UNIT-III

(BAYESIAN LEARNING)

Supervised Learning

Topics :

- Bayes Theorem
- Naive Bayes Classifier
  - Examples

BAYES THEOREM

- In order to explain **Naive Bayes** we need to first explain **Bayes theorem**.
- Bayes' Theorem, named **after 18th-century British mathematician Thomas Bayes,** is a mathematical formula for determining **conditional probability**.
- **Conditional probability** is known as the **possibility of an event or outcome happening**, based on the **existence of a previous event or outcome.**
- **Eg: Coin a Toss, Roll a Dice, Playing Cards etc**.

CONTD..

- When you **flip a fair coin**, there is an **equal chance of getting either heads or tails.** So you can say the probability of getting **heads is 50%.**
- Similarly what would be the probability of getting **a 1 when you roll a dice with 6 faces?** Assuming the dice is fair, the **probability of 1/6 = 0.166.**
- There fore these are the **classical examples** of **Conditional Probability.**
- So, when you say the **conditional probability of A given B**, it denotes the **probability of A occurring given that B has already occurred.**

CONTD..

- **Mathematically, Conditional probability of A given B can be computed as:**

$$P(A|B) = P(A \text{ AND } B) / P(B)$$

- **Eg: School Example** Consider a **school with a total population of 100 persons**. These 100 persons can be seen either as **'Students' and 'Teachers' or as a population of 'Males' and 'Females'.**
- With below **tabulation of the 100 people**, what is the **conditional probability** that a certain member of the **school is a 'Teacher' given that he is a 'Man'?**

CONTD..

	Female	Male	Total
Teacher	8	12	20
Student	32	48	80
Total	40	60	100

- **To calculate this**, you may filter the **sub-population of 60 males and focus on the 12 (male) teachers.** So the required **conditional probability**

  P(Teacher | Male) = 12 / 60 = 0.2.

$$P(Teacher \mid Male) = \frac{P(Teacher \cap Male)}{P(Male)} = 12/60 = 0.2$$

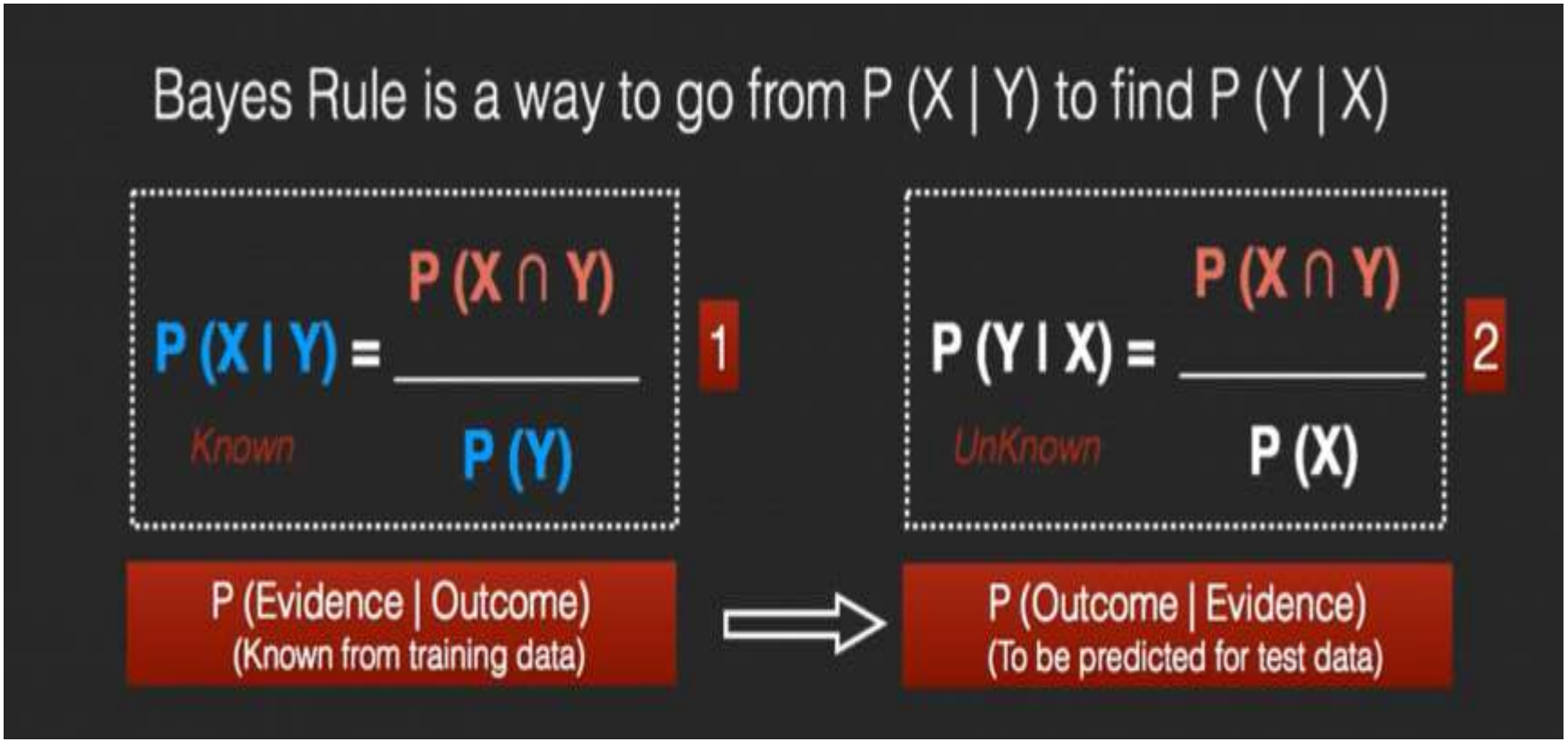
CONTD..

- This can be represented as the **intersection of Teacher (A) and Male (B) divided by Male (B).**
- Likewise, the c**onditional probability of B given A can be computed**.
- The Bayes Rule that we use for Naive Bayes, can be derived from these two notations.

$$P(A \mid B) = \frac{P(A \cap B)}{P(B)} \qquad (1)$$

$$P(B \mid A) = \frac{P(A \cap B)}{P(A)} \qquad (2)$$

CONTD..


Bayes Rule is a way to go from P (X | Y) to find P (Y | X)
P (X I Y) = P (X ∩ Y) / P (Y)
Known
1
P (Y I X) = P (X ∩ Y) / P (X)
UnKnown
2
P (Evidence | Outcome)
(Known from training data)
P (Outcome | Evidence)
(To be predicted for test data)

CONTD..

- Bayes Theorem is also used widely in **machine learning,** where it is a **simple, effective way to predict classes with precision and accuracy.**
- The Bayesian method of **calculating conditional probabilities** is used in **machine learning applications** that **involve classification tasks**.
- A **simplified version of the Bayes Theorem**, known as the **Naive Bayes Classification**, is used to **reduce computation time and costs.**

NAIVE BAYES CLASSIFIER

- **Naïve Bayes algorithm** is a **supervised learning algorithm,** which is based on **Bayes theorem** and used for **solving classification problems and also solving Regression Problems**.
- It is mainly used in **text classification** that includes a **high-dimensional training dataset.**
- **Naïve Bayes Classifier** is one of the simple and most **effective Classification algorithms** which helps in **building the fast machine learning models that can make quick predictions.**

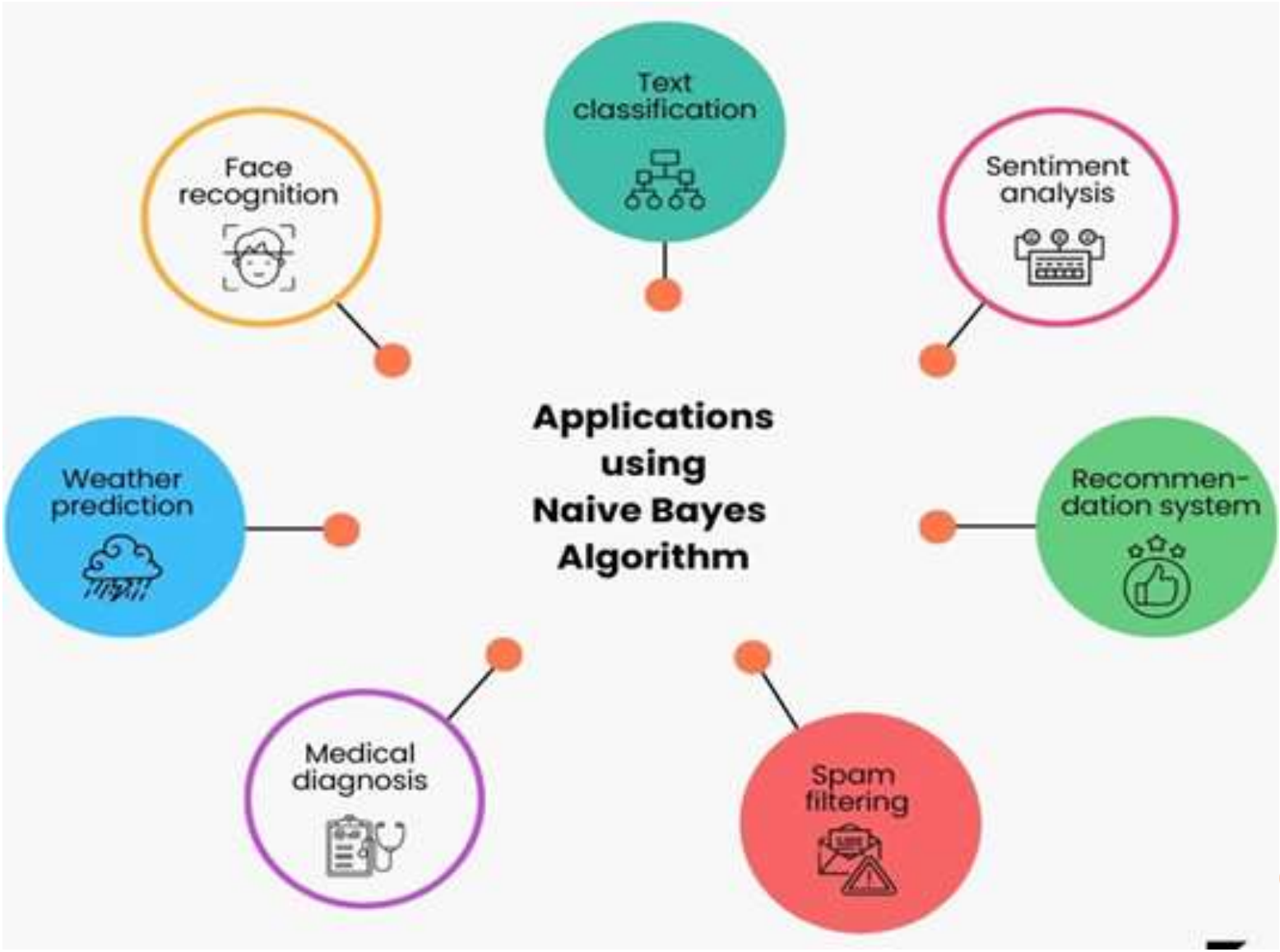
CONTD..

- The Naïve Bayes algorithm is comprised of **two words Naïve and Bayes,** Which can be described as:
- **Naïve:** It is called Naïve because it assumes that the occurrence of a **certain feature is independent of the occurrence of other features.**
- Such as if the **fruit is identified on the bases of color, shape, and taste, then red, spherical, and sweet fruit** is recognized as **an apple**.
- Hence **each feature individually contributes** to **identify that it is an apple without depending on each other.**

CONTD..

- **Bayes:** It is called Bayes because it depends on the principle of Bayes' Theorem.
- **Naive Bayes mainly used in**
  - 1. Face Recognition
  - 2. Weather Prediction
  - 3. Medical Diagnosis
  - 4. News Classification etc.

CONTD..


Text classification
Face recognition
Sentiment analysis
Applications using Naive Bayes Algorithm
Weather prediction
Recommen-dation system
Medical diagnosis
Spam filtering

ADVANTAGES OF NAIVE BAYES

The following are some of the **benefits of the Naive Bayes classifier:**

1. It is simple and easy to implement
2. It doesn't require **as much training data**
3. It handles both **continuous and discrete data**
4. It is fast and can be used to **make real-time predictions**

USE CASE

- **Text classification** is one of the most popular **applications of a Naive Bayes classifier.**
- **Problem statement:** To perform text classification of **news headlines and classify news into different topics for a news website.**

ALGORITHM

- The Naive Bayes Classifier is inspired by **Bayes Theorem** which states the **following equation:**

$$P(A|B) = \frac{P(B|A) * P(A)}{P(B)}$$

- This **equation can be rewritten** using **X (input variables)** and **y (output variable)** to make it easier to understand.
- In **plain English**, **this equation is solving** for the **probability of y given input features X**.

$$P(A|B) = \frac{P(B|A) * P(A)}{P(B)}$$

- **Re write as**

$$P(y|X) = \frac{P(X|y) * P(y)}{P(X)}$$

- Here **x1, x2…, xn** represents the **Features (or) Attributes . i.e, color, shape, size etc.** and
- **'y'** is the **Class Labels. i.e, Yes or No etc.**

$$X = (x_1, x_2, x_3, ..., x_n)$$

- Because of the **naive assumption** (hence the name) that **variables are independent given the class**, we can **rewrite P(X|y)** as follows:

$$P(X|y) = P(x_1|y) * P(x_2|y) * ... * P(x_n|y)$$

- By **this equation,**

$$P(y|X) = \frac{P(X|y)P(y)}{P(X)}$$

- By **substituting** for **X** and **expanding using the chain rule** **we get,**

$$P(y|x_1, ..., x_n) = \frac{P(x_1|y)P(x_2|y)...P(x_n|y)P(y)}{P(x_1)P(x_2)...P(x_n)}$$

- Now, you can obtain the values for each by looking at the **dataset** and **substitute them into the equation**.

- For **all entries in the dataset**, the **denominator does not change,** it **remains static**.
- Therefore, the **denominator can be removed** and **proportionality can be injected**.

$$P(y|x_1, \ldots, x_n) \propto P(y) \prod_{i=1}^{n} P(x_i|y)$$

- Finally, we have to **find** the **class variable (y)** with **maximum probability.**

$$y = argmax_y P(y) \prod_{i=1}^{n} P(x_i|y)$$

- **Here, Argmax** is simply **an operation** that finds the **argument that gives the maximum value from a target function.**

NAÏVE BAYES ALGORITHM STEPS

- **Step-1:** Convert the **data set** into a **frequency table**
- **Step-2:** Next **calculate** the **Likely hood table** from the **Frequency Table**
- **Step-3:** Now, use **Naive Bayesian equation** to **calculate** the **posterior probability for each class**.
- Finally, The class with the **highest posterior probability** is the **outcome of the prediction.**

EXAMPLE-1

- Suppose you tracked the weather conditions for **14 days** and based on the **weather conditions**, you decided whether to **play golf or not play golf.**

Outlook	Temperature	Humidity	Windy	Play?
sunny	hot	high	FALSE	no
sunny	hot	high	TRUE	no
overcast	hot	high	FALSE	yes
rainy	mild	high	FALSE	yes
rainy	cool	normal	FALSE	yes
rainy	cool	normal	TRUE	no
overcast	cool	normal	TRUE	yes
sunny	mild	high	FALSE	no
sunny	cool	normal	FALSE	yes
rainy	mild	normal	FALSE	yes
sunny	mild	normal	TRUE	yes
overcast	mild	high	TRUE	yes
overcast	hot	normal	FALSE	yes
rainy	mild	high	TRUE	no

CONTD..

- **Step-1:** **First, we need to convert this into a Frequency Table**

outlook			temperature		
	yes	no		yes	no
sunny	2	3	hot	2	2
overcast	4	0	mild	4	2
rainy	3	2	cool	3	1

humidity			windy			play?	
	yes	no		yes	no	yes	no
high	3	4	FALSE	6	2	9	5
normal	6	1	TRUE	3	3		

CONTD..

- **Step-2:** Next, convert this into a Likelihood Table. i.e, we want to convert the frequencies into ratios or conditional probabilities:

outlook			temperature		
	yes	no		yes	no
sunny	2/9	3/5	hot	2/9	2/5
overcast	4/9	0/5	mild	4/9	2/5
rainy	3/9	2/5	cool	3/9	1/5

humidity			windy			play?	
	yes	no		yes	no	yes	no
high	3/9	4/5	FALSE	6/9	2/5	9/14	5/14
normal	6/9	1/5	TRUE	3/9	3/5		

CONTD..

- **Step-3:** **Finally, we can use the proportionality equation to predict y, given X.**
- Imagine that **X = {outlook: sunny, temperature: mild, humidity: normal, windy: false}.**
- First, we'll calculate the **probability** that you will **play golf given X, P(yes|X)** followed by the **probability that you won't play golf given X, P(no|X).**

CONTD..

- Using **the chart above**, we can get the **following information:**

$$P(yes) = 9/14$$

$$P(outlook = sunny|yes) = 2/9$$

$$P(temperature = mild|yes) = 4/9$$

$$P(humidity = normal|yes) = 6/9$$

$$P(windy = false|yes) = 6/9$$

CONTD..

