light	Yes	No

$$E(A) = \sum \frac{p_i + n_i}{p+n} (E(p,n))$$

Step 2: Calculate Information Gain of each attribute

i) Hair

	Yes	No	E.G
Blonde	2	2	1
Brown	0	3	0
Red	1	0	0

$$E.G(\text{Blonde}) = -\frac{2}{4}\log_2\left(\frac{2}{4}\right) - \frac{2}{4}\log_2\left(\frac{2}{4}\right)$$

$$= 0.5 + 0.5 = 1$$

$$E.G(\text{Brown}) = -\frac{0}{3}\log_2\left(\frac{0}{3}\right) - \frac{3}{3}\log_2\left(\frac{3}{3}\right)$$

$$= 0$$

$$E.G(\text{Red}) = -\frac{1}{1}\log_2\left(\frac{1}{1}\right) - \frac{0}{1}\log_2\left(\frac{0}{1}\right)$$

$$= 0$$

$$\text{Gain} = E.G - \text{Entropy(Hair)} = 0.9544 - \left(\frac{4}{8}\times 1 + \frac{3}{8}\times 0 + \frac{1}{8}\times 0\right)$$

$$= 0.454$$

Name	Hair	Height	Weight	Location	Class
Sam	Blonde	Average	Light	No	Yes
Jacob	Blonde	Tall	Average	Yes	No
Alfred	Brown	Short	Average	Yes	No
Jennifer	Blonde	Short	Average	No	Yes
Ross	Red	Average	Heavy	No	Yes
Smith	Brown	Tall	Heavy	No	No
Roselin	Brown	Average	Heavy	No	No
Cavin	Blonde	Short	Light	Yes	No

ii) Height

	Yes	No	E.G
Tall	0	2	0
Average	2	1	0.91
Short	1	2	0.91

$$E(\text{Tall}) = -\frac{0}{2}\log_2\left(\frac{0}{2}\right) - \frac{2}{2}\log_2\left(\frac{2}{2}\right)$$

$$= 0$$

$$E(\text{Average}) = -\frac{2}{3}\log_2\left(\frac{2}{3}\right) - \frac{1}{3}\log_2\left(\frac{1}{3}\right)$$

$$= 0.91$$

$$E(\text{Short}) = -\frac{1}{3}\log_2\left(\frac{1}{3}\right) - \frac{2}{3}\log_2\left(\frac{2}{3}\right)$$

$$= 0.91$$

$$E(A) = \sum \frac{p_i + n_i}{p+n} (I(p,n))$$

$$E(\text{Height}) = \sum \frac{p_i + n_i}{p+n} (I(p,n))$$

$$= \frac{2}{8} \times I(0,2) + \frac{3}{8} \times I(2,1) + \frac{3}{8} I(1,2)$$

$$= 0 + \frac{3}{8} \times 0.91 + \frac{3}{8} \times 0.91 = 0.6825$$

$$\text{Gain} = I(a - \text{Entropy}(\text{Height}))$$

$$= 0.9544 - 0.6825$$

$$= 0.2719$$

Name	Hair	Height	Weight	Location	Class
Sam	Blonde	Average	Light	No	Yes
Jacob	Blonde	Tall	Average	Yes	No
Alfred	Brown	Short	Average	Yes	No
Jennifer	Blonde	Short	Average	No	Yes
Ross	Red	Average	Heavy	No	Yes
Smith	Brown	Tall	Heavy	No	No
Roselin	Brown	Average	Heavy	No	No
Cavin	Blonde	Short	Light	Yes	No

$$E(A) = \sum \frac{p_i + n_i}{p+n} (E(p,n))$$

iii) Weight

	Yes	No	E(A)
Light	1	1	1
Average	1	2	0.91
Heavy	1	2	0.91

$$E(A(\text{Light})) = -\frac{1}{2}\log_2\left(\frac{1}{2}\right) - \frac{1}{2}\log_2\left(\frac{1}{2}\right)$$

