Practical machine learning

DECISION TREE

DECISION TREE

OUTLINE

- **Basic Concepts of Decision Tree**

- **Build Decision Tree – General Procedure**

- **Information Theory**
  - **Basic concept of entropy in information theory**
  - **Mathematical formulation of entropy**
  - **Calculation of entropy of a training set**

- **Decision Tree induction algorithms**
  - **ID3** (Iterative Dichotomiser 3)
  - **C50**
  - **CART (Classification and Regression Tree) - (Reading Exercise)**

BASIC CONCEPTS & BUILDING OF A DECISION TREE

DECISION TREE - INTRODUCTION

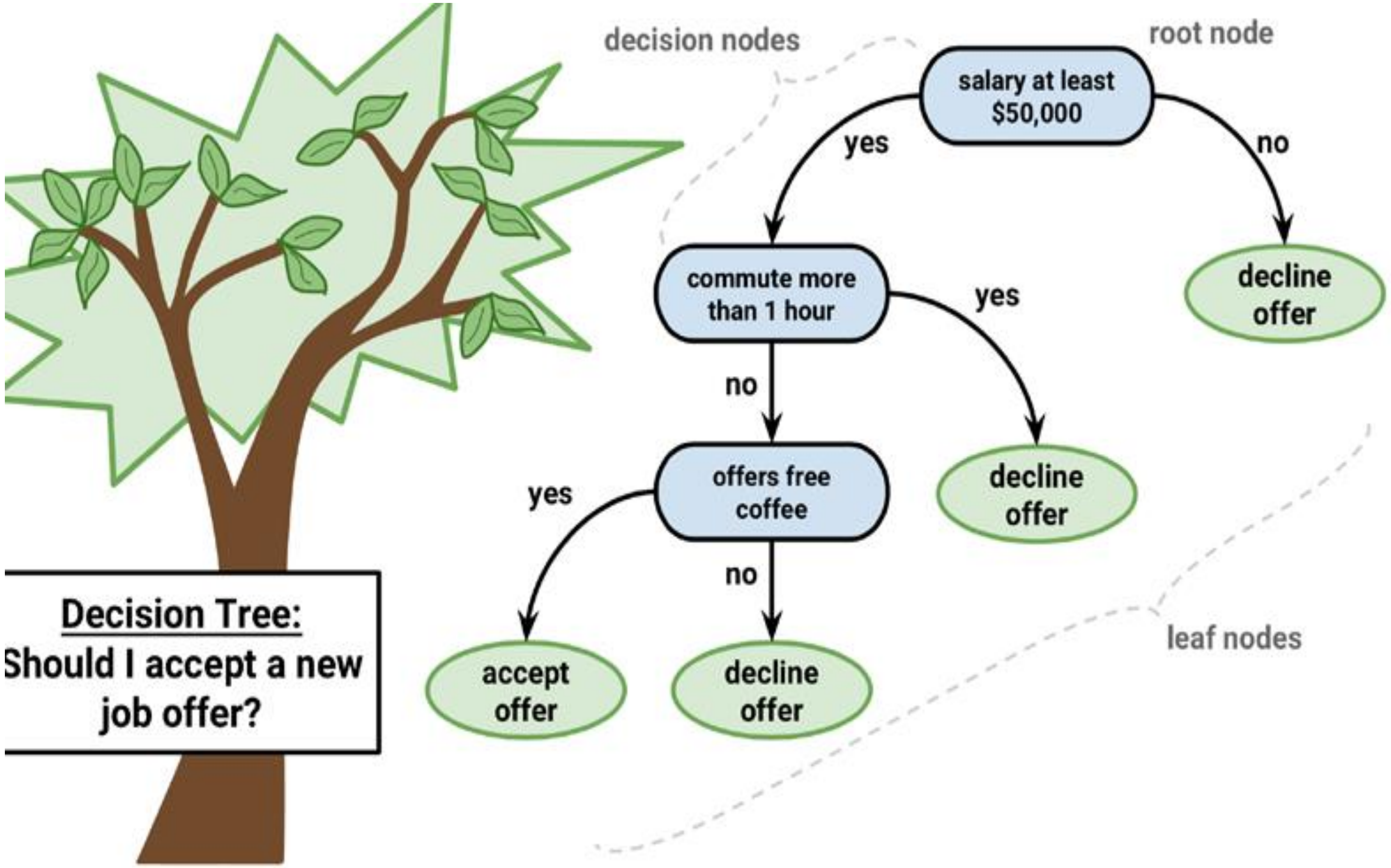
- An emergency room in a hospital measures 20 variables (e.g., blood pressure, age, etc) of newly admitted patients.
- A decision is needed: Whether to put a new patient in an intensive-care unit.
- Due to the high cost of ICU, those patients who may survive less than a month are given higher priority.
- **Problem:** to predict high-risk patients and discriminate them from low-risk patients.

UNDERSTANDING DECISION TREE

"Decision tree learners are powerful classifiers, which utilize a **tree structure** to model the relationships among the features and the potential outcomes"

- Its classification accuracy is competitive with other methods

- The classification model is a tree, called decision tree.
- C5.0 (Latest and used by industry) by Ross Quinlan is perhaps the best known system.

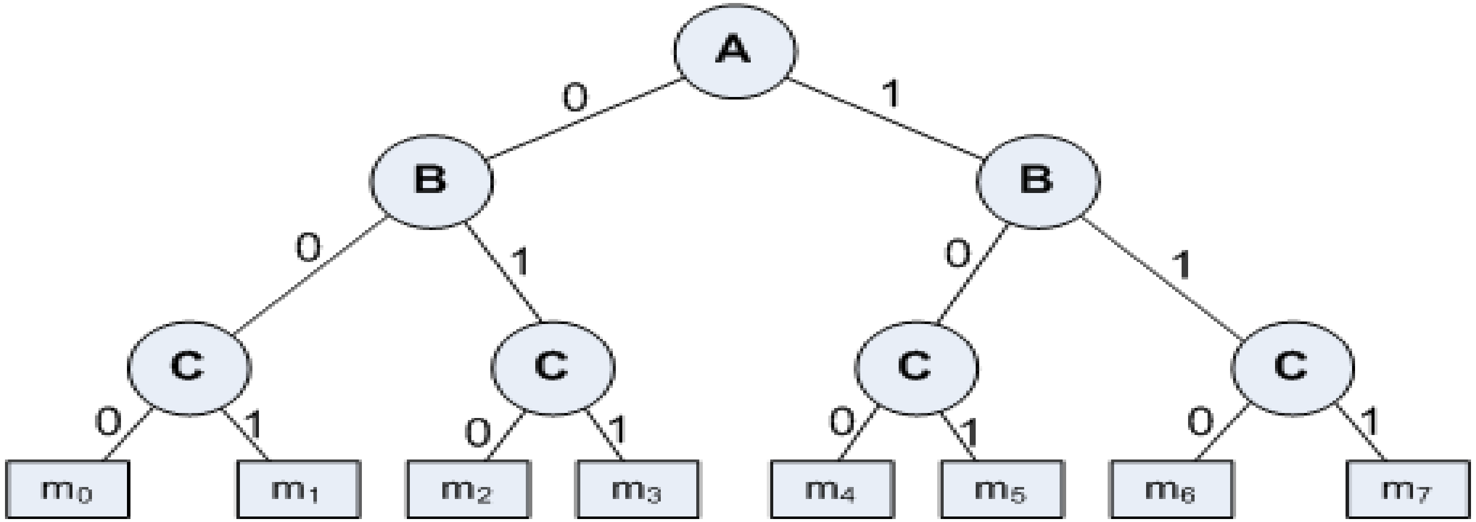


BASIC CONCEPTS – HOW TO CONSTRUCT A TREE?

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

Example: Binary Decision Tree with binary data or categorical data

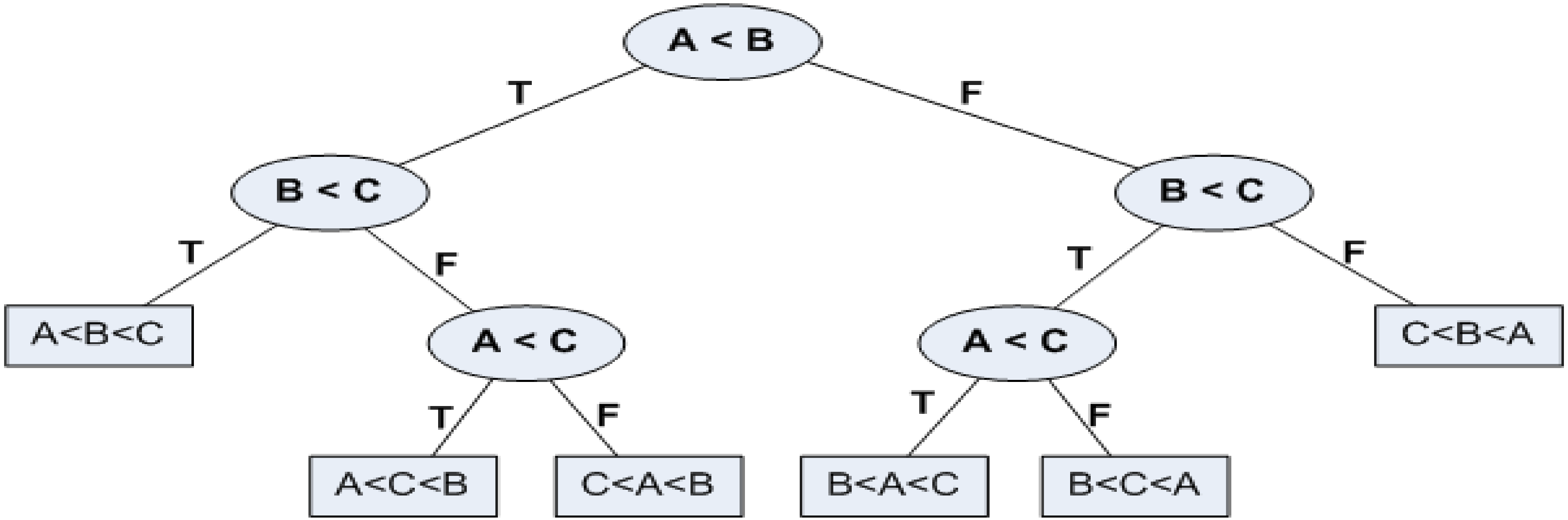
A	*B*	*C*	*f*
0	0	0	m_0
0	0	1	m_1
0	1	0	m_2
0	1	1	m_3
1	0	0	m_4
1	0	1	m_5
1	1	0	m_6
1	1	1	m_7



HOW TO CONSTRUCT A TREE?

- Decision tree is also possible where attributes are of continuous data type

Example: Decision Tree with numeric data



DEFINITION OF DECISION TREE

Decision Tree

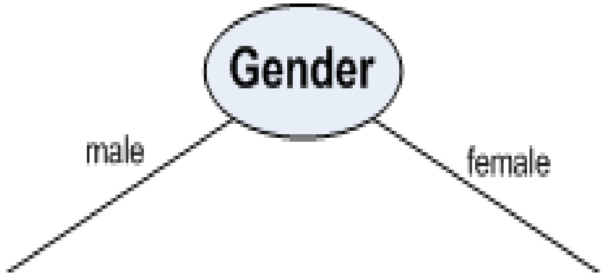
Given a dataset $D = \{t_1, t_2, \ldots\ldots, t_n\}$, where t_i denotes a tuple, which is defined by a set of attribute $A = \{A_1, A_2, \ldots\ldots, A_m\}$. Also, given a set of classes $C = \{c_1, c_2, \ldots\ldots, c_k\}$.

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

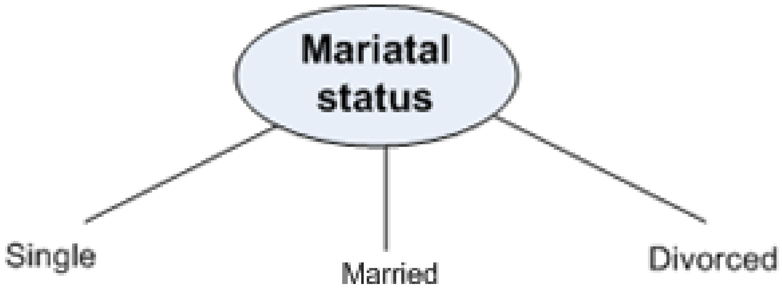
- Each **internal node** is an attribute A_i
- Each **edge** is a predicate that can be applied to the attribute associated with the parent node of it
- Each leaf node is labeled with class c_j

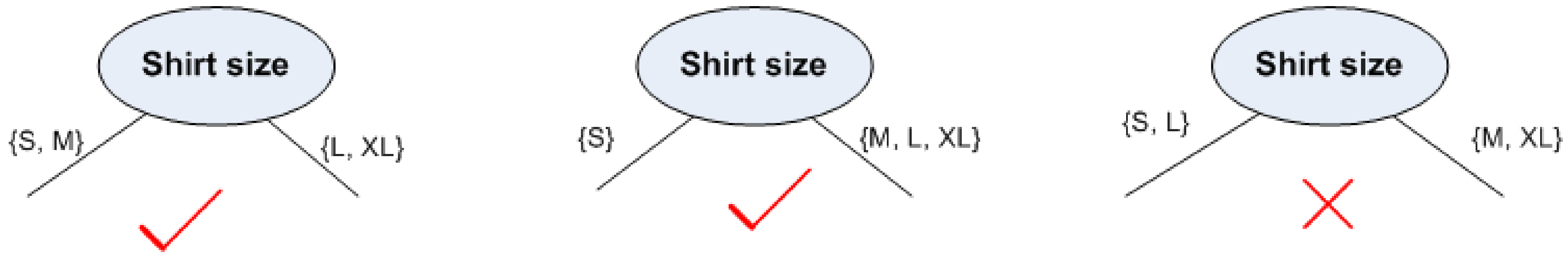
NODE SPLITTING

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



- **Case: Binary attribute & Ordinal Attribute**
  - This is the simplest case of node splitting
  - The test condition for a binary attribute generates only two outcomes
  - We can group the values of an attribute properly to split.
  - Multiway split is required if an attribute has multiple categories.





