INTRODUCTION TO DATA ANALYTICS

Class # 19

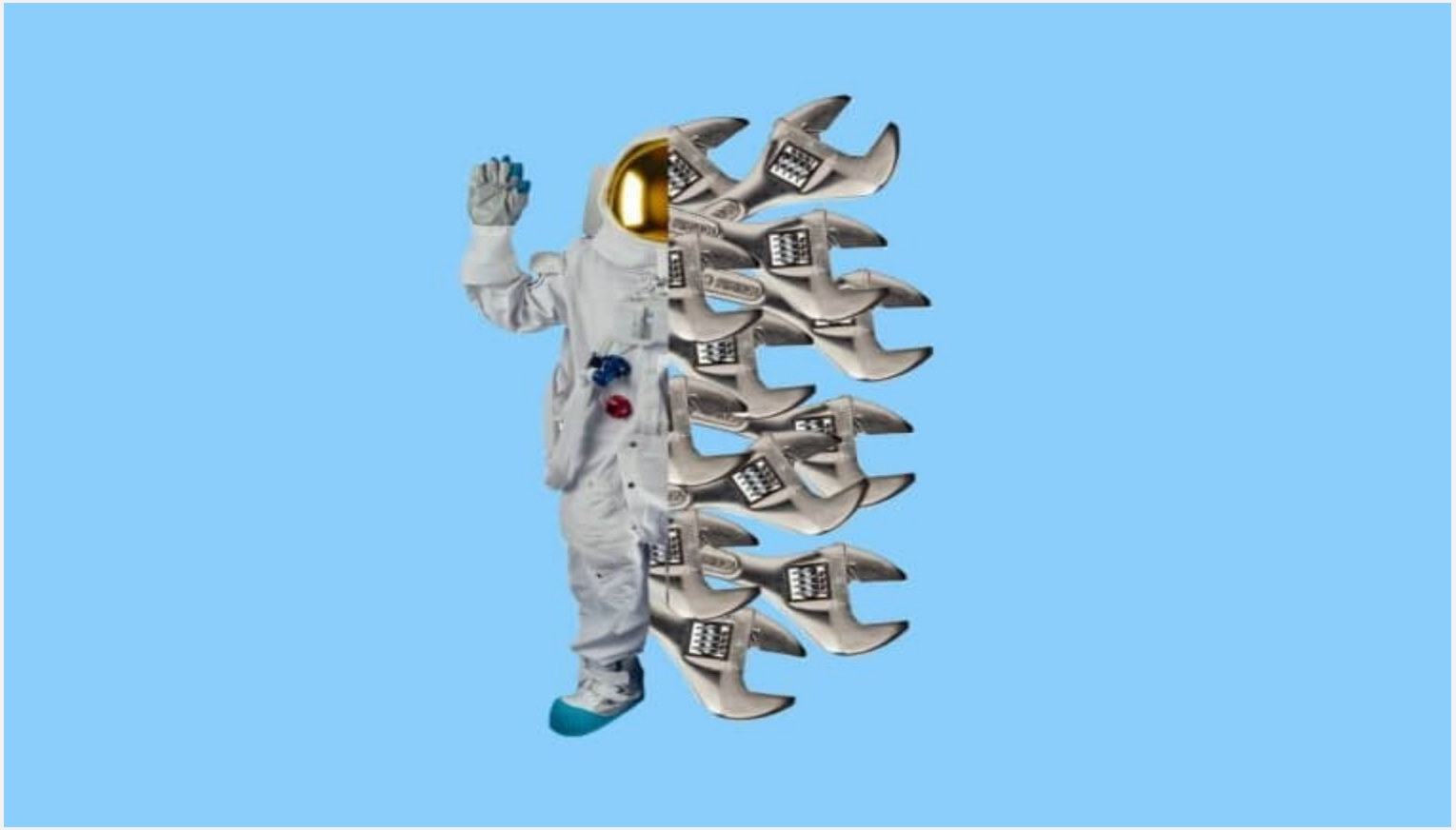
Classification: Decision Tree Induction

Dr. Sreeja S R

Assistant Professor

Indian Institute of Information Technology

IIIT Sri City



Thanks to 3D printing, NASA can basically “email” tools to astronauts.
Getting new equipment to the Space Station used to take months or years, but the new technology means the tools are ready within hours.