$$= 1$$

$$E(A(\text{Average})) = -\frac{1}{3}\log_2\left(\frac{1}{3}\right) - \frac{2}{3}\log_2\left(\frac{1}{3}\right)$$

$$= 0.91$$

$$E(A(\text{Heavy})) = -\frac{1}{3}\log_2\left(\frac{1}{3}\right) - \frac{2}{3}\log_2\left(\frac{2}{3}\right)$$

$$= 0.91$$

$$\text{Gain} = E(A) - \text{Entropy(weight)}$$

$$= 0.9544 - \left(\frac{2}{8} * 1 + \frac{3}{8} * 0.91 + \frac{3}{8} * 0.91\right)$$

$$= 0.0219$$

Name	Hair	Height	Weight	Location	Class
Sam	Blonde	Average	Light	No	Yes
Jacob	Blonde	Tall	Average	Yes	No
Alfred	Brown	Short	Average	Yes	No
Jennifer	Blonde	Short	Average	No	Yes
Ross	Red	Average	Heavy	No	Yes
Smith	Brown	Tall	Heavy	No	No
Roselin	Brown	Average	Heavy	No	No
Cavin	Blonde	Short	Light	Yes	No

$$E(A) = \sum \frac{p_i + n_i}{p+n} (E(p,n))$$

iv) Location :-

	Yes	No	E G
Yes	0	3	0
No	3	2	0.97

$$E G(\text{Yes}) = -\frac{0}{3}\log_2\left(\frac{0}{3}\right) - \frac{3}{3}\log_2\left(\frac{3}{3}\right)$$

$$= 0$$

$$E G(\text{No}) = -\frac{3}{5}\log_2\left(\frac{3}{5}\right) - \frac{2}{5}\log_2\left(\frac{2}{5}\right)$$