NODE SPLITTING…

- **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

ILLUSTRATION WITH SAMPLE DATA

Example : Illustration of Building a decision tree

- 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} // Continuous attribute

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

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

ILLUSTRATION WITH SAMPLE DATA

- To build 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>**

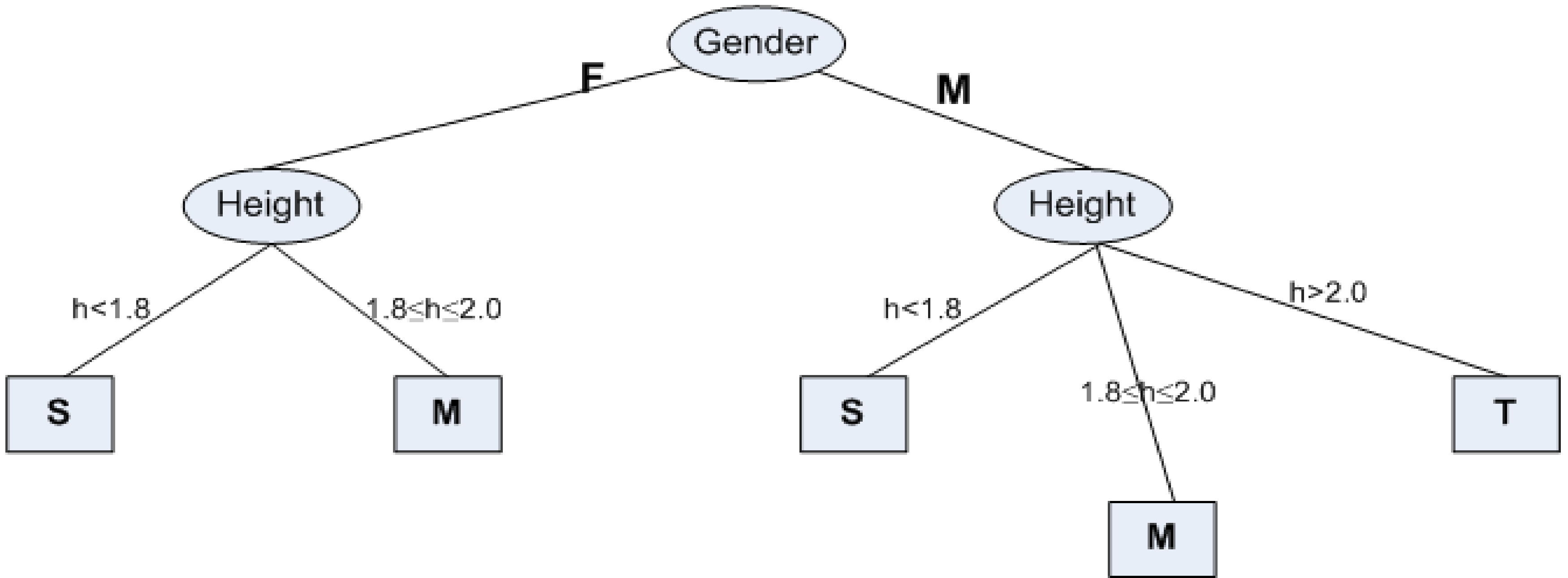


ILLUSTRATION WITH SAMPLE DATA

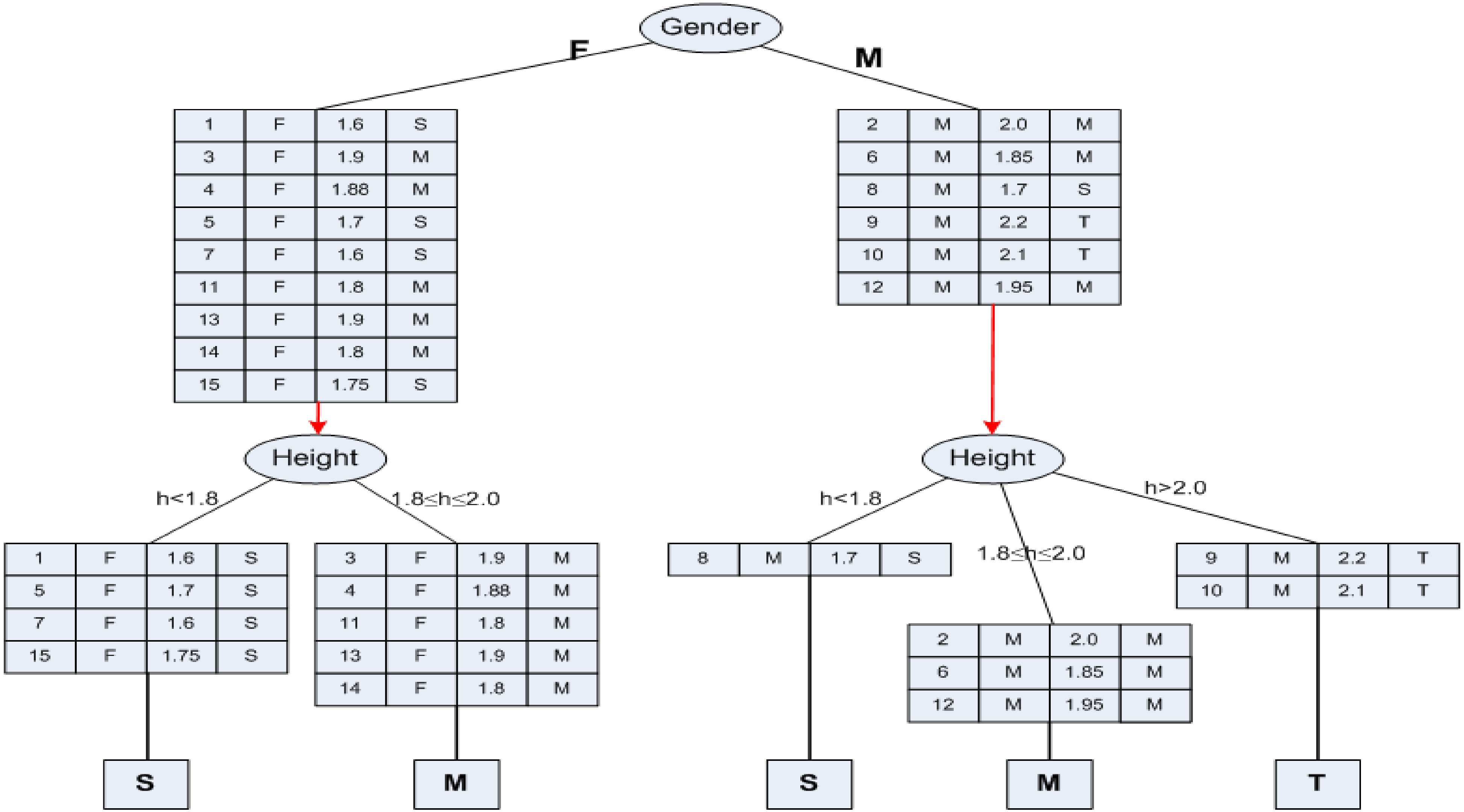
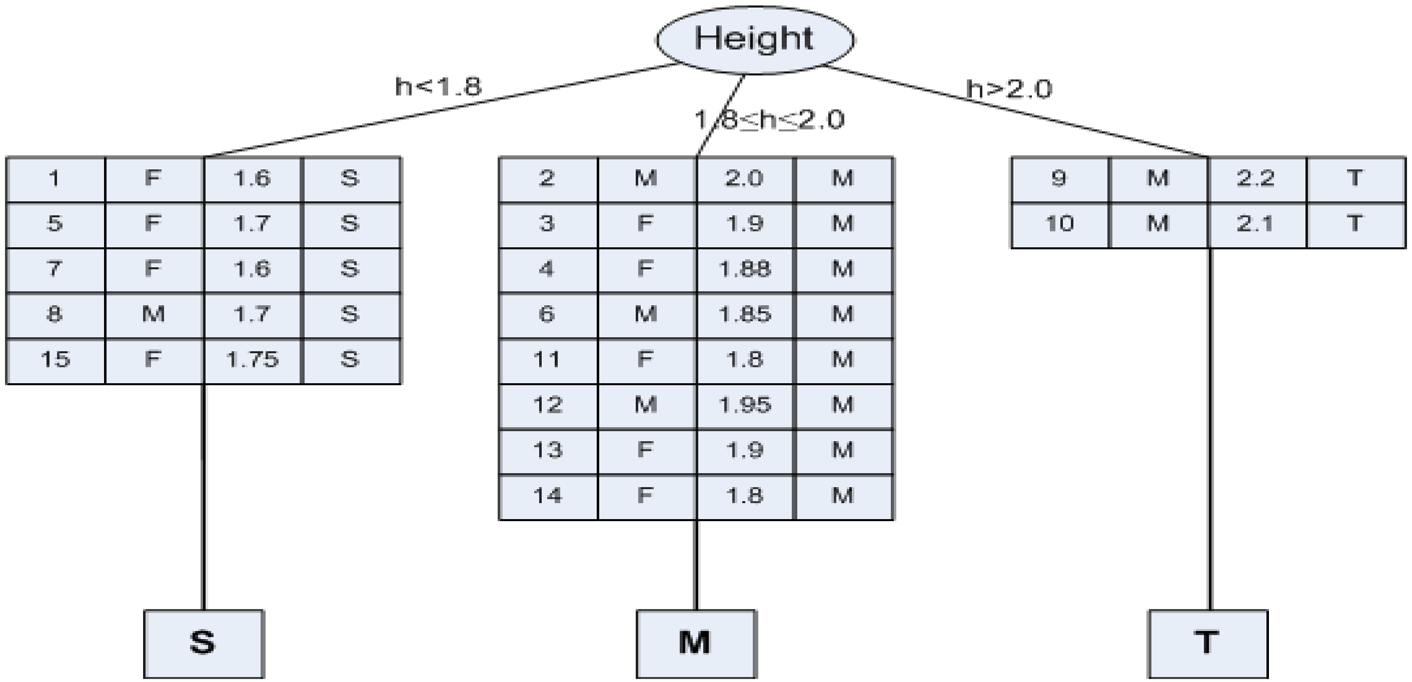


ILLUSTRATION WITH SAMPLE DATA

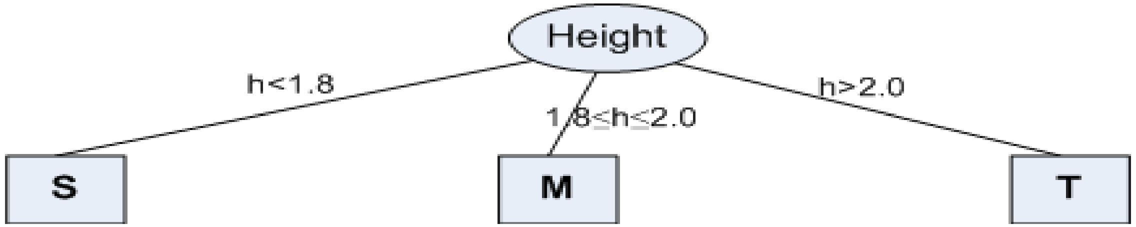
- **Approach 2 : <Height, Gender>**



1	F	1.6	S
5	F	1.7	S
7	F	1.6	S
8	M	1.7	S
15	F	1.75	S

2	M	2.0	M
3	F	1.9	M
4	F	1.88	M
6	M	1.85	M
11	F	1.8	M
12	M	1.95	M
13	F	1.9	M
14	F	1.8	M

9	M	2.2	T
10	M	2.1	T



Which approach we shall consider?

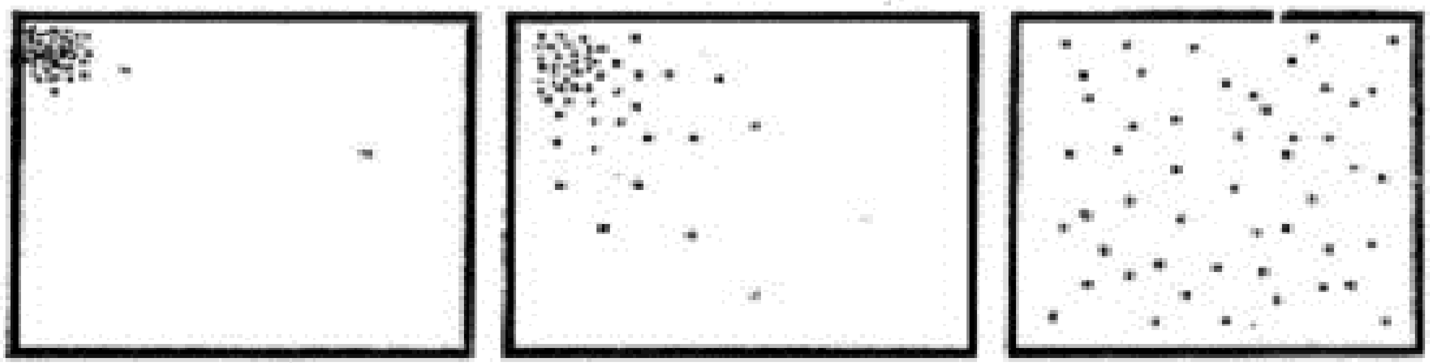
Answer:

Compute Entropy

Let us learn Information Theory

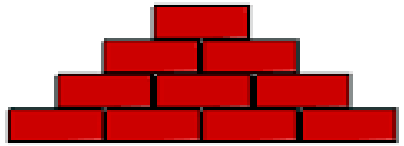
BASIC CONCEPTS OF ENTROPY

CONCEPT OF ENTROPY

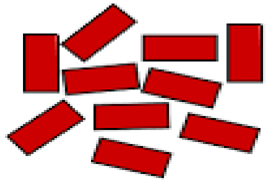


If a point represents a gas molecule, then which system has the more entropy?

