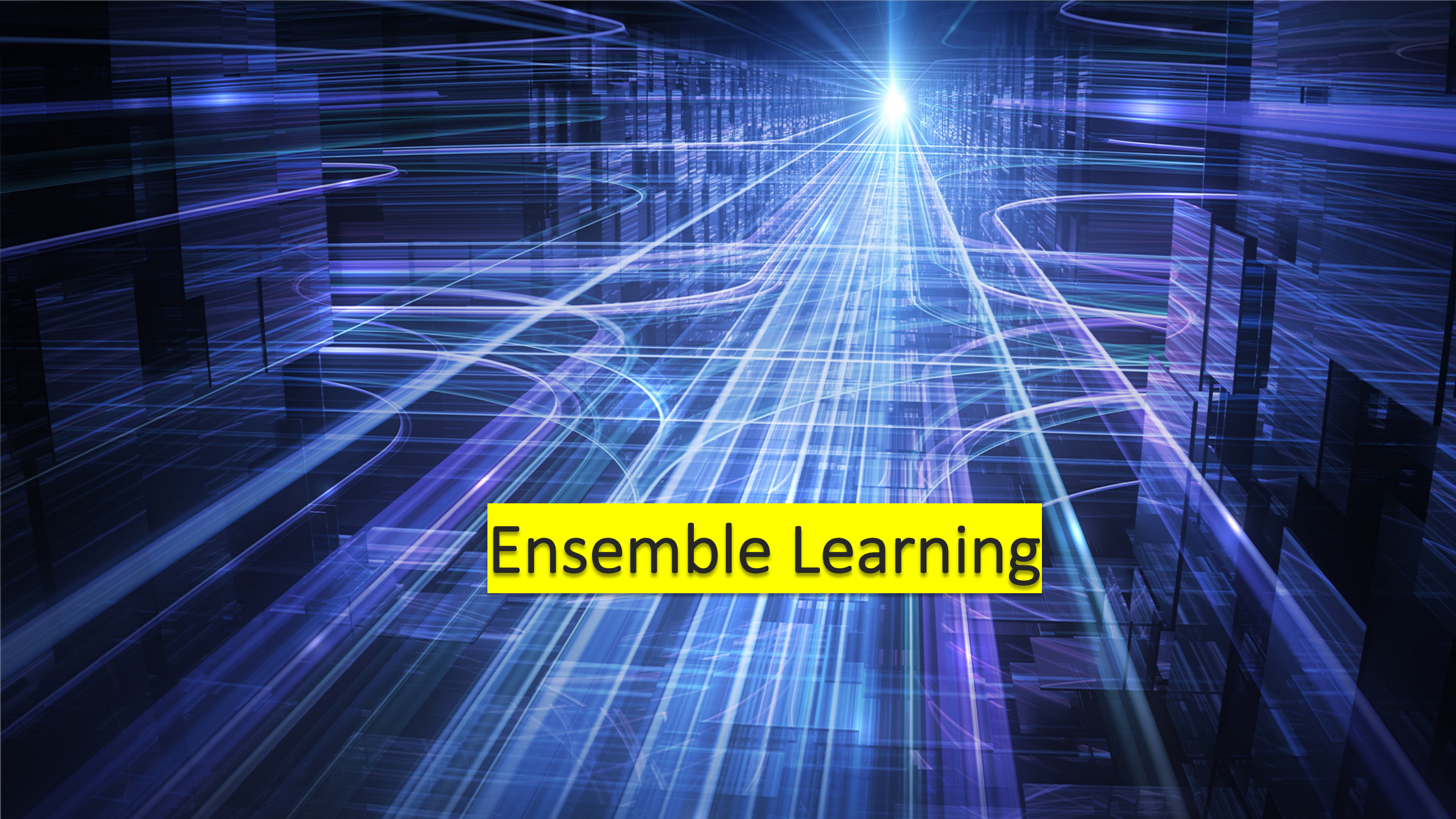

Ensemble Learning

Ensemble Learning

Bagging: RandomForest (n_estimators = 100)

Classification
eg: Spam vs Ham

DT1 → m1 → Ham
DT2 → m2 → Spam
⋮
DT100 → m100 → Ham

⇓

mode: Ham: 75
Spam: 25

∴ o/p is Ham.

