SUPERVISED AND UNSUPERVISED LEARNING

Ms. Krishna Modi, DCS

Types of Learning

- **Supervised (inductive) learning**
  - Training data includes desired outputs

- **Unsupervised learning**
  - Training data does not include desired outputs

SUPERVISED LEARNING

Ms. Krishna Modi, DCS

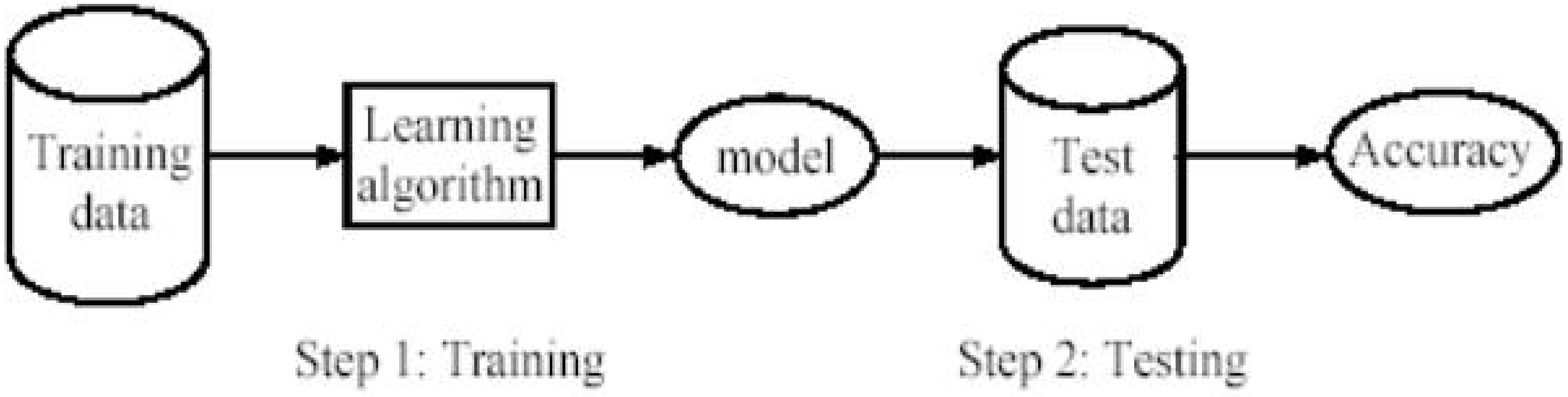
Supervised learning

- discover patterns in the data that relate data attributes with a target (class) attribute.
- These patterns are then utilized to predict the values of the target attribute in future data instances.
- Supervision: The data (observations, measurements, etc.) are labeled with pre-defined classes. It is like that a “teacher” gives the classes (supervision).
- Test data are classified into these classes too

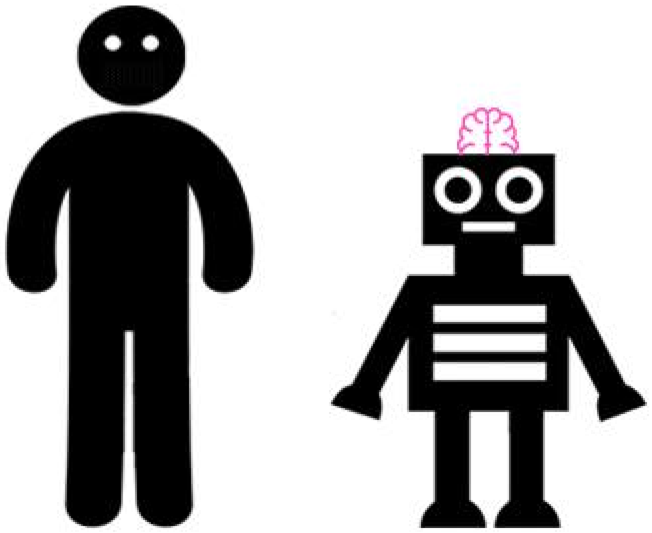
Supervised learning process: two steps

- Learning (training): Learn a model using the training data
- Testing: Test the model using unseen test data to assess the model accuracy

$$Accuracy = \frac{\text{Number of correct classifications}}{\text{Total number of test cases}},$$



Supervised Learning



An example application

- An emergency room in a hospital measures 17 variables (e.g., blood pressure, age, etc) of newly admitted patients.
- A decision is needed: whether to put a new patient in an intensive-care unit.
- Due to the high cost of ICU, those patients who may survive less than a month are given higher priority.
- Problem: to predict high-risk patients and discriminate them from low-risk patients.

Another application

- A credit card company receives thousands of applications for new cards. Each application contains information about an applicant,
  - age
  - Marital status
  - annual salary
  - outstanding debts
  - credit rating
  - etc.
- Problem: to decide whether an application should approved, or to classify applications into two categories, approved and not approved.

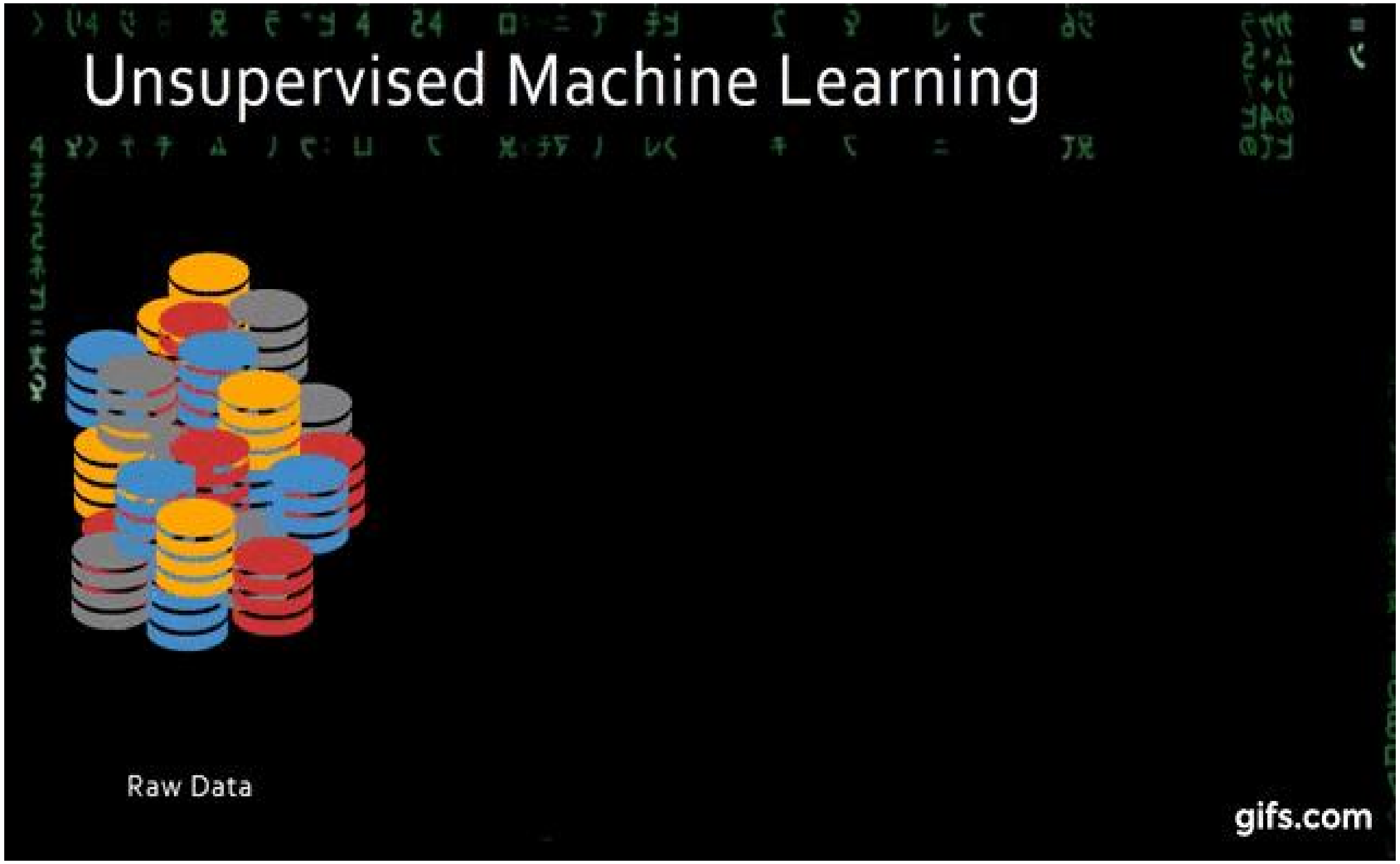
UNSUPERVISED LEARNING

Ms. Krishna Modi, DCS

Unsupervised Learning

- The data have no target attribute.
  - We want to explore the data to find some intrinsic structures in them.
  - Class labels of the data are unknown.
  - Given a set of data, the task is to establish the existence of classes or clusters in the data
  - Also known as clustering.