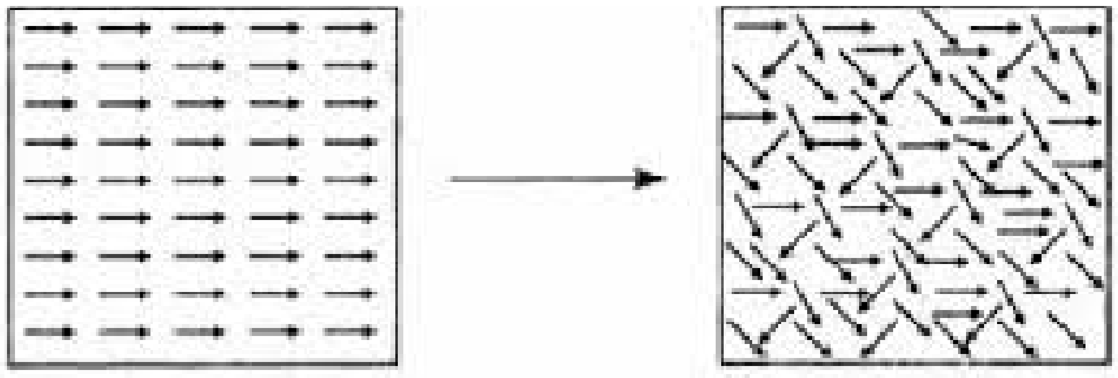
How to measure? $\Delta S = \frac{\Delta Q}{T}$?



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



Less ordered
higher entropy



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

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

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
  - **Entropy** - "an information-theoretic **"measure of uncertainty"** contained in a training data.

ENTROPY IN INFORMATION THEORY

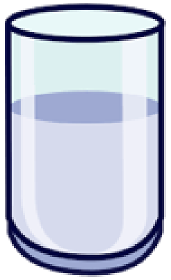
- The entropy concept in information theory was first time coined by Claude Shannon (1850).
- The first time it was used to **measure the "information content" in messages**.
- According to his concept of entropy, presently entropy is widely being used as a way of representing messages for efficient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

"A Mathematical Theory of Communication", 1948, **Claude Shannon** introduced the revolutionary notion of *Information Entropy*.

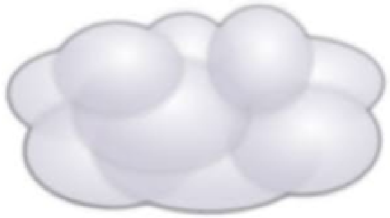
Entropy In Information Theory

- if the particles inside a system have:
  - many possible positions to move around, then the system has → **high entropy**,
  - if they have to stay rigid, then the system has → **low entropy**.

ICE
Low Entropy



WATER
Medium Entropy



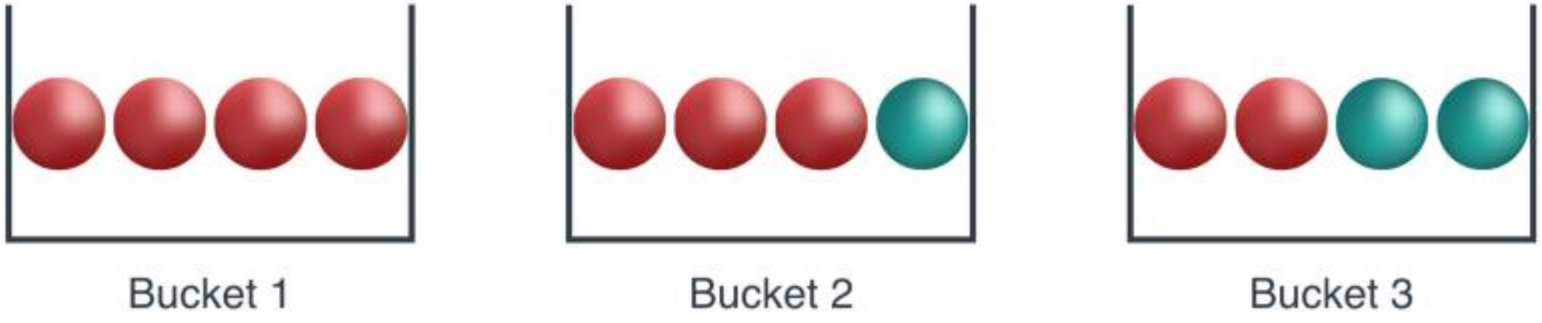
WATER VAPOR
High Entropy

Entropy in Information Theory

Entropy and Information

Let's say we have 3 buckets with 4 balls each. The balls have the following colors:

- Bucket 1: 4 red balls
- Bucket 2: 3 red balls, and 1 blue ball
- Bucket 3: 2 red balls, and 2 blue balls

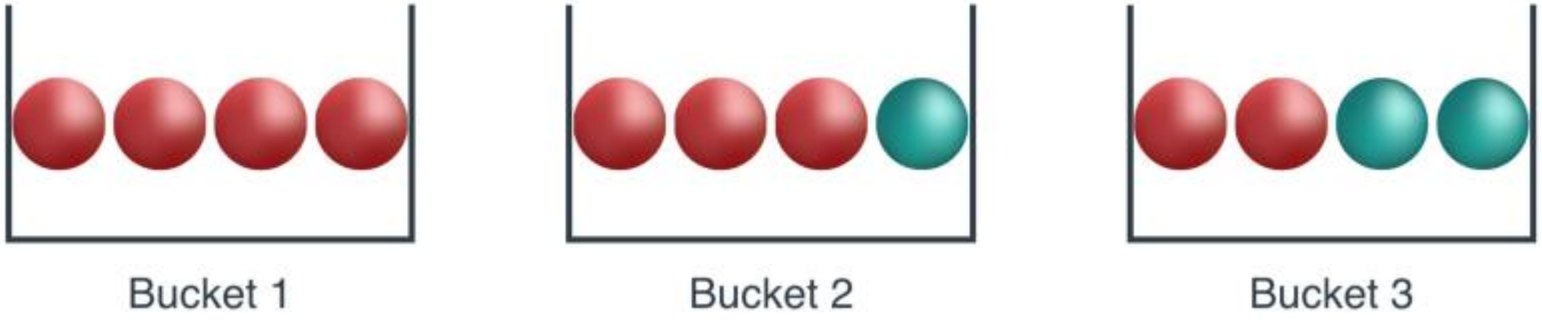


How much information we have on the color of a ball drawn at random?

Entropy in Information Theory

Entropy and Information

- In the **first bucket**, we'll know for sure that the **ball coming out is red**.
- In the **second bucket**, we know with **75% certainty that the ball is red**, and with 25% certainty that it's blue.
- In the **third bucket**, we know with **50% certainty that the ball is red**, and with the same certainty that it's blue.



How much information we have on the color of a ball drawn at random?

Entropy in Information Theory

Entropy and Information

- Bucket 1 gives us the most amount of "knowledge" about what ball we'll draw (because we know for sure it's red), that Bucket 2 gives us some knowledge, and that Bucket 3 will give us the least amount of knowledge.
- **Entropy is in some way, the opposite of knowledge.** So we'll say that Bucket 1 has the least amount of entropy, Bucket 2 has medium entropy, and Bucket 3 has the greatest amount of entropy.

How much information we have on the color of a ball drawn at random?

Definition Of Entropy

Definition: Entropy

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

Entropy Calculation - Example

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

  - There are two approaches:
    - Brute –force approach
    - **Entropy Approach**

Example : Answer to the City Quiz

Suppose, There 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 **as City of Joy**”**?**

Entropy Calculation - 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 – It is Kolkata

Note:

Approach 2is 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.

- It is no coincidence that $8 = 2^3$, and the minimum number of yes/no questions needed is 3.
- If m = 16, then $16 = 2^4$, 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.

Entropy Calculation

Entropy calculation

The minimum number of *yes/no* questions needed to identify an unknown object from $m = 2^n$ equally likely possible object is n.

If m is not a power of 2, then the entropy of a set of m distinct objects that are equally likely is $\boldsymbol{\log_2 m}$

Entropy In Messages

- **In this point, we can note that to identify an object, if it is encoded with bits, then we have to ask questions in an alternative way. For example**

  - **Is the first bit 0?**
  - **Is the second bit 0?**
  - **Is the third bit 0?** **and so on**

- **Thus, we need *n* questions, if *m* objects are there such that $m = 2^n$.**

- The above leads to (an alternative) and equivalent definition of entropy

> Definition: **Entropy**
>
> The entropy of a set of *m* distinct values is the number of bits needed to encode all the values in the most efficient way.

Entropy Of A Training Set - Example

Example: OPTHAL dataset

Consider the OTPHAL data shown in the following table with total 24 instances in it.

Age	Eye sight	Astigmatic	Use Type	Class
1	1	1	1	3
1	1	1	2	2
1	1	2	1	3
1	1	2	2	1
1	2	1	1	3
1	2	1	2	2
1	2	2	1	3
1	2	2	2	1
2	1	1	1	3
2	1	1	2	2
2	1	2	1	3
2	1	2	2	1
2	2	1	1	3
2	2	1	2	2
2	2	2	1	3
2	2	2	2	3
3	1	1	1	3
3	1	1	2	3
3	1	2	1	3
3	1	2	2	1
3	2	1	1	3
3	2	1	2	2
3	2	2	1	3
3	2	2	2	3

A coded forms for all values of attributes are used to avoid the cluttering in the table.

WHEN M != 2^N ?

Entropy Calculation (General Formula)

The entropy of a set of m distinct objects is $\log_2 m$ even when m is not exactly a power of 2.

Till now, we are assuming that all m objects are equally probable.

What if all *m*objects are not equally probable?

Suppose, p_i denotes the frequency with which the i^{th} object of m objects occurs, where $0 \le p_i \le 1$ for all p_i such that

$$\sum_{i=1}^{m} p_i = 1$$

INFORMATION CONTENT

Based on the previous discussion we can easily prove the following lemma.

Information content

If an object occurs with frequency p, then the most efficient way to represent it with $\log_2(^1/_p)$ bits.

Example: Information content

- A which occurs with frequency $\frac{1}{2}$ is represented by 1-bit,
- B which occurs with frequency $\frac{1}{4}$ represented by 2-bits
- C and D which occurs with frequency $\frac{1}{8}$ are represented by 3 bits each.

ENTROPY CALCULATION

We can generalize the above understanding as follows.

- If there are m objects with frequencies $p_1, p_2 \ldots\ldots, p_m$, thenthe average number of bits (i.e. questions) that need to be examined a value, that is, **entropy is the frequency of occurrence of the i^{th} value multiplied by the number of bits that need to be determined, summed up values of i from *1* to m.**

Entropy calculation

If p_i denotes the frequencies of occurrences of m distinct objects, then the entropy E is

$$\boldsymbol{E = \sum_{i=1}^{m} p_i \log\left({}^{1}/_{p_i}\right) \ and \sum_{i=1}^{m} p_i = 1}$$

Note:

- If all are equally likely, then $p_i = \frac{1}{m}$ and $E = \log_2 m$; it is the special case.

ENTROPY OF A TRAINING SET

- If there are k classes $c_1, c_2 \ldots\ldots, c_k$ and p_i for $i = 1\ to\ k$ denotes the number of occurrences of classes c_i divided by the total number of instances (i.e., the frequency of occurrence of c_i) in the training set, then entropy of the training set is denoted by

$$\boxed{E = -\sum_{i=1}^{m} p_i \log_2 p_i}$$

Here, E is measured in "bits" of information.

Note:

- The above formula should be summed over the non-empty classes only, that is, classes for which $p_i \neq 0$
- E is always a positive quantity
- E takes it's minimum value (zero) if and only if all the instances have the same class (i.e., the training set with only **one** non-empty class, for which the probability 1).
- Entropy takes its maximum value when the instances are equally distributed among k possible classes. In this case, the maximum value of E is $log_2\, k$.

Entropy Of A Training Set - Example

Example: OPTHAL dataset

Consider the OTPHAL data shown in the following table with total 24 instances in it.

Age	Eye sight	Astigmatic	Use Type	Class
1	1	1	1	3
1	1	1	2	2
1	1	2	1	3
1	1	2	2	1
1	2	1	1	3
1	2	1	2	2
1	2	2	1	3
1	2	2	2	1
2	1	1	1	3
2	1	1	2	2
2	1	2	1	3
2	1	2	2	1
2	2	1	1	3
2	2	1	2	2
2	2	2	1	3
2	2	2	2	3
3	1	1	1	3
3	1	1	2	3
3	1	2	1	3
3	1	1	2	1
3	2	1	1	3
3	2	2	2	2
3	2	2	1	3
3	2	2	2	3

A coded forms for all values of attributes are used to avoid the cluttering in the table.

ENTROPY OF A TRAINING SET…

Age	Eye Sight	Astigmatic	Use Type
1: Young	1: Myopia	1: No	1: Frequent
2: Middle-aged	2: Hypermetropia	2: Yes	2: Less
3: Old			

ENTROPY OF A TRAINING SET…

Age	Eye Sight	Astigmatic	Use Type
1: Young	1: Myopia	1: No	1: Frequent
2: Middle-aged	2: Hypermetropia	2: Yes	2: Less
3: Old			

Class: 1: Contact Lens 2:Normal glass 3: Nothing

In the OPTH database, there are 3 classes and 4 instances with class 1, 5 instances with class 2 and 15 instances with class 3. Hence, entropy *E* of the database is:

$$E = -\frac{4}{24}\log_2\frac{4}{24} - \frac{5}{24}\log_2\frac{5}{24} - \frac{15}{24}\log_2\frac{15}{24} = 1.3261$$

Class 1 – Contact Lens – 4 people
Class 2 – Normal Glass – 5 People
Class 3 – Nothing – 15 People

Coming back to Decision Tree…

DECISION TREE INDUCTION TECHNIQUES

- Decision tree induction is a top-down, recursive and divide-and-conquer approach.

- The procedure is to choose an attribute and split it into from a larger training set into smaller training sets.

- Different algorithms have been proposed to take a good control over

  1. Choosing the best attribute to be splitted, and

  2. Splitting criteria

- Several algorithms have been proposed for the above tasks. In this lecture, we shall limit our discussions into three important of them
  - **ID3**
  - **C 4.5 / C5.0**
  - **CART**

Algorithm1: ID3 – Iterative Dichotomizer 3

ID3

- Quinlan [1986] introduced the ID3, a popular short form of **I**terative **D**ichotomizer 3 for decision trees from a set of training data.

- In ID3, each node corresponds to a splitting attribute and each arc is a possible value of that attribute.

- At each node, the splitting attribute is selected to be the most informative (Entropy) among the attributes not yet considered in the path starting from the root.

ID3…

- In ID3, entropy is used to measure how informative a node is.