Regression
eg: Predict sal. of an employee

⋮ Same process of 100 DT

⇓

Avg: Rs. 29357/-

=. Pred. Sal. will be Rs. 29357/-

Both r using n_estimators = 100 i.e. power of 100 trees to predict.
Hence we call it Bagging

Boosting: Sequential learning happens.

eg: ITeach Joshua (pvt tutions) – XII Std. (Mathematics)

Chapters: Linear Alg, Derivatives, Integration, Limits, Vectors, Matrices } Initial dataset
1 2 3 4 5 6 7

↓ Test 1

Res: Excellent: 1, 7
Poor: 2, 3, 4, 5, 6

↓

Test 2: QP → majority Q's r from Ch. 2, 3, 4, 5, 6 & v. few from Ch. 1, 7 } Adjusted dataset 1

Res: Excellent: 1, 2, 3, 7
Poor: 4, 5, 6

↓

Test 3: QP → majority Q's r from Ch. 4, 5, 6 & v. few from 1, 2, 3, 7 } Adjusted Dataset 2

Res: Excellent: all
Poor: None

∴ I STOP HERE.

→ Bagging
→ Boosting

INSAID

Course Overview

You are here...

Term	CDF	GCD	GCDAI	PGPDSAI
Term 1	Data Analytics with Python	Data Analytics with Python	Data Analytics with Python	Data Analytics with Python
Term 2	Data Visualization Techniques	Data Visualization Techniques	Data Visualization Techniques	Data Visualization Techniques
Term 3	EDA & Data Storytelling	EDA & Data Storytelling	EDA & Data Storytelling	EDA & Data Storytelling
		Minor Project	Minor Project	Minor Project
Term 4		Machine Learning Foundation	Machine Learning Foundation	Machine Learning Foundation
Term 5		Machine Learning Intermediate	Machine Learning Intermediate	Machine Learning Intermediate
Term 6		Machine Learning Advanced (Mandatory)	Machine Learning Advanced (Mandatory)	Machine Learning Advanced (Mandatory)
		Data Visualization with Tableau (Elective - I)	Data Visualization with Tableau (Elective - I)	Data Visualization with Tableau (Elective - I)
		Data Analytics with R (Elective - II)	Data Analytics with R (Elective - II)	Data Analytics with R (Elective - II)
		Capstone Project	Capstone Project	Capstone Project
Term 7		Bonus: Industrial ML (ML – 4 & 5)	Basics of AI, TensorFlow, and Keras	Basics of AI, TensorFlow, and Keras
Term 8			Deep Learning Foundation	Deep Learning Foundation
Term 9			NPL – I/CV – I	CV – I
Term 10			NLP – II/CV – II	NLP – I
			Capstone Project	Capstone Project
Term 11				CV – II
Term 12				NLP – II
				NLP – III + CV – III
				AutoVision & AutoNLP
				Building AI product

INSAID

Term Context

- K – Nearest Neighbor
- K-means Clustering
- **Ensemble Learning**  You are here…
- Optimization

INSAID

Agenda

1. **Ensemble Learning**
2. Ensemble Learning Types
3. Bagging
4. Boosting
5. Random Forest
6. Voting
7. Stacking