Unsupervised Learning



- Clustering is a technique for finding similarity groups in data, called **clusters**. I.e.,
  - it groups data instances that are similar to (near) each other in one cluster and data instances that are very different (far away) from each other into different clusters.
- Clustering is often called an **unsupervised learning** task as no class values denoting an *a priori* grouping of the data instances are given, which is the case in supervised learning.

Example:

Given a collection of text documents, we want to organize them according to their content similarities,

- To produce a topic hierarchy

- In fact, clustering is one of the most utilized data mining techniques.
  - It has a long history, and used in almost every field, e.g., medicine, psychology, botany, sociology, biology, archeology, marketing, insurance, libraries, etc.
  - In recent years, due to the rapid increase of online documents, text clustering becomes important.

Decision Tree

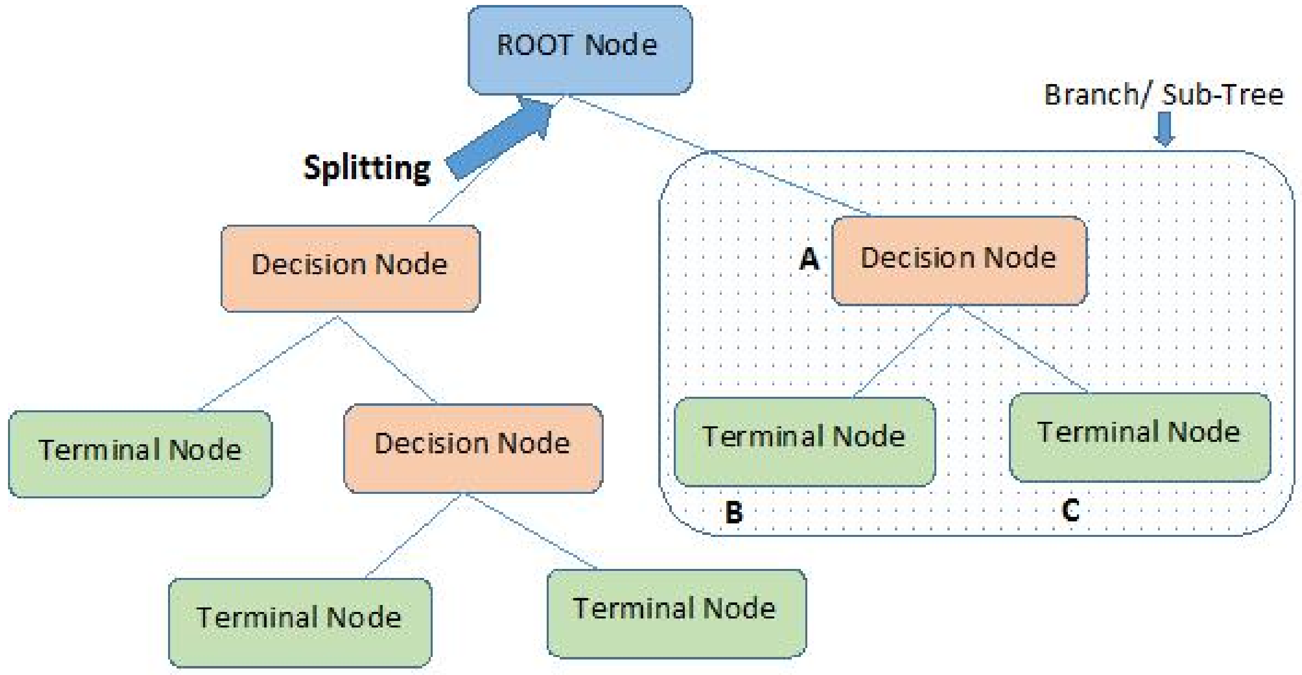
It has three main parts.

Root Node : The node that performs the first split.

Terminal Nodes/Leaves :Nodes that predict the outcome.

Branches : arrows connecting nodes, showing the flow from question to answer.

Decision Tree

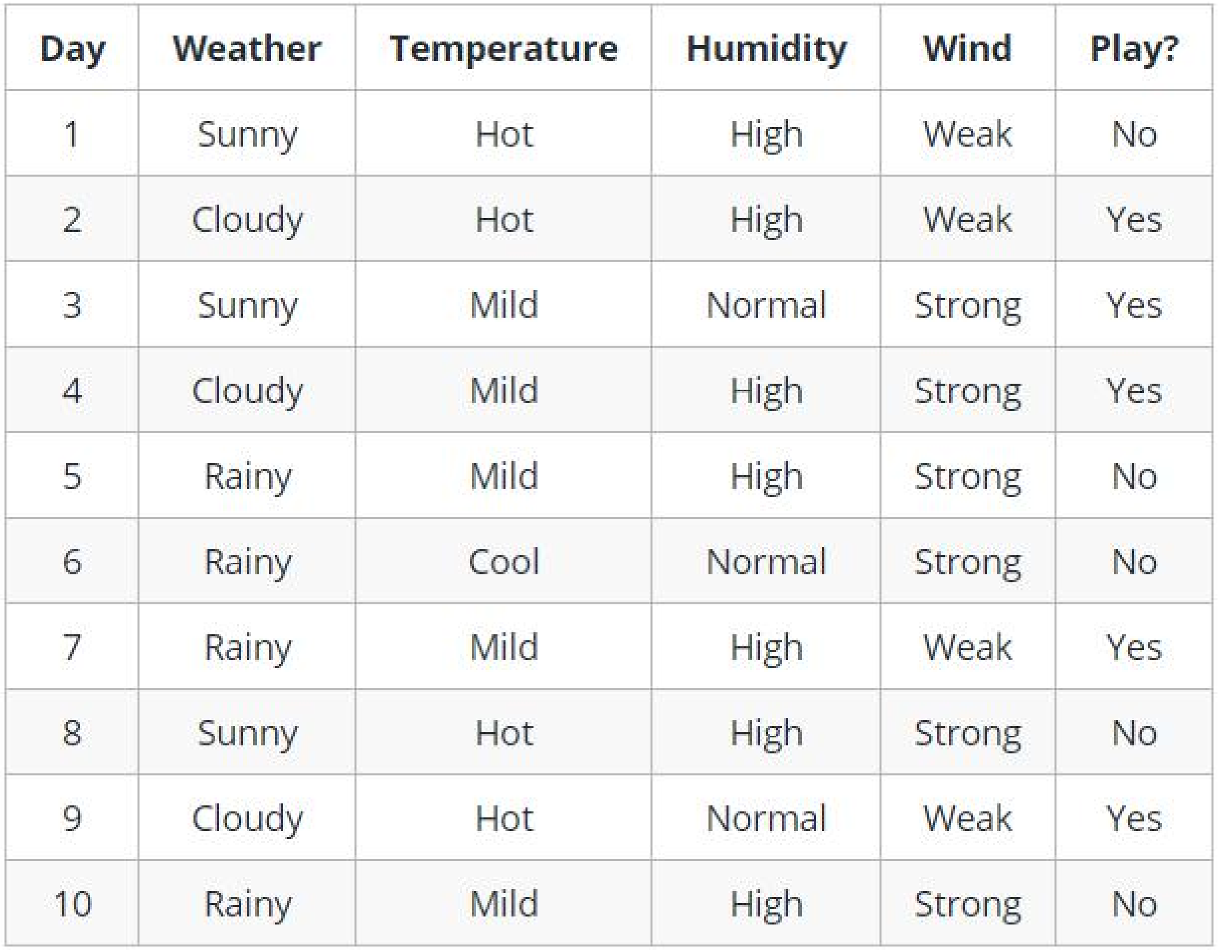


Note:- A is parent node of B and C.