THIS PRESENTATION SLIDES INCLUDES…

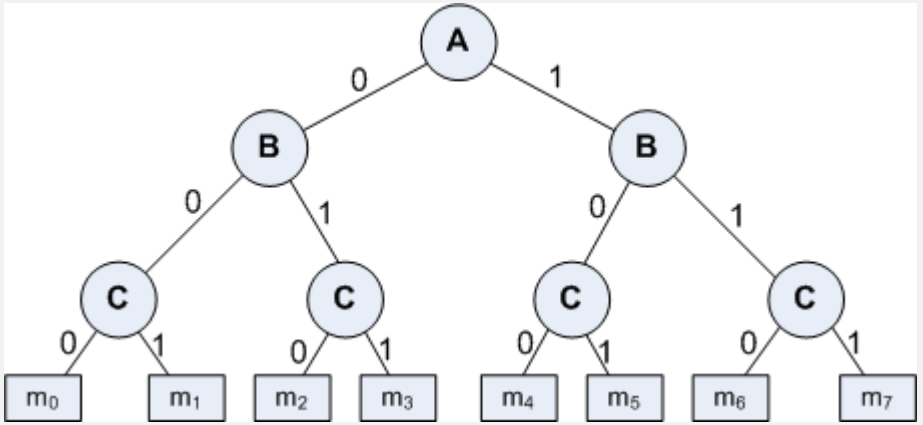
- Concept of Decision Tree
- Use of Decision Tree to classify data
- Basic algorithm to build Decision Tree
  - Some illustrations
- Concept of Entropy
  - Basic concept of entropy in information theory
  - Mathematical formulation of entropy
  - Calculation of entropy of a training set
- Decision Tree induction algorithms
  - ID3
  - CART
  - C4.5

BASIC CONCEPT

- A Decision Tree is an important data structure known to solve many computational problems

Example 19.1: Binary Decision Tree

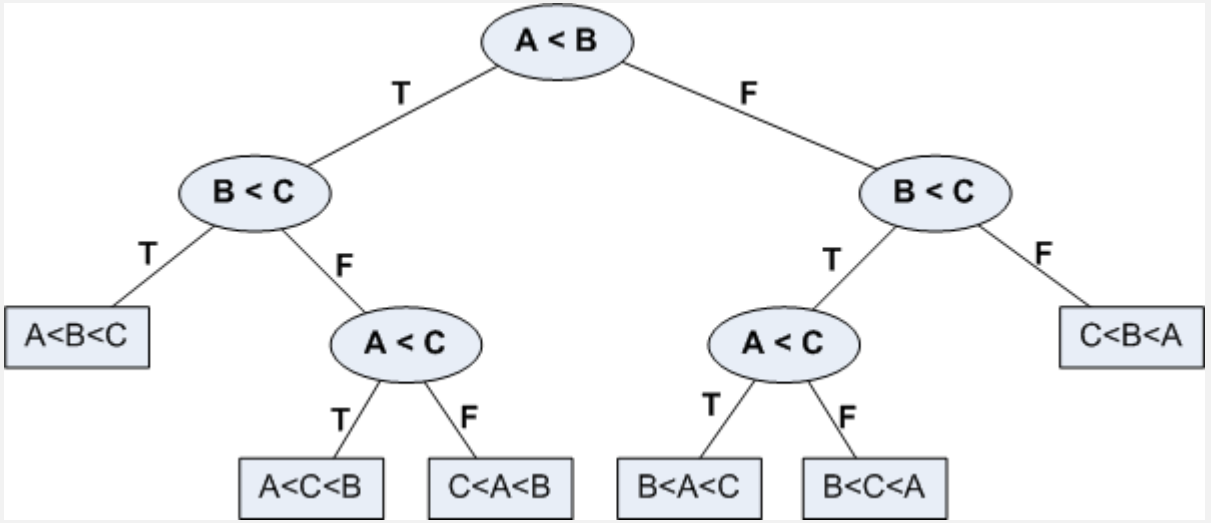
A	B	C	f
0	0	0	m_0
0	0	1	m_1
0	1	0	m_2
0	1	1	m_3
1	0	0	m_4
1	0	1	m_5
1	1	0	m_6
1	1	1	m_7



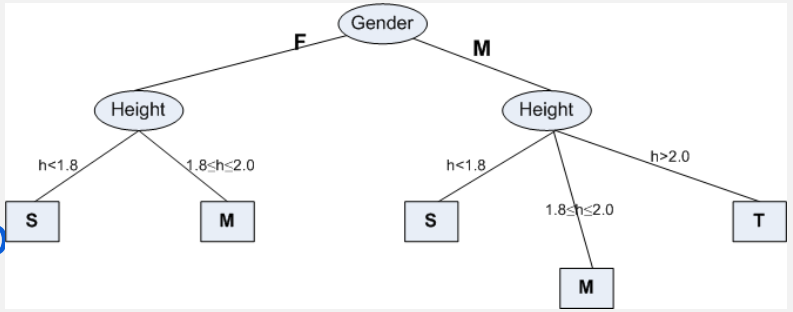
BASIC CONCEPT

- In Example 19.1, we have considered a decision tree where values of any attribute if binary only. Decision tree is also possible where attributes are of continuous data type