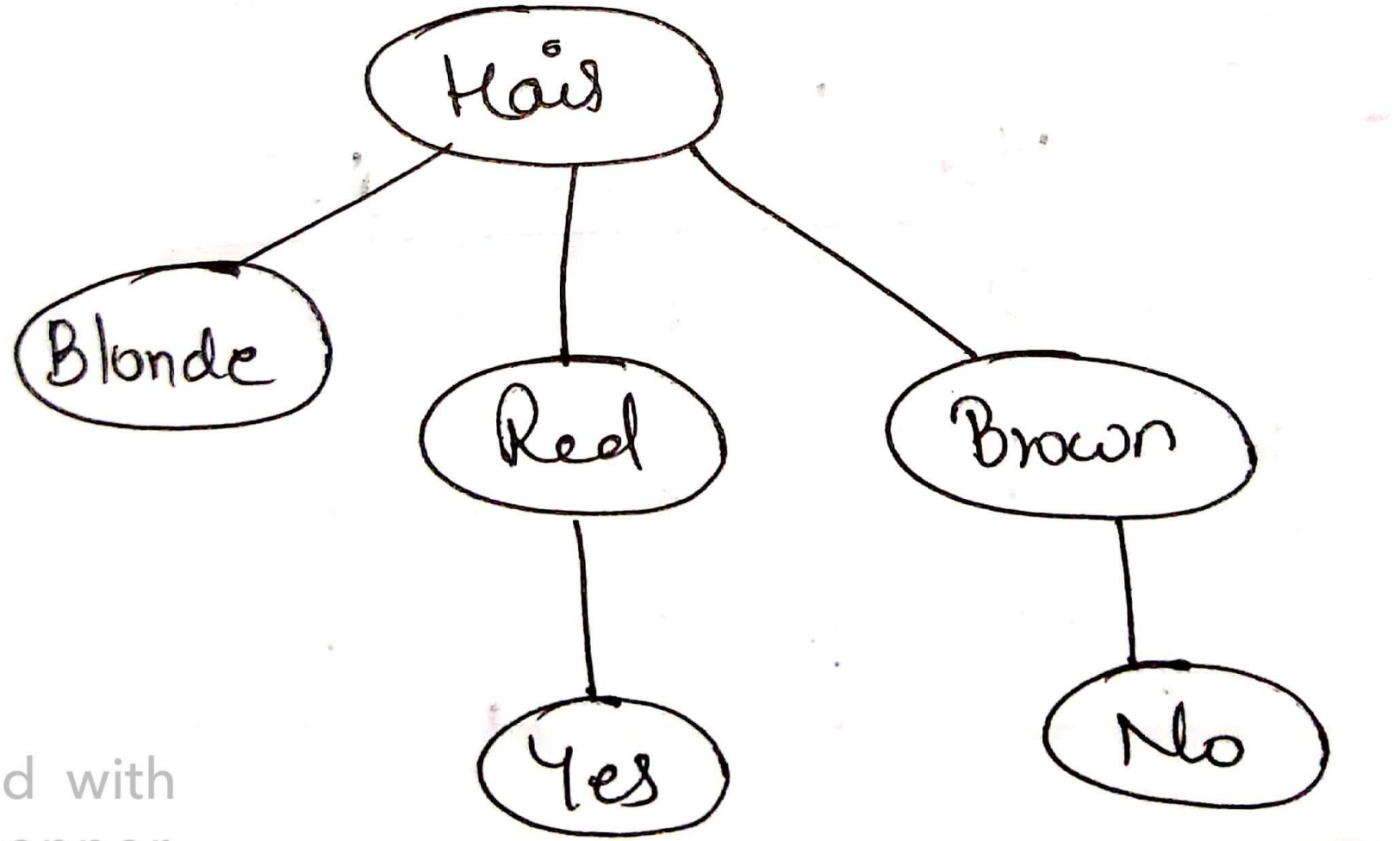
$$= 0.97$$

$$\text{Gain} = E.G - \text{Entropy(Location)}$$

$$= \left(0.9544 - \left(\frac{3}{8} * 0 + \frac{5}{8} * 0.97\right)\right)$$

$$= 0.3481$$

Attribute	Gain
Hair	0.454
Height	0.2719
Weight	0.0219
Location	0.3481



Name	Hair	Height	Weight	Location	Class
Sam	Blonde	Average	Light	No	Yes
Jacob	Blonde	Tall	Average	Yes	No
Alfred	Brown	Short	Average	Yes	No
Jennifer	Blonde	Short	Average	No	Yes
Ross	Red	Average	Heavy	No	Yes
Smith	Brown	Tall	Heavy	No	No
Roselin	Brown	Average	Heavy	No	No
Cavin	Blonde	Short	Light	Yes	No

Hair	Height	Weight	Location	Class
Blonde	Average	Light	No	Yes
Blonde	Tall	Average	Yes	No
Blonde	Short	Average	Yes	Yes
Blonde	Short	Light	Yes	No

Step 3: Now, we need to calculate info gain for remaining attributes

Hair	Height	Weight	Location	Class
Blonde	Average	Light	No	Yes
Blonde	Tall	Average	Yes	No
Blonde	Short	Average	Yes	Yes
Blonde	Short	Light	Yes	No

Yes = 2 No = 2

$$IG = -\frac{2}{4}\log_2\left(\frac{2}{4}\right) - \frac{2}{4}\log_2\left(\frac{2}{4}\right)$$

$$= 1$$

Hair	Height	Weight	Location	Class
Blonde	Average	Light	No	Yes
Blonde	Tall	Average	Yes	No
Blonde	Short	Average	Yes	Yes
Blonde	Short	Light	Yes	No

$$E(A) = \sum \frac{p_i + n_i}{p+n} (I(p,n))$$

ii) Height

	Yes	No	I.G
Tall	0	1	0
Average	1	0	0
Short	1	1	1

Gain = I.G − Entropy (Height)