Types of decision tree

- Tree models where the target variable can take a discrete set of values are called **classification trees**; in these tree structures, leaves represent class labels and branches represent conjunctions of features that lead to those class labels.
- Decision trees where the target variable can take continuous values (typically real numbers) are called **regression trees**.

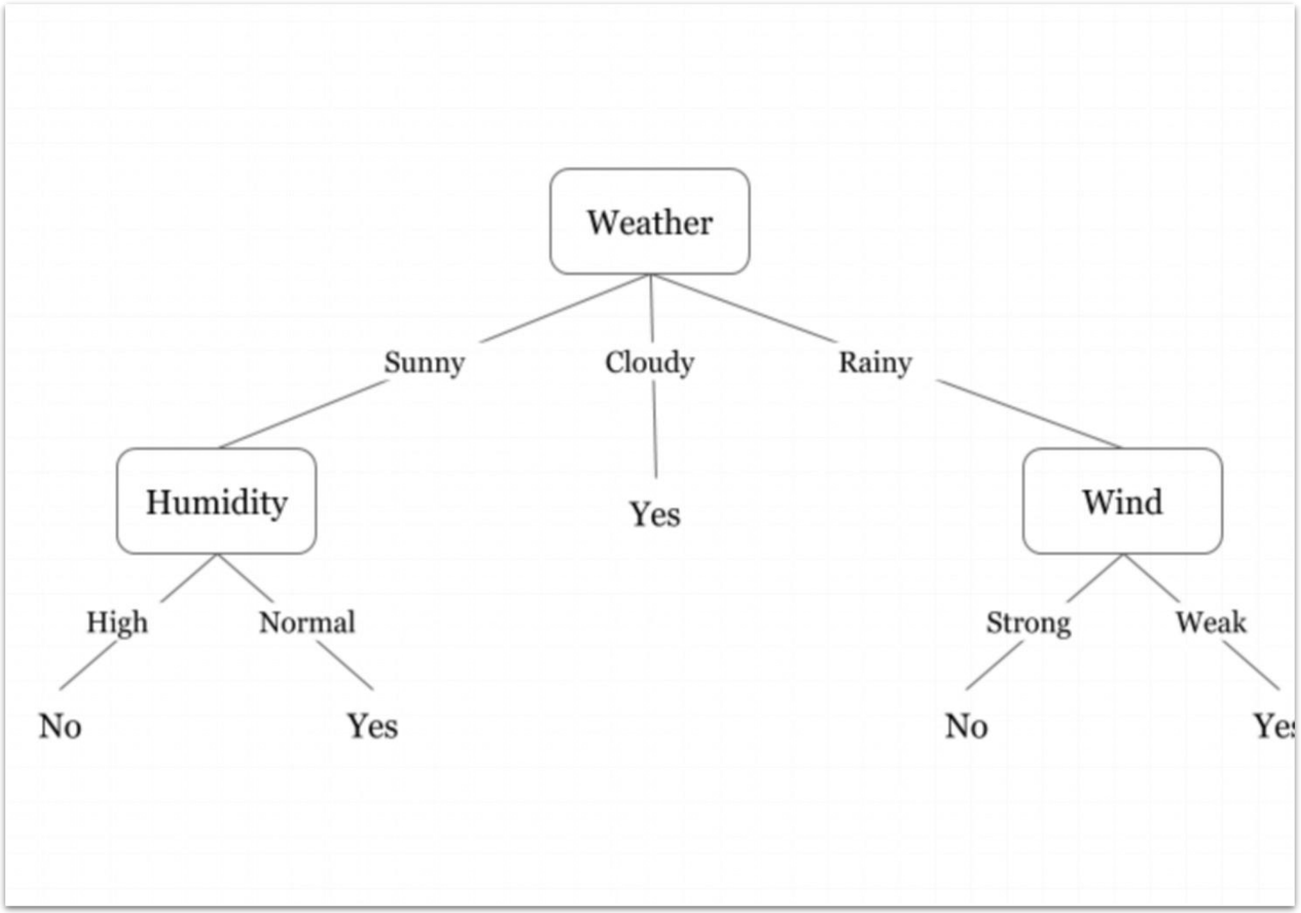
Example



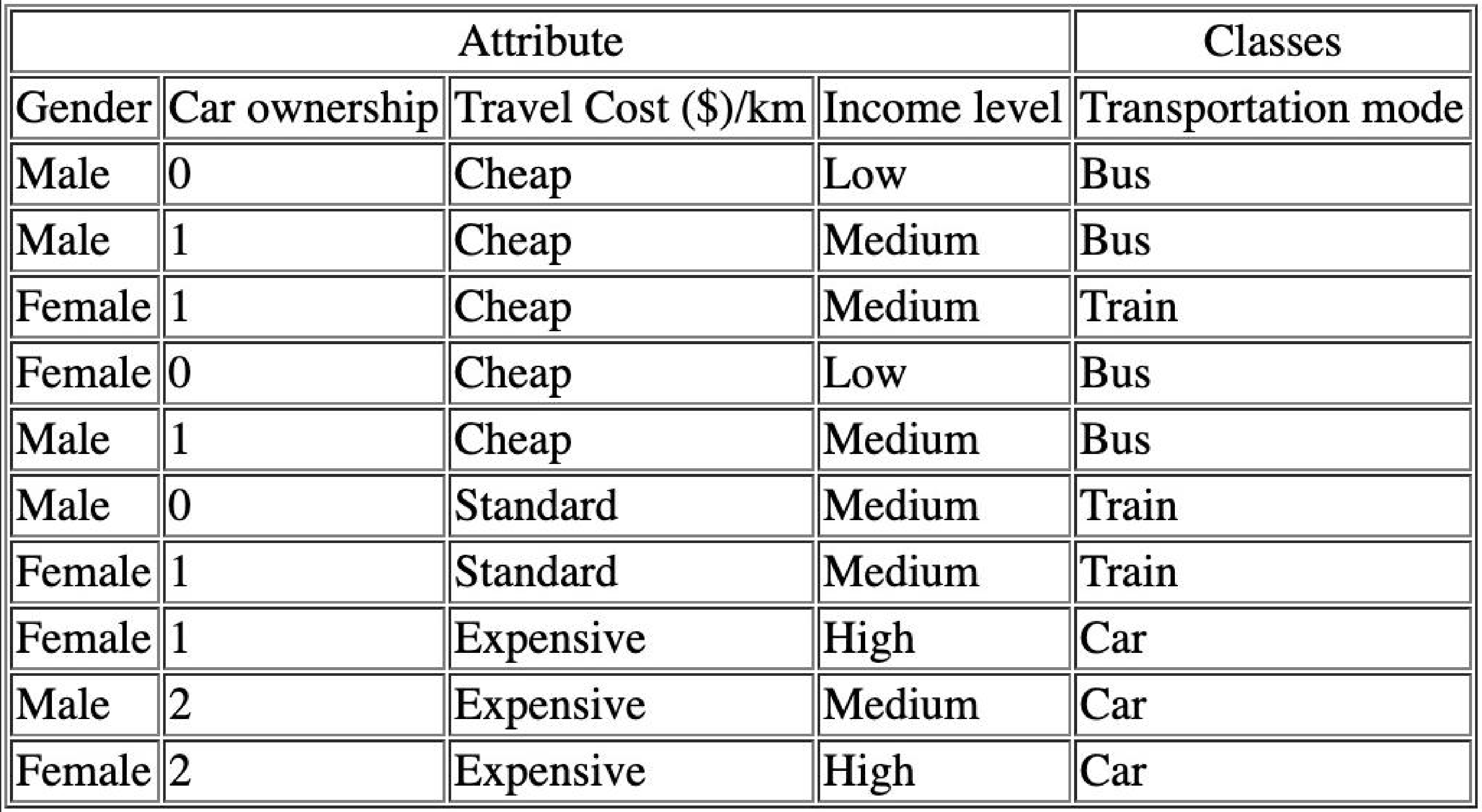
Day	Weather	Temperature	Humidity	Wind	Play?
1	Sunny	Hot	High	Weak	No
2	Cloudy	Hot	High	Weak	Yes
3	Sunny	Mild	Normal	Strong	Yes
4	Cloudy	Mild	High	Strong	Yes
5	Rainy	Mild	High	Strong	No
6	Rainy	Cool	Normal	Strong	No
7	Rainy	Mild	High	Weak	Yes
8	Sunny	Hot	High	Strong	No
9	Cloudy	Hot	Normal	Weak	Yes
10	Rainy	Mild	High	Strong	No

Decision Tree

Whether to play Tennis or not?