- ID3 algorithm defines a measurement of a splitting called **Information Gain** to determine the goodness of a split.
  - The attribute with the largest value of information gain is chosen as the splitting attribute and
  - it partitions into a number of smaller training sets based on the distinct values of attribute under split.

DEFINING INFORMATION GAIN

- We consider the following symbols and terminologies to define information gain, which is denoted as α.
  - $D\equiv$ denotes the training set at any instant
  - $|D|\equiv$ denotes the size of the training set D
  - $E(D)\equiv$ denotes the entropy of the training set D
- The entropy of the training set D

$$E(D) = -\sum_{i=1}^{k} p_i \log_2(p_i)$$

  - where the training set D has $c_1, c_2, \dots, c_k$, the k number of distinct classes and
  - p_i, $0 < p_i \leq 1$ is the probability that an arbitrary tuple in D belongs to class c_i ($i = 1, 2, \dots, k$).

Defining Information Gain

Definition: **Weighted Entropy**

The weighted entropy denoted as $E_A(D)$ for all partitions of D with respect to A is given by:

$$E_A(D) = \sum_{j=1}^{m} \frac{|D_j|}{|D|} E(D_j)$$

Here, the term $\frac{|D_j|}{|D|}$ denotes the weight of the j-th training set.

More meaningfully, $E_A(D)$ is the expected information required to classify a tuple from D based on the splitting of A.

Defining Information Gain…

- Our objective is to take A on splitting to produce an exact classification (also called pure), that is, all tuples belong to one class.
- However, it is quite likely that the partitions is impure, that is, they contain tuples from two or more classes.
- In that sense, $E_A(D)$is a measure of impurities (or purity). A lesser value of $E_A(D)$implying more power the partitions are.

Definition 9.5: **Information Gain**

Information gain, $\alpha(A, D)$ of the training set D splitting on the attribute A is given by

$$\boldsymbol{\alpha(A, D) = E(D) - E_A(D)}$$

In other words, $\alpha(A, D)$ **gives us an estimation how much would be gained by splitting on** A. The attribute A with the highest value of α should be chosen as the splitting attribute for D.

ID3 – Example1 – OPTH Dataset

Entropy Of A Training Set - Example

Example: OPTHAL dataset

Consider the OTPHAL data shown in the following table with total 24 instances in it.

Age	Eye sight	Astigmatic	Use Type	Class
1	1	1	1	3
1	1	1	2	2
1	1	2	1	3
1	1	2	2	1
1	2	1	1	3
1	2	1	2	2
1	2	2	1	3
1	2	2	2	1
2	1	1	1	3
2	1	1	2	2
2	1	2	1	3
2	1	2	2	1
2	2	1	1	3
2	2	1	2	2
2	2	2	1	3
2	2	2	2	3
3	1	1	1	3
3	1	1	2	3
3	1	2	1	3
3	1	2	2	1
3	2	1	1	3
3	2	1	2	2
3	2	2	1	3
3	2	2	2	3

A coded forms for all values of attributes are used to avoid the cluttering in the table.

Information Gain Calculation

Example: Information gain on splitting OPTH

- Let us refer to the OPTH database discussed .
- Splitting on **Age** at the root level, it would give three subsets D_1, D_2 and D_3 as shown in the tables in the following three slides.
- The entropy $E(D_1), E(D_2)$ and $E(D_3)$ of training sets D_1, D_2 and D_3 and corresponding weighted entropy $E_{Age}(D_1), E_{Age}(D_2)$ and $E_{Age}(D_3)$ are also shown alongside.
- The Information gain $\alpha\ (Age, OPTH)$ is then can be calculated as **0.0394**.
- Recall that entropy of OPTH data set, we have calculated as *E(OPTH)* = **1.3261**

Information Gain Calculation

Example 9.11 : Information gain on splitting OPTH

Age	Eye-sight	Astigmatism	Use type	Class
1	1	1	1	3
1	1	1	2	2
1	1	2	1	3
1	1	2	2	1
1	2	1	1	3
1	2	1	2	2
1	2	2	1	3
1	2	2	2	1

$$E(D_1) = -\frac{2}{8} log_2(\frac{2}{8}) - \frac{2}{8} log_2(\frac{2}{8}) - \frac{4}{8} log_2(\frac{4}{8}) = \mathbf{1.5}$$

$$E_{Age}(D_1) = \frac{8}{24} \times 1.5 = \mathbf{0.5000}$$

Training set: D_1(Age = 1)

Calculating Information Gain

Training set: D_2(Age = 2)

Age	Eye-sight	Astigmatism	Use type	Class
2	1	1	1	3
2	1	1	2	2
2	1	2	1	3
2	1	2	2	1
2	2	1	1	3
2	2	1	2	2
2	2	2	1	3
2	2	2	2	3

$$E(D_2) = -\frac{1}{8}log_2(\frac{1}{8}) - \frac{2}{8}log_2(\frac{2}{8}) - \frac{5}{8}log_2(\frac{5}{8})$$
$$= \mathbf{1.2988}$$

$$E_{Age}(D_2) = \frac{8}{24} \times 1.2988 = \mathbf{0.4329}$$

Calculating Information Gain

Training set: D_3(Age = 3)

Age	Eye-sight	Astigmatism	Use type	Class
3	1	1	1	3
3	1	1	2	3
3	1	2	1	3
3	1	2	2	1
3	2	1	1	3
3	2	1	2	2
3	2	2	1	3
3	2	2	2	3

$$E(D_3) = -\frac{1}{8}log_2(\frac{1}{8}) - \frac{1}{8}log_2(\frac{1}{8}) - \frac{6}{8}log_2(\frac{6}{8}) = \mathbf{1.0613}$$

$$E_{Age}(D_3) = \frac{8}{24} \times 1.0613 = \mathbf{0.3504}$$

$$\boldsymbol{\alpha\ (Age, D) = 1.3261 - (0.5000 + 0.4329 + 0.3504) = 0.0394}$$

INFORMATION GAIN FOR DIFFERENT ATTRIBUTES…

- In the same way, we can calculate the information gains, when splitting the OPTH database on Eye-sight, Astigmatic and Use Type. The results are summarized below.
- Splitting attribute: Age

$$\alpha(Age, OPTH) = 0.0394$$

- Splitting attribute: Eye-sight

$$\alpha(Eye - sight, OPTH) = 0.0395$$

- Splitting attribute: Astigmatic

$$\alpha(\text{Astigmatic}, OPTH) = 0.3770$$

- Splitting attribute: Use Type

$$\alpha(\text{Use Type}, OPTH) = 0.5488$$

Decision Tree Induction : ID3 Way

- The ID3 strategy of attribute selection is to choose to split on the attribute that gives the greatest reduction in the weighted average entropy
  - The one that maximizes the value of information gain
- In the example with OPTH database, the larger values of information gain is $\alpha(\text{Use Type}, OPTH) = 0.5488$
  - Hence, the attribute should be chosen for splitting is "Use Type".
- The process of splitting on nodes is repeated for each branch of the evolving decision tree, and the final tree, which would look like is shown in the following slide and calculation is left for practice.

DECISION TREE INDUCTION: ID3 WAY

$OPTH$

Age | Eye-sight | Use Type | Astigmatic

$\alpha = 0.0394$

$D1$ $Frequent(1)$

$E(D_1) = ?$

Age	Eye	Ast	Use	Class
1	1	1	1	3
1	1	2	1	3
1	2	1	1	3
1	2	2	1	3
2	1	1	1	3
2	2	1	1	3
2	2	2	1	3
3	1	1	1	3
3	1	2	1	3
3	2	1	1	3
3	2	2	1	3

Age | Eye-sight | Astigmatic

$D2$ $Less(2)$

$E(D_2) = ?$

Age	Eye	Ast	Use	Class
1	1	1	2	2
1	1	2	2	1
1	2	1	2	2
1	2	2	2	1
2	1	1	2	2
2	1	2	2	1
2	2	1	2	2
3	1	1	2	3
3	1	2	2	3
3	2	1	2	2
3	2	2	2	3

Age | Eye-sight | Astigmatic

ID3 – Example2 – Loan Application Data Set

AN EXAMPLE

- Data: Loan application data
- Task: Predict whether a loan should be approved or not.
- Performance measure: Accuracy.

No learning: classify all future applications (test data) to the majority class (i.e., Yes):

Accuracy = 9/15 = 60%.

- We can do better than 60% with learning.

EXAMPLE 2 – LOAN APPROVAL PREDICTION

Loan Application (Class: Approved or not - Yes or No)

ID	Age	Has_JOB	Own_House	Credit_Rating	Class
1	Young	False	False	Fair	No
2	Young	False	False	Good	No
3	Young	True	False	Good	Yes
4	Young	True	True	Fair	Yes
5	Young	False	False	Fair	No
6	Middle	False	False	Fair	No
7	Middle	False	False	Good	No
8	Middle	True	True	Good	Yes
9	Middle	False	True	Excellent	Yes
10	Middle	False	True	Excellent	Yes
11	Old	False	True	Excellent	Yes
12	Old	False	True	Good	Yes
13	Old	True	False	Good	Yes
14	Old	True	False	Excellent	Yes
15	Old	False	False	Fair	No

Age:

Young, Middle, Old

Has_Job:

True, False

Own_House:

True, False

Credit Rating:

Fair, Good, Excellent

CHOOSE AN ATTRIBUTE TO PARTITION DATA - BEST SPLIT

- The *key* to building a decision tree - **which attribute to choose in order to branch.?**
- The objective is to **reduce impurity** or uncertainty in data as much as possible.
  - A subset of data is pure if all instances belong to the same class.
- The *heuristic* in ID3 is to **choose the attribute with the maximum Information Gain**

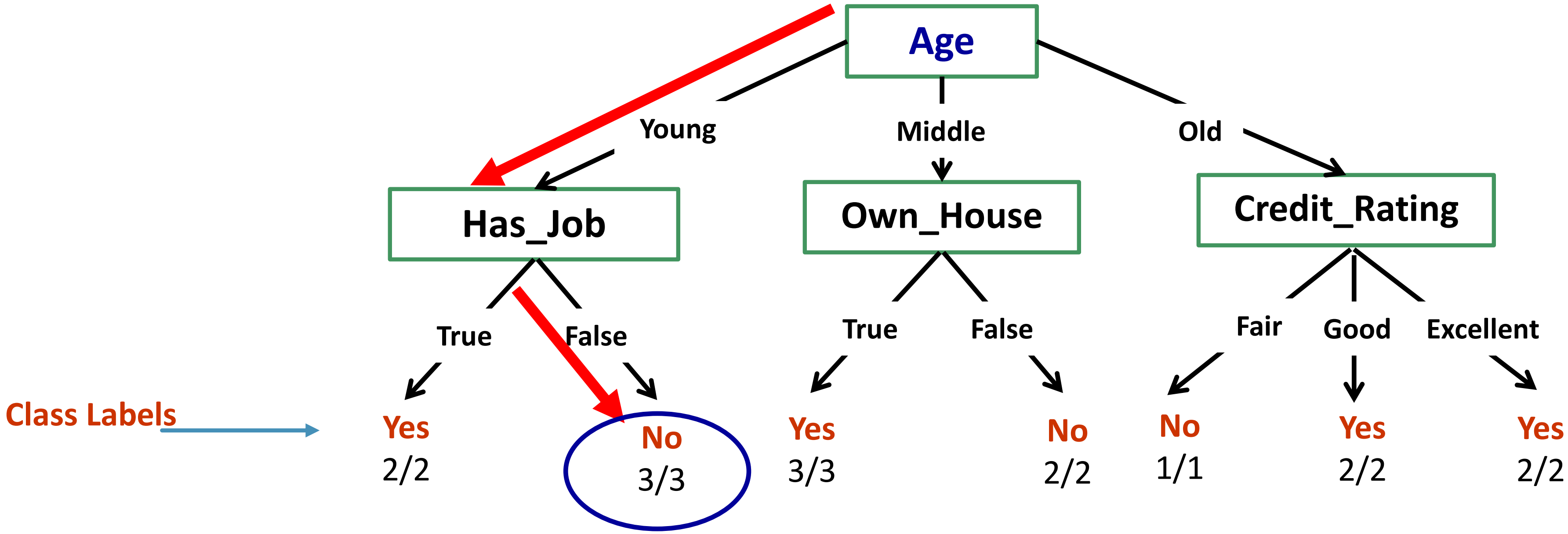
EXAMPLE 2 – LOAN APPROVAL PREDICTION

Decision Tree Representation of Loan Data

New Applicant Details

Age	Has_Job	Own_House	Credit_Rating	Class
Young	False	False	Good	?

No



Decision nodes and leaf nodes (classes – Approved or Not)

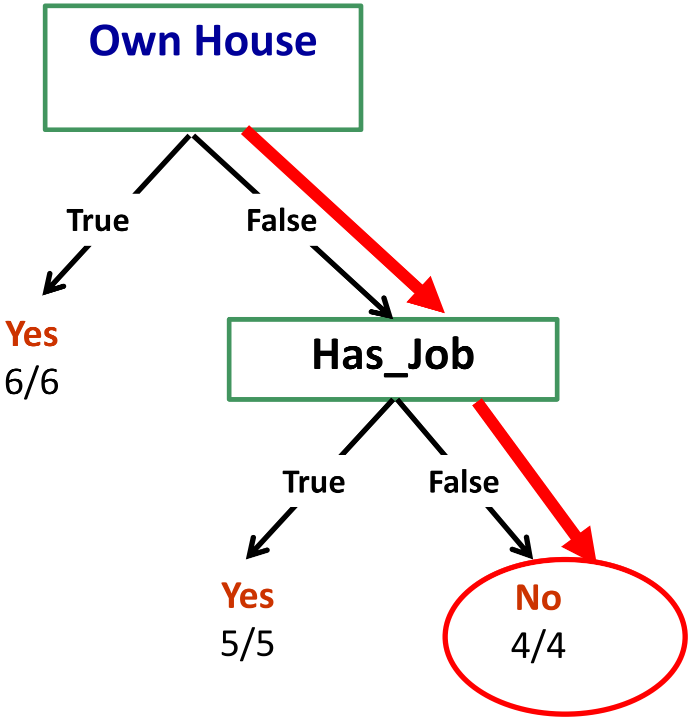
EXAMPLE 2 – LOAN APPROVAL PREDICTION

Is this tree unique?