$= 1 - \left[\frac{1}{4} * 0 + \frac{1}{4} * 0 + \frac{2}{4} * 1\right]$

$= 0.5$

Hair	Height	Weight	Location	Class
Blonde	Average	Light	No	Yes
Blonde	Tall	Average	Yes	No
Blonde	Short	Average	Yes	Yes
Blonde	Short	Light	Yes	No

$$E(A) = \sum \frac{p_i + n_i}{p+n} (I(p,n))$$

ii) Weight

	Yes	No	I(p,n)
Light	1	1	1
Average	1	1	1
Heavy	0	0	0

$$\text{Gain} = 1 - \left[\frac{2}{4} * 1 + \frac{2}{4} * 1 + \frac{0}{4} * 0\right]$$

$$= 0$$

Hair	Height	Weight	Location	Class
Blonde	Average	Light	No	Yes
Blonde	Tall	Average	Yes	No
Blonde	Short	Average	Yes	Yes
Blonde	Short	Light	Yes	No

iii) Location:-

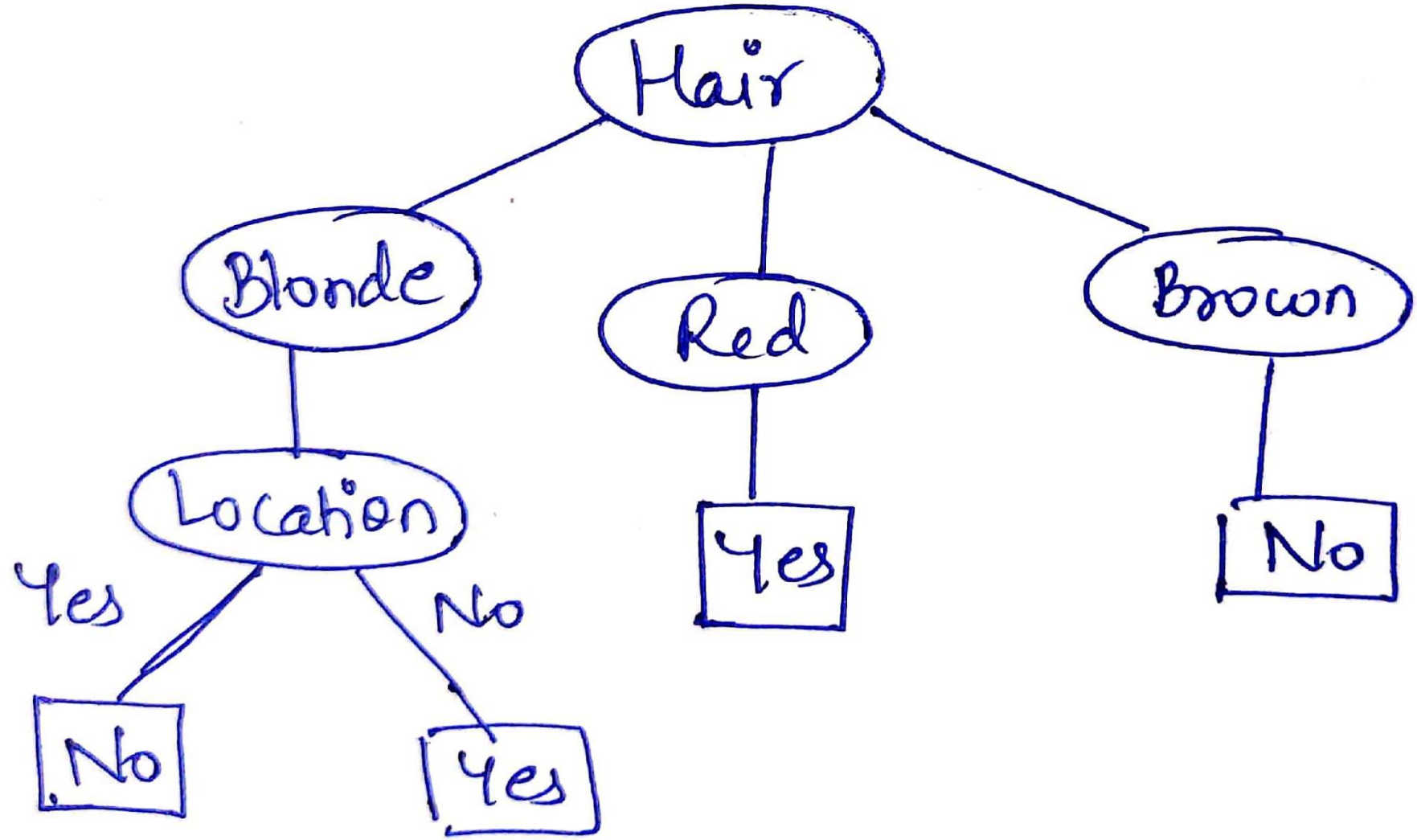
$$E(A) = \sum \frac{p_i + n_i}{p+n} (E(p,n))$$