INSAID

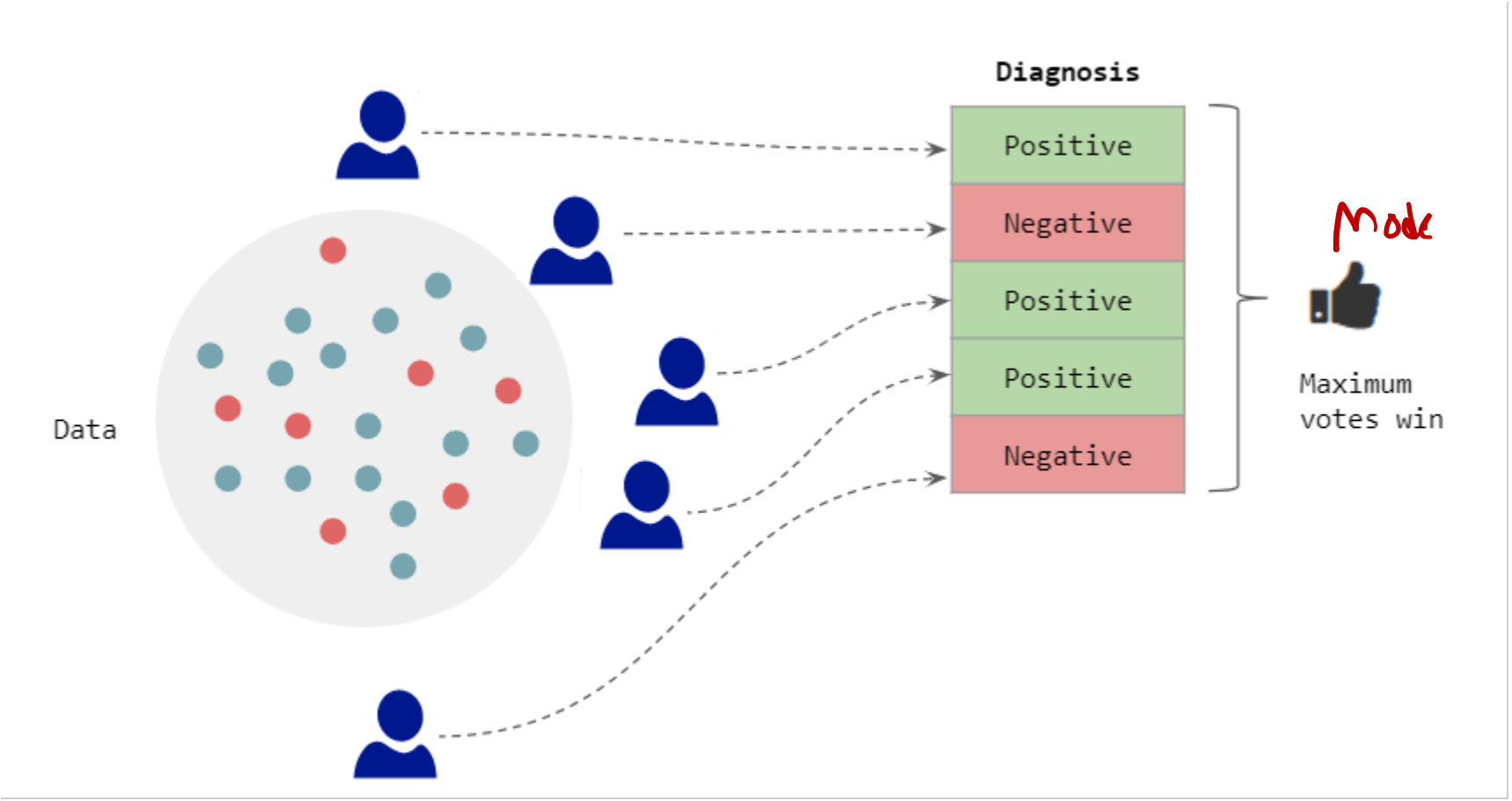
Ensemble Learning

- A process by which **multiple models** (classifiers or experts), are **strategically generated** and combined.
- Provide a solution to a particular **computational intelligence problem**.
- Used to **improve** the **performance** of a model, or **reduce** the likelihood of an **unfortunate selection** of a poor one.

INSAID

Ensemble Learning Example

- **Health Diagnosis:** The result obtained from one classifier may be biased but not from the majority of votes.