No. We can build a simple Tree.

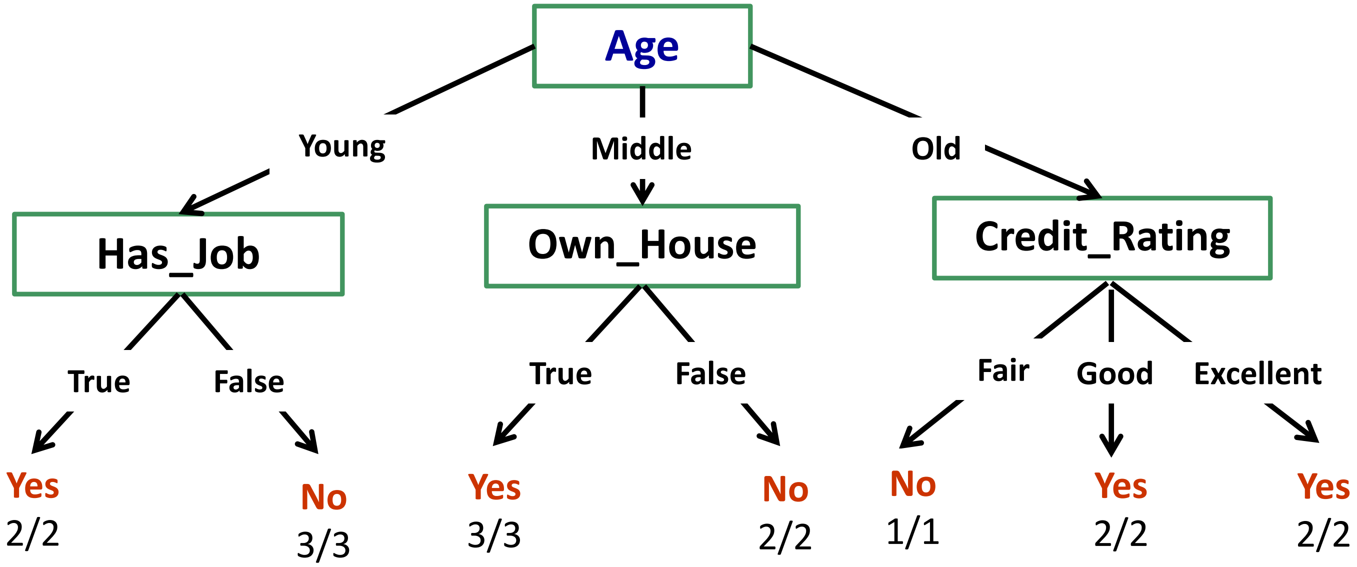
Age	Has_Job	Own_House	Credit_Rating	Class
Young	False	False	Good	?

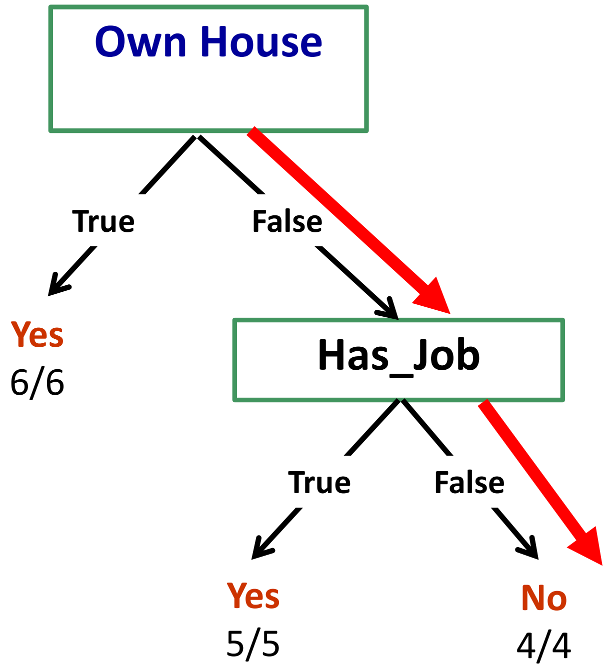
No



I am Not using the features “Age” and “credit_rating” in my tree. These features may not be significant.

TWO POSSIBLE ROOTS, WHICH IS BETTER?





EXAMPLE 2 – LOAN APPROVAL PREDICTION

Two possible roots. Which is better? Use Information Gain

- **Information Theory - The entropy formula:**

$$entropy(D) = -\sum_{j=1}^{|C|} \Pr(c_j)\log_2 \Pr(c_j)$$

$$\sum_{j=1}^{|C|} \Pr(c_j) = 1,$$

- $\Pr(c_j)$ is the **probability** of class $\boldsymbol{c_j}$ in data set *D*
- We use entropy as a measure of impurity or disorder of data set *D*. (Or, a measure of information in a tree)
- As the data becomes purer and purer, the entropy values becomes smaller and smaller

EXAMPLE 2 – LOAN APPROVAL PREDICTION

Two possible roots. Which is better? Use Information Gain

- **Information Theory – Information Gain Formula:**

$$\boxed{InformationGain(D, A_i) = Entropy(D) - EntropyA_i(D)}$$

- It is the difference between the entropy of the total data set and the entropy of the attribute A_i
- We should choose an attribute with more Information Gain as the root.

EXAMPLE 2 – LOAN APPROVAL PREDICTION

Which Should be the Root – Age or Own_House?

Compute Entropy of the total data set D

$$entropy(D) = -\sum_{j=1}^{|C|} \Pr(c_j)\log_2 \Pr(c_j)$$

Class - Yes	Class – No	Total
9	6	15

$$entropy(D) = -\frac{6}{15} \times \log_2(\frac{6}{15}) - \frac{9}{15} \times \log_2(\frac{9}{15})$$

$$= 0.971$$

EXAMPLE 2 – LOAN APPROVAL PREDICTION

Which Should be the Root – Age or Own_House?

$$entropy(D) = -\sum_{j=1}^{|C|} \Pr(c_j)\log_2 \Pr(c_j)$$

Compute Entropy – A metric to decide the root.

$$entropy_{Age}(D) = \frac{5}{15}\times entropy(D_1) + \frac{5}{15}\times entropy(D_2) + \frac{5}{15}\times entropy(D_3)$$

$$= \frac{5}{15}\times 0.971 + \frac{5}{15}\times 0.971 + \frac{5}{15}\times 0.722$$

$$= 0.888$$

Values (Age)	Class - Yes	Class – No	Entropy(Di)
D1 – young (5/15)	2	3	0.971
D2 – Middle (5/15)	3	2	0.971
D3 – Old (5/15)	4	1	0.722

$$entropy_{Own\_house}(D) = \frac{6}{15}\times entropy(D_1) + \frac{9}{15}\times entropy(D_2)$$

$$= \frac{6}{15}\times 0 + \frac{9}{15}\times 0.918$$

$$= 0.551$$

Values (Own_House)	**Class - Yes**	**Class - No**	**Entropy(Di)**
D1 – False (9/15)	6	3	0.918
D2 – True (6/15)	6	0	0

EXAMPLE 2 – LOAN APPROVAL PREDICTION

Which Should be the Root – Age or Own_House?

Compute Entropy – A metric to decide the root.

$$entropy(D) = -\sum_{j=1}^{|C|} \Pr(c_j)\log_2 \Pr(c_j)$$

$$entropy(D) = 0.971$$

$$entropy_{Age}(D) = 0.888$$

$$entropy_{Own\_house}(D) = 0.551$$

Attribute "Own_House" has more information gain (0.420)
When compared to "Age" (0.083)

Hence, Choose Own_House as Root

Information Gain(D, Age) = Entropy(D) – Entropy_Age(D)
= 0.971 – 0.888
= 0.083

Information Gain(D, Age) = Entropy(D) – Entropy_OwnHouse(D)
= 0.971 – 0.551
= 0.420

PROBLEMS IN ID3

- Cannot handle the attributes with unknown or missing values
- Cannot handle continuous values
- Pruning is very difficult
- It focuses on attributes with more values…biased…suffers from overfitting problem

Algorithm: C4.5 / C5.0

LIMITATIONS OF ID3…

- J. Ross Quinlan, a researcher in machine learning developed a decision tree induction algorithm in 1984 known as ID3 (Iterative Dichotometer 3).

- Quinlan later presented C4.5, a successor of ID3, addressing some limitations in ID3.

- ID3 uses information gain measure, which is, in fact biased towards splitting attribute having a large number of outcomes.

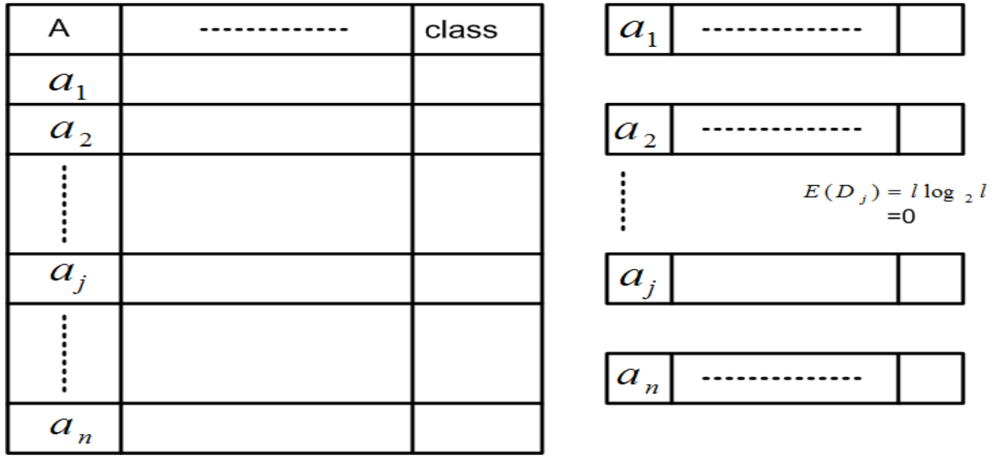
- For example, if an attribute has distinct values for all tuples, then it would result in a large number of partitions, each one containing just one tuple.

  - In such a case, note that each partition is pure, and hence the purity measure of the partition, that is $E_A(D) = 0$

LIMITATIONS OF ID3…

Example 9.18 : Limitation of ID3

In the following, each tuple belongs to a unique class. The splitting on A is shown.



$$E_A(D) = \sum_{j=1}^{n} \frac{|D_j|}{|D|} . E(D_j) = \sum_{j=1}^{n} \frac{1}{|D|} . 0 = 0$$

Thus, $\alpha(A, D) = E(D) - E_A(D)$ is maximum in such a situation.

Algorithm: C 4.5 : Introduction

- The overfitting problem in ID3 is due to the measurement of information gain.

- In order to reduce the effect of the use of the bias due to the use of information gain, **C4.5 uses a different measure called Gain Ratio**, denoted as $\boldsymbol{\beta}$.

- Gain Ratio is a kind of normalization to information gain using a **split information**.

Algorithm: C 4.5 : Gain Ratio

Definition: **Gain Ratio**

The gain ratio can be defined as follows. We first define split information $E_A^*(D)$ as

$$E_A^*(D) = -\sum_{j=1}^{m} \frac{|D_j|}{|D|} . \log \frac{|D_j|}{|D|}$$

Here, *m* is the number of distinct values in *A*.

The gain ratio is then defined as $\beta(A, D) = \frac{\alpha(A,D)}{E_A^*(D)}$, where $\alpha(A, D)$ denotes the information gain on splitting the attribute *A* in the dataset *D*.

Split Information

SplitinformationE^{*}_{A}(D)

- The value of split information depends on
  - the number of (distinct) values an attribute has and
  - how uniformly those values are distributed.

- In other words, it represents the potential information generated by splitting a data set D into m partitions, corresponding to the m outcomes of on attribute A.

Split Information…

Example: Split information $E_A^*(\mathrm{D})$

- To illustrate $E_A^*(\mathrm{D})$, let us examine the case where there are 32 instances and splitting an attribute A which has a_1, a_2, a_3 and a_4 sets of distinct values.

  - Distribution 1 : Highly non-uniform distribution of attribute values

	a_1	a_2	a_3	a_4
Frequency	32	0	0	0

$$E_A^*(\mathrm{D}) = -\frac{32}{32} log_2(\frac{32}{32}) = -log_2 1 = 0$$

  - Distribution 2

	a_1	a_2	a_3	a_4
Frequency	16	16	0	0

$$E_A^*(\mathrm{D}) = -\frac{16}{32} log_2(\frac{16}{32}) - \frac{16}{32} log_2(\frac{16}{32}) = log_2 2 = 1$$

Split Information…

- **Distribution 3**

	a_1	a_2	a_3	a_4
Frequency	16	8	8	0

$$E_A^*(\mathrm{D}) = -\frac{16}{32}log_2(\frac{16}{32}) - \frac{8}{32}log_2(\frac{8}{32}) - \frac{8}{32}log_2(\frac{8}{32}) = 1.5$$

- **Distribution 4**

	a_1	a_2	a_3	a_4
Frequency	16	8	4	4

$$E_A^*(\mathrm{D}) = 1.75$$

- **Distribution 5: Uniform distribution of attribute values**

	a_1	a_2	a_3	a_4
Frequency	8	8	8	8

$$E_A^*(\mathrm{D}) = (-\frac{8}{32}log_2(\frac{8}{32}))*4 = -log_2(\frac{1}{4}) = 2.0$$

Split Information…vs…information Gain

- **Information gain signifies how much information will be gained on partitioning the values of attribute *A***
  - **"Higher information gain means splitting of *A* is more desirable."**
- **On the other hand, split information forms the denominator in the gain ratio formula.**
  - **This implies that higher the value of split information is, lower the gain ratio.**
  - **In turns, it decreases the information gain.**
- **Further, information gain is large when there are many distinct attribute values.**
  - **When many distinct values, split information is also a large value.**
  - **This way split information reduces the value of gain ratio, thus resulting a balanced value for information gain.**
- **Like information gain (in ID3), the attribute with the maximum gain ratio is selected as the splitting attribute in C4.5.**

Comparing ID3, CART and C4.5/5.0

COMPARE ID3, CART AND C4.5/C5.0

Algorithm	Splitting Criteria	Remark
ID3	**Information Gain** $\alpha(A,D) = E(D) - E_A(\text{D})$ Where $E(D)$ = Entropy of D (a measure of uncertainty) = $-\sum_{i=1}^{k} p_i \log_2 p_i$ where D is with set of k classes c_1, c_2, … , c_k and p_i = $\frac{\lvert C_{i,D}\rvert}{\lvert D\rvert}$;Here, $C_{i,D}$ is the set of tuples with class c_i in D. E_A(D) = Weighted average entropy when D is partitioned on the values of attribute A = $\sum_{j=1}^{m} \frac{\lvert D_j\rvert}{\lvert D\rvert} E(D_j)$ Here, m denotes the distinct values of attribute A.	• The algorithm calculates $\alpha(A_i,D)$ for all A_i *in* D and choose that attribute which has **maximum**$\alpha(A_i,D)$. • The algorithm can handle both categorical and numerical attributes. • It favors splitting those attributes, which has a large number of distinct values.