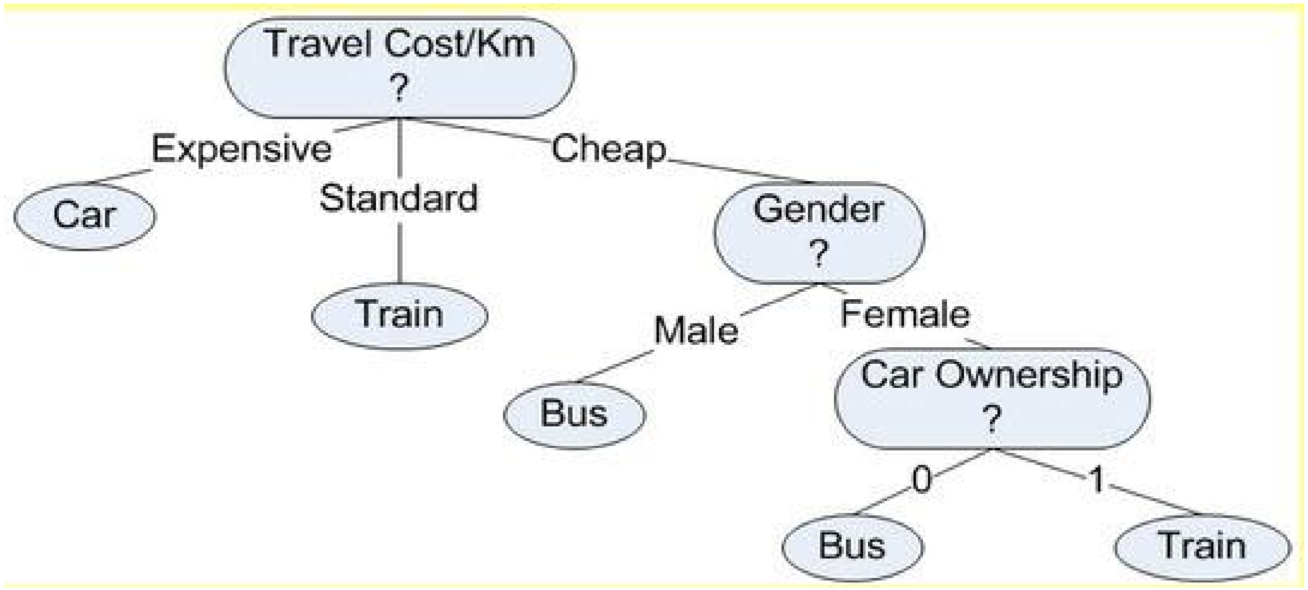
Example



Attribute				Classes
Gender	Car ownership	Travel Cost ($)/km	Income level	Transportation mode
Male	0	Cheap	Low	Bus
Male	1	Cheap	Medium	Bus
Female	1	Cheap	Medium	Train
Female	0	Cheap	Low	Bus
Male	1	Cheap	Medium	Bus
Male	0	Standard	Medium	Train
Female	1	Standard	Medium	Train
Female	1	Expensive	High	Car
Male	2	Expensive	Medium	Car
Female	2	Expensive	High	Car

Decision Tree

Transportation Mode



ID3 ALGORITHM – ITERATIVE DICHOTOMISER 3

How decision Tree works?

Used to generate a decision tree from dataset.

Ms. Krishna Modi, DCS

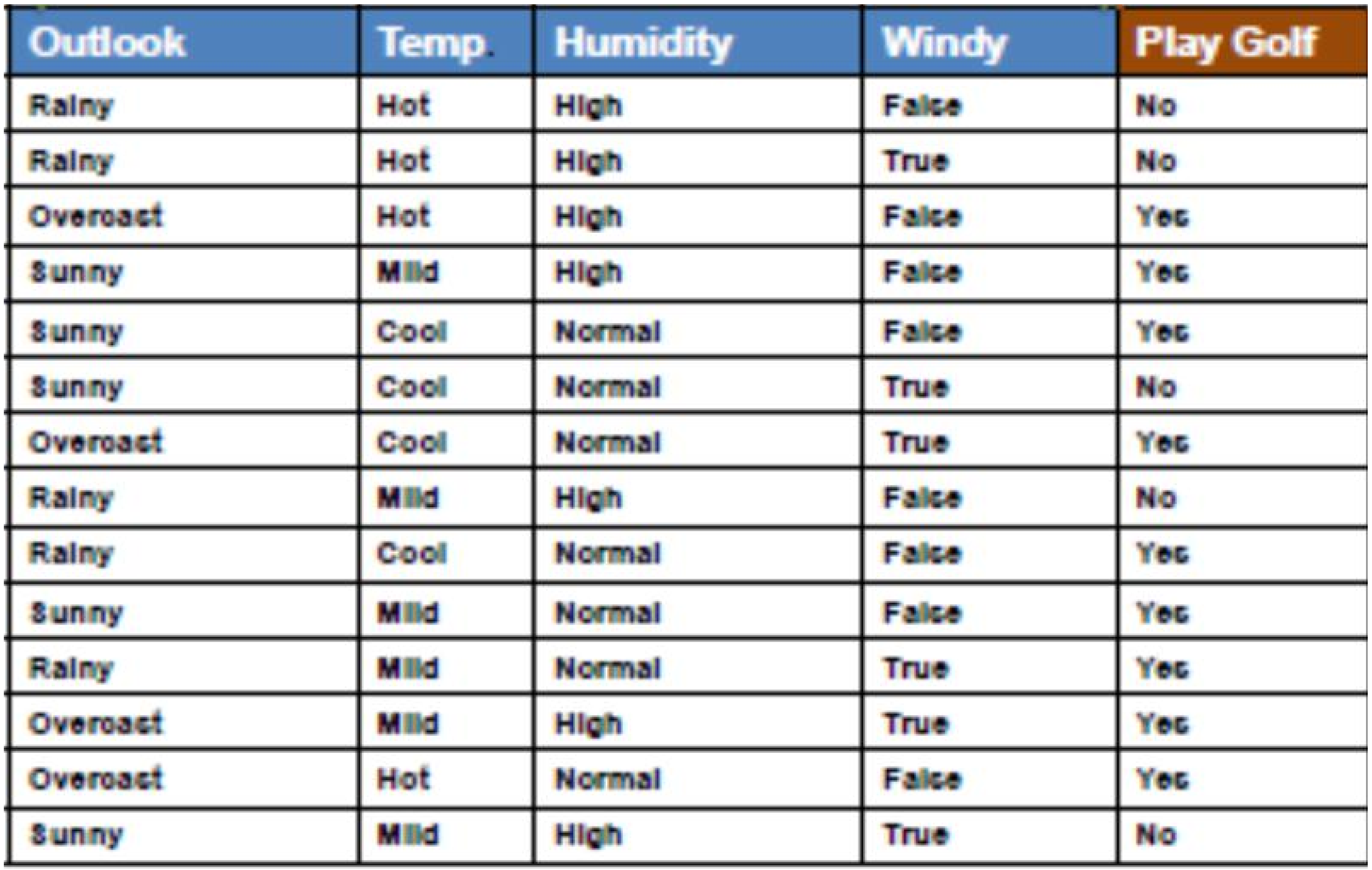
Entropy

- A decision tree is built top-down from a root node and involves partitioning the data into subsets that contain instances with similar values (homogeneous).
- ID3 algorithm uses entropy to calculate the homogeneity of a sample.
- If the sample is completely homogeneous the entropy is zero.
- if the sample is equally divided then it has entropy of one.
- Value of entropy is always between 0 to 1.