Example 19.2: Decision Tree with numeric data



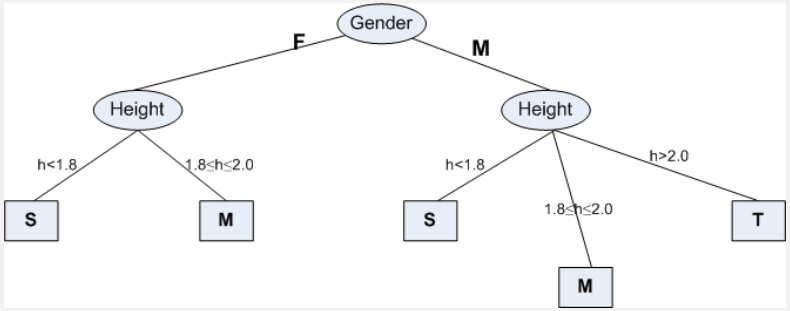
SOME CHARACTERISTICS



- Decision tree may be *n*-ary, $n \geq 2$.
- There is a special node called root no
- All nodes drawn with circle (ellipse) are called internal nodes.
- All nodes drawn with rectangle boxes are called terminal nodes or leaf nodes.
- Edges of a node represent the outcome for a value of the node.
- In a path, a node with same label is never repeated.
- Decision tree is not unique, as different ordering of internal

DECISION TREE AND CLASSIFICATION TASK

- Decision tree helps us to classify data.
  - Internal nodes are some attribute
  - Edges are the values of attributes
  - External nodes are the outcome of classification



- Such a classification is, in fact, made by posing questions starting from the root node to each terminal node.

DECISION TREE AND CLASSIFICATION TASK

Example 19.3 : Vertebrate Classification

Name	Body Temperature	Skin Cover	Gives Birth	Aquatic Creature	Aerial Creature	Has Legs	Hibernates	Class
Human	Warm	hair	yes	no	no	yes	no	**Mammal**
Python	Cold	scales	no	no	no	no	yes	**Reptile**
Salmon	Cold	scales	no	yes	no	no	no	**Fish**
Whale	Warm	hair	yes	yes	no	no	no	**Mammal**
Frog	Cold	none	no	semi	no	yes	yes	**Amphibian**
Komodo	Cold	scales	no	no	no	yes	no	**Reptile**
Bat	Warm	hair	yes	no	yes	yes	yes	**Mammal**
Pigeon	Warm	feathers	no	no	yes	yes	no	**Bird**
Cat	Warm	fur	yes	no	no	yes	no	**Mammal**
Leopard	Cold	scales	yes	yes	no	no	no	**Fish**
Turtle	Cold	scales	no	semi	no	yes	no	**Reptile**
Penguin	Warm	feathers	no	semi	no	yes	no	**Bird**
Porcupine	Warm	quills	yes	no	no	yes	yes	**Mammal**
Eel	Cold	scales	no	yes	no	no	no	**Fish**
Salamander	Cold	none	no	semi	no	yes	yes	**Amphibian**

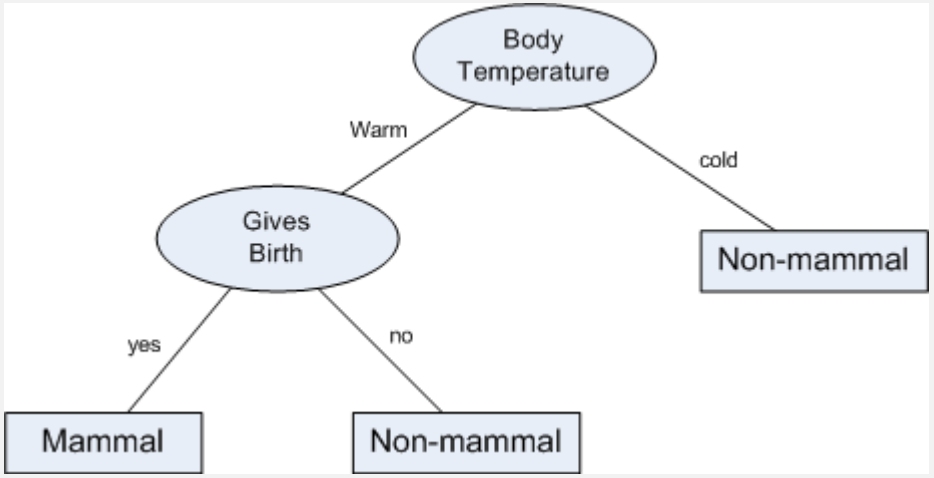
DECISION TREE AND CLASSIFICATION TASK

Example 19.3 : Vertebrate Classification

- Suppose, a new species is discovered as follows.

Name	Body Temperature	Skin Cover	Gives Birth	Aquatic Creature	Aerial Creature	Has Legs	Hibernates	Class
Gila Monster	cold	scale	no	no	no	yes	yes	**?**

Decision tree that can be inducted based on the data (in Example 19.3) is as follows.



DECISION TREE AND CLASSIFICATION TASK

- Example 19.3 illustrates how we can solve a classification problem by asking a series of question about the attributes.
  - Each time we receive an answer, a follow-up question is asked until we reach a conclusion about the class-label of the test.
- The series of questions and their answers can be organized in the form of a decision tree
  - As a hierarchical structure consisting of nodes and edges
- Once a decision tree is built, it is applied to any test to classify it.

DEFINITION OF DECISION TREE

Definition: **Decision Tree**

Given a database D = here denotes a tuple, which is defined by a set of attribute set of classes C = .