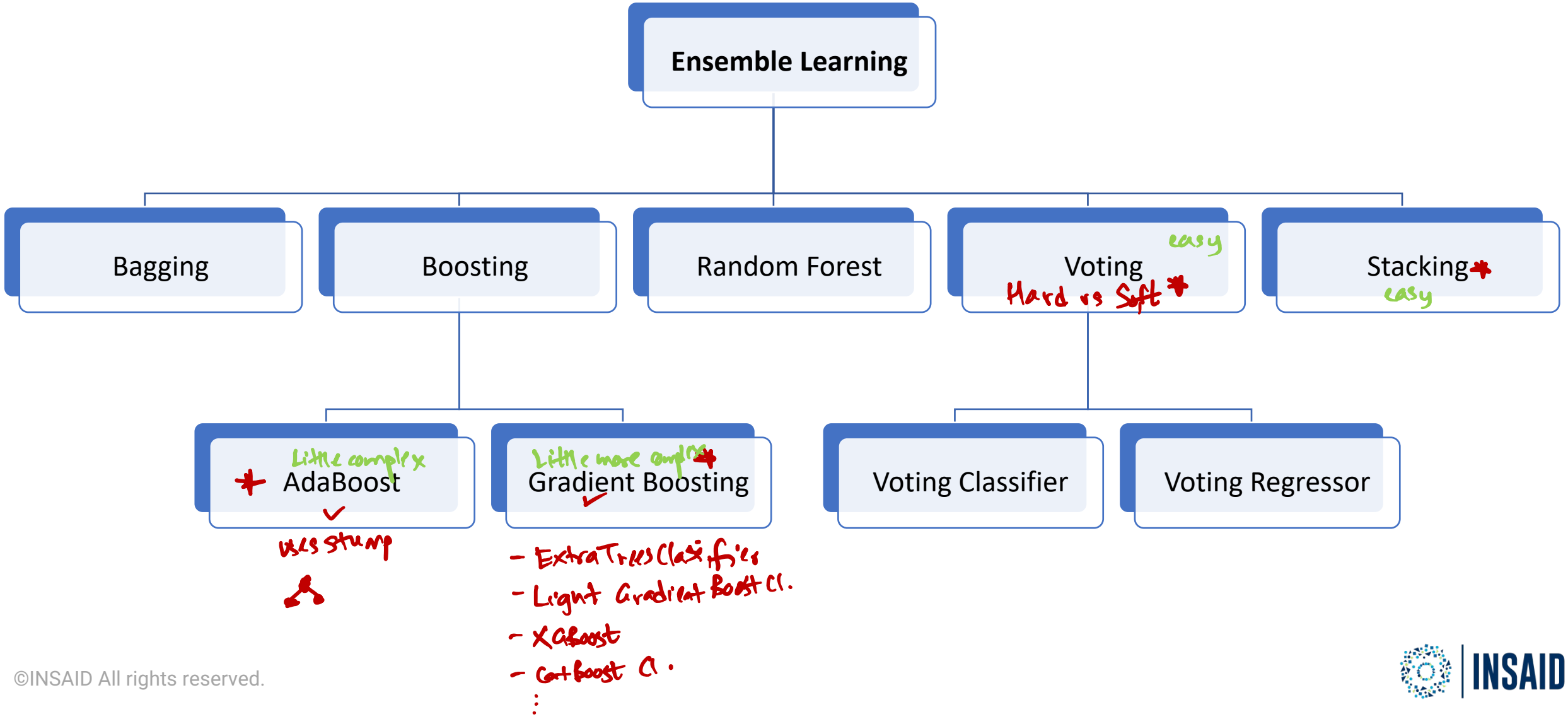
INSAID

Agenda

1. Ensemble Learning
2. **Ensemble Learning Types**
3. Bagging
4. Boosting
5. Random Forest
6. Voting
7. Stacking

INSAID

Ensemble Learning Types



INSAID

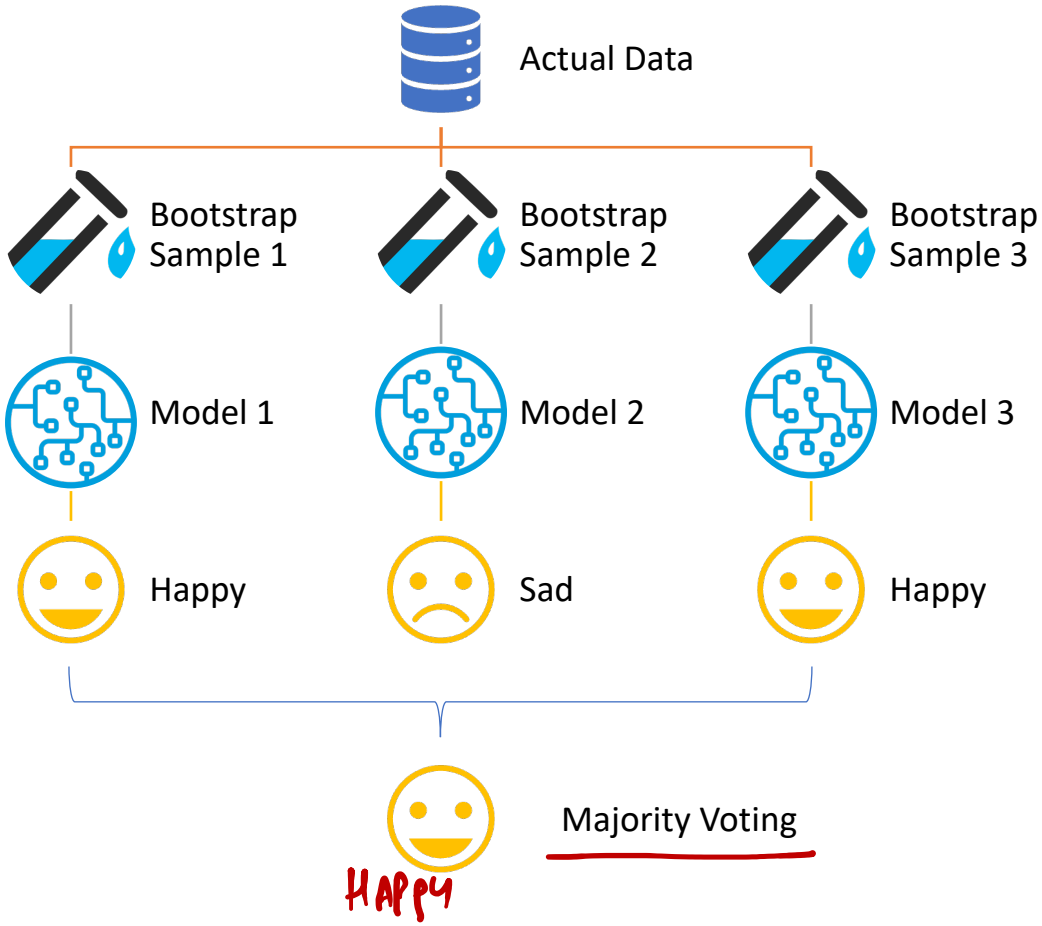
Agenda

1. Ensemble Learning
2. Ensemble Learning Types
3. **Bagging**
4. Boosting
5. Random Forest
6. Voting
7. Stacking

INSAID

Bagging

- It builds several instances of a **black-box estimator** on **random subsets** of the original training set.
- **Aggregate** their **individual predictions** to form a final prediction.