To build a decision tree, we need to calculate two types of entropy using frequency tables.

- Step 1: Calculate entropy of the target.

Example



Outlook	Temp.	Humidity	Windy	Play Golf
Rainy	Hot	High	False	No
Rainy	Hot	High	True	No
Overcast	Hot	High	False	Yes
Sunny	Mild	High	False	Yes
Sunny	Cool	Normal	False	Yes
Sunny	Cool	Normal	True	No
Overcast	Cool	Normal	True	Yes
Rainy	Mild	High	False	No
Rainy	Cool	Normal	False	Yes
Sunny	Mild	Normal	False	Yes
Rainy	Mild	Normal	True	Yes
Overcast	Mild	High	True	Yes
Overcast	Hot	Normal	False	Yes
Sunny	Mild	High	True	No

a) Entropy using the frequency table of one attribute:

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

Play Golf	
Yes	No
9	5

Entropy(PlayGolf) = Entropy (5,9)

= Entropy (0.36, 0.64)

= - (0.36 $\log_2$ 0.36) - (0.64 $\log_2$ 0.64)

= 0.94

***Step 2*:**

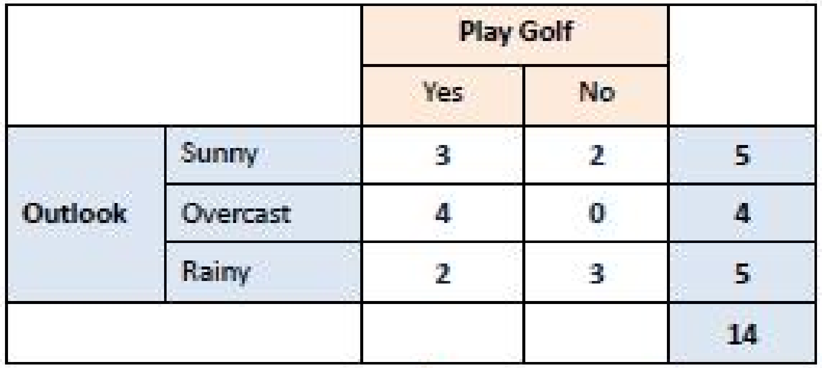
The dataset is then split on the different attributes. The entropy for each branch is calculated.

Then it is added proportionally, to get total entropy for the split.

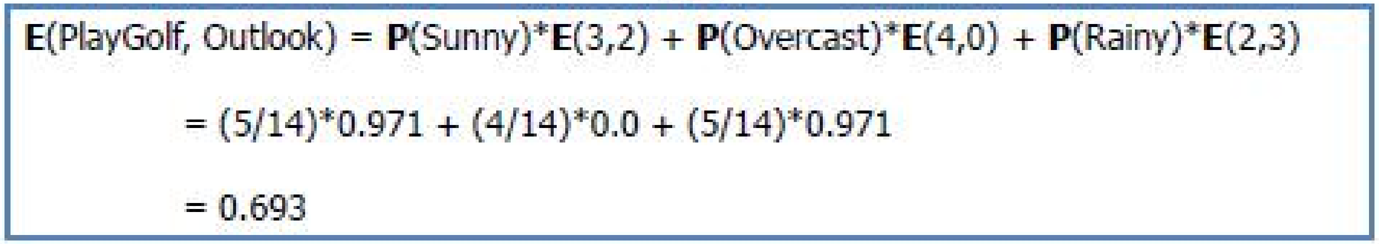
The resulting entropy is subtracted from the entropy before the split. The result is the Information Gain, or decrease in entropy.

b) Entropy using the frequency table of two attributes:

$$E(T,X)=\sum_{c\in X}P(c)E(c)$$



		Play Golf		
		Yes	No	
Outlook	Sunny	3	2	5
	Overcast	4	0	4
	Rainy	2	3	5
				14



E(PlayGolf, Outlook) = **P**(Sunny)***E**(3,2) + **P**(Overcast)***E**(4,0) + **P**(Rainy)***E**(2,3)

= (5/14)*0.971 + (4/14)*0.0 + (5/14)*0.971

= 0.693

Information Gain

- Statistical property that measures how well a given attribute separates the training examples according to their target classification.
- To define information gain, we need to define a measure **Entropy.**

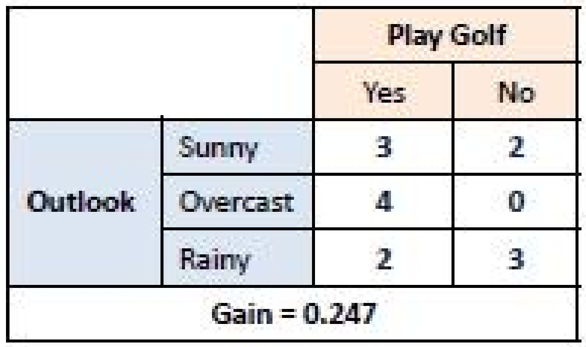
Information Gain

$$Gain(T,X) = Entropy(T) - Entropy(T,X)$$

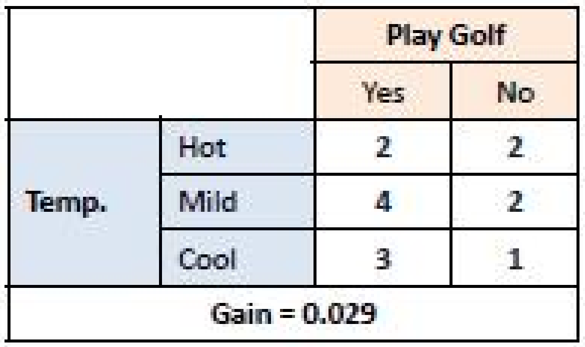
G(PlayGolf, Outlook) = **E**(PlayGolf) – **E**(PlayGolf, Outlook)

= 0.940 – 0.693 = 0.247

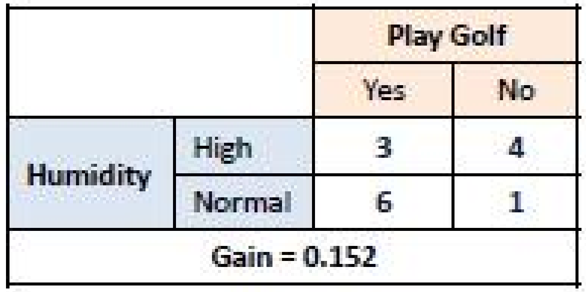
Information Gain is calculated for each attribute.