A decision tree ***T*** is a tree associated with D that has the following properties:

- Each internal node is labeled with an attribute A_i
- Each edges is labeled with predicate that can be applied to the attribute associated with the parent node of it
- Each leaf node is labeled with class c_j

BUILDING DECISION TREE

- In principle, there are exponentially many decision tree that can be constructed from a given database (also called training data).
  - Some of the tree may not be optimum
  - Some of them may give inaccurate result
- Two approaches are known
  - **Greedy strategy**
    - A top-down recursive divide-and-conquer
  - **Modification of greedy strategy**
    - ID3
    - C4.5
    - CART, etc.

BUILT DECISION TREE ALGORITHM

- **Algorithm BuiltDT**
- Input: D : Training data set
- Output: T : Decision tree

Steps

1. If all tuples in D belongs to the same class C_j

   Add a leaf node labeled as C_j

   Return *// Termination condition*
2. **Select** an attribute A_i (so that it is not selected twice in the same branch)
3. **Partition** $D = \{ D_1, D_2, \ldots, D_p \}$ based on p different values of A_i in D
4. For each $D_k \in D$

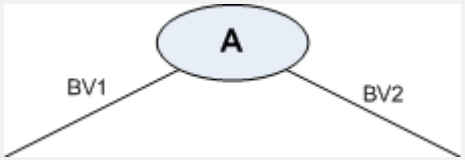
   Create a node and add an edge between D and D_k with label as the A_i's attribute value in D_k
5. For each $D_k \in D$

   BuildDT(D_k) *// Recursive call*
6. Stop

NODE SPLITTING IN **BUILDDT** ALGORITHM

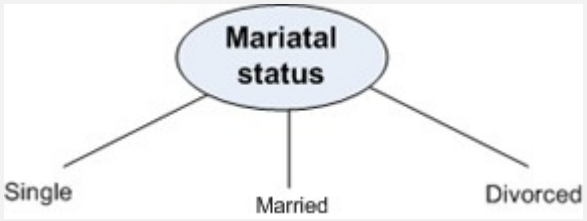
- BuildDT algorithm must provides a method for expressing an attribute test condition and corresponding outcome for different attribute type

- **Case: Binary attribute**
  - This is the simplest case of node splitting
  - The test condition for a binary attribute generates only two outcomes

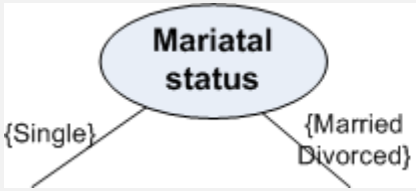


NODE SPLITTING IN **BUILDDT** ALGORITHM

- **Case: Nominal attribute**
  - Since a nominal attribute can have many values, its test condition can be expressed in two ways:
    - A multi-way split
    - A binary split
  - Muti-way split: Outcome depends on the number of distinct values for the corresponding attribute

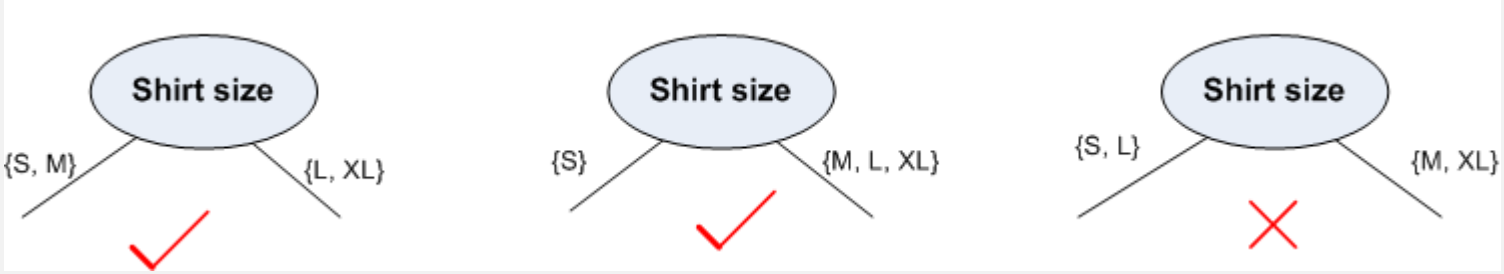


  - Binary splitting by grouping attribute values



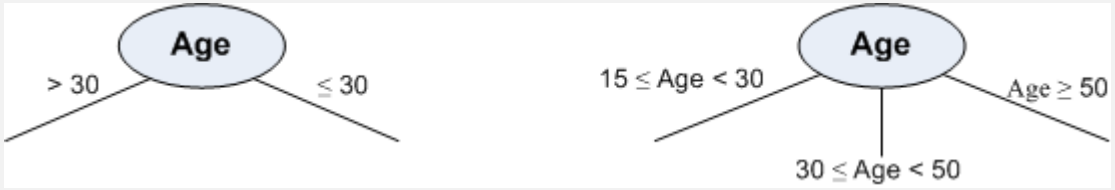
NODE SPLITTING IN **BUILDDT** ALGORITHM

- **Case: Ordinal attribute**
  - It also can be expressed in two ways:
    - A multi-way split
    - A binary split
  - Multi-way split: It is same as in the case of nominal attribute
  - Binary splitting attribute values should be grouped maintaining the order property of the attribute values



NODE SPLITTING IN **BUILDDT** ALGORITHM

- **Case: Numerical attribute**
  - For numeric attribute (with discrete or continuous values), a test condition can be expressed as a comparison set
    - Binary outcome: $A > v$ or $A \leq v$
      - In this case, decision tree induction must consider all possible split positions
    - Range query : $v_i \leq A < v_{i+1}$ for $i = 1, 2, \ldots, q$ (if q number of ranges are chosen)
      - Here, q should be decided a priori



- For a numeric attribute, decision tree induction is a combinatorial optimization problem

ILLUSTRATION : **BUILDDT** ALGORITHM

Example 19.4: Illustration of BuildDT Algorithm

- Consider a training data set as shown.