INSAID

Agenda

1. Ensemble Learning
2. Ensemble Learning Types
3. Bagging
4. **Boosting**
5. Random Forest
6. Voting
7. Stacking

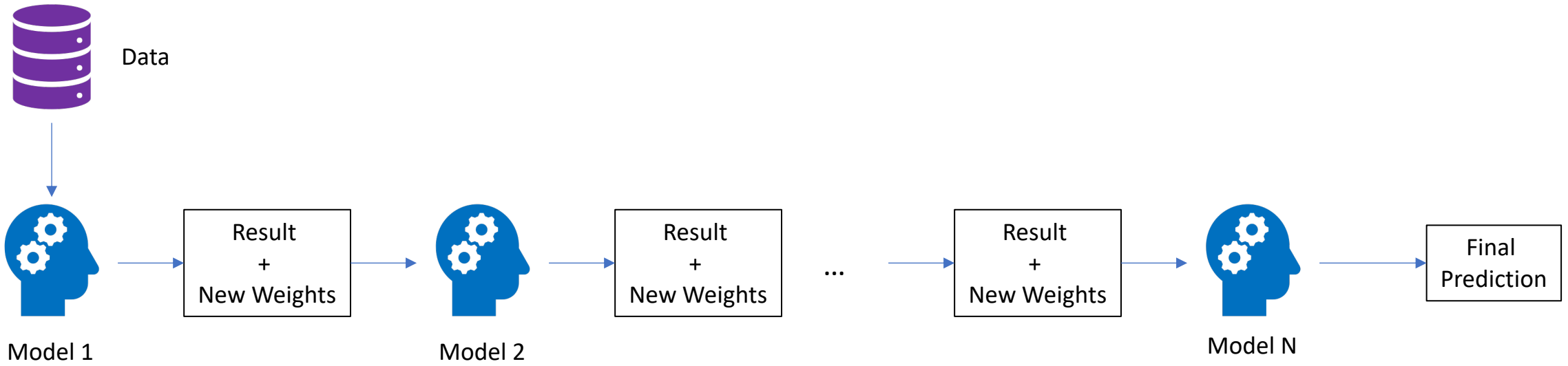
INSAID

Boosting

- The idea of boosting is to **train weak learners sequentially**, each trying to correct its predecessor.
- There are two variants of Boosting that exists i.e. **AdaBoost** and **Gradient Boosting**.

AdaBoost

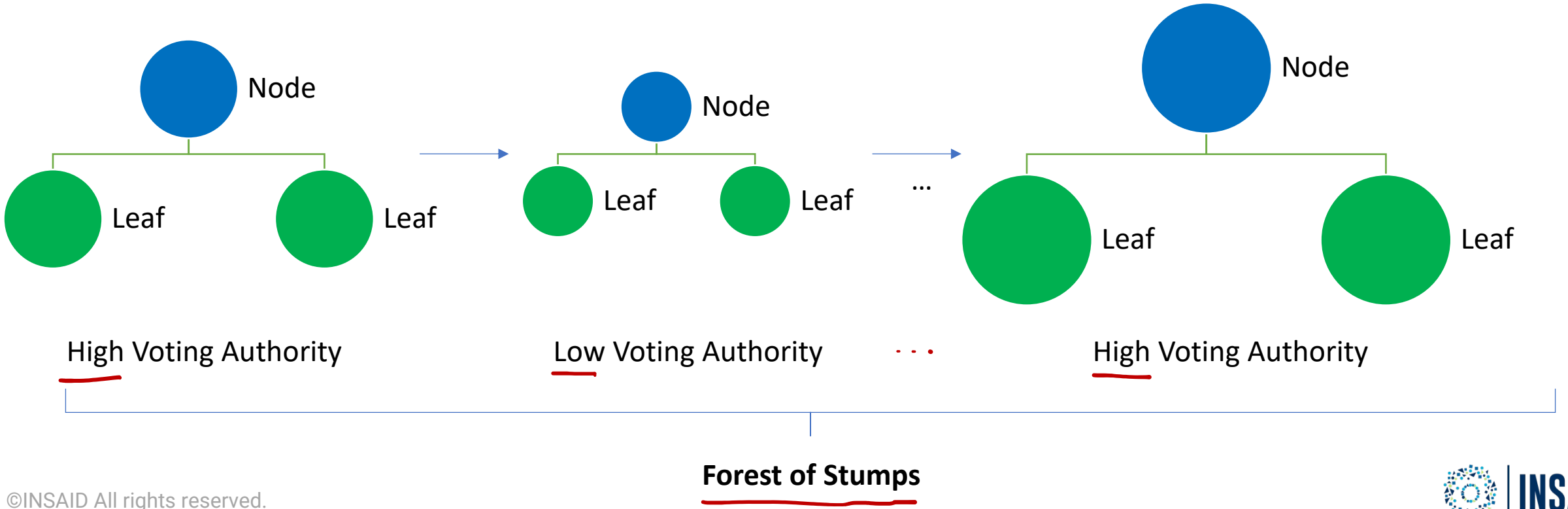
- Here the core principle is to **fit** a **sequence of stumps** (weak learners) on repeatedly modified versions of the data.
- Accumulated **predictions** are **combined** through a **weighted majority vote** (or sum) to produce the final prediction.