	Yes	No	E(A)
Yes	0	2	0
No	2	0	0

$$\text{Gain (location)} = 1 - \left[\frac{2}{4} \times 0 + \frac{2}{4} \times 0\right]$$

$$= 1$$

Maximum Gain is wrt location


Hair
Blonde
Red
Brown
Location
Yes
No
Yes
No
No
Yes

Classification and Regression Trees (CART)

- CART can handle both classification and regression tasks.
- This algorithm uses a metric named gini index to create decision points for classification tasks. It stores sum of squared probabilities of each class. We can formulate it as illustrated below.

$$\text{Gini} = 1 - \Sigma\,(Pi)^2 \text{ for } i=1 \text{ to number of classes}$$

- Tree models where the target variable can take a discrete set of values are called **classification trees.**
- Decision trees where the target variable can take continuous values are called **regression trees**.

Classification and Regression Trees (CART)

The main elements of CART (and any decision tree algorithm) are:

1.Rules for splitting data at a node based on the value of one variable;

2.Stopping rules for deciding when a branch is terminal and can be split no more; and

3.Finally, a prediction for the target variable in each terminal node.

Example: Find the decision tree using CART.

Day	Outlook	Temperature	Humidity	Wind	PlayTennis
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

➢ **Outlook**

Outlook is a nominal feature. It can be sunny, overcast or rain.

Outlook	Yes	No	Number of instances
Sunny	2	3	5
Overcast	4	0	4
Rain	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 weighted sum of gini indexes for outlook feature.

Gini(Outlook) = $(5/14) \times 0.48 + (4/14) \times 0 + (5/14) \times 0.48 = 0.171 + 0 + 0.171 = 0.342$

➢ **Temperature**

Similarly, temperature is a nominal feature and it could have 3 different values: Cool, Hot and Mild. Let's summarize decisions for temperature feature.

Temperature	Yes	No	Number of instances
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) = $(4/14) \times 0.5 + (4/14) \times 0.375 + (6/14) \times 0.445 = 0.142 + 0.107 + 0.190 = 0.439$

- **Humidity**

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

Humidity	Yes	No	Number of instances
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) = $1 - (6/7)^2 - (1/7)^2 = 1 - 0.734 - 0.02 = 0.244$