Person	Gender	Height	Class
1	F	1.6	S
2	M	2.0	M
3	F	1.9	M
4	F	1.88	M
5	F	1.7	S
6	M	1.85	M
7	F	1.6	S
8	M	1.7	S
9	M	2.2	T
10	M	2.1	T
11	F	1.8	M
12	M	1.95	M
13	F	1.9	M
14	F	1.8	M
15	F	1.75	S

Attributes:

Gender = {Male(M), Female (F)} // Binary attribute
Height = {1.5, …, 2.5} // Continuou attribute

Class = {Short (S), Medium (M), Tall (T)}

Given a person, we are to test in which class s/he belongs

ILLUSTRATION : **BUILDDT** ALGORITHM

- To built a decision tree, we can select an attribute in two different orderings: <Gender, Height> or <Height, Gender>
- Further, for each ordering, we can choose different ways of splitting
- Different instances are shown in the following.
- **Approach 1 : <Gender, Height>**

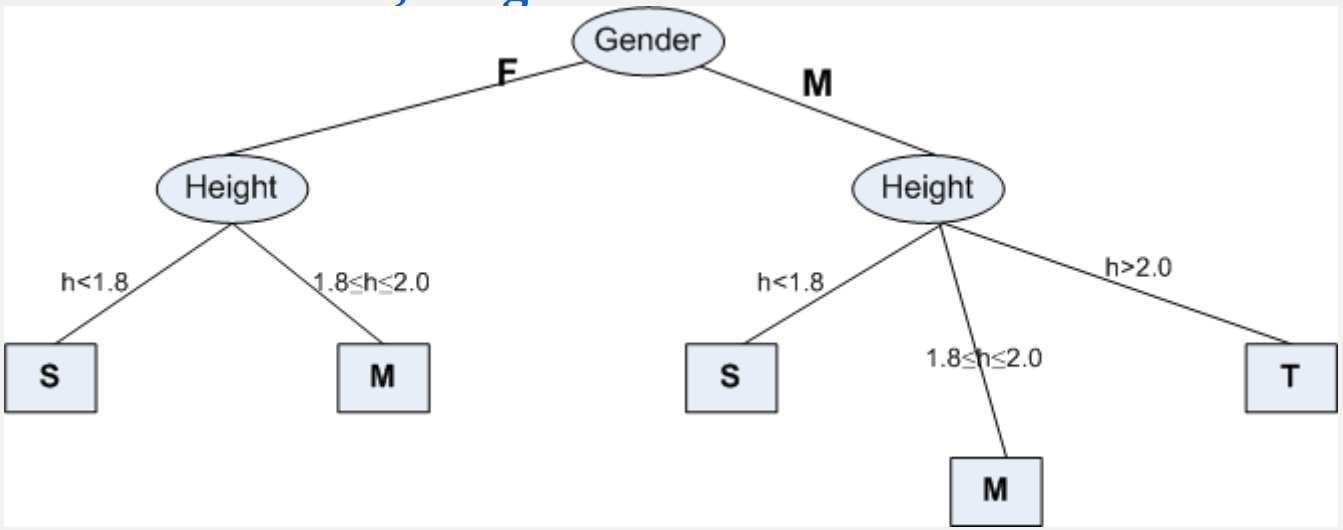


ILLUSTRATION : **BUILDDT** ALGORITHM

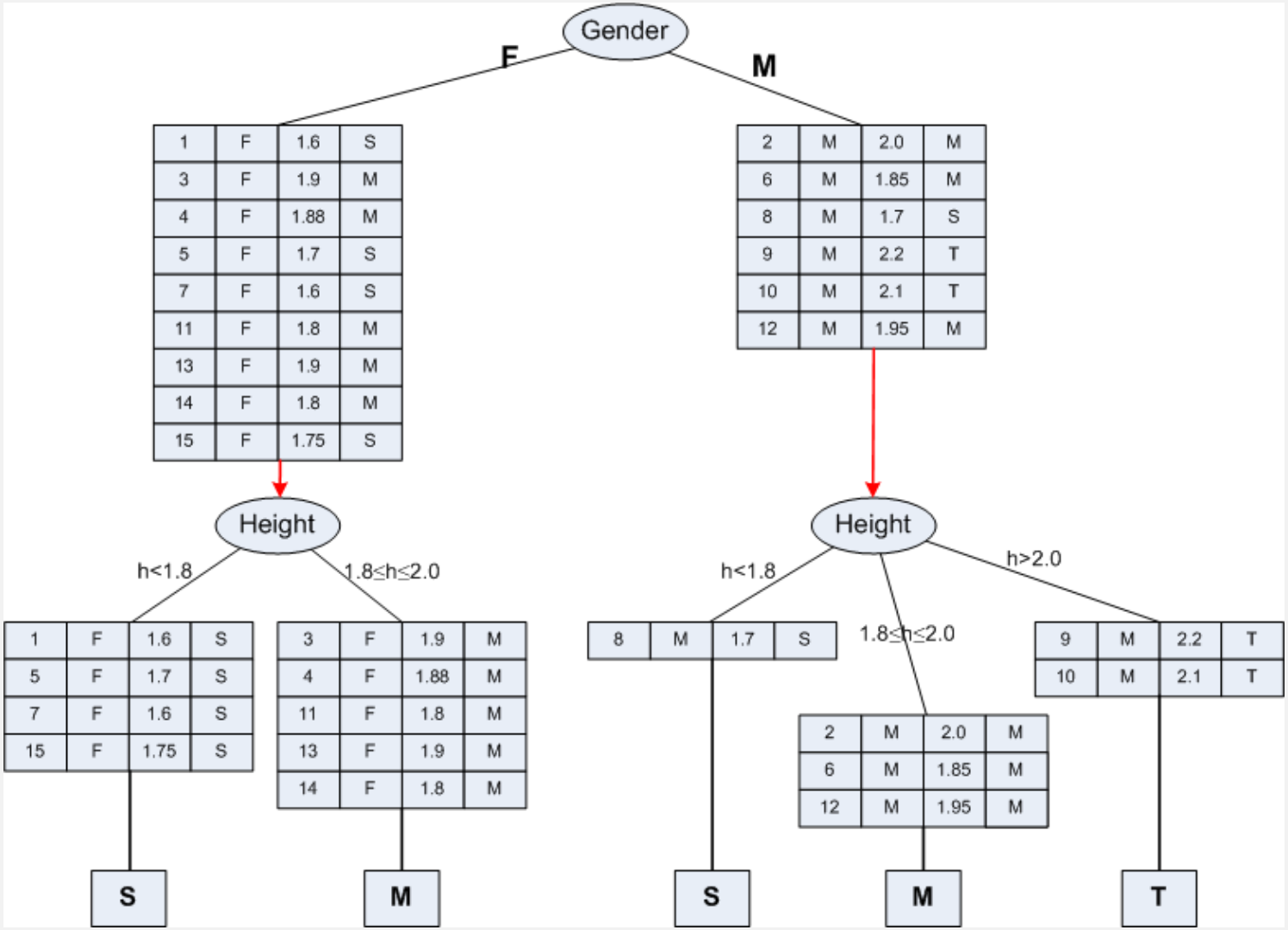
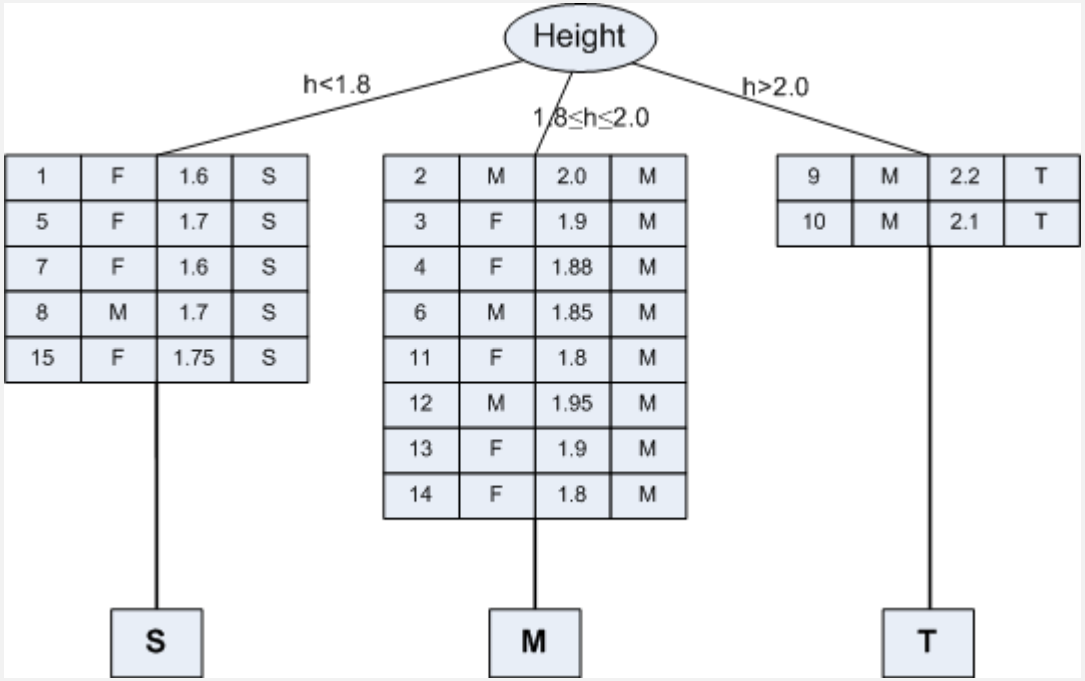