- Now we can simply **input this information** into the following **formula:**

$$P(yes|X) \propto P(X|y) * P(y)$$

$$P(yes|X) \propto P(x_1|y) * P(x_2|y) * P(x_3|y) * P(x_4|y) * P(y)$$

$$P(yes|X) \propto P(sunny|yes) * P(mild|yes) * P(normal|yes) * P(false|yes) * P(yes)$$

$$P(yes|X) \propto \frac{2}{9} * \frac{4}{9} * \frac{6}{9} * \frac{6}{9} * \frac{9}{14}$$

$$\mathbf{P(yes|X) \propto 0.0282}$$

Similarly, you would complete the **same sequence of steps for P(no|X).**

$$P(no|X) \propto 0.0069$$

Since **P(yes|X) > P(no|X)**, then you can predict that this person **would play golf** g**iven that the outlook is sunny, the temperature is mild, the humidity is normal and it's not windy.**

EXAMPLE-2

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

Frequency Table

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

		Play Golf	
		Yes	No
Humidity	High	3	4
	Normal	6	1

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

		Play Golf	
		Yes	No
Windy	False	6	2
	True	3	3

Likelihood Table

		Play Golf	
		Yes	No
Outlook	Sunny	3/9	2/5
	Overcast	4/9	0/5
	Rainy	2/9	3/5

		Play Golf	
		Yes	No
Humidity	High	3/9	4/5
	Normal	6/9	1/5

		Play Golf	
		Yes	No
Temp.	Hot	2/9	2/5
	Mild	4/9	2/5
	Cool	3/9	1/5

		Play Golf	
		Yes	No
Windy	False	6/9	2/5
	True	3/9	3/5

Example 2:

In this example we have 4 inputs (predictors). The final posterior probabilities can be standardized between 0 and 1.

Outlook	Temp	Humidity	Windy	Play
Rainy	Cool	High	True	?

$$P(Yes \mid X) = P(Rainy \mid Yes) \times P(Cool \mid Yes) \times P(High \mid Yes) \times P(True \mid Yes) \times P(Yes)$$

$$P(Yes \mid X) = 2/9 \times 3/9 \times 3/9 \times 3/9 \times 9/14 = 0.00529 \rightarrow 0.2 = \frac{0.00529}{0.02057 + 0.00529}$$

$$P(No \mid X) = P(Rainy \mid No) \times P(Cool \mid No) \times P(High \mid No) \times P(True \mid No) \times P(No)$$

$$P(No \mid X) = 3/5 \times 1/5 \times 4/5 \times 3/5 \times 5/14 = 0.02057 \rightarrow 0.8 = \frac{0.02057}{0.02057 + 0.00529}$$

- The class with the **highest posterior probability is the outcome of prediction. Ie, Not to play Golf**

NAIVE BAYES CLASSIFIER

EXAMPLE - 3

Example No.	Color	Type	Origin	Stolen?
1	Red	Sports	Domestic	Yes
2	Red	Sports	Domestic	No
3	Red	Sports	Domestic	Yes
4	Yellow	Sports	Domestic	No
5	Yellow	Sports	Imported	Yes
6	Yellow	SUV	Imported	No
7	Yellow	SUV	Imported	Yes
8	Yellow	SUV	Domestic	No
9	Red	SUV	Imported	No
10	Red	Sports	Imported	Yes

$$p(Yes) = \frac{5}{10} = 0.5$$

$$p(No) = \frac{5}{10} = 0.5$$

Color	Yes	No
Red	3/5	2/5
Yellow	2/5	3/5

Type	Yes	No
Sports	4/5	2/5
SUV	1/5	3/5

Origin	Yes	No
Domestic	2/5	3/5
Imported	3/5	2/5

$$New\ Instance = (Red, SUV, Domestic)$$

$$P(Yes|New\ Instance) = p(Yes) * P(Color = Red|Yes) * P(Type = SUV|Yes) * P(Origin = Domestic|Yes)$$

$$P(Yes|New\ Instance) = \frac{5}{10} * \frac{3}{5} * \frac{1}{5} * \frac{2}{5} = \frac{3}{125} = 0.024$$

$$P(No|New\ Instance) = p(No) * P(Color = Red|No) * P(Type = SUV|No) * P(Origin = Domestic|No)$$

$$P(No|New\ Instance) = \frac{5}{10} * \frac{2}{5} * \frac{3}{5} * \frac{3}{5} = \frac{9}{125} = 0.072$$

$$P(No|New\ Instance) > P(Yes|New\ Instance)$$

Ans : NO

Example-4

Given all the previous patients I've seen (below are their symptoms and diagnosis)...

chills	runny nose	headache	fever	flu?
Y	N	Mild	Y	N
Y	Y	No	N	Y
Y	N	Strong	Y	Y
N	Y	Mild	Y	Y
N	N	No	N	N
N	Y	Strong	Y	Y
N	Y	Strong	N	N
Y	Y	Mild	Y	Y

Do I believe that a patient with the following symptoms has the flu?

chills	runny nose	headache	fever	flu?
Y	N	Mild	Y	?

Example-5

age	income	student	Credit rating	Buys computer
<=30	high	no	fair	no
<=30	high	no	excellent	no
31...40	high	no	fair	yes
>40	medium	no	fair	yes
>40	low	yes	fair	yes
>40	low	yes	excellent	no
31...40	low	yes	excellent	yes
<=30	medium	no	fair	no
<=30	low	yes	fair	yes
>40	medium	yes	fair	yes
<=30	medium	yes	excellent	yes
31...40	medium	no	excellent	yes
31...40	high	yes	fair	yes
>40	medium	no	excellent	no

- Naïve Bayes for data with nominal attributes
- Given the training data in the table (Buy Computer data), predict the class of the following new example using Naïve Bayes classification:
- age<=30, income=medium, student=yes, credit-rating=fair

THANK YOU

UNIT-III

(Measuring Classifier Accuracy)

Topics :

- Measuring Classifier Accuracy
  - Confusion Matrix
  - Accuracy
  - Precision
  - Recall or Sensitivity
  - F-Score

Measuring Classifier Accuracy

- **Accuracy** is one metric for **evaluating classification models.**
- Accuracy is perhaps the **best-known machine learning model** used in **classification problems**.
- Every machine learning task can be broken down to either **Regression or Classification.**
- There are **many performance measures** to **calculate the Accuracy. Some of them are:**

CONTD..

- **Measuring Classifier Accuracy**
  - **Confusion Matrix**
  - **Accuracy**
  - **Precision**
  - **Recall or Sensitivity**
  - **F-Score**

CONFUSION MATRIX

- The **confusion matrix** is one of the most popular and widely used **performance measurement techniques** for **classification models.**
- Confusion Matrix as the name suggests gives us a **matrix as output** and describes the **complete performance of the model.**
- And it also used to determine the **performance of the classification models** for a **given set of test data.**

REAL TIME EXAMPLE (SCENARIO)

- To consider is if **a person is Pregnant or Not Pregnant**
- So here **Machine Leaning** is just need to **classify** the **person is Pregnant or Not Pregnant.**
- Here, we have tak**en more scenarios**, ie, we gave taken **4 scenarios.** And how do we put these **4 scenarios into the Confusion Matrix**.
- **So, Let's see the scenarios one by one.**

SCENARIO-1


TRUE POSITIVE (TP)
You are Pregnant
A person who is actually pregnant (positive) and classified as pregnant (positive). This is called TRUE POSITIVE (TP)

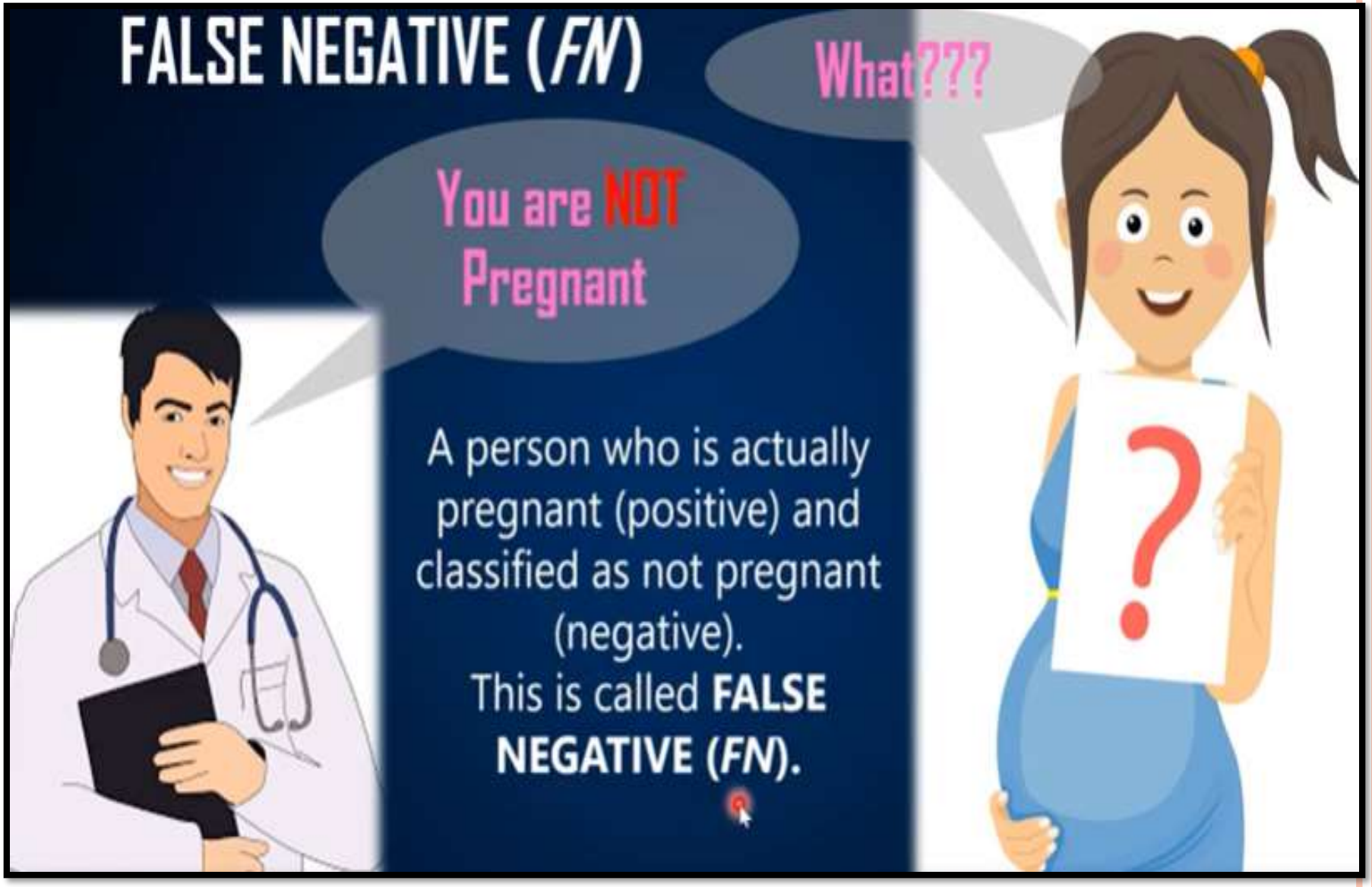
SCENARIO-2


TRUE NEGATIVE (TN)
You are NOT Pregnant
A person who is actually not pregnant (negative) and classified as not pregnant (negative).
This is called TRUE NEGATIVE (TN).
NOT PREGNANT
JUST FAT

SCENARIO-3


FALSE POSITIVE (FP)
You are Pregnant
A person who is actually not pregnant (negative) and classified as pregnant (positive).
This is called FALSE POSITIVE (FP)

SCENARIO-4


FALSE NEGATIVE (FN)
What???
You are NOT Pregnant
A person who is actually pregnant (positive) and classified as not pregnant (negative).
This is called FALSE NEGATIVE (FN).

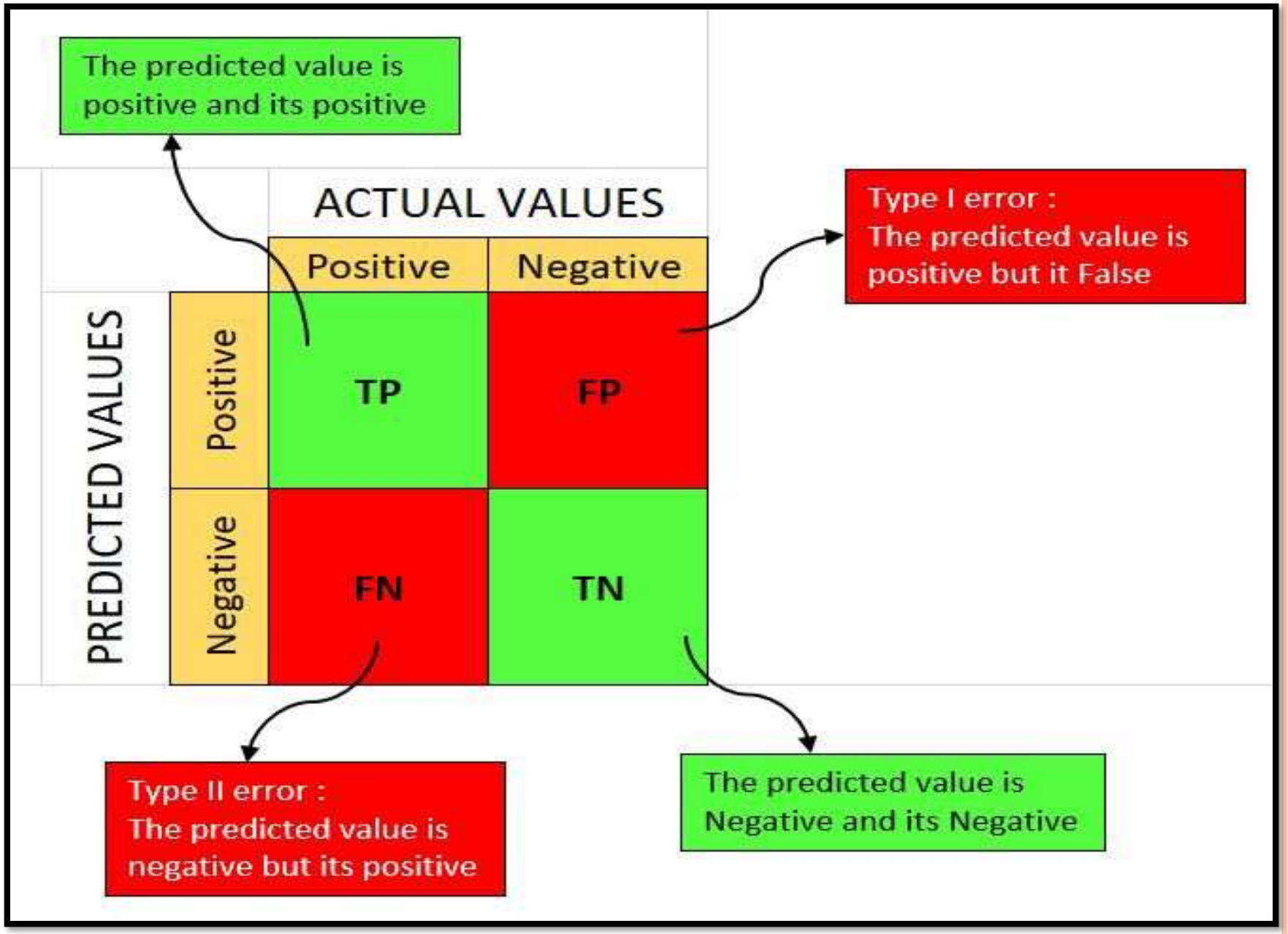
CONTD..