Weighted sum for humidity feature will be calculated next

Gini(Humidity) = $(7/14) \times 0.489 + (7/14) \times 0.244 = 0.367$

- **Wind**

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

Wind	Yes	No	Number of instances
Weak	**6**	**2**	**8**
Strong	**3**	**3**	**6**

Gini(Wind=Weak) = $1 - (6/8)^2 - (2/8)^2 = 1 - 0.5625 - 0.062 = 0.375$

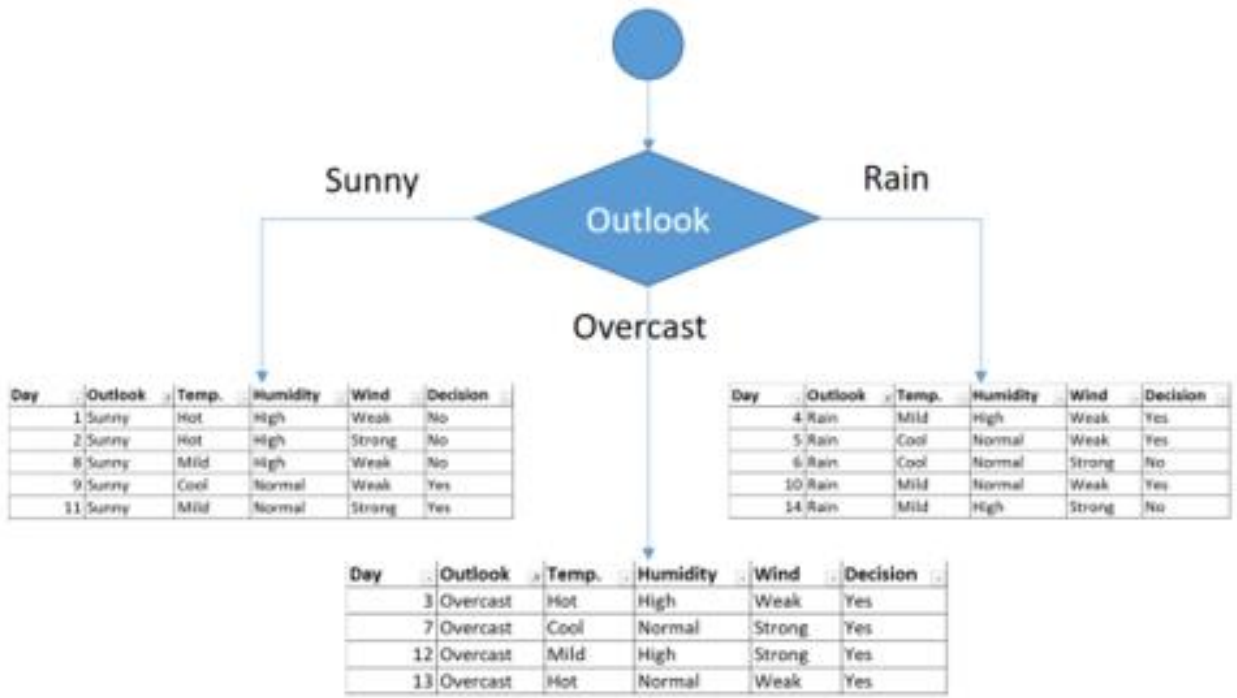
Gini(Wind=Strong) = $1 - (3/6)^2 - (3/6)^2 = 1 - 0.25 - 0.25 = 0.5$

Gini(Wind) = $(8/14) \times 0.375 + (6/14) \times 0.5 = 0.428$

We've calculated gini index values for each feature. The winner will be outlook feature because its cost is the lowest.

Feature	Gini index
Outlook	0.342
Temperature	0.439
Humidity	0.367
Wind	0.428

We'll put outlook decision at the top of the tree.



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

We will apply same principles to those sub datasets in the following steps.