ILLUSTRATION : **BUILDDT** ALGORITHM

- **Approach 2 : <Height, Gender>**



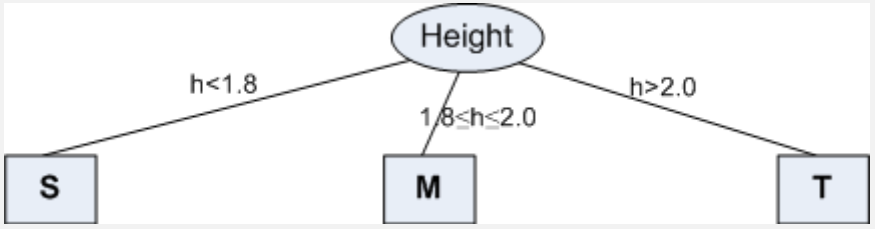


ILLUSTRATION : **BUILDDT** ALGORITHM

Example 19.5: Illustration of BuildDT Algorithm

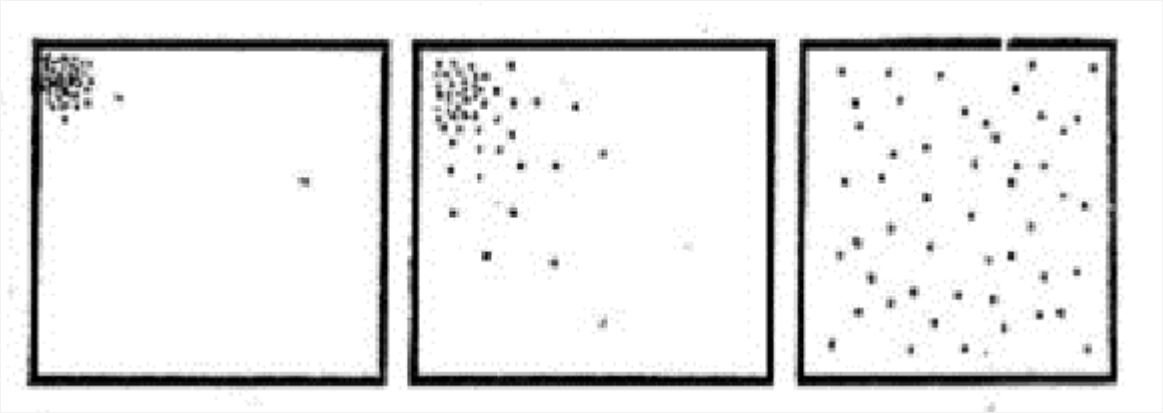
- Consider an anonymous database as shown.

A1	A2	A3	A4	Class
a11	a21	a31	a41	C1
a12	a21	a31	a42	C1
a11	a21	a31	a41	C1
a11	a22	a32	a41	C2
a11	a22	a32	a41	C2
a12	a22	a31	a41	C1
a11	a22	a32	a41	C2
a11	a22	a31	a42	C1
a11	a21	a32	a42	C2
a11	a22	a32	a41	C2
a12	a22	a31	a41	C1
a12	a22	a31	a42	C1

- Is there any “clue” that enables to select the “best” attribute first?
- Suppose, following are two attempts:
  - A1□A2□A3□A4 [naïve]
  - A3□A2□A4□A1 [Random]
- Draw the decision trees in the above-mentioned two cases.
- Are the trees different to classify any test data?
- If any other sample data is added into the database, is that likely to alter the decision tree already obtained?

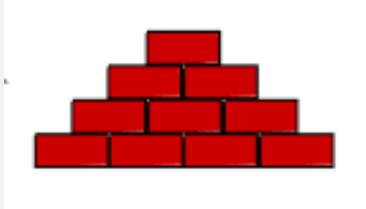
CONCEPT OF ENTROPY

CONCEPT OF ENTROPY

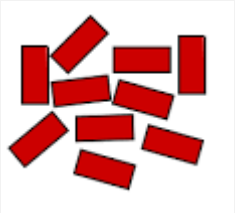


If a point represents a gas molecule, then which system has the more entropy?

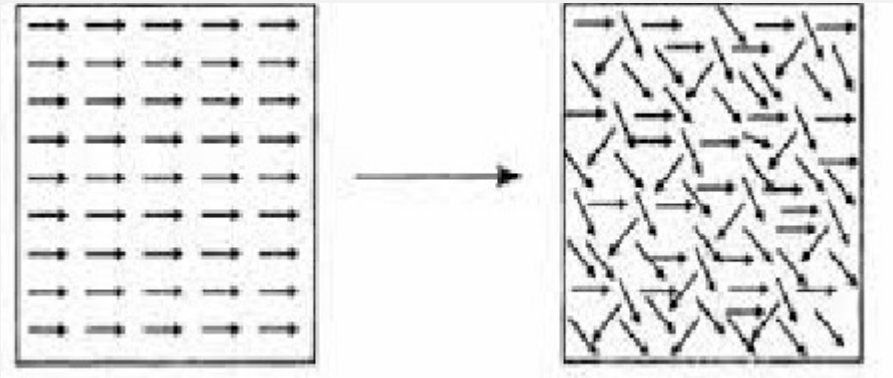
How to measure? ?