- Since it **shows the errors** in the model performance in the **form of a matrix**, hence also known as an **Error Matrix**. Some features of Confusion matrix are given below:
- **1.** For the **2 prediction classes of classifiers**, the matrix is of **2*2 table**, for **3 classes, it is 3*3 table**, and so on.
- **2.** The **matrix** is **divided into two dimensions,** that are **predicted values** and **actual values** along with the **total number of predictions.**

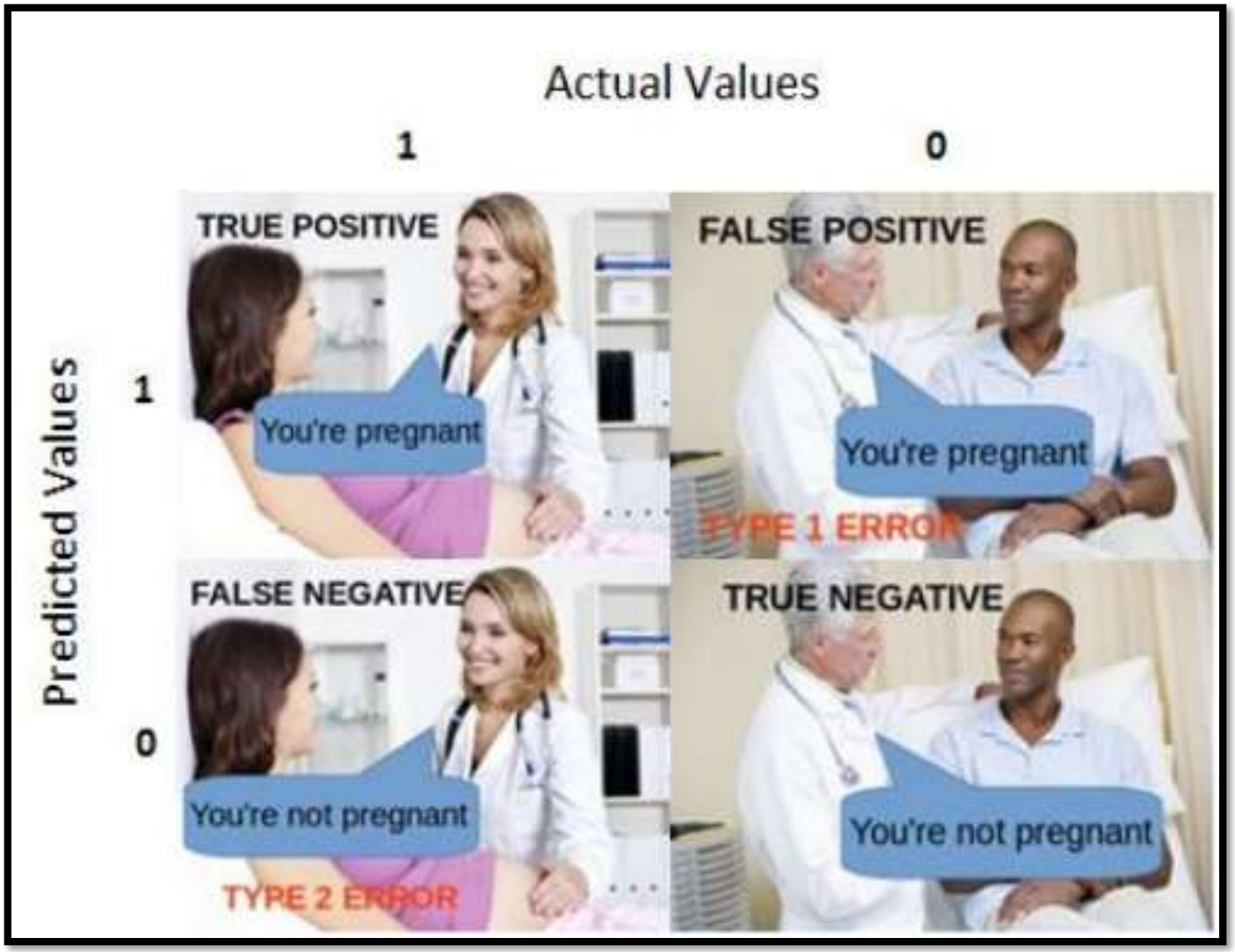
CONTD..

- **3. Predicted values are those values**, which are **predicted by the model**, and **actual values** are the **true values for the given observations**.
- **It looks like the below table:**
- For a **binary classification problem, we would have a 2 x 2 matrix as shown below with 4 values**:
- The target variable has two values: **Positive** or **Negative**
- The **columns** represent the **actual values** of the target variable
- The **rows** represent the **predicted values** of the target variable

CONTD..


The predicted value is positive and its positive
ACTUAL VALUES
Positive
Negative
PREDICTED VALUES
Positive
Negative
TP
FP
FN
TN
Type I error : The predicted value is positive but it False
Type II error : The predicted value is negative but its positive
The predicted value is Negative and its Negative

EXAMPLE-1


Actual Values
1
0
Predicted Values
1
0
TRUE POSITIVE
You're pregnant
FALSE POSITIVE
You're pregnant
TYPE 1 ERROR
FALSE NEGATIVE
You're not pregnant
TYPE 2 ERROR
TRUE NEGATIVE
You're not pregnant

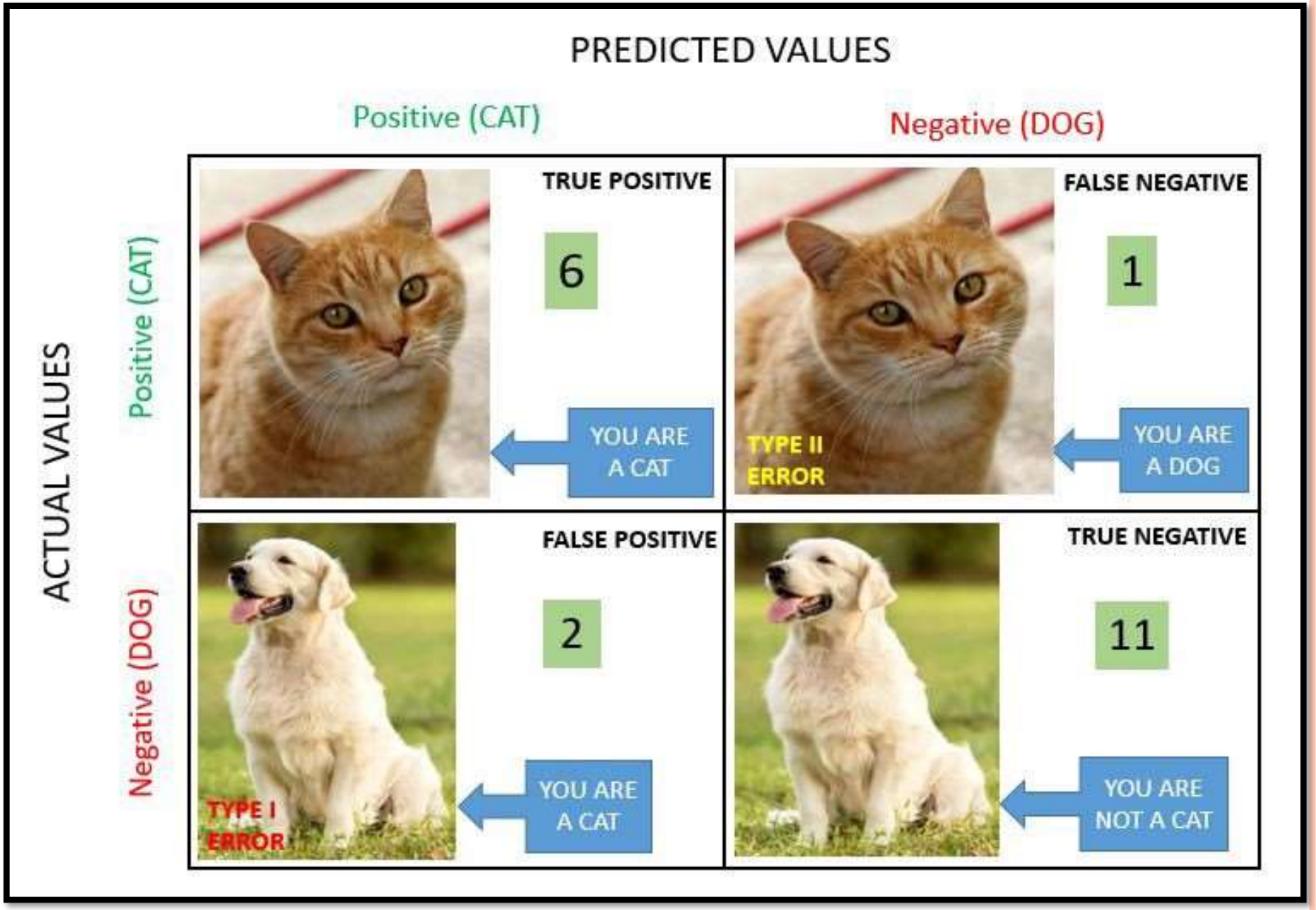
CONTD..

- The following **4 are the basic terminology** which will help us in **determining the metrics**. Those are:
- **1. True Positives (TP):** when the actual value is **Positive and predicted is also Positive.**
- **2. True negatives (TN):** when the actual value is **Negative and prediction is also Negative.**
- **3. False positives (FP):** When the **actual is negative but prediction is Positive.** Also known as the **Type 1 error**
- **4. False negatives (FN):** When the **actual is Positive but the prediction is Negative.** Also known as the **Type 2 error.**

EXAMPLE-2

- Based on the **Confusion Matrix**, We have to **calculate the Measures.**
- **In this Example**, We have a **total of 20 cats and dogs** and **our model predicts** whether **it is a cat or not.**
- **Actual values = ['dog', 'cat', 'dog', 'cat', 'dog', 'dog', 'cat', 'dog', 'cat', 'dog', 'dog', 'dog', 'dog', 'cat', 'dog', 'dog', 'cat', 'dog', 'dog', 'cat']**
- **Predicted values = ['dog', 'dog', 'dog', 'cat', 'dog', 'dog', 'cat', 'cat', 'cat', 'cat', 'dog', 'dog', 'dog', 'cat', 'dog', 'dog', 'cat', 'dog', 'dog', 'cat'].**

CONTD..


PREDICTED VALUES
Positive (CAT)
Negative (DOG)
ACTUAL VALUES
Positive (CAT)
Negative (DOG)
TRUE POSITIVE
6
YOU ARE A CAT
FALSE NEGATIVE
1
TYPE II ERROR
YOU ARE A DOG
FALSE POSITIVE
2
TYPE I ERROR
YOU ARE A CAT
TRUE NEGATIVE
11
YOU ARE NOT A CAT

CONTD..

- 1. **True Positive (TP) = 6**

You predicted positive and it's true. You predicted that an animal is a cat and it actually is.

- 2. **True Negative (TN) = *11***

You predicted negative and it's true. You predicted that animal is not a cat and it actually is not (it's a dog).

- 3. **False Positive (Type 1 Error) (FP) = 2**

You predicted positive and it's false. You predicted that animal is a cat but it actually is not (it's a dog).

- 4. **False Negative (Type 2 Error) (FN) = *1***

You predicted negative and it's false. You predicted that animal is not a cat but it actually is.

Classification Measures

- Basically, it is an **extended version** of the **confusion matrix.** There are **measures** other than the confusion matrix which can help **achieve better understanding** and **analysis of our model** and **its performance.**
- **Those are:**
  - **Accuracy**
  - **Precision**
  - **Recall or Sensitivity**
  - **F-Score**

1. ACCURACY

- It is one of the **important parameters** to determine the **accuracy** of the **classification problems**.
- It defines how often the **model predicts the correct output.**
- It can be calculated as the It's the **ratio between the number of correct predictions** and **the total number of predictions.**

$$\text{Accuracy} = \frac{\text{Number of correct predictions}}{\text{Total number of predictions}}$$

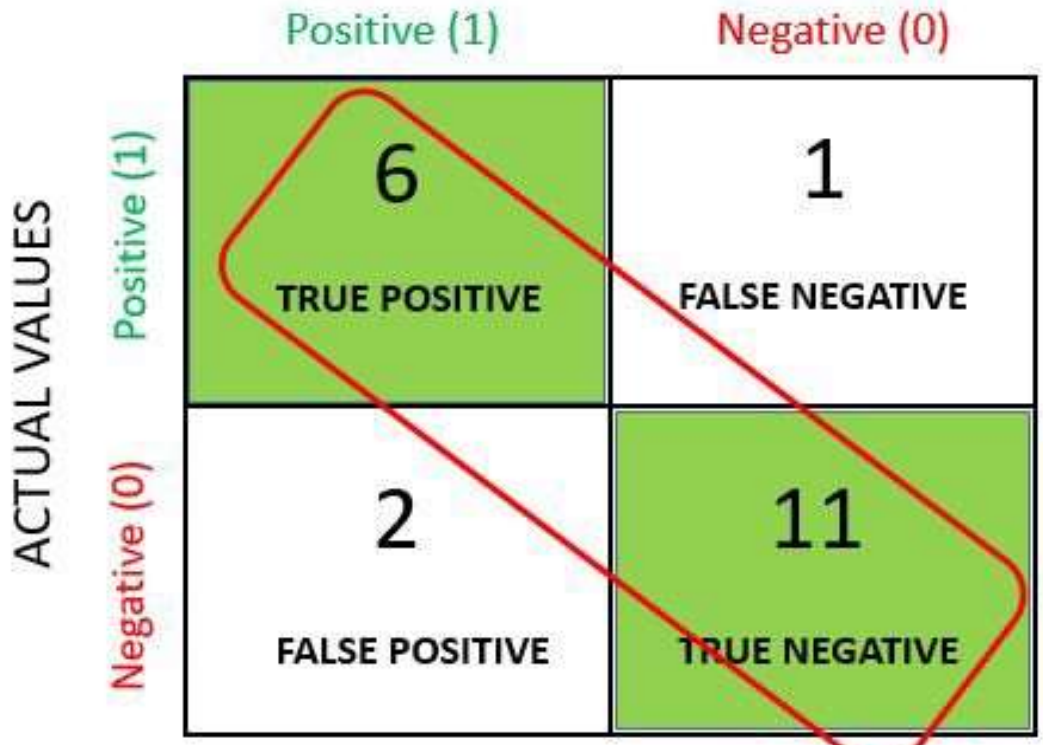
CONTD..

For binary classification, accuracy can also be calculated in terms of positives and negatives as follows:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

Where *TP* = True Positives, *TN* = True Negatives, *FP* = False Positives, and *FN* = False Negatives.

CONTD..



Accuracy

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} = \frac{\text{Correct Predictions}}{\text{Total Predictions}} = \frac{6 + 11}{6 + 11 + 2 + 1} = 85\%$$

2. PRECISION

- **Precision** is the share of **true positive predictions** in **all positive predictions**. In other words, it shows how often the **model is right** when **it predicts the target class.**
- It can be calculated using the below **formula:**

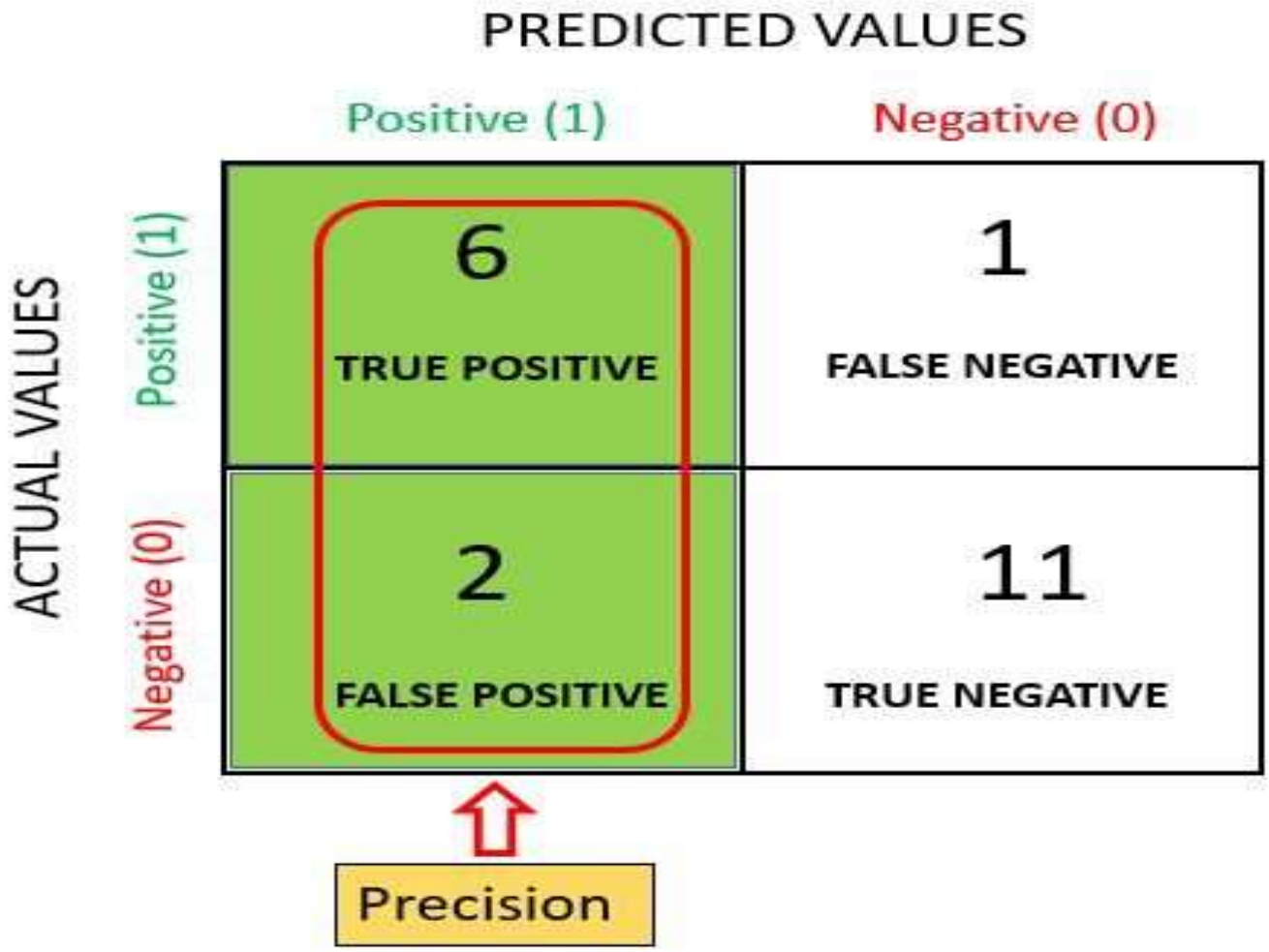
$$\text{Precision} = \frac{\text{True Positive}}{\text{Actual Results}} \quad \text{or} \quad \frac{\text{True Positive}}{\text{True Positive + False Positive}}$$