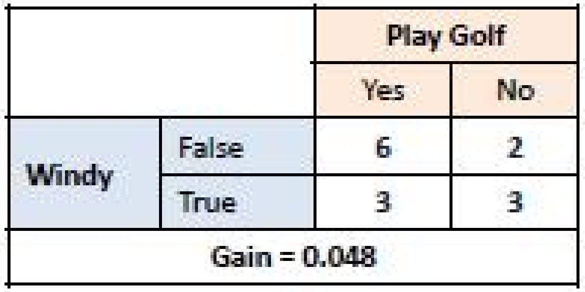
		Play Golf	
		Yes	No
Outlook	Sunny	3	2
	Overcast	4	0
	Rainy	2	3
Gain = 0.247			



		Play Golf	
		Yes	No
Temp.	Hot	2	2
	Mild	4	2
	Cool	3	1
Gain = 0.029			

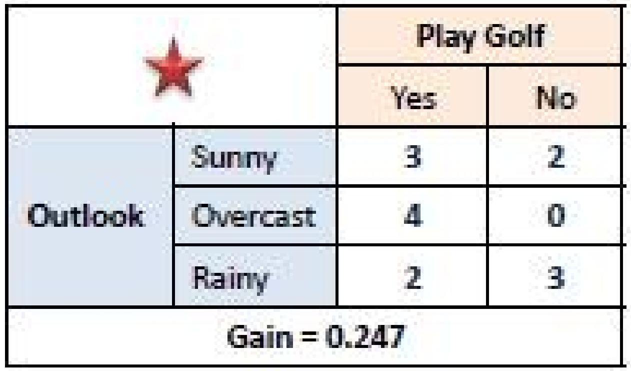


		Play Golf	
		Yes	No
Humidity	High	3	4
	Normal	6	1
Gain = 0.152			

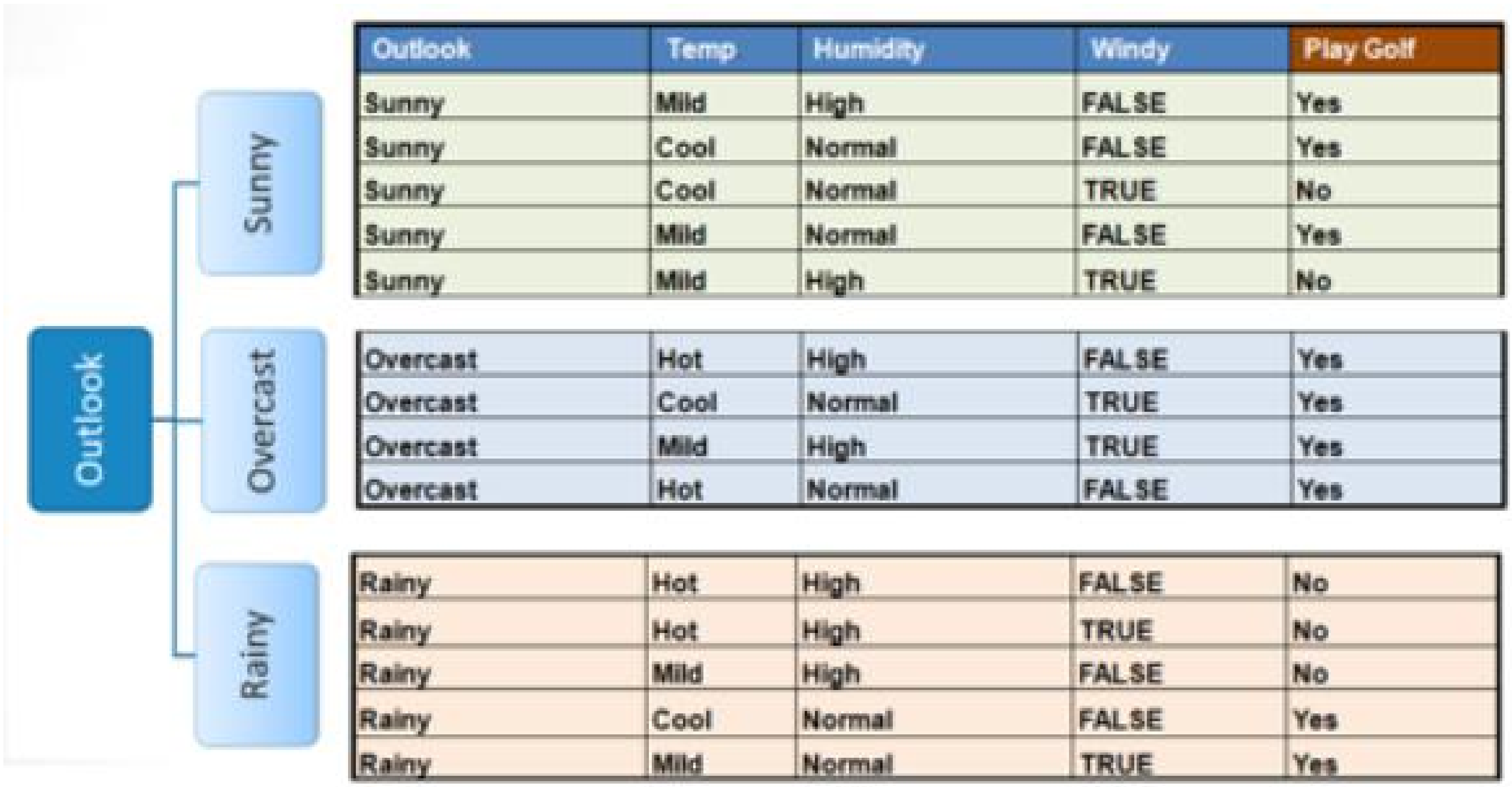


		Play Golf	
		Yes	No
Windy	False	6	2
	True	3	3
Gain = 0.048			

Step 3: Choose attribute with the largest information gain as the decision node, divide the dataset by its branches and repeat the same process on every branch.



★		Play Golf	
		Yes	No
Outlook	Sunny	3	2
	Overcast	4	0
	Rainy	2	3
Gain = 0.247			



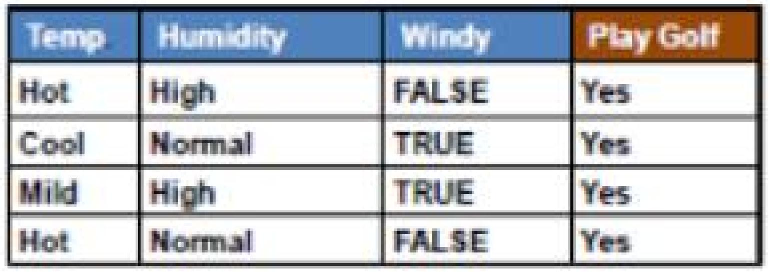
Outlook	Temp	Humidity	Windy	Play Golf
Sunny	Mild	High	FALSE	Yes
Sunny	Cool	Normal	FALSE	Yes
Sunny	Cool	Normal	TRUE	No
Sunny	Mild	Normal	FALSE	Yes
Sunny	Mild	High	TRUE	No
Overcast	Hot	High	FALSE	Yes
Overcast	Cool	Normal	TRUE	Yes
Overcast	Mild	High	TRUE	Yes
Overcast	Hot	Normal	FALSE	Yes
Rainy	Hot	High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

Step 4 :

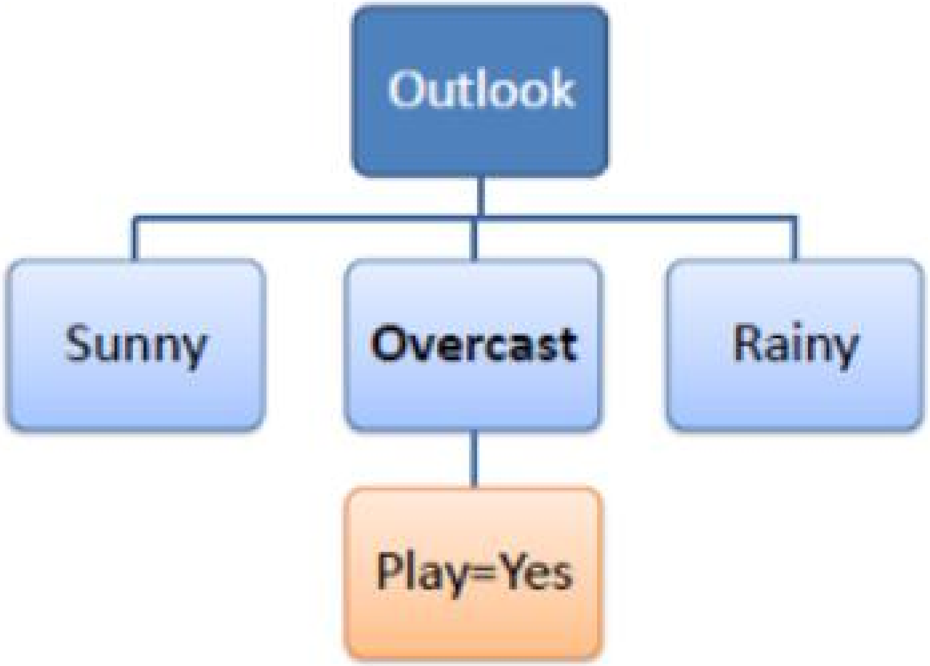
A branch with entropy 0 is a leaf node.