COMPARE ID3, CART AND C4.5/C5.0

Algorithm	Splitting Criteria	Remark
CART	**Gini Index** $\gamma(A, D) = G(D) - G_A(\mathrm{D})$ where $G(D)$ = Gini index (a measure of impurity) $= 1 - \sum_{i=1}^{k} {p_i}^2$ Here, $p_i = \frac{\|C_{i,D}\|}{\|D\|}$ and D is with k number of classes and $G_A(D) = \frac{\|D_1\|}{\|D\|} G(D_1) + \frac{\|D_2\|}{\|D\|} G(D_2)$, when D is partitioned into two data sets D_1 and D_2 based on some values of attribute A.	• The algorithm calculates all binary partitions for all possible values of attribute A and choose that binary partition which has the **maximum** $\gamma(A, D)$. • The algorithm is computationally very expensive when the attribute A has a large number of values.

COMPARE ID3, CART AND C4.5/C5.0

Algorithm	Splitting Criteria	Remark
C4.5	**Gain Ratio** $$\beta(A,D) = \frac{\alpha(A,D)}{E_A^*(\mathrm{D})}$$ where $\alpha(A,D)$ = Information gain of D (same as in ID3, and $E_A^*(\mathrm{D})$ =splitting information $= -\sum_{j=1}^{m} \frac{\lvert D_j \rvert}{\lvert D \rvert} log_2 \frac{\lvert D_j \rvert}{\lvert D \rvert}$ when D is partitioned into D_1, D_2, … , D_m partitions corresponding to m distinct attribute values of A.	• The attribute A with **maximum** value of $\beta(A,D)$ is selected for splitting. • Splitting information is a kind of normalization, so that it can check the biasness of information gain towards the choosing attributes with a large number of distinct values.

In addition to this, we also highlight few important characteristics of decision tree induction algorithms in the following.

Problems of Decision Trees

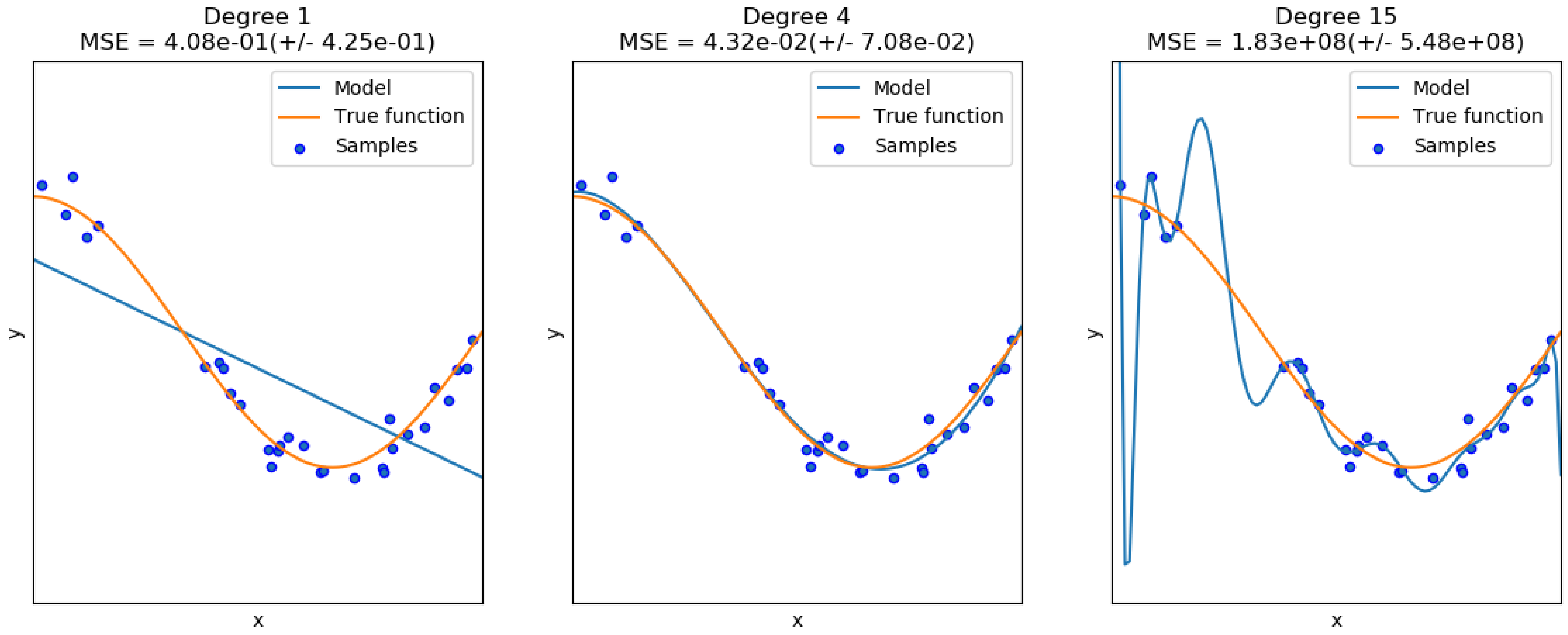
ISSUES IN DECISION TREE LEARNING

- Overfitting
- Handling skewed distributions
- Handling attributes and classes with different costs
- Etc.

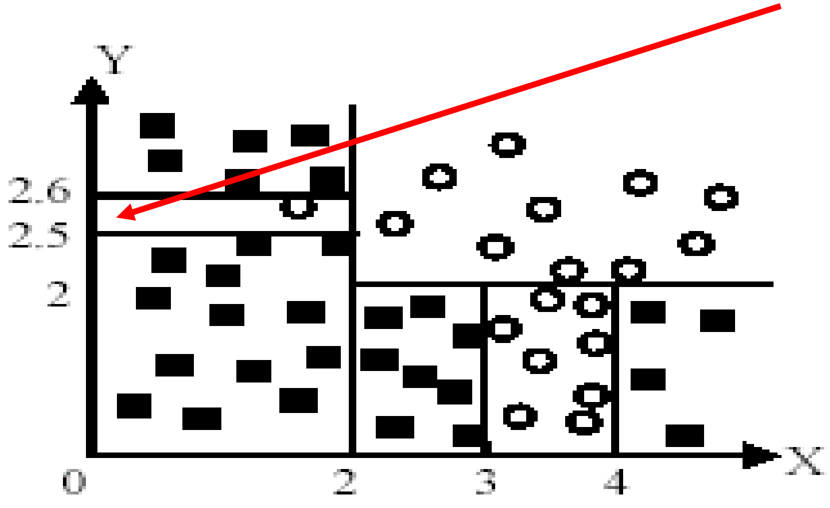
AVOID OVERFITTING IN CLASSIFICATION

- Overfitting: A tree may overfit the training data
  - **Good accuracy on training data but poor on test data**
  - Symptoms: tree too deep and too many branches, some may reflect anomalies due to noise or outliers
- A decision tree can continue to grow indefinitely, choosing splitting features and dividing the data into smaller and smaller partitions until each example is perfectly classified or the algorithm runs out of features to split on.
- Two approaches to avoid overfitting (Pruning)
  - Pruning is a solution
  - The process of pruning a decision tree involves reducing its size of the tree such that it generalizes better to unseen data.

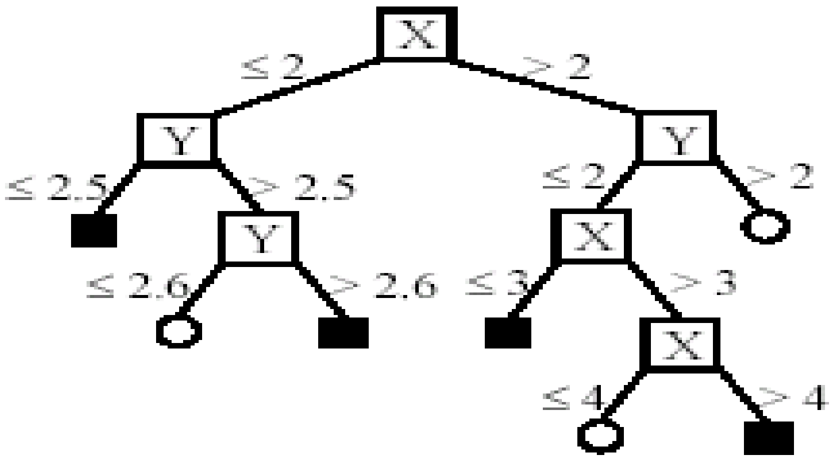
Example Of Over Fit – General Case



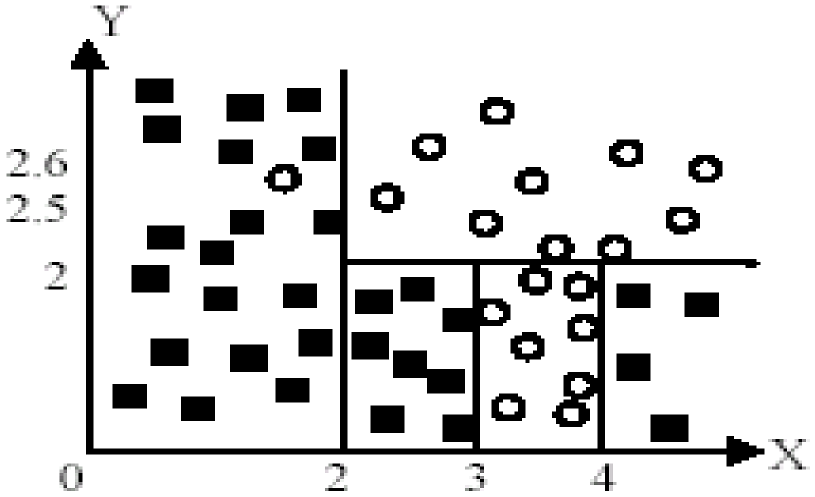
Example Of Over Fit – Decision Tree

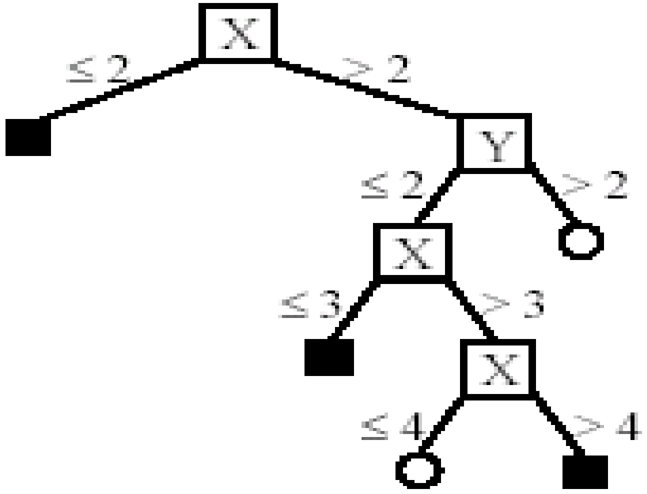


(A) A partition of the data space



(B). The decision tree





AVOID OVERFITTING IN CLASSIFICATION

- Two approaches to avoid overfitting (Pruning)
  - Pre-pruning: Halt tree construction early
    - Stop the tree from growing once it reaches a certain number of decisions or when the decision nodes contain only a small number of examples.
    - Difficult to decide because we do not know what may happen subsequently if we keep growing the tree.
  - Post-pruning: Remove branches or sub-trees from a "fully grown" tree.
    - This method is commonly used. C5.0 uses a statistical method to estimates the errors at each node for pruning.
    - A validation set may be used for pruning as well.

Evaluation Methods

Evaluation Methods

1. **Holdout set:** The available data set *D* is divided into two disjoint subsets,
   - the *training set* D_{train} (for learning a model)
   - the *test set* D_{test} (for testing the model)
- **Important:** training set should not be used in testing and the test set should not be used in learning.
  - Unseen test set provides a unbiased estimate of accuracy.
- The test set is also called the holdout set. (the examples in the original data set *D* are all labeled with classes.)
- This method is mainly used when the data set *D* is large.

Evaluation Methods (Cont…)

2. n-fold cross-validation: The available data is partitioned into *n* equal-size disjoint subsets.

- Use each subset as the test set and combine the rest *n*-1 subsets as the training set to learn a classifier.
- The procedure is run *n* times, which give *n* accuracies.
- The final estimated accuracy of learning is the average of the *n* accuracies.
- 10-fold and 5-fold cross-validations are commonly used.
- This method is used when the available data is not large.