CONTD..

$$Precision = \frac{TruePositive}{TruePositive + FalsePositive}$$

$$precision = \frac{TP}{TP+FP}$$

CONTD..



$$\text{Precision} = \frac{TP}{TP + FP} = \frac{\text{Predictions Actually Positive}}{\text{Total Predicted positive}} = \frac{6}{6 + 2} = 0.75$$

3. Recall or Sensitivity

- **Recall** shows the share of **true positive predictions made by the model** out **of all positive samples in the dataset.**
- It can be calculated using the below **formula:**

$$\text{Recall} = \frac{\text{True Positive}}{\text{Predicted Results}} \quad \text{or} \quad \frac{\text{True Positive}}{\text{True Positive + False Negative}}$$

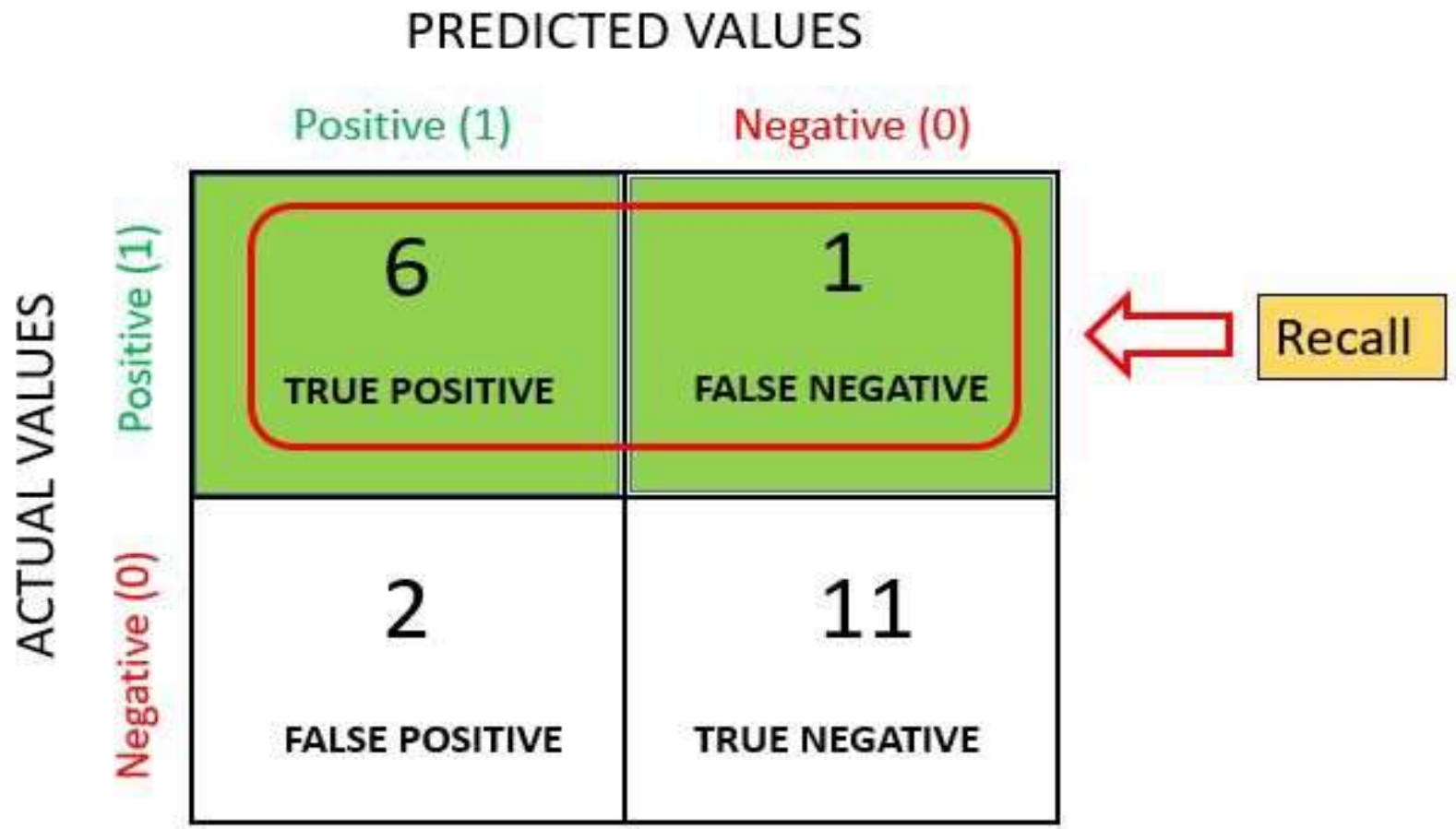
CONTD..

$$\text{Recall} = \frac{True\ Positive}{True\ Positive + False\ Negative}$$

Sensitivity or
Recall or
True Positive Rate

$$= \frac{TP}{(TP+FN)}$$

CONTD..



$$\text{Recall} = \frac{TP}{TP + FN} = \frac{\text{Predictions Actually Positive}}{\text{Total Actual positive}} = \frac{6}{6 + 1} = 0.85$$

4. F- Score / F- Measure

- An **F-Score** is a way to **measure a model's accuracy** based on **recall and precision**.
- The **F1 score** is a number between **0 and 1** and is the ***harmonic mean of precision and recall.***

$$\text{F1-Score} = 2^* \frac{(\text{Recall}^*\text{Precision})}{(\text{Recall} + \text{Precision})} = 2^* \frac{(0.85^*0.75)}{(0.85 + 0.75)} = 0.79$$

THANK YOU

UNIT-III

(EVALUATING HYPOTHESIS)

Topics :

- **Estimating Hypothesis Accuracy**
  - **Sample Error and**
  - **True Error**

ESTIMATING HYPOTHESIS ACCURACY

- **Hypothesis** is used to **classify the future instances.**
- In this classification we need to **estimate the accuracy of this Hypothesis.**
- **For Eg:** In the Class, we are having **60 Students**. Among 60 students, we have **30 students boys and 30 students girls.**
- Here, we have assumed hypothesis. i.e., 30 boys and 30 girls. **Is it true? Up to 90% true or 80% true etc.**
- Ie, here we need to **evaluate the hypothesis .**
- Finally we conclude that we have to **find the accuracy to all the learning examples.**

CONTD..

- Evaluating the hypothesis , we can mainly focus on **3 Questions:**
- **1.** Hypothesis is accurate over **limited sample of data**, then what about **additional data**.
- **2.** we have **2 hypothesis** – one is better than other when used on **sample of data**.
- **3.** When we have **limited data** , what is the best way to use this data for both **learning and estimating hypothesis**.
- So, here we can use this **Probable Error**, which is used to **compute the Accuracy**.(i.e, what error bars to associate with this Estimate).

MOTIVATION

- Sometimes, **very precise hypothesis** is required. In that cases estimating the accuracy is very important.
- Eg: **Medical Treatment.**

ESTIMATING HYPOTHESIS ACCURACY

- **Let us make some assumptions:**
- There is some space of possible instances "X" over which **various target functions** may be defined.
- Different instances will have the different instances. Based on the frequency , each instance in '**x1,x2---xn'** will have some **unknown probability**.
- **Eg: Rolling a Die. Rolling '1' for 10 times.**
- Ie, here the **probability is unknown.**
- So here **Trainer will teach the machine about the training examples of "Target Function".**

CONTD..

- So , here **trainer will teach** the **particular instance**, **"xi".**

  Ie, one particular instance for the set of instances.
- Suppose here the target function **is $f(x)= x^2$**
- So, in this target function , the output is generated

  F(x) = (0,1).
- So, here classifies , each instance into **different categories based on requirement.**
- Now, after classification, what is the **Probable error** in this estimation made.

CONTD..

- **Errors are 2 types : 1. Sample Error 2. True Error**
- **Sample Error:** The error rate of hypothesis over a **sample of data.**
- Ie, Estimate of **true error calculated on a data sample.**

The sample error of h with respect to target function f and data sample S is the proportion of examples h misclassifies

$$error_S(h) \equiv \frac{1}{n} \sum_{x \in S} \delta\left(f(x) \neq h(x)\right)$$

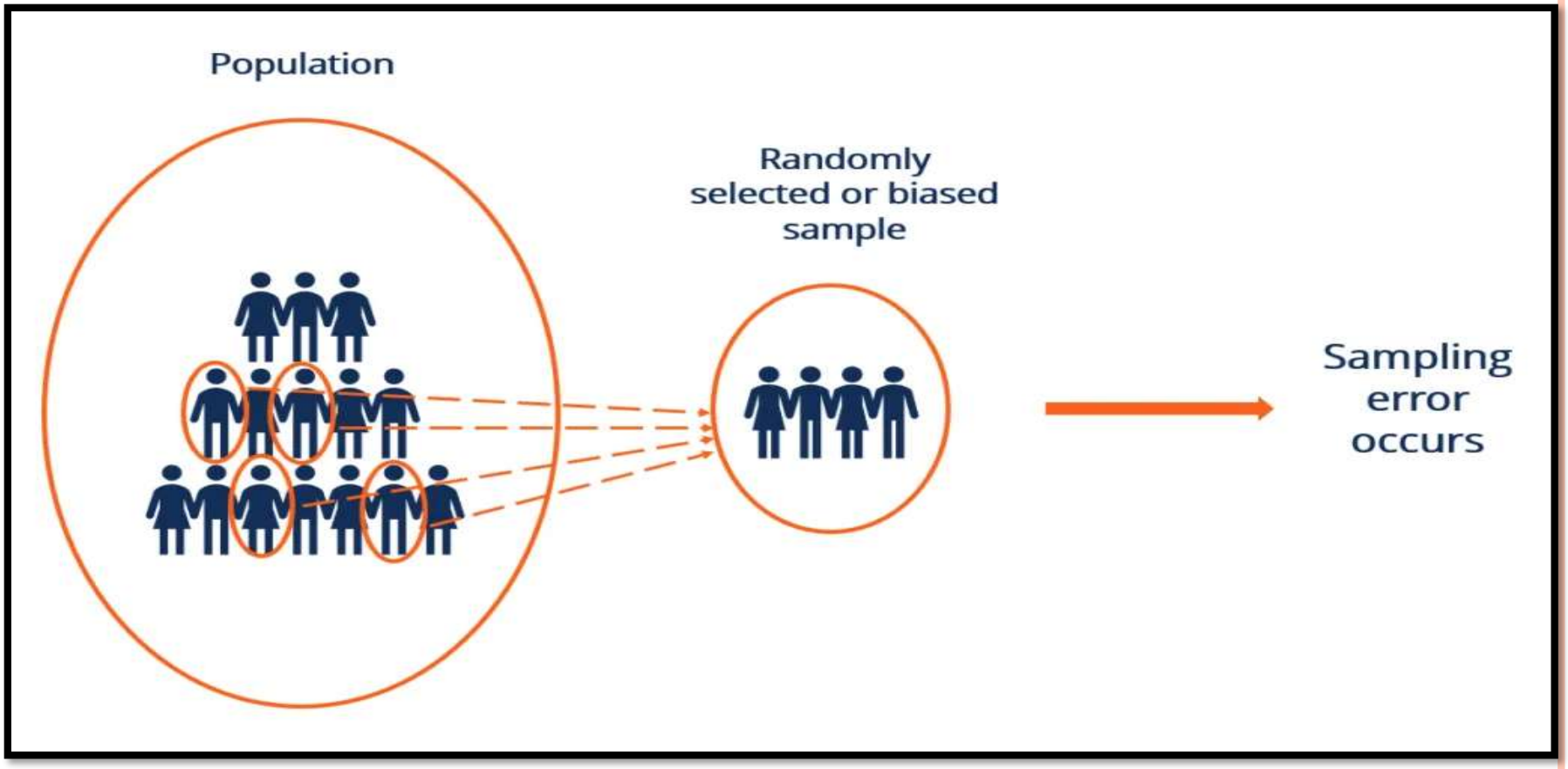
where $\delta\left(f(x) \neq h(x)\right)$ is 1 if $f(x) \neq h(x)$, and 0 otherwise

CONTD..

- Here, f(x) = **Learned Target Function**

  h(x) = **Predicted Function by the Hypothesis**

- **Sample Error** is the error Calculated w.r.t ***Data Sample Set S***.



CONTD..

- **Categories of Sample Errors:**
- 1. **Population Specification Error –** Happens when the **analysts do not understand** who to survey.
- **For example**, for a survey of breakfast cereals, the population can be the mother, children, or the entire family.
- 2. **Sample Frame Error –** Occurs when a sample is selected from the wrong population data.

CONTD..

2. True Error:

The true error of hypothesis h with respect to target function f and distribution D is the probability that h will misclassify an instance drawn at random according to D.

$$error_D(h) \equiv \Pr_{x \in D}[f(x) \neq h(x)]$$

For Eg: True Error is the **error** Calculated with respect to ***Data Distribution D***. Here we can see that **Sample Error** is 0.2 (i.e 1 out of 5) from the *OUR SAMPLE* circle & **True Error** is 0.5 (i.e 10 out of 20) from all the DATA DISTRIBUTION

CONTD..

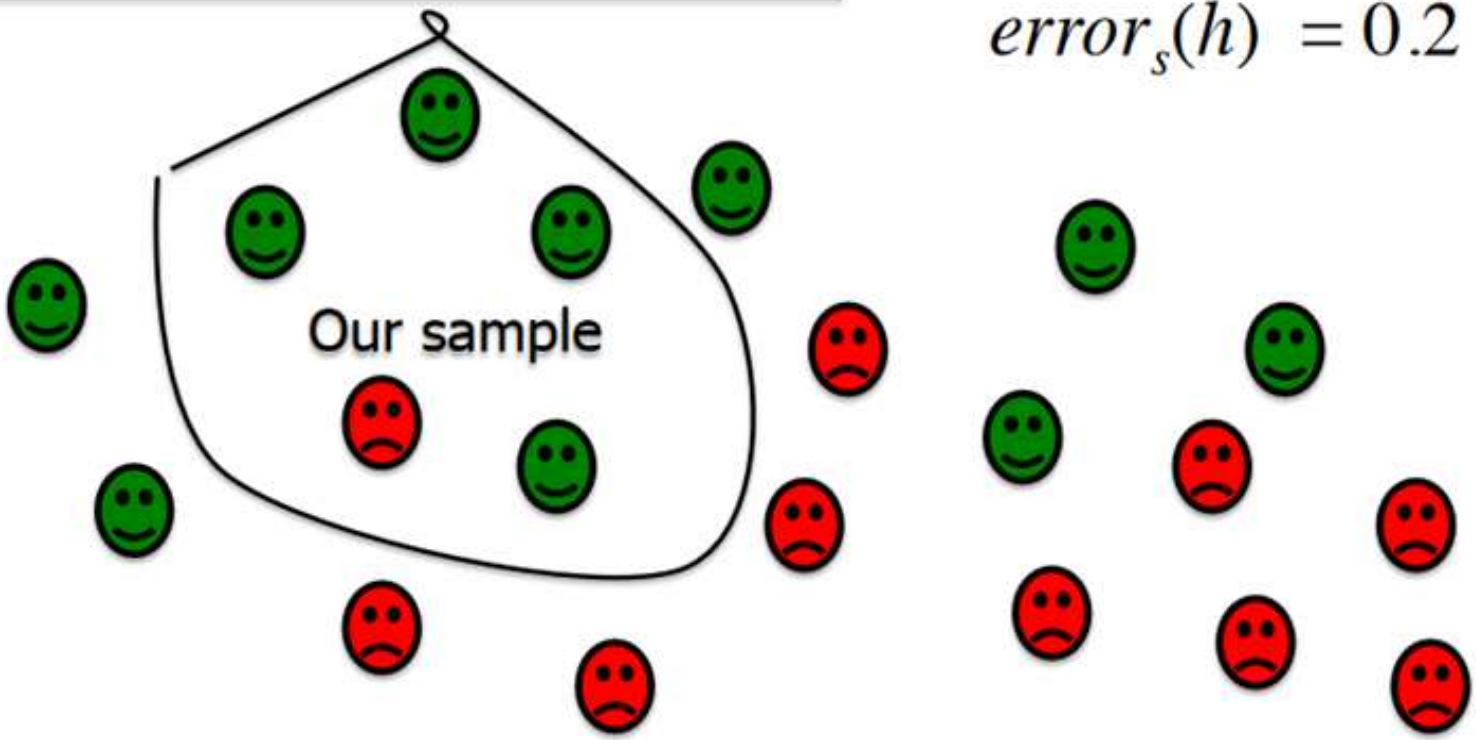
True Error vs Sample Error

Misclassified: h(x) != f(x)

Correctly classified: h(x) = f(x)

$$error_D(h) = 0.5$$

$$error_s(h) = 0.2$$



THANK YOU

UNIT-III

(ENSEMBLE METHODS)

Topics :

- Ensemble Methods
  - Bagging
  - Boosting

ENSEMBLE METHODS

- In **simple English**, Ensemble is nothing but **group of items.**
- **Ensemble methods** is a machine learning technique that **combines several base models** in order to **produce one optimal predictive model.**
- It is a powerful method to **improve the performance of the model.**
- The main Aim is to combine all these models , to **increase the Accuracy.**

CONTD..