A branch with entropy more than 0 needs further splitting.

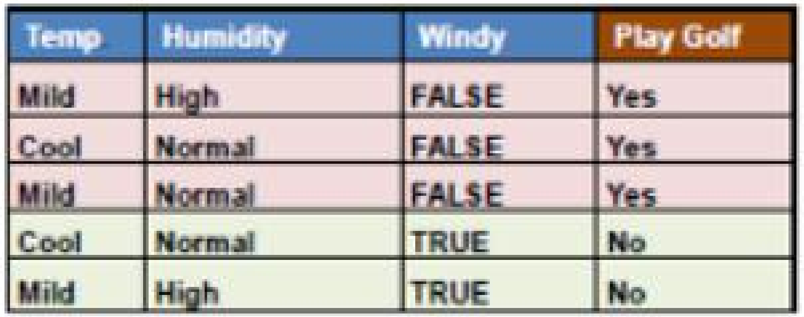
A branch with entropy of 0 is a leaf node.



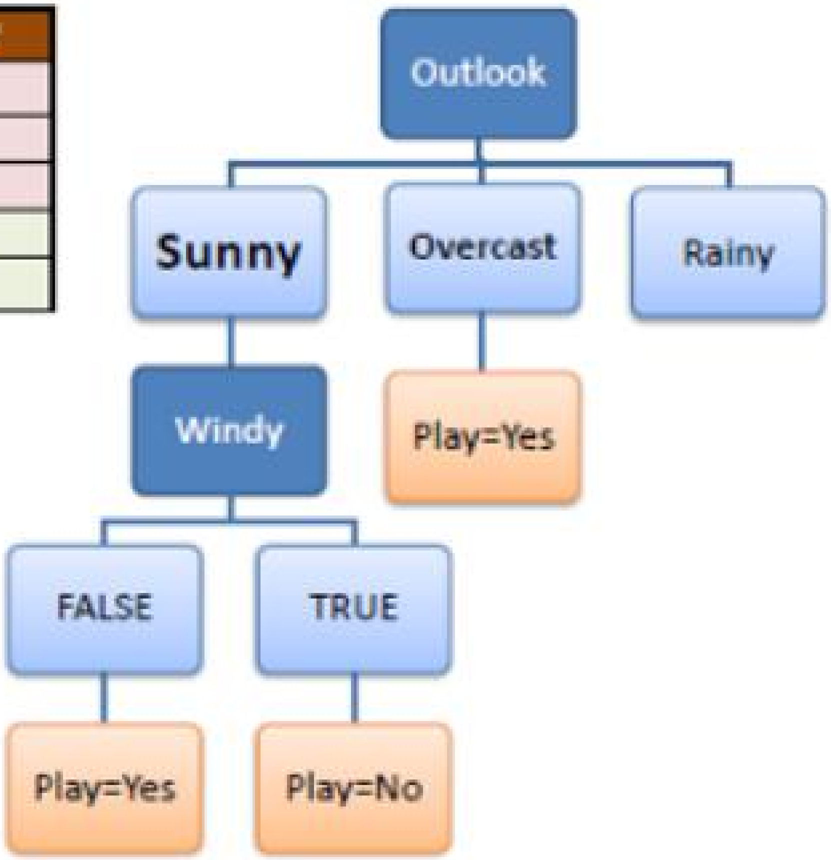
Temp	Humidity	Windy	Play Golf
Hot	High	FALSE	Yes
Cool	Normal	TRUE	Yes
Mild	High	TRUE	Yes
Hot	Normal	FALSE	Yes



A branch with entropy more than 0 needs further splitting.



Temp	Humidity	Windy	Play Golf
Mild	High	FALSE	Yes
Cool	Normal	FALSE	Yes
Mild	Normal	FALSE	Yes
Cool	Normal	TRUE	No
Mild	High	TRUE	No

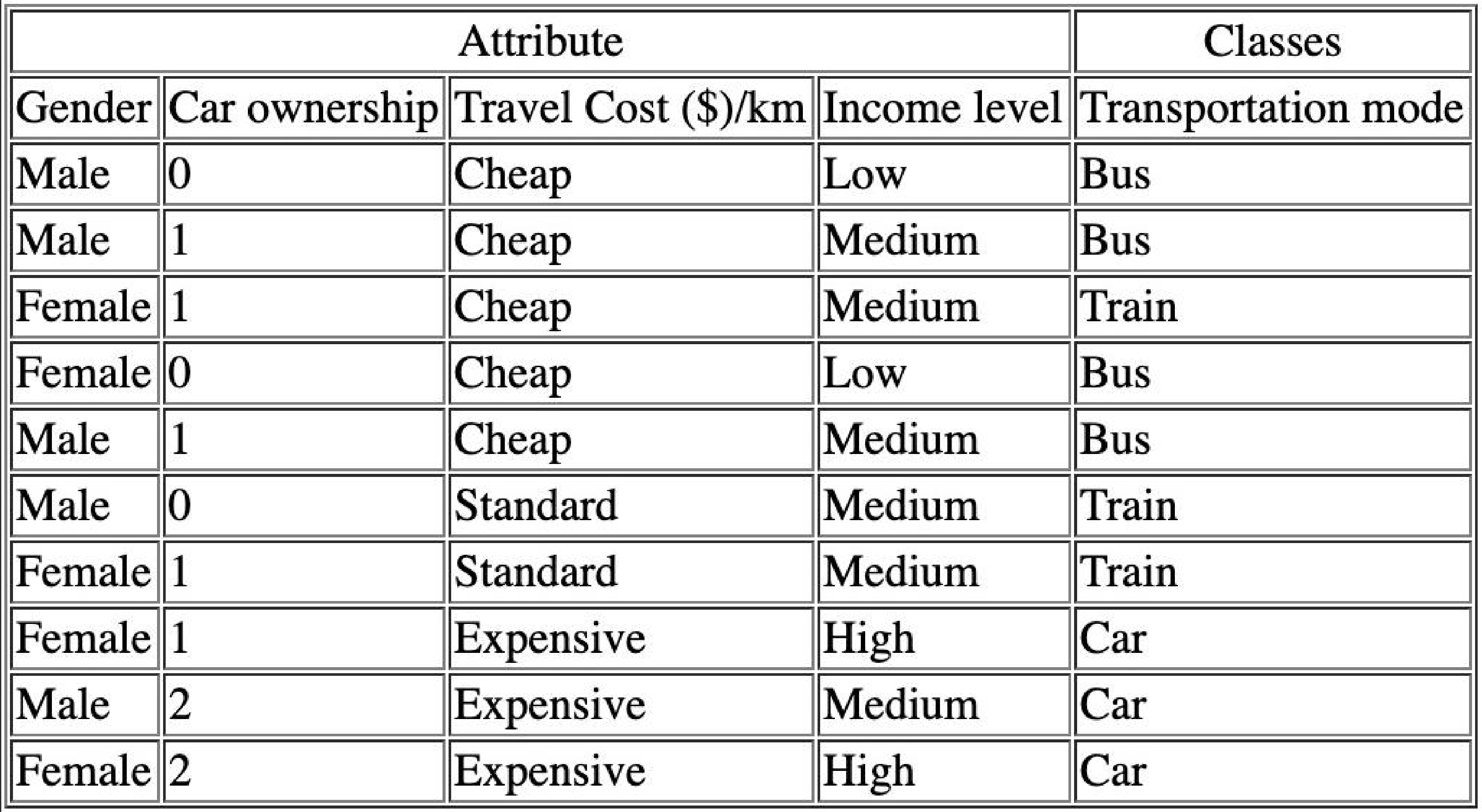


Step 5: The ID3 algorithm is run recursively on the non-leaf branches, until all data is classified.