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 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) = $1 - (1/3)^2 - (2/3)^2 = 0.266$

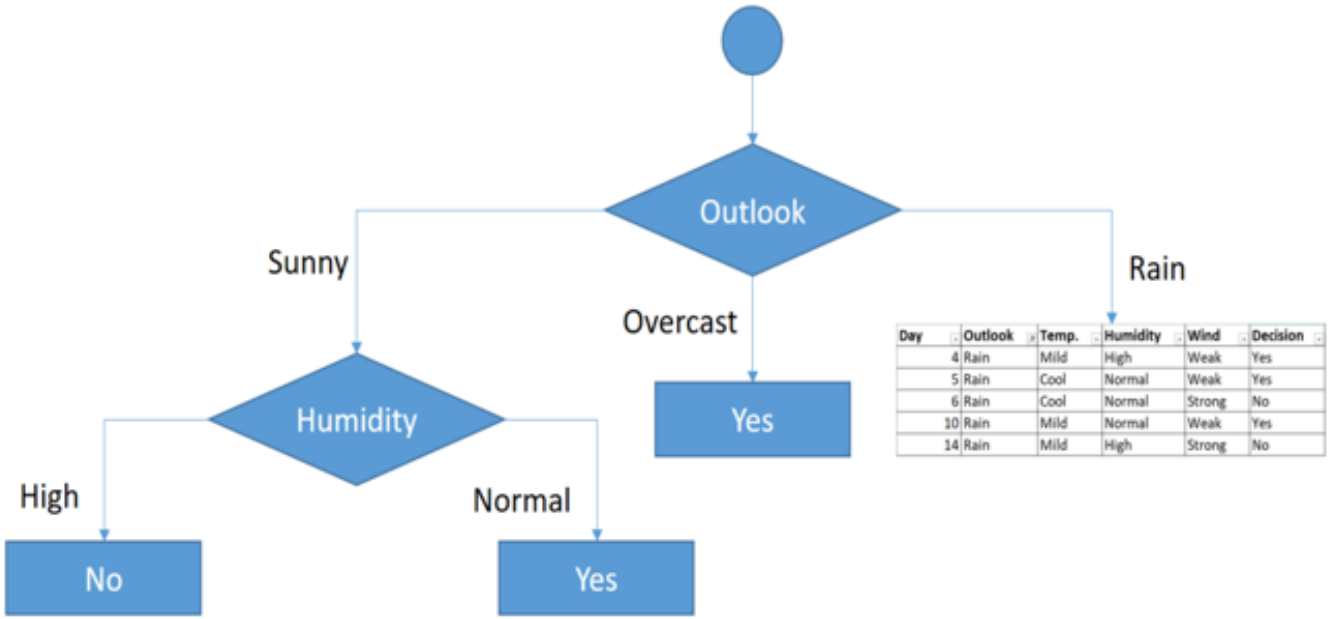
Gini(Outlook=Sunny and Wind=Strong) = $1 - (1/2)^2 - (1/2)^2 = 0.2$

Gini(Outlook=Sunny and Wind) = $(3/5)\text{x}0.266 + (2/5)\text{x}0.2 = 0.466$

➢ Decision for sunny outlook

We've calculated gini index scores for feature when outlook is sunny. The winner is humidity because it has the lowest value.

Feature	Gini index
Temperature	0.2
Humidity	0
Wind	0.466



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

Now, we need to focus on rain outlook.