- **For Eg:** When you want to **purchase a laptop**, will you simply **walk up to the store and pick any laptop**? It's **highly unlikely.**

- Generally, You **would likely browse a few web portals where people have posted their reviews and compare different laptop models, checking for their features, specifications and prices**. You will also probably ask your **friends and colleagues for their opinion**.

CONTD..

- In short, you **wouldn't directly reach a conclusion**, but will **instead make a decision considering the opinions of other people as well.** Here also we can follow the **Ensemble Learning .**
- *An ensemble is itself a supervised learning algorithm,* because it can be **trained and then used to make predictions.**
- **By using the Ensemble Learning, we have 2 objectives:**
- **1. Reducing the Error**
- **2. Maintaining the Generalizations.**

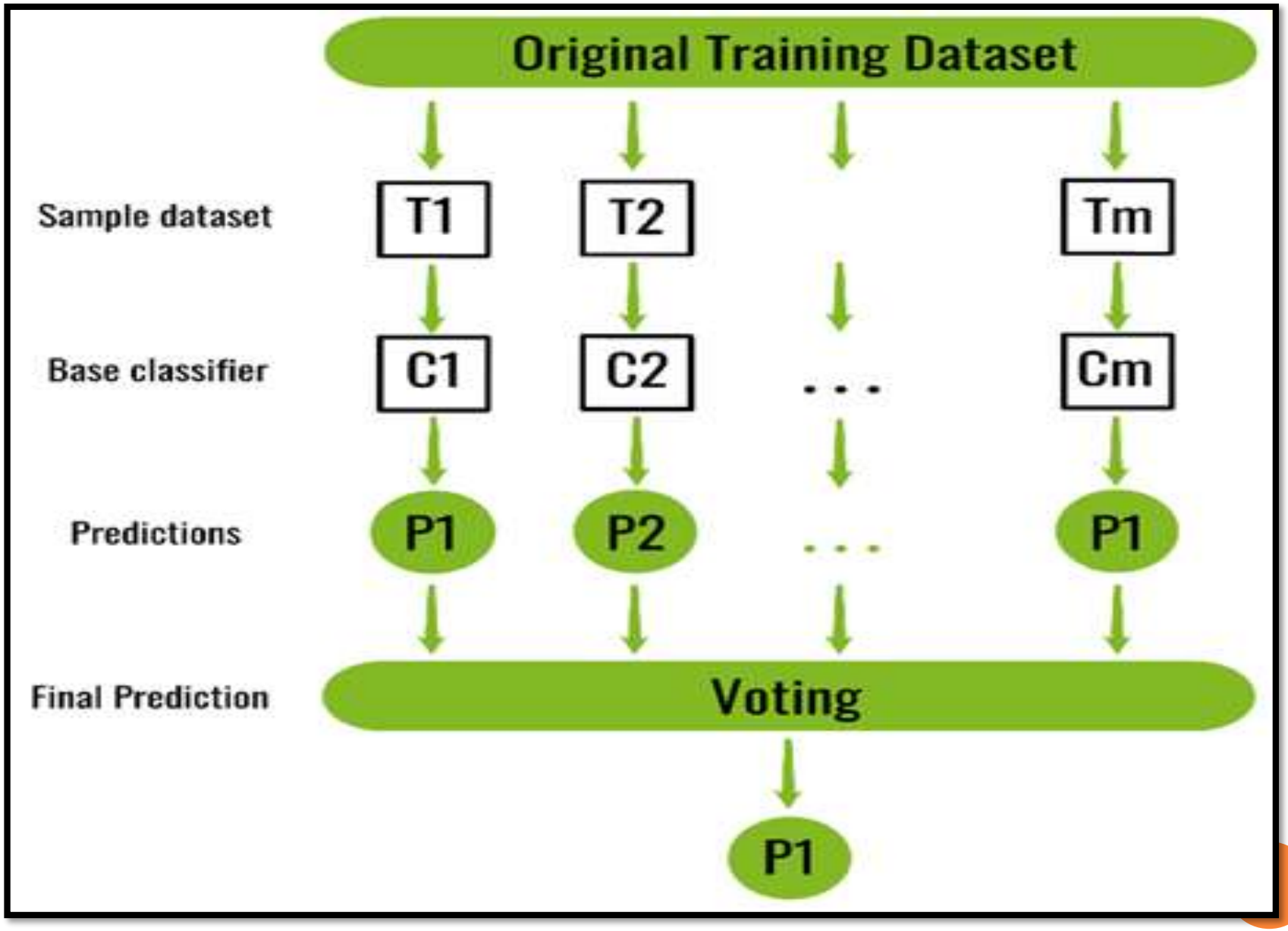
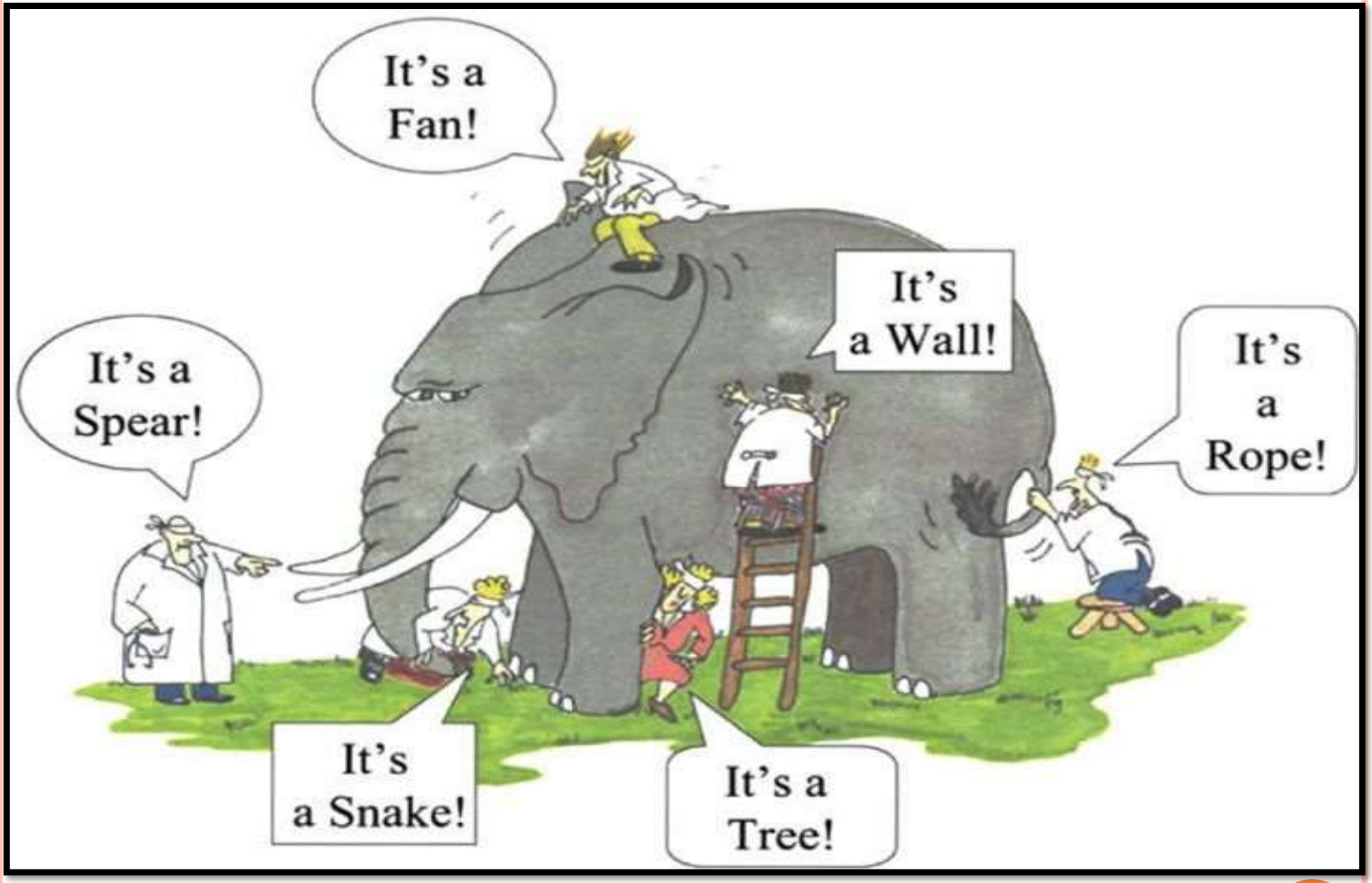


Figure: Ensemble Methods

Real Time Example

- Consider the fable of the **blind men** and the **elephant depicted** in the image below.
- The blind men are each describing an elephant from their **own point of view.**
- Their descriptions are **all correct but incomplete.** Their understanding of the elephant would be **more accurate and realistic** if they came together to discuss and combined their descriptions.



The main principle of ensemble methods is to **combine weak and strong learners to form strong learners**.

Types of Ensemble Techniques

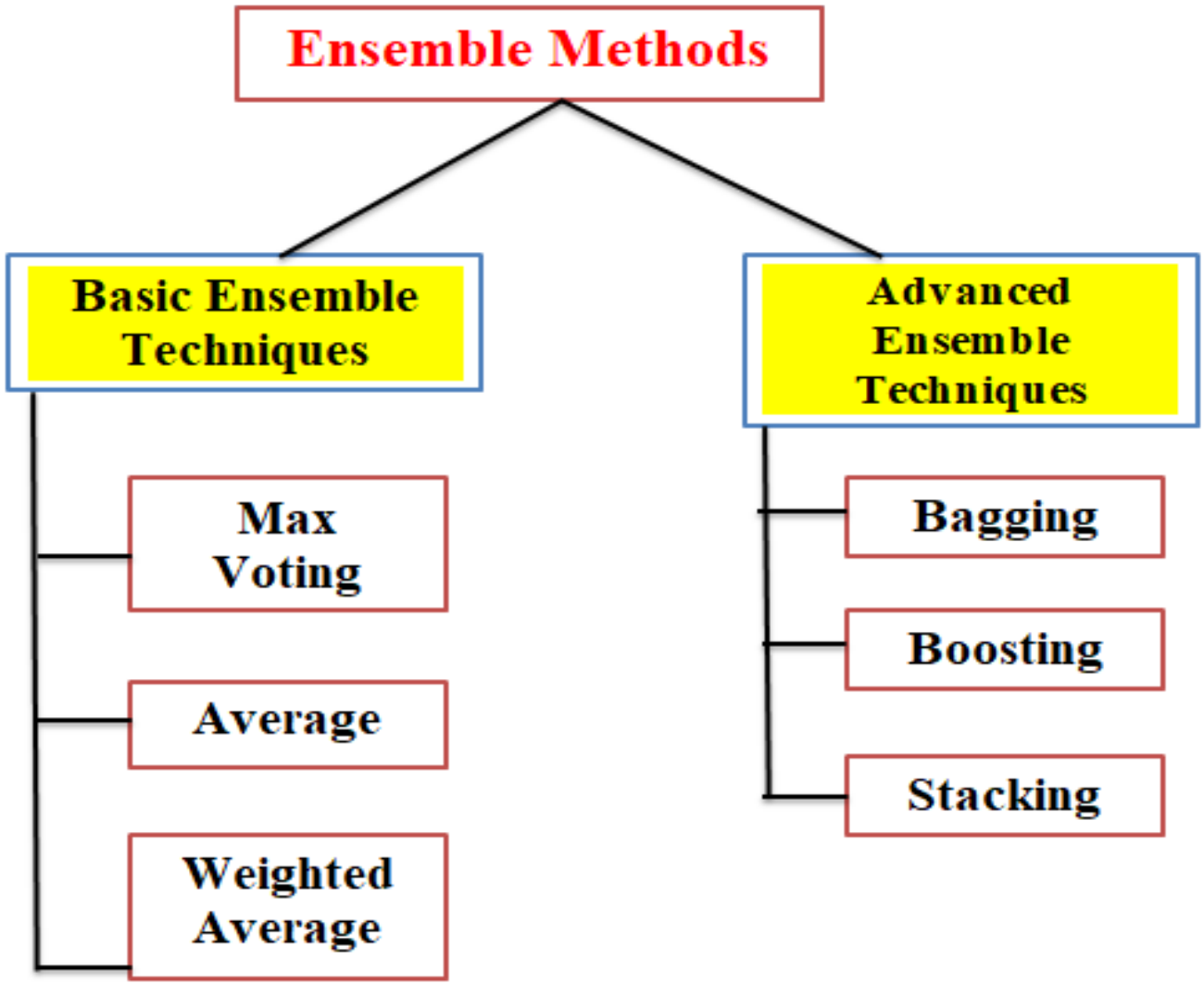


Figure: Types of Ensemble Techniques

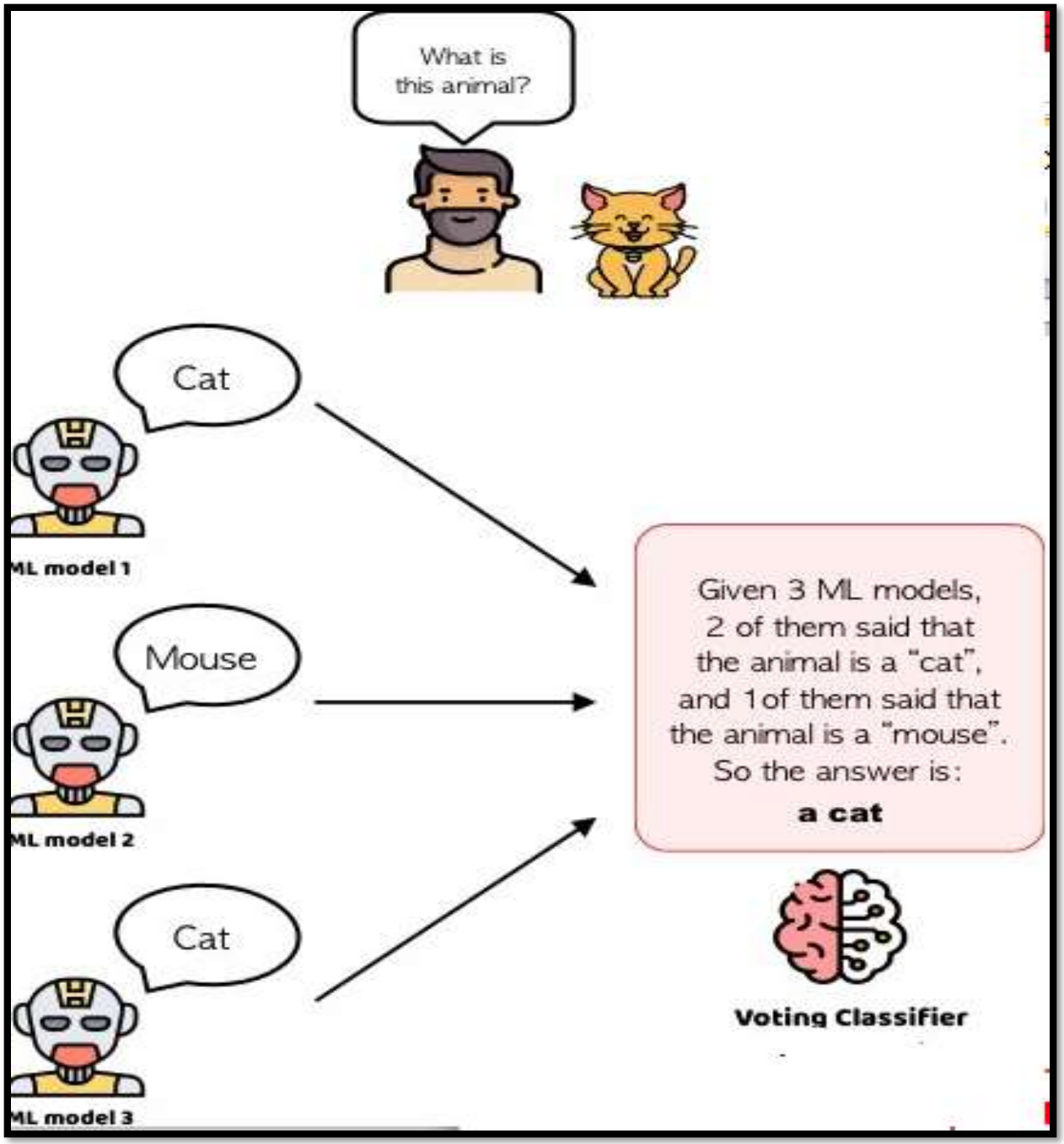
Basic Ensemble Techniques

- **1. Max Voting:** The **max voting** method is generally used for **classification problems.** In this technique, **multiple models are used to make predictions for each data point.**
- The predictions by each model are considered as **a 'vote'.**
- The predictions which we get from the **majority of the models are used as the final prediction.**
- **For Example:** when you **asked 5 of your colleagues** to **rate your movie (out of 5);**
- we'll assume **three of them rated it as 4** while **two of them** gave **it a 5**.