Evaluation Methods (Cont…)

3. Leave-one-out cross-validation: This method is used when the data set is very small.

- It is a special case of cross-validation
- Each fold of the cross validation has only a single test example and all the rest of the data is used in training.
- If the original data has *m* examples, this is *m*-fold cross-validation

Classification Evaluation Metrics/Measures

Classification Measures - Accuracy

- Accuracy is only one measure (error = 1-accuracy).
- **Accuracy is not suitable in some applications**.
- In classification involving skewed or highly imbalanced data, e.g., network intrusion and financial fraud detections, we are interested only in the minority class.
  - High accuracy does not mean any intrusion is detected.
  - E.g., 1% intrusion. Achieve 99% accuracy by doing nothing.
- The class of interest is commonly called the **positive class**, and the rest **negative classes.**

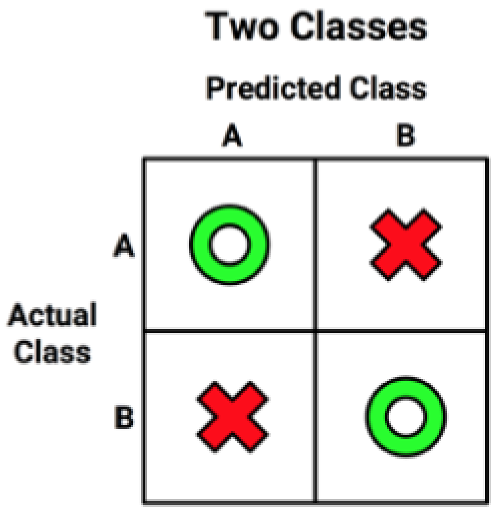
Classification measures – Precision (P) and Recall (R)

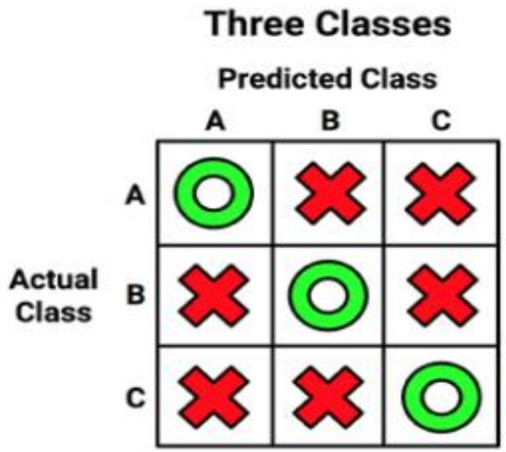
- Used in classification.
- We use a confusion matrix to introduce them.

	Classified Positive	Classified Negative
Actual Positive	TP	FN
Actual Negative	FP	TN

where

TP: the number of correct classifications of the positive examples (**true positive**),
FN: the number of incorrect classifications of positive examples (**false negative**),
FP: the number of incorrect classifications of negative examples (**false positive**), and
TN: the number of correct classifications of negative examples (**true negative**).





Classification measures – Precision and recall…

	Classified Positive	Classified Negative
Actual Positive	TP	FN
Actual Negative	FP	TN

$$p = \frac{TP}{TP + FP}. \qquad r = \frac{TP}{TP + FN}.$$

- Precision p (Positive predicted value) is the number of correctly classified positive examples divided by the total number of examples that are classified as positive.

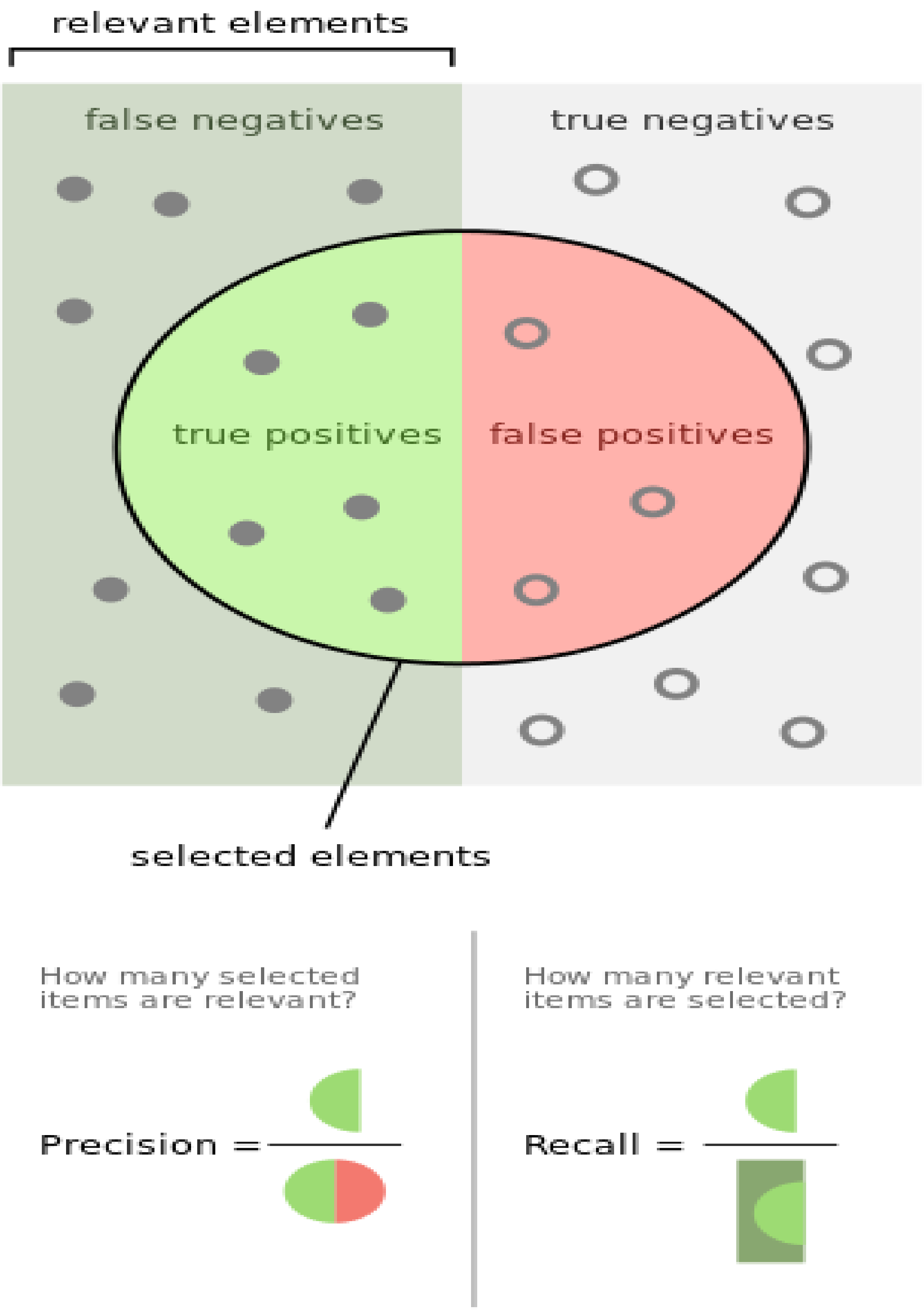
- Recall r (Sensitivity) is the number of correctly classified positive examples divided by the total number of actual positive examples in the test set.

Classification measures – Precision and r

	Classified Posi
Actual Positive	TP
Actual Negative	FP

$$p = \frac{TP}{TP + FP}. \qquad r =$$

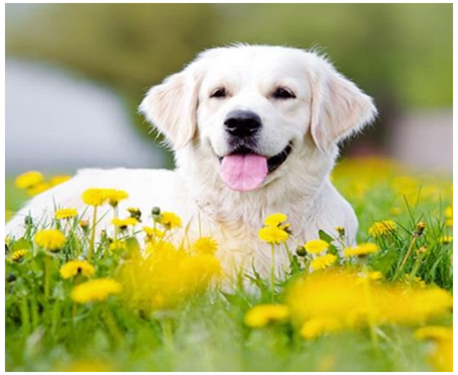
- p - Relevant instances among all the retrieved instances
- R - Fraction of relevant instances out of total relevant instances



Precision & Recall - Example

- Example:

  A data set has 12 dogs and 5 cats

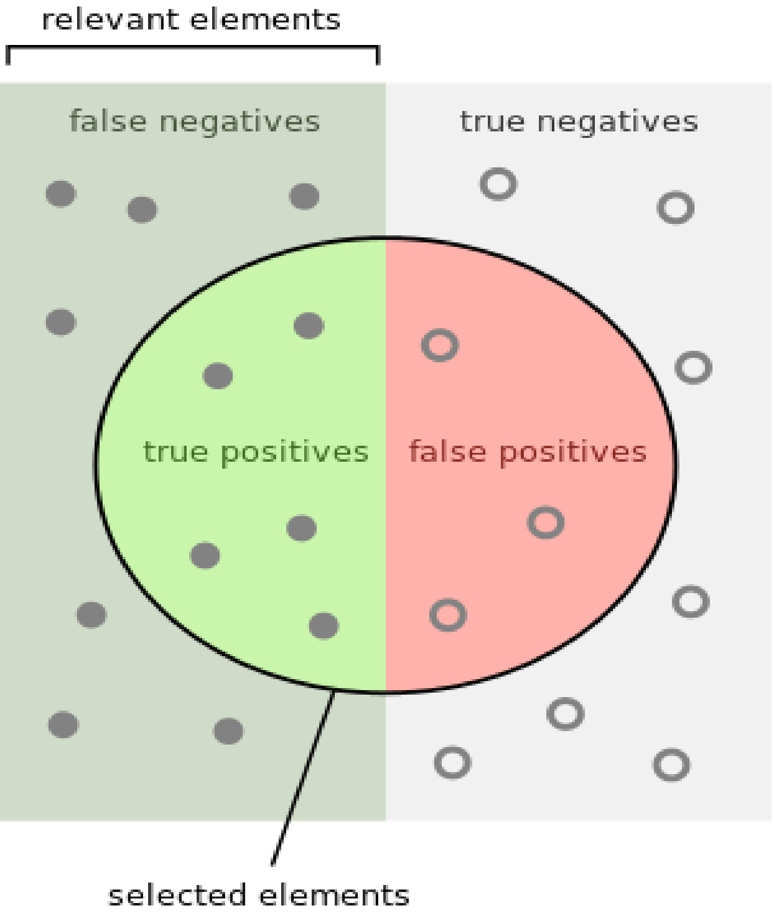


Decision Tree identified 8 dogs.

Out of 8 dogs which are identified:

5 - Actual Dogs

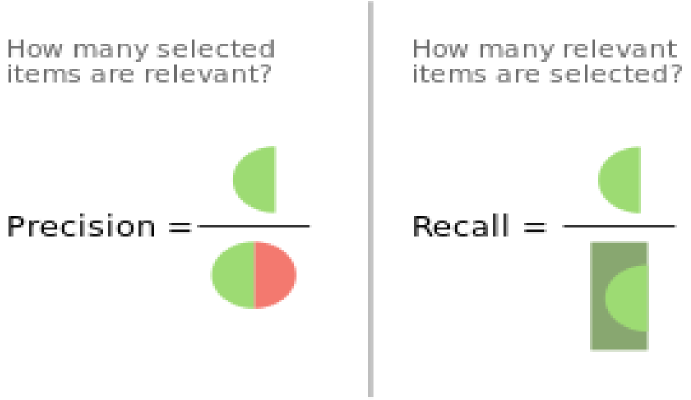
3 - Cats



Precision p = Actual Dogs/(Actual Dogs + Wrongly identified as Dogs)

= 5 / (5 + 3) = 5/8

Recall r = Actual Dogs/ Total Dogs = 5/12



CLASSIFICATION MEASURES – PRECISION AND RECALL…

	Classified Positive	Classified Negative
Actual Positive	1	99
Actual Negative	0	1000

$$p = \frac{TP}{TP + FP}. \qquad r = \frac{TP}{TP + FN}.$$

- This confusion matrix gives
  - precision p = 100% and
  - recall r = 1%

  because we only classified one positive example correctly and no negative examples wrongly.
- Note: precision and recall only measure classification on the positive class.

CLASSIFICATION MEASURES – F1 SCORE

- It is hard to compare two classifiers using two measures. F_1 score combines precision and recall into one measure

$$F_1 = \frac{2pr}{p + r}$$

F_1-score is the harmonic mean of precision and recall.

$$F_1 = \frac{2}{\frac{1}{p} + \frac{1}{r}}$$

- The harmonic mean of two numbers tends to be closer to the smaller of the two.
- For F_1-value to be large, both *p* and *r* much be large.

RECEIVER OPERATING CHARACTERISTICS CURVE (AUC-ROC)

- It is commonly called the AUC-ROC curve.
- It is also written as AUROC **(Area Under the Receiver Operating Characteristics)**
- It is a plot of the true positive rate (TPR) against the false positive rate (FPR).
- True positive rate (TPR):

$$\text{TPR / Recall / Sensitivity} = \frac{TP}{TP + FN}$$

- False positive rate (FPR):

$$\text{FPR} = 1 - \text{Specificity}$$

$$= \frac{FP}{TN + FP}$$

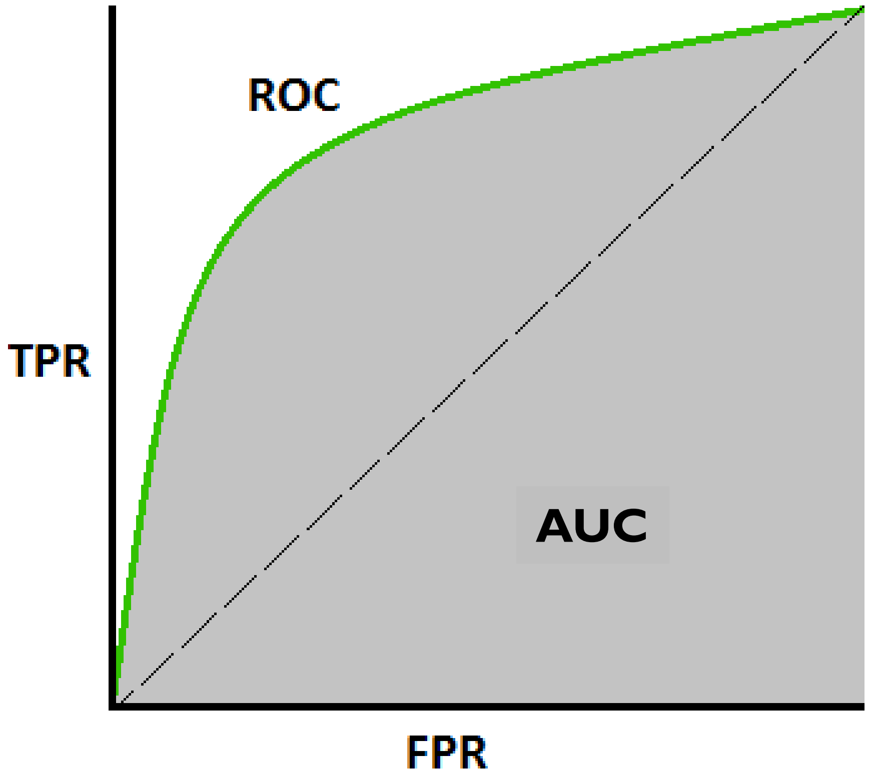
$$\text{Specificity} = \frac{TN}{TN + FP}$$

SENSITIVITY AND SPECIFICITY

- AUC near to the 1 → Excellent Model - it has good measure of separability.