More **ordered**
less **entropy**



Less ordered
higher entropy



More organized or
ordered (less **probable**)

Less organized or
disordered (**more** probable)

CONCEPT OF ENTROPY

Universe!

What was its entropy value at its starting point?

ENTROPY AND ITS MEANING

- Entropy is an important concept used in Physics in the context of heat and thereby uncertainty of the states of a matter.
- At a later stage, with the growth of Information Technology, entropy becomes an important concept in Information Theory.
- To deal with the classification job, entropy is an important concept, which is considered as
  - an information-theoretic measure of the “uncertainty” contained in a training data
    - due to the presence of more than one classes.

ENTROPY IN INFORMATION THEORY

- The entropy concept in information theory first time coined by Claude Shannon (1850).
- The first time it was used to measure the “information content” in messages.
- According to his concept of entropy, presently entropy is widely being used as a way of representing messages for efficient transmission by Telecommunication Systems.

MEASURE OF INFORMATION CONTENT

- People, in general, are information hungry!
- Everybody wants to acquire information (from newspaper, library, nature, fellows, etc.)
  - Think how a crime detector do it to know about the crime from crime spot and criminal(s).
  - Kids annoyed their parents asking questions.
- In fact, fundamental thing is that we gather information asking questions (and decision tree induction is no exception).
- We may note that information gathering may be with certainty or uncertainty.

MEASURE OF INFORMATION CONTENT

Example 19.6

a) Guessing a birthday of your classmate

It is with uncertainty ~

Whereas guessing the day of his/her birthday is .

This uncertainty, we may say varies between 0 to 1, both inclusive.

b) As another example, a question related to event with eventuality (or impossibility) will be answered with 0 or 1 uncertainty.

- Does sun rises in the East? (answer is with 0 uncertainty)
- Will mother give birth to male baby? (answer is with ½ uncertainty)
- Is there a planet like earth in the galaxy? (answer is with an extreme uncertainty)

DEFINITION OF ENTROPY

Suppose there are *m* distinct objects, which we want to identify by asking a series of **Yes/No** questions. Further, we assume that *m* is an exact power of 2, say , where .

Definition: **Entropy**

The entropy of a set of *m* distinct values is the minimum number of yes/no questions needed to determine an unknown values from these *m* possibilities.

ENTROPY CALCULATION

- **How can we calculate the minimum number of questions, that is, entropy?**

  - There are two approaches:
    - Brute –force approach
    - Clever approach.

Example 19.7: City quiz

Suppose, Thee is a quiz relating to guess a city out of 8 cities, which are as follows:

Bangalore, Bhopal, Bhubaneshwar, Delhi, Hyderabad, Kolkata, Madras, Mumbai

The question is, “Which city is called **city of joy**”**?**

APPROACH 1: BRUTE-FORCE SEARCH

- Brute force approach
  - We can ask “Is it city *X*?”,
  - if *yes* stop, else ask next …

In this approach, we can ask such questions randomly choosing one city at a time. As a matter of randomness, let us ask the questions, not necessarily in the order, as they are in the list.

Q.1: Is the city Bangalore? No

Q.2: Is the city Bhubaneswar? No

Q.3: Is the city Bhopal? No

Q.4: Is the city Delhi? No

Q.5: Is the city Hyderabad? No

Q.6: Is the city Madras? No

Q.7: Is the city Mumbai? No

No need to ask further question! Answer is already out by the Q.7. If asked randomly, each of these possibilities is equally likely with probability . Hence on the average, we need

questions.

APPROACH 2: CLEVER APPROACH

- Clever approach (binary search)
  - In this approach, we divide the list into two halves, pose a question for a half
  - Repeat the same recursively until we get *yes* answer for the unknown.

Q.1: Is it Bangalore, Bhopal, Bhubaneswar or Delhi? No

Q.2: Is it Madras or Mumbai? No

Q.3: Is it Hyderabad? No

So after fixing 3 questions, we are able to crack the answer.

Note:

Approach 2 is considered to be the best strategy because it will invariably find the answer and will do so with a minimum number of questions on the average than any other strategy.

Approach 1 occasionally do better (when you are lucky enough!)

- It is no coincidence that , and the minimum number of yes/no questions needed is 3.
- If m = 16, then , and we can argue that we need 4 questions to solve the problem. If m = 32, then 5 questions, m = 256, then 8 questions and so on.

REFERENCE

- The detail material related to this lecture can be found in

Data Mining: Concepts and Techniques, (3rd Edn.), Jiawei Han, Micheline Kamber, Morgan Kaufmann, 2015.

Introduction to Data Mining, Pang-Ning Tan, Michael Steinbach, and Vipin Kumar, Addison-Wesley, 2014

Any question?