CONTD..

- Since the **majority gave a rating of 4,** the **final rating will be taken as 4.** You can consider this as taking the mode of all the predictions.
- **The result of max voting would be something like this:**

Colleague 1	Colleague 2	Colleague 3	Colleague 4	Colleague 5	Final Rating
5	4	5	4	4	4

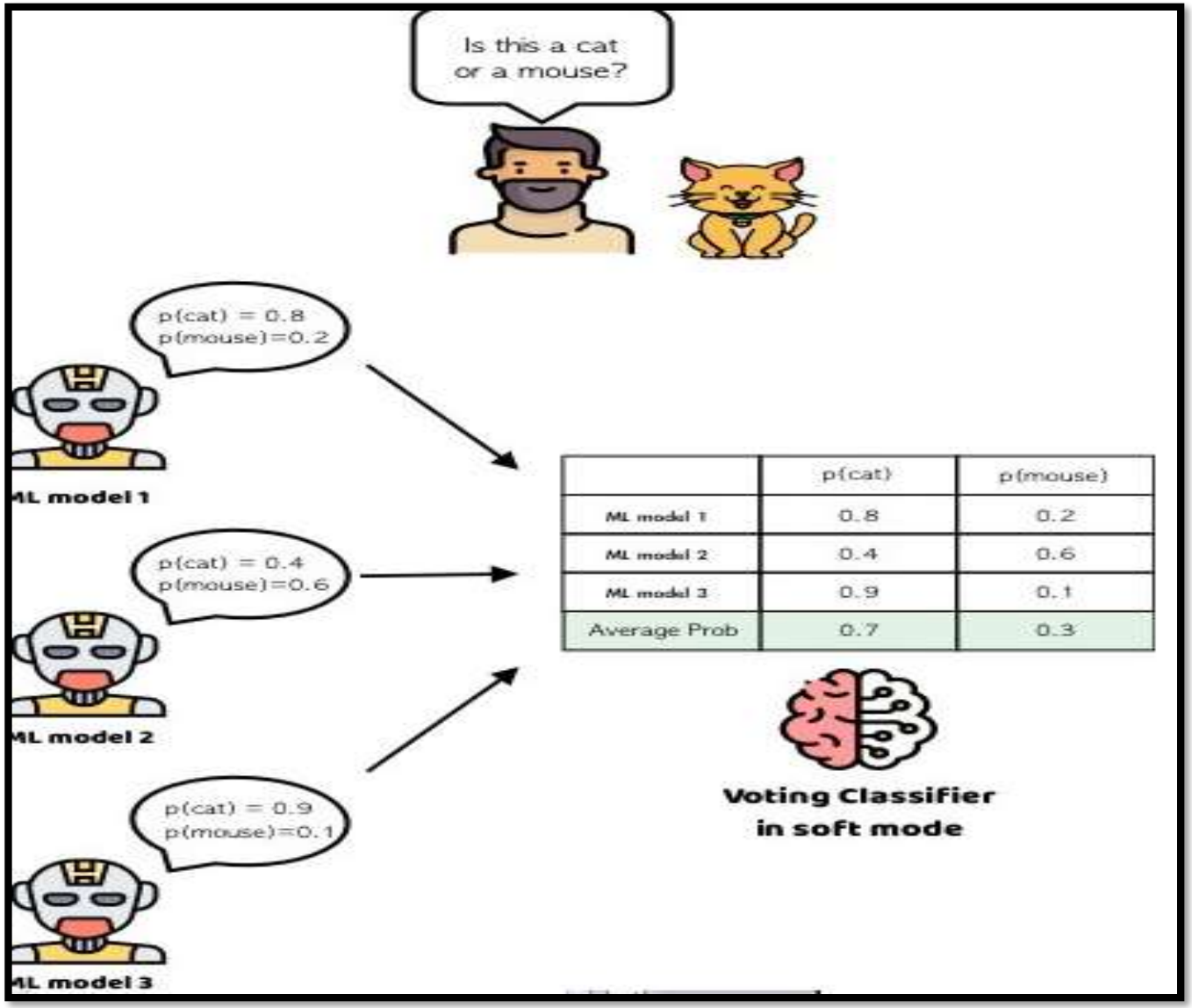

What is this animal?
Cat
ML model 1
Mouse
ML model 2
Cat
ML model 3
Given 3 ML models, 2 of them said that the animal is a "cat", and 1of them said that the animal is a "mouse". So the answer is:
a cat
Voting Classifier

CONTD..

- **2. Average Voting:** Similar to the **max voting technique**, **multiple predictions are made for each data point in averaging.**
- In this method, we take **an average of predictions from all the models** and use it to make the **final prediction**.
- Averaging can be used for making predictions in **regression problems** or while **calculating probabilities** for **classification problems.**
- **For example**, in the below case, the **averaging method would take the average of all the values.**

i.e. (5+4+5+4+4)/5 = 4.4

Colleague 1	Colleague 2	Colleague 3	Colleague 4	Colleague 5	Final Rating
5	4	5	4	4	4.4



CONTD..

- **3. Weighted Average:** This is an **extension of the averaging method**.
- All models are assigned **different weights** defining the **importance of each model for prediction**.
- For instance, if two of your **colleagues are critics**, while others have **no prior experience in this field**, then the answers by these **two friends are given more importance as compared to the other people.**

Colleague 1	Colleague 2	Colleague 3	Colleague 4	Colleague 5	Final Rating	
Weight	0.23	0.23	0.18	0.18	0.18	
Rating	5	4	5	4	4	4.41

- The result is calculated as

[(5*0.23) + (4*0.23) + (5*0.18) + (4*0.18) + (4*0.18)] = 4.41.

Advanced Ensemble Techniques

- **Bagging and Boosting** are the two very important ensemble methods to **improve the measure of accuracy** in **predictive models** which is widely used.
- While performing a machine learning algorithm we might come across **various errors** such as **noise** . To **overcome these errors** we apply **ensemble methods.**

CONTD..

- **1.Bagging:** Bagging is an **acronym** for ***Bootstrapped Aggregation*.** And it is a **simple and very powerful ensemble method.**
- It is mainly used in **Decision Trees.**
- **For Eg:** In the analogy, each friend will take the test separately, and then **their answers will be combined** to **form the final answer on the exam**. There are many ways of combining the answers, but the **most common way is voting.**
- **Bagging is mainly used in Random Forest Algorithm.**

How Bagging Works

- In 1996, Leo Breiman introduced the bagging algorithm, which has three basic steps:
  - **1. Bootstrapping**
  - **2. Parallel Training**
  - **3. Aggregation**
- **1. Bootstrapping:** Bootstrapping is a **sampling technique** in which **subsets of observations** from the **original dataset are generated.**
- This means that **each time you select a data point** from the **training dataset**, you are able to **select the same instance multiple times.**

CONTD..

2. Parallel Training: These bootstrap samples are then **trained independently** and **in parallel with each other using base learners**.

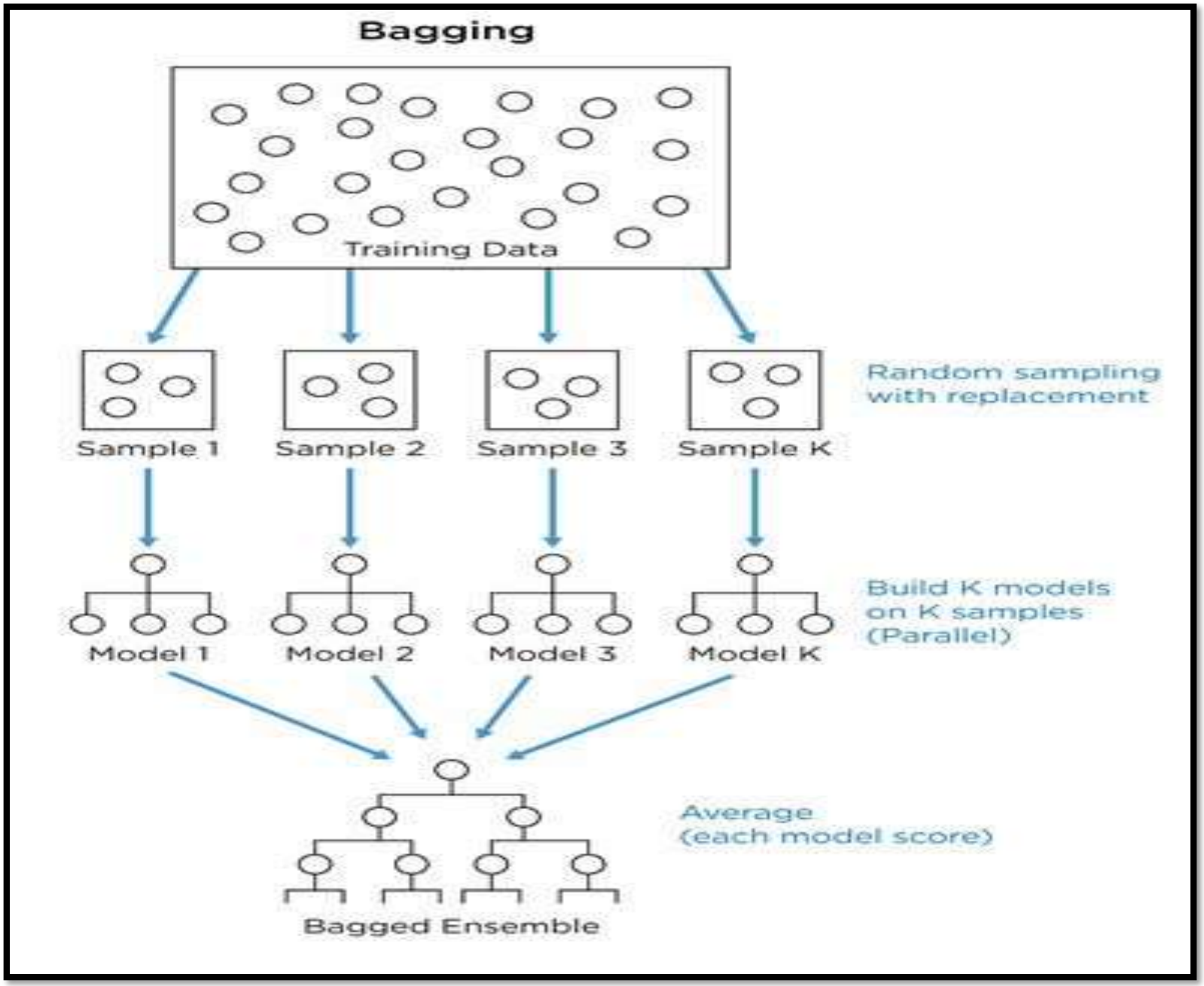
3. Aggregation: Finally, **depending on the task**(i.e. **regression or classification**), an **average or a majority of the predictions** are **taken** to compute the **Accurate Results.**

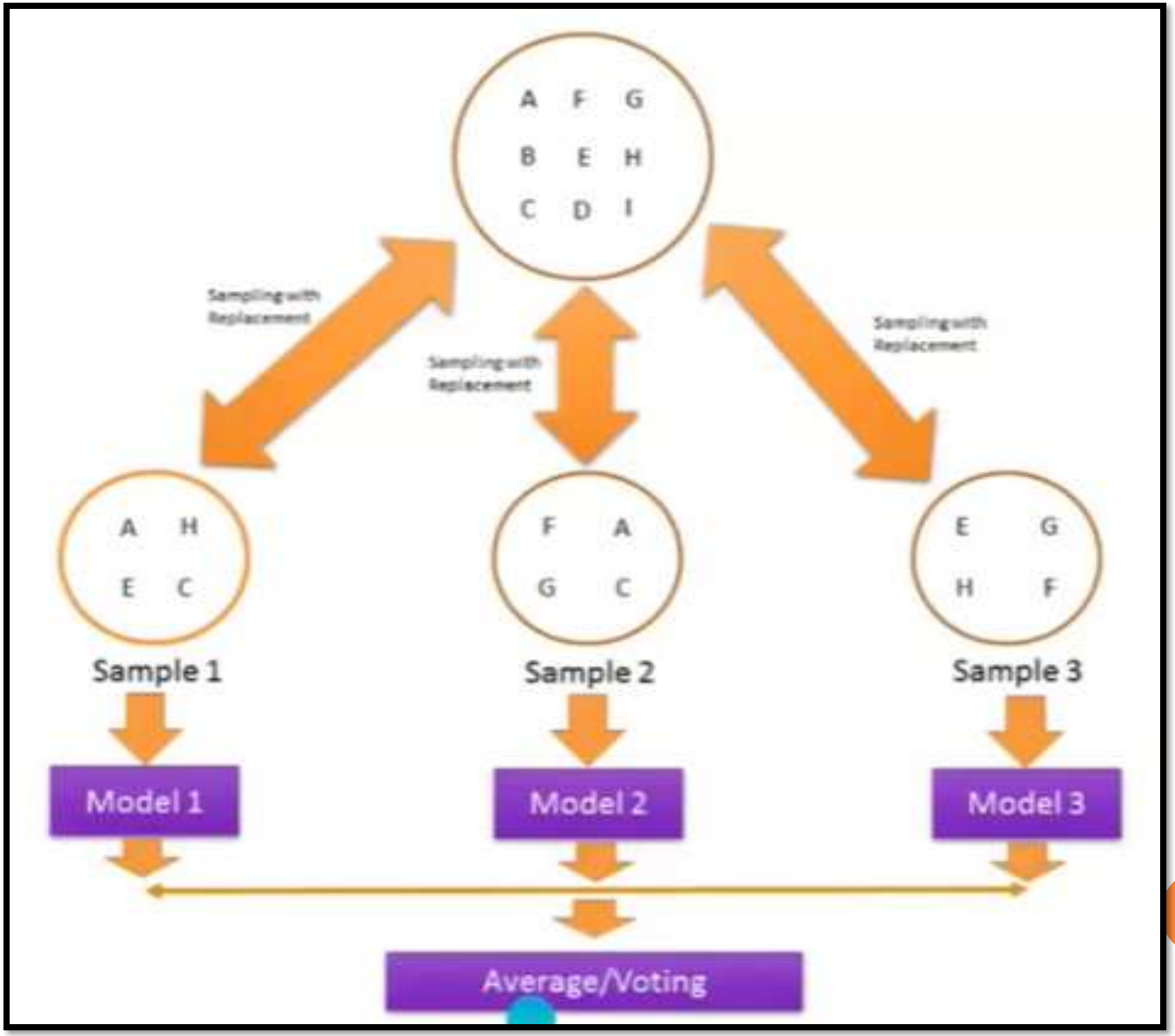
CONTD..