Inductive bias

- inductive bias is the set of assumptions that, together with the training data, deductively justify the classifications assigned by the learner to future instances.
- Given a collection of training examples, there are typically many decision trees consistent with these examples.
- **Describing the inductive bias of ID3 therefore consists of describing the basis by which it chooses one of these consistent hypotheses over the others.**

Example



Attribute				Classes
Gender	Car ownership	Travel Cost ($)/km	Income level	Transportation mode
Male	0	Cheap	Low	Bus
Male	1	Cheap	Medium	Bus
Female	1	Cheap	Medium	Train
Female	0	Cheap	Low	Bus
Male	1	Cheap	Medium	Bus
Male	0	Standard	Medium	Train
Female	1	Standard	Medium	Train
Female	1	Expensive	High	Car
Male	2	Expensive	Medium	Car
Female	2	Expensive	High	Car

Which of decision trees ID3 select?

(a) selects in favor of shorter trees over longer ones, and

(b) selects trees that place the attributes with highest information gain closest to the root.

- for example there are many more 500-node decision trees than 5-node decision trees. Given a small set of 20 training examples, we might expect to be able to find many 500-node decision trees consistent with these, whereas we would be more surprised if a 5-node decision tree could perfectly fit this data. We might therefore believe the 5-node tree is less likely to be a statistical coincidence and prefer this hypothesis over the 500-node hypothesis.

Issues in decision tree learning

1. **how deeply to grow the decision tree?**

Issues in decision tree learning

2. **handling continuous attributes**

Issues in decision tree learning

3. choosing an appropriate attribute selection measure

Issues in decision tree learning

4. handling training data with missing attribute values

Issues in decision tree learning

5. improving computational efficiency

K – NEAREST NEIGHBOR (KNN)

Classification algorithm

Ms. Krishna Modi, DCS

Example

- For example, movie ratingYou have a bunch of movies that have been classified as “thumbs up” or “thumbs down,”.
- You want to classify the movie “Spiderman"
  - Step 1: Define the attributes of “Spiderman"
    - length of movie
    - genre
    - number of action scenes
    - number of Oscar-winning actors in it
    - budget
  - Step 2: find the majority rating of other movies with similar attributes
  - Step 3: Assign rating to “Spiderman"
    - If there is a majority rating, then the rating of " Spiderman " should have the same rating.
    - If there’s a tie, then the rating of " Spiderman " is randomly

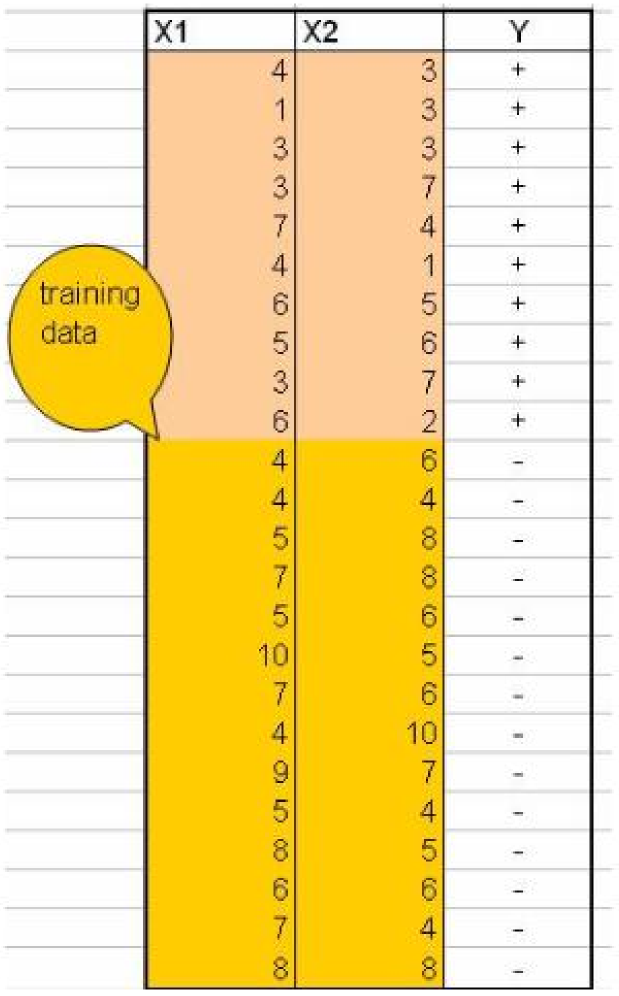
Basics of KNN

- Training Method :
  - Save the training example.
- At prediction time :
  - Find the k training examples (x1,y1),… , (xk,yk) that are closest to the test example x.
  - Classification: Predict the most frequent class among those yi’s.
  - Regression : Predict the average of yi’s.
  -

K – Nearest neighbor

- It is used when
  - you have a bunch of objects that have been classified, and
  - you want a way to automatically classify similar objects that haven’t gotten classified.
- Procedure of automation
  - Step 1: Define similarity of two objects
  - Step 2: Compare the similarity of a given unrated object with all the classified objects.
  - Step 3: Take the most similar objects and call them neighbors, who each have a “label.”
  - Step 4: Determine the "label" of the unrated object.

Example


training data

X1	X2	Y
4	3	+
1	3	+
3	3	+
3	7	+
7	4	+
4	1	+
6	5	+
5	6	+
3	7	+
6	2	+
4	6	-
4	4	-
5	8	-
7	8	-
5	6	-
10	5	-
7	6	-
4	10	-
9	7	-
5	4	-
8	5	-
6	6	-
7	4	-
8	8	-

Predict value for (9,6)

Working of kNN Algorithm

- **Step 1** – For implementing any algorithm, we need dataset. So during the first step of KNN, we must load the training as well as test data.
- **Step 2** – Next, we need to choose the value of K i.e. the nearest data points. K can be any integer.
- **Step 3** – For each point in the test data do the following –
  - **3.1** – Calculate the distance between test data and each row of training data with the help of any of the method namely: Euclidean, Manhattan or Hamming distance. The most commonly used method to calculate distance is Euclidean.
  - **3.2** – Now, based on the distance value, sort them in ascending order.
  - **3.3** – Next, it will choose the top K rows from the sorted array.
  - **3.4** – Now, it will assign a class to the test point based on most frequent class of these rows.
- **Step 4** – End

Advantages

- The algorithm is simple and easy to implement.
- There’s no need to build a model, tune several parameters, or make additional assumptions.
- The algorithm is versatile. It can be used for classification, regression, and search (as we will see in the next section).

Disadvantages

- The algorithm gets significantly slower as the number of examples and/or predictors/independent variables increase.

Example – use 3 Nearest Neighbor

Height (in cms)	Weight (in kgs)	T Shirt Size
158	58	M
158	59	M
158	63	M
160	59	M
165	65	L
168	62	L
168	63	L
168	66	L
170	63	L
170	64	L
170	68	L
161	61	?

Example

Sepal length	Sepal width	Petal length	Petal width	Species
5.1	3.5	1.4	0.2	Iris-Setosa
4.9	3	1.4	0.2	Iris-Setosa
5.7	3.8	1.7	0.3	Iris-setosa
5.1	3.8	1.5	0.3	Iris-setosa
5.6	2.9	3.6	1.3	Iris-versicolor
5.6	3	4.5	1.5	Iris-versicolor
5.8	2.7	4.1	1	Iris-versicolor
6.4	3.1	5.5	1.8	Iris-virginica
6	3	4.8	1.8	Iris-virginica
7.7	3	6.1	2.3	???

References

- https://medium.com/@rishabhjain_22692/decision-trees-it-begins-here-93ff54ef134

- https://people.revoledu.com/kardi/tutorial/KNN/HowTo_KNN.html
- https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_ml/py_knn/py_knn_understanding/py_knn_understanding.html
- https://www.csee.umbc.edu/~tinoosh/cmpe650/slides/K_Nearest_Neighbor_Algorithm.pdf
- http://www.saedsayad.com/k_nearest_neighbors.htm