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**

$$Gini(Outlook=Rain\ and\ Temp.=Cool) = 1 - (1/2)^2 - (1/2)^2 = 0.5$$

$$Gini(Outlook=Rain\ and\ Temp.=Mild) = 1 - (2/3)^2 - (1/3)^2 = 0.444$$

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

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 wind for rain outlook

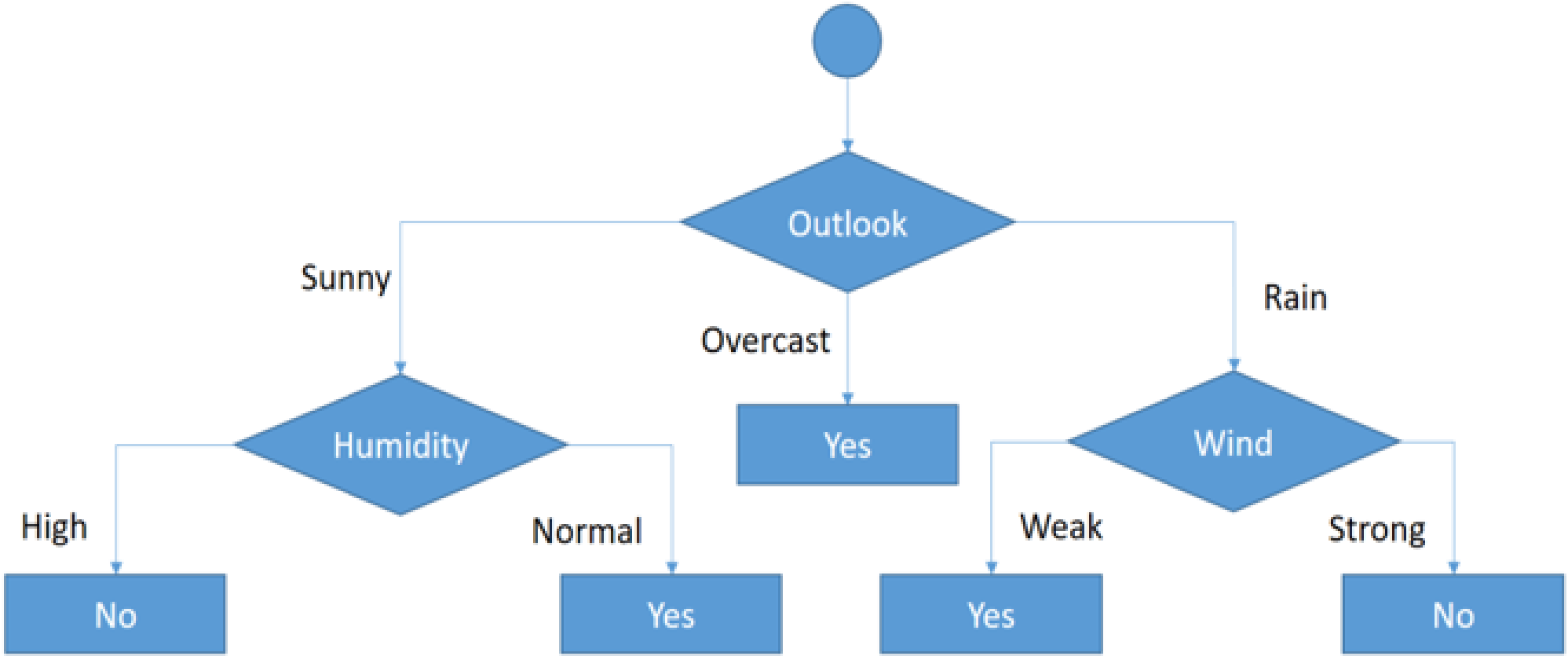
Wind	Yes	No	Number of instances
Weak	3	0	3
Strong	0	2	2

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) = $(3/5)x0 + (2/5)x0 = 0$

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



Final form of the decision tree built by CART algorithm

Advantages of ID3

- Understandable prediction rules are created from the training data.
- Builds a short tree in relatively small time.
- It only needs to test enough attributes until all data is classified.
- Finding leaf nodes enables test data to be pruned, reducing the number of tests.

Disadvantages of ID3

- Data may be over-fitted or over-classified, if a small sample is tested.
- Only one attribute at a time is considered for making a decision.

Advantages of CART

- Decision trees can inherently perform ***multiclass classification***.
- They provide ***most model interpretability*** because they are simply series of if-else conditions.
- They can handle both ***numerical*** and ***categorical data***.
- ***Nonlinear relationships*** among features do not affect the performance of the decision trees.

Disadvantages of CART

- A small change in the dataset can make the tree structure unstable which can cause variance.
- Decision tree learners create ***underfit trees*** if some classes are imbalanced. It is therefore recommended to balance the data set prior to fitting with the decision tree.