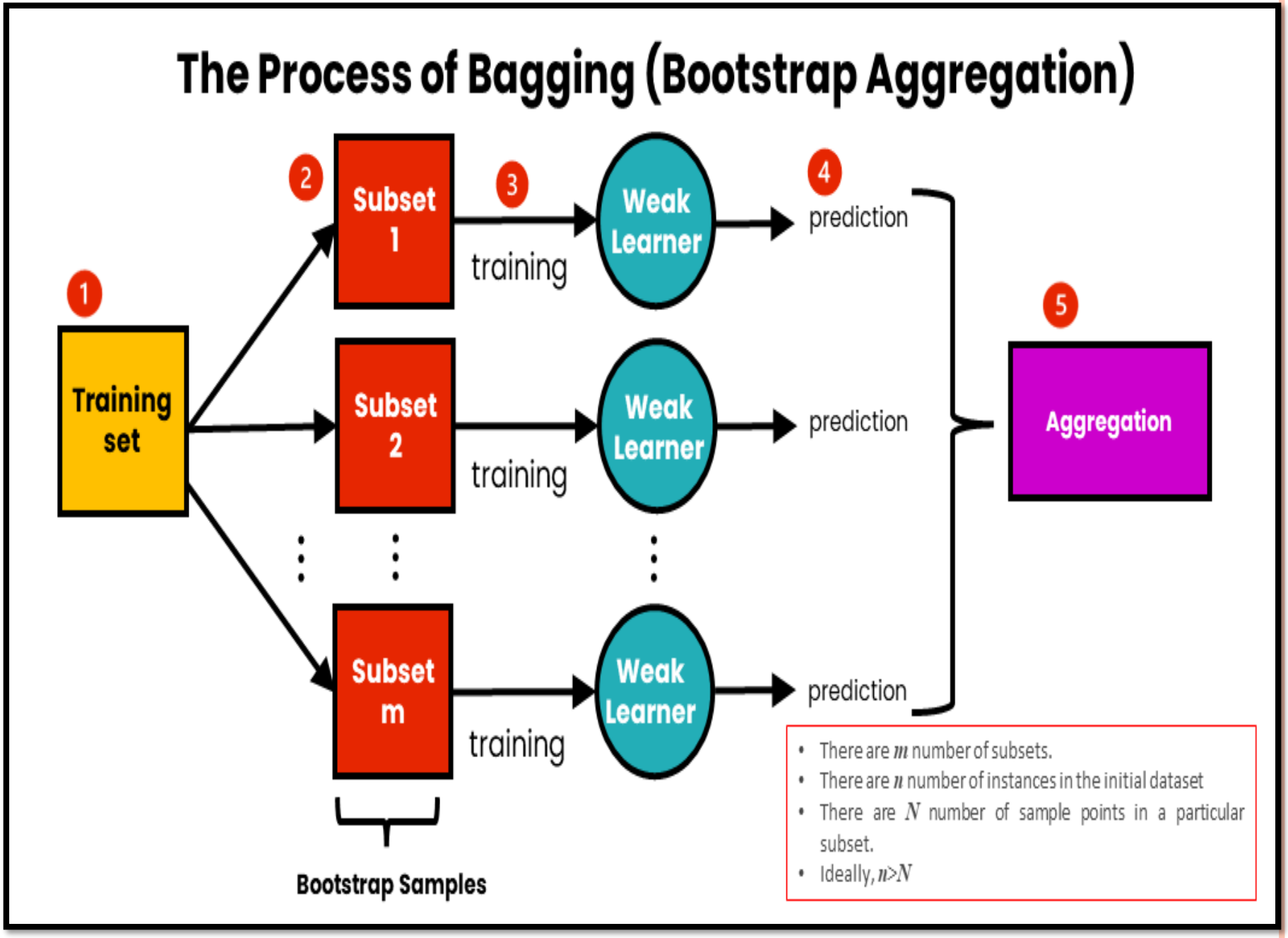
- In the **case of regression**, an **average is taken of all the outputs predicted** by the **individual classifiers**; this is known as **soft voting**.
- For **classification problems**, the class with the **highest majority of votes is accepted;** this is known as **hard voting or majority voting.**

Implementation Steps of Bagging

- **Step 1:** Multiple subsets are created from the original data set with equal tuples, selecting observations with replacement.
- **Step 2:** A base model is created on each of these subsets.
- **Step 3:** Each model is learned in parallel with each training set and independent of each other.
- **Step 4:** The final predictions are determined by combining the predictions from all the models.


Bagging
Training Data
Sample 1
Sample 2
Sample 3
Sample K
Random sampling with replacement
Model 1
Model 2
Model 3
Model K
Build K models on K samples (Parallel)
Average (each model score)
Bagged Ensemble


A F G
B E H
C D I
Sampling with Replacement
Sampling with Replacement
Sampling with Replacement
A H
E C
F A
G C
E G
H F
Sample 1
Sample 2
Sample 3
Model 1
Model 2
Model 3
Average/Voting


The Process of Bagging (Bootstrap Aggregation)
1
Training set
2
Subset 1
Subset 2
Subset m
3
training
training
training
Weak Learner
Weak Learner
Weak Learner
4
prediction
prediction
prediction
5
Aggregation
Bootstrap Samples
There are m number of subsets.
There are n number of instances in the initial dataset
There are N number of sample points in a particular subset.
Ideally, n>N

Random Forest Algorithm

- **Random Forest** is a popular **machine learning algorithm** that belongs to the **supervised learning technique.**
- It can be used for both **Classification and Regression** problems in ML.
- It is based on the concept of **ensemble learning**, which is a process of **combining multiple classifiers** to solve a **complex problem and to improve the performance of the model.**

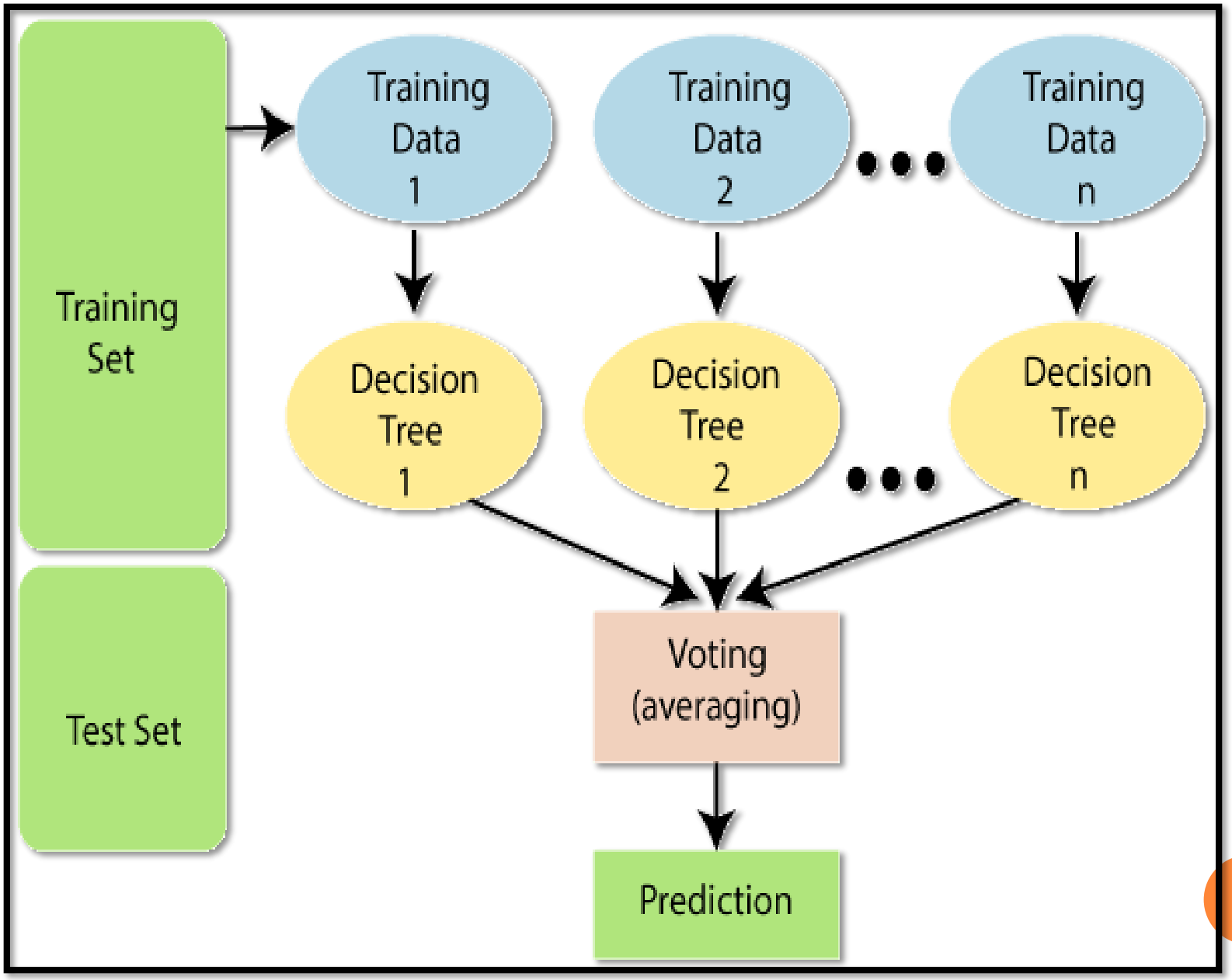
CONTD..

- As the name suggests, **"Random Forest" is a classifier** that contains a **number of decision trees** on **various subsets of the given dataset** and **takes the average to improve the predictive accuracy of that dataset**.
- Here the random forest takes **the prediction from each tree and based on the majority votes of predictions**, and it **predicts the final output.**
- The **main Aim is** the **greater number of trees in the forest leads to higher accuracy** and **prevents the problem of overfitting.**

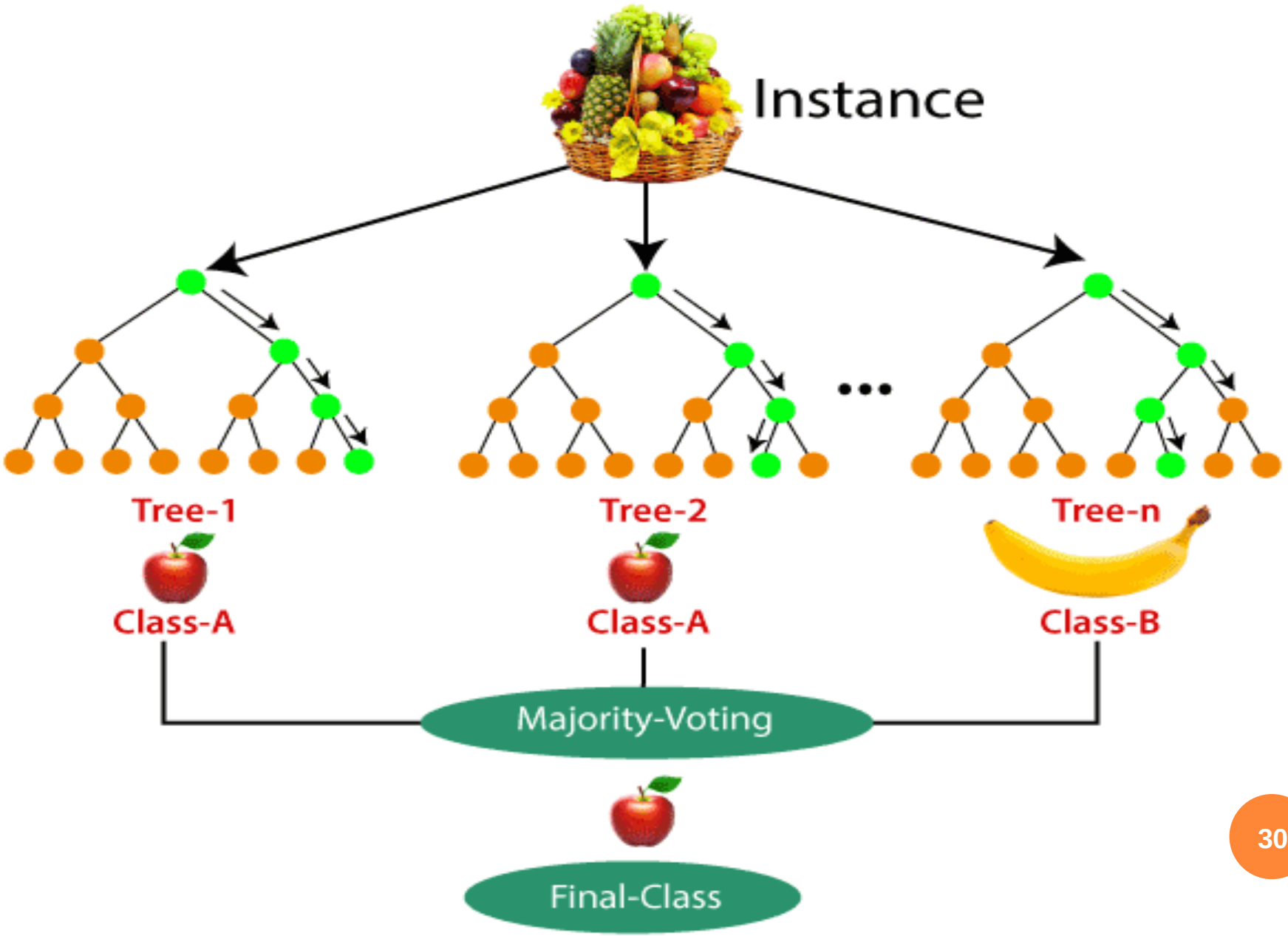
CONTD..

RANDOM FOREST

Random Forest: Algorithm Steps


Training Set
Test Set
Training Data 1
Training Data 2
Training Data n
Decision Tree 1
Decision Tree 2
Decision Tree n
Voting (averaging)
Prediction

Example:


Instance
Tree-1
Tree-2
Tree-n
Class-A
Class-A
Class-B
Majority-Voting
Final-Class

Applications of Random Forest

There are mainly **four sectors** where **Random forest mostly used:**

1. Banking: Banking sector mostly uses this algorithm for the **identification of loan risk.**

2. Medicine: With the help of this algorithm, **disease trends and risks of the disease can be identified.**

3. Land Use: We can identify the **areas of similar land use by this algorithm.**

4. Marketing: **Marketing trends** can be identified using this algorithm.

CONTD..

- **2.Boosting:** Gradient Boosting is an **ensemble learning technique** in which **models are not run independently** but **sequentially.**
- Boosting is a kind of algorithm that is able to **convert weak learners to strong learners.**
- In Boosting Technique, Each Algorithm. Ie, **base learners are trained sequentially and at every time** the **next learner is trying to reduce the error** by **updating the parameters** and perform **better comparison to the previous learners.**

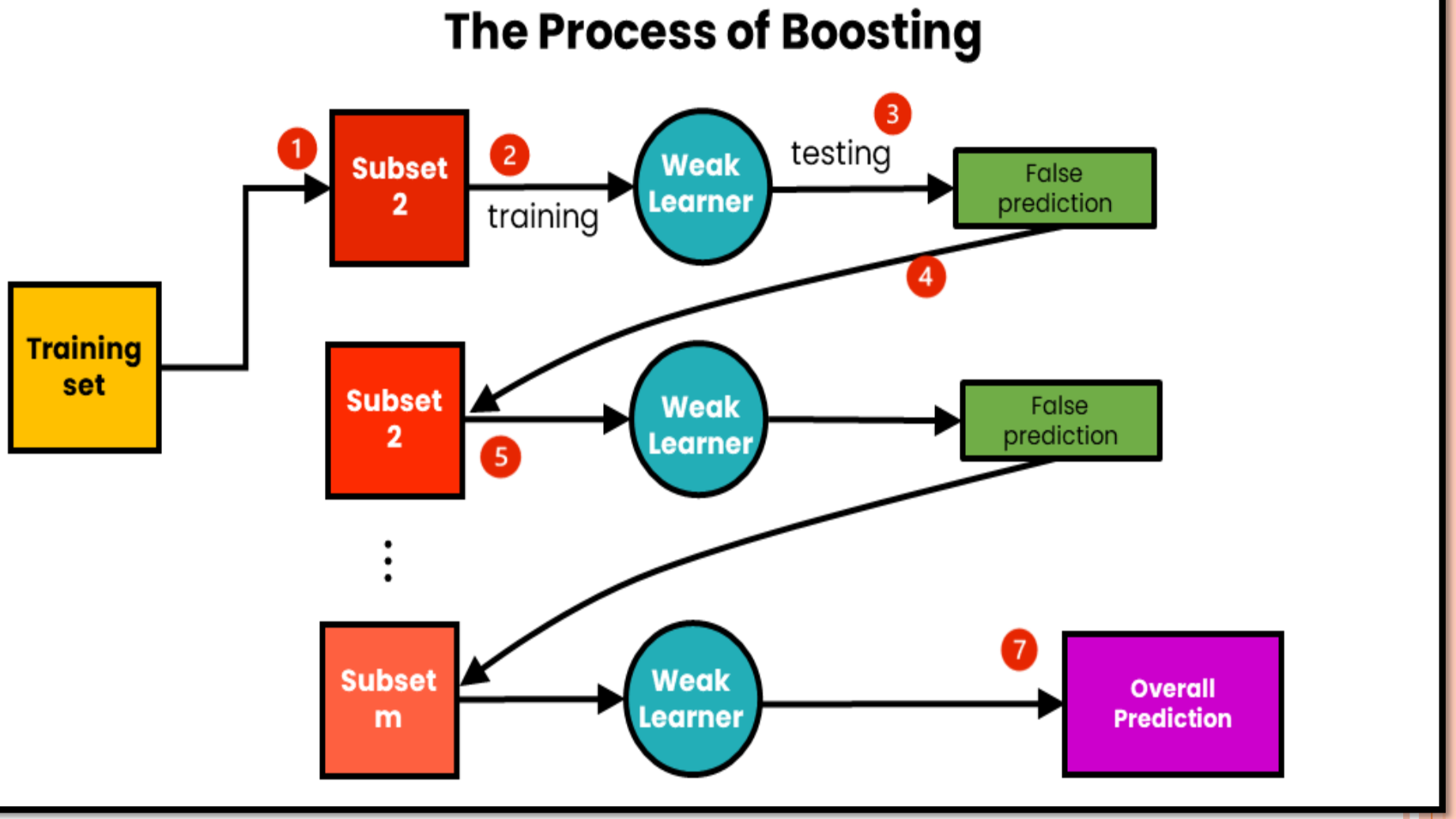
How Boosting Works

- Here's **what the entire process looks like:**
  - **1. Create a subset from the original data,**
  - 2. **Build an initial model with this data,**
  - 3. **Run predictions on the whole data set,**
  - 4. **Calculate the error using the predictions and the actual values, Assign more weight to the incorrect predictions.**