- AUC near to 0 → A poor model - which means it has worst measure of separability.

- AUC = 0.5 → model has no class separation capacity whatsoever.

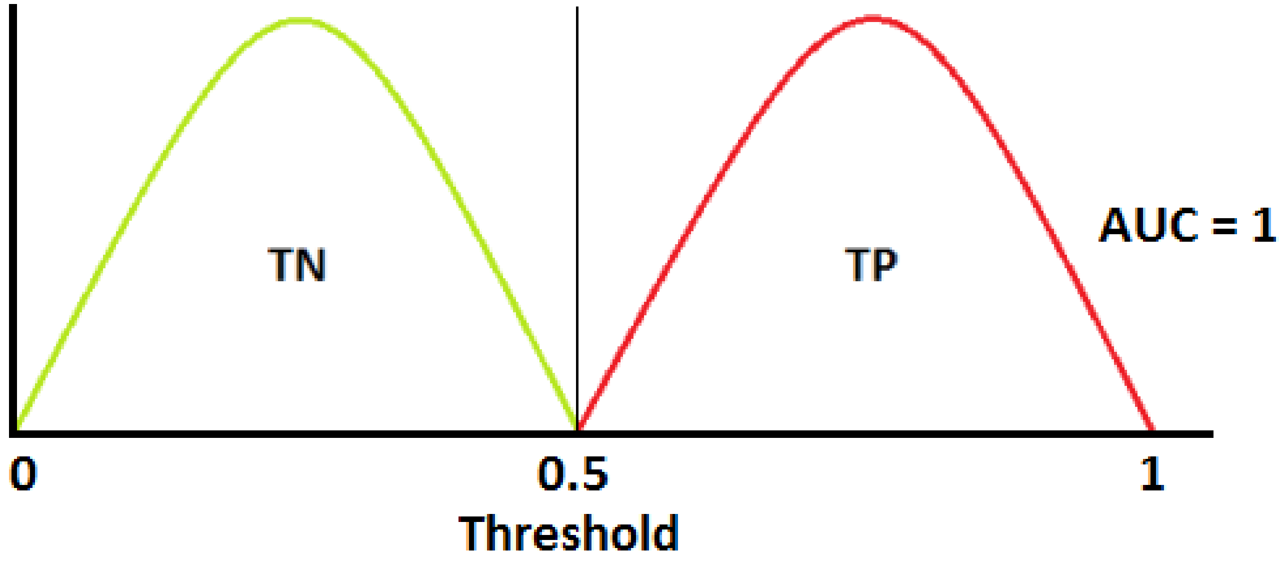


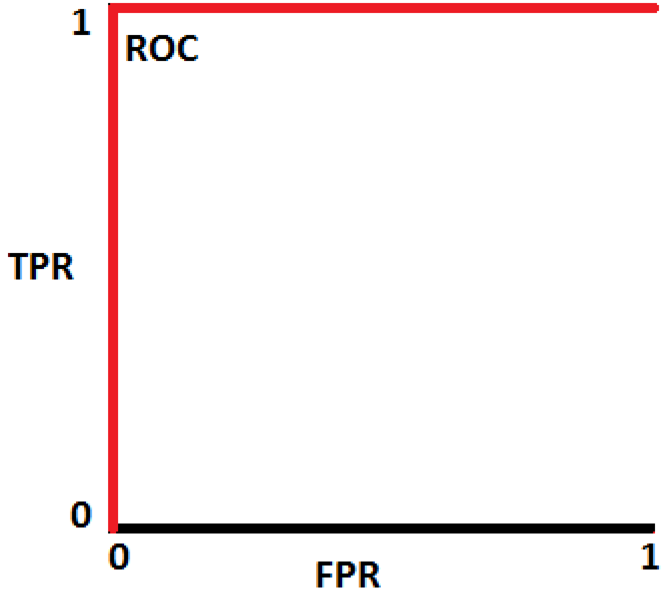
ROC is a curve of probability.

EXAMPLE ROC CURVES

Example:

- Red distribution curve is of the positive class (patients with disease)
- **Green** distribution curve is of negative class(patients with no disease).



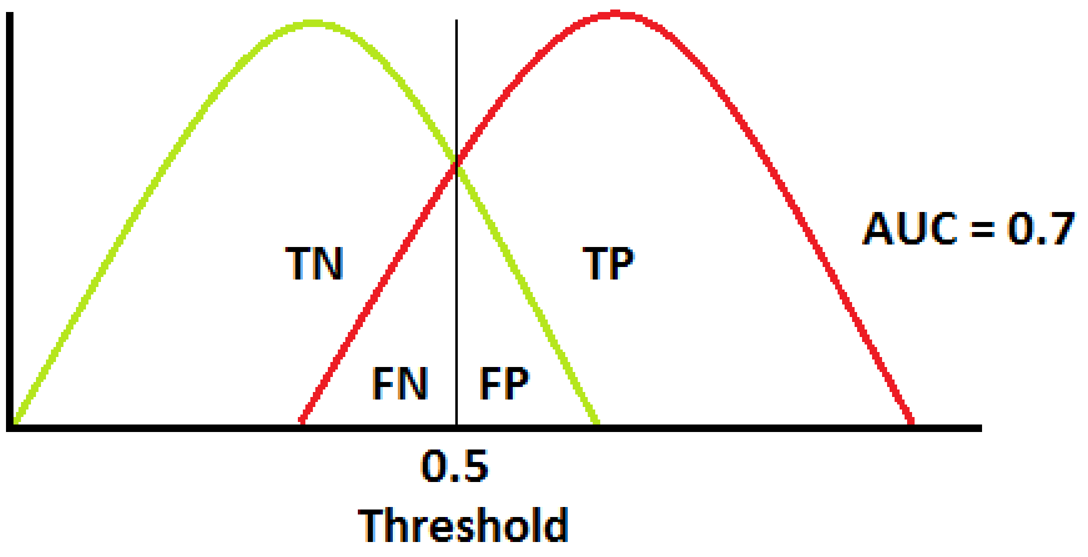


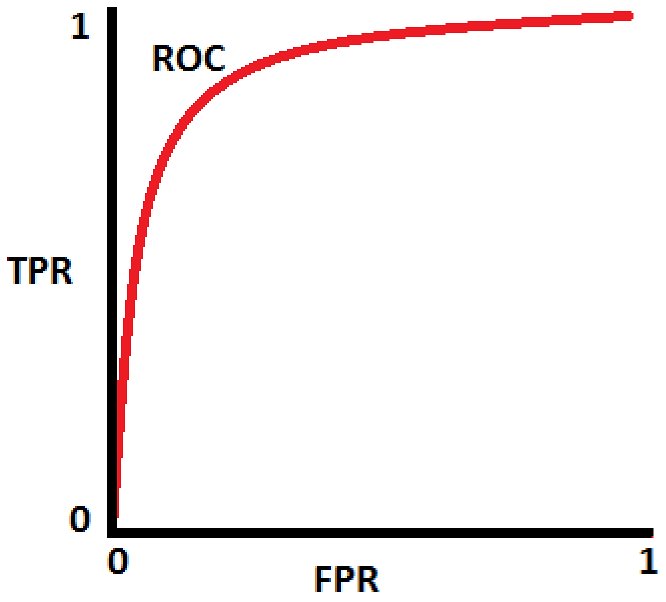
This is an ideal situation. When two curves don't overlap at all means **model has an ideal measure of separability.** It is perfectly able to distinguish between positive class and negative class.

EXAMPLE ROC CURVES

Example:

- Red distribution curve is of the positive class (patients with disease)
- **Green** distribution curve is of negative class(patients with no disease).



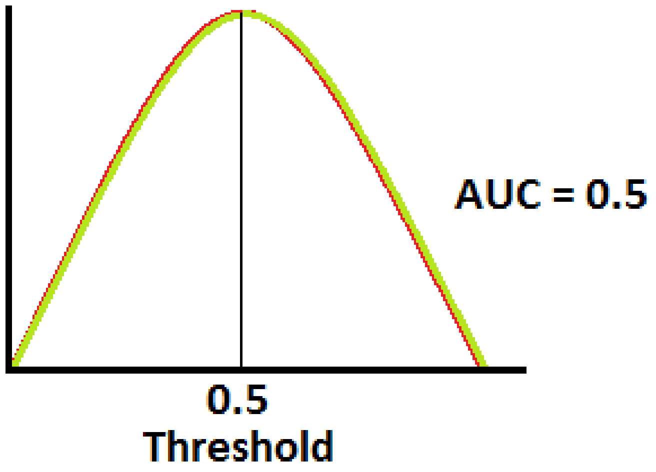


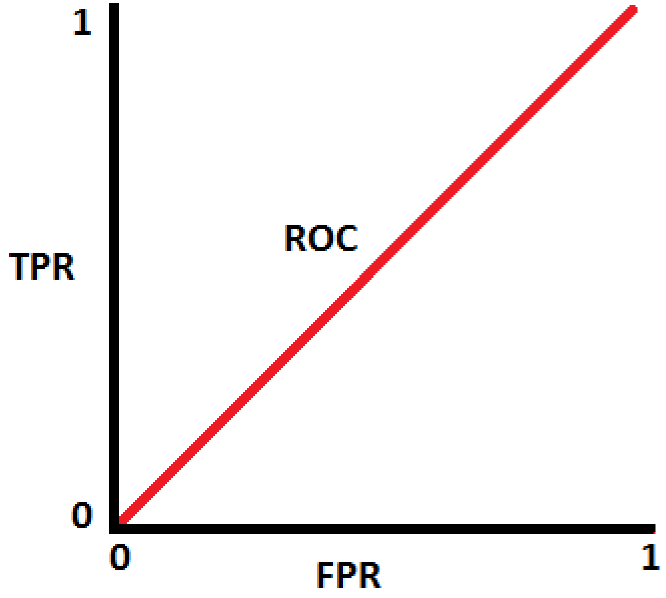
When AUC is 0.7, it means there is **70% chance that model will be able to distinguish** between positive class and negative class.

EXAMPLE ROC CURVES

Example:

- Red distribution curve is of the positive class (patients with disease)
- **Green** distribution curve is of negative class(patients with no disease).



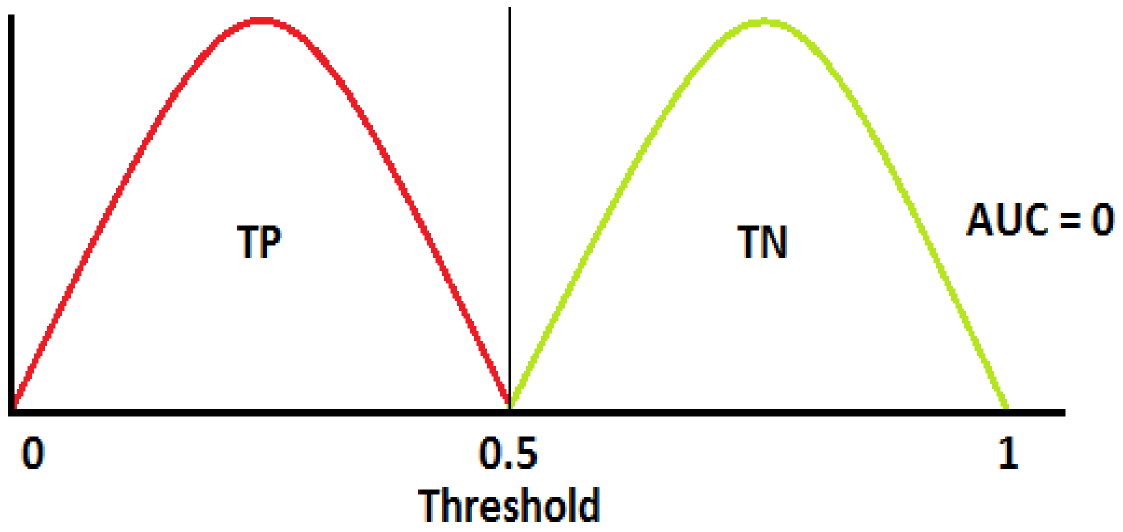


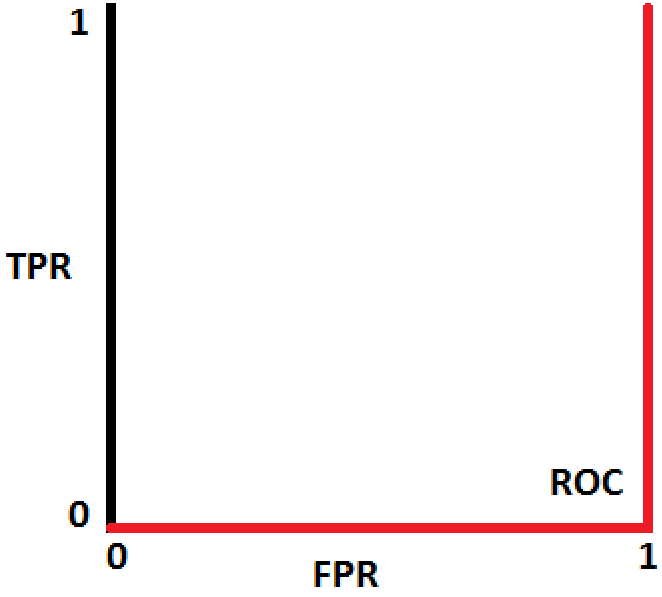
This is the worst situation. When AUC is approximately 0.5, **model has no discrimination capacity to distinguish between positive class and negative class.** Model may be doing all random guessess.

EXAMPLE ROC CURVES

Example:

- Red distribution curve is of the positive class (patients with disease)
- **Green** distribution curve is of negative class(patients with no disease).





When AUC is approximately 0, **model is actually reciprocating the classes**. It means, model is predicting negative class as a positive class and vice versa. (Completely wrong predictions but model is able discriminate the classes in completely wrong way)

Thank You