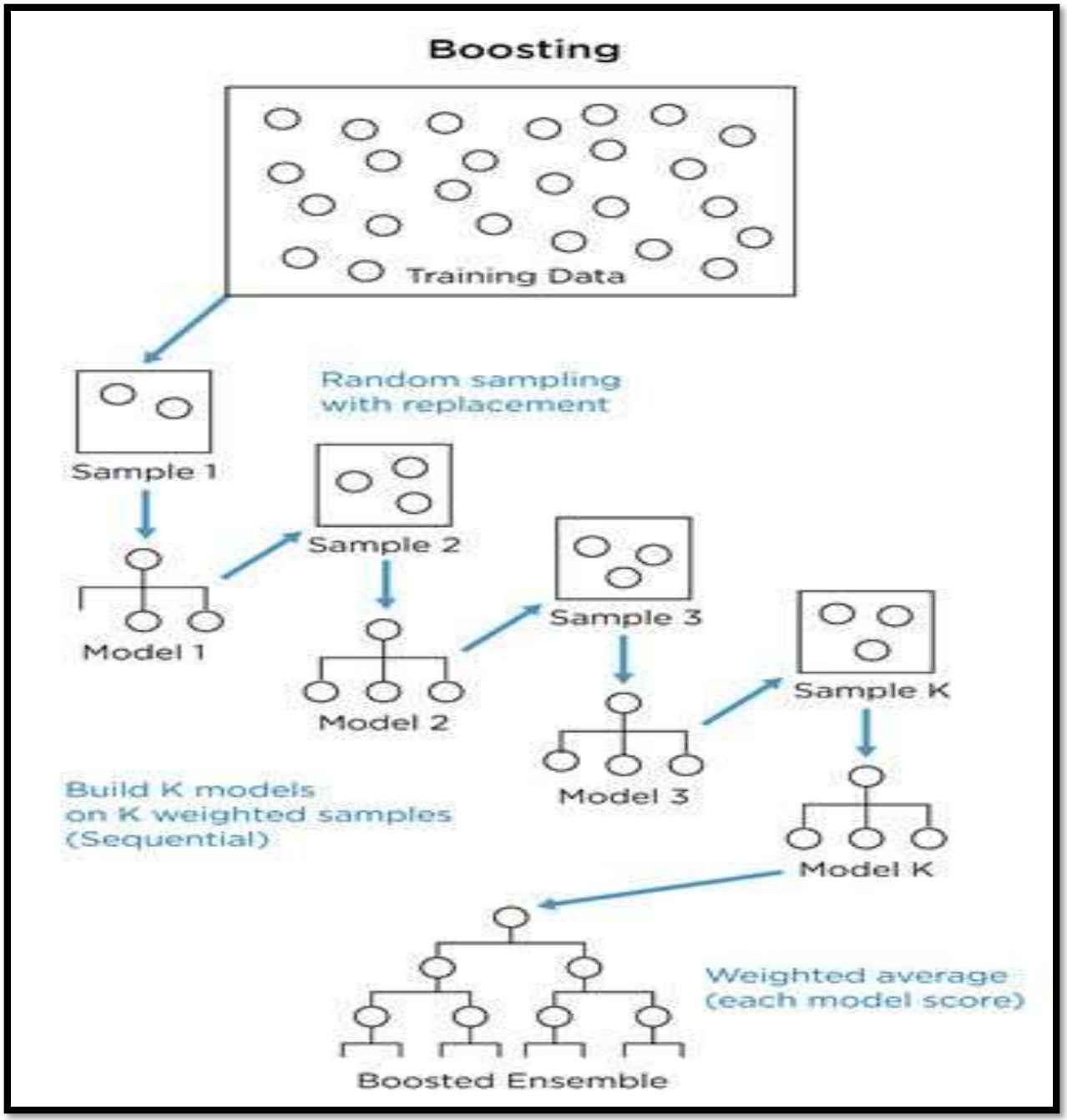
CONTD..

- Create another model that attempts **to fix errors from the last model,**
- Run predictions on the **entire dataset with the new model**,
- Create several models with each model aiming at correcting the errors generated by the previous one, Obtain the **final model by weighting the mean of all the models.**
- Boosting algorithms is the family of algorithms that combine weak learners into a strong learner.

CONTD..


The Process of Boosting
Training set
1
Subset 2
2
training
Weak Learner
3
testing
False prediction
4
Subset 2
5
Weak Learner
False prediction
Subset m
Weak Learner
7
Overall Prediction

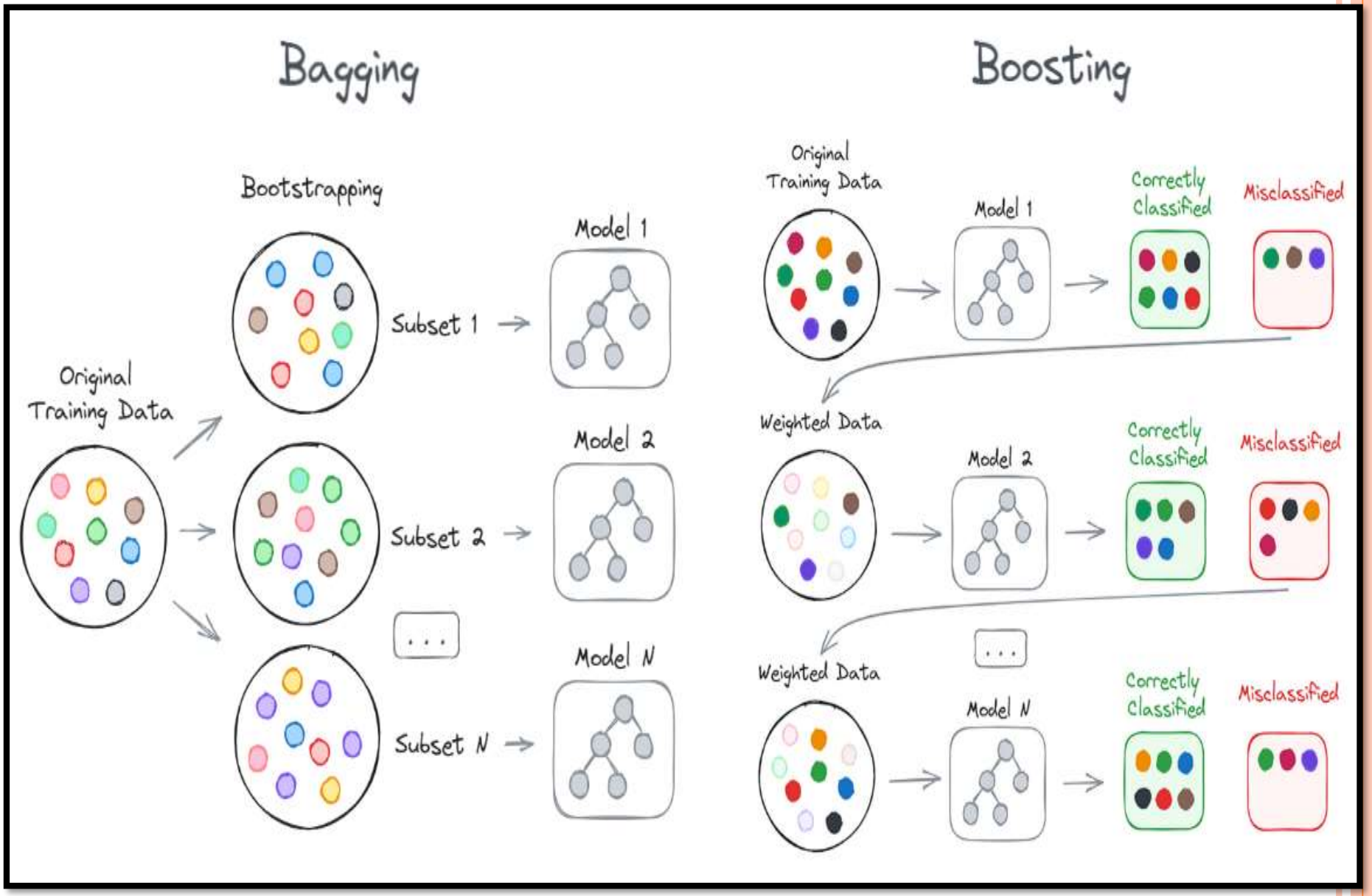
CONTD..


Boosting
Training Data
Random sampling
with replacement
Sample 1
Sample 2
Sample 3
Sample K
Model 1
Model 2
Model 3
Model K
Build K models
on K weighted samples
(Sequential)
Weighted average
(each model score)
Boosted Ensemble

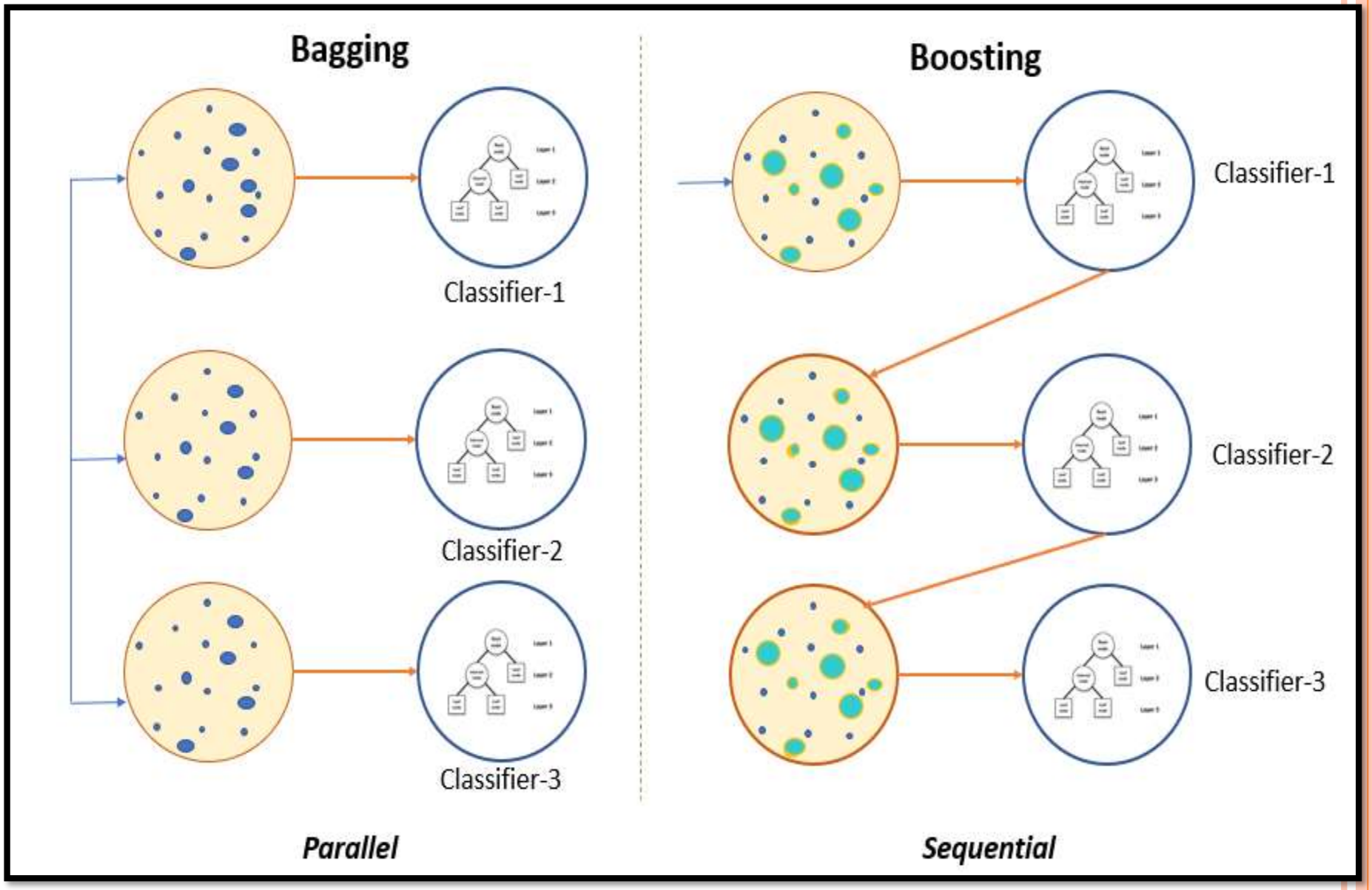
Differences between Bagging And Boosting

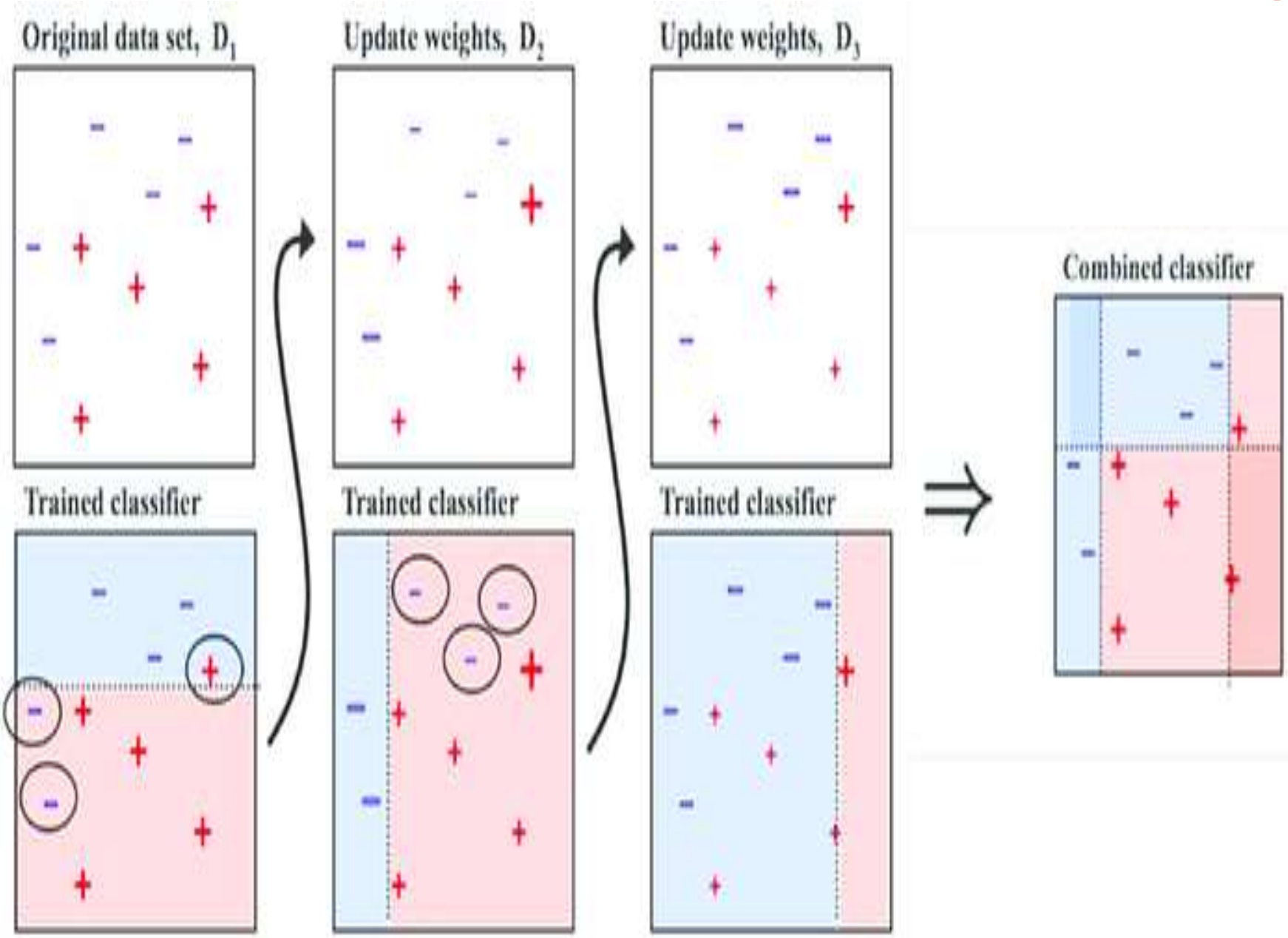
	Bagging	Boosting
Similarities	• Uses voting • Combines models of the same type	
Differences	Individual models are built separately	Each new model is influenced by the performance of those built previously
	Equal weight is given to all models	Weights a model's contribution by its performance

Differences between Bagging And Boosting



Differences between Bagging And Boosting




Original data set, D_1
Update weights, D_2
Update weights, D_3
Trained classifier
Trained classifier
Trained classifier
Combined classifier

CONTD..

Main Steps involved in boosting are :

- **Train model A** on the **whole set**
- **Train the model B** with exaggerated data on the **regions in which A performs poorly**
- …
- Instead of **training the models in *parallel***, we can **train them *sequentially*.** This is the **main idea of Boosting!**

CONTD..

- In order to provide you with an **idea about mail spam detection problem.**
- The **spam detection problem** can be **divided into the following steps.**
  - If the email **contains only an image?**
  - **Who is the sender?**
  - How **caps lock was used in the email?**
  - Check the **subject line in the email**

THANK YOU