INSAID

AdaBoost Working

- It creates a series of stump (weak learners) i.e. a node and two leaves.
- Some **stumps** may have high **voting authority** than other stumps in the final classification.
- Each stump is made by taking the **previous stump's mistake** into account.



INSAID

AdaBoost Working

- Let's say we have the following data and want to predict the **heart disease** given its features.
- Before anything else we will **define weights** (equal at initial) for each data point.

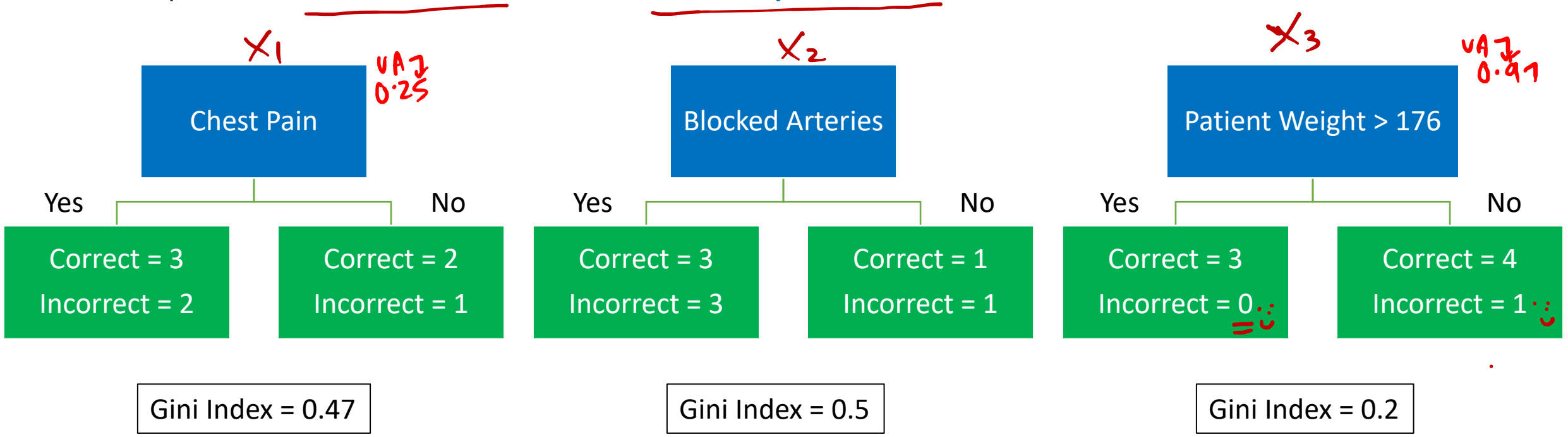
Chest Pain	Blocked Arteries	Patient Weight	Heart Disease	Weight
Yes	Yes	205	Yes	1 ÷ 8
No	Yes	180	Yes	1 ÷ 8
Yes	No	210	Yes	1 ÷ 8
Yes	Yes	167	Yes	1 ÷ 8
No	Yes	156	No	1 ÷ 8
No	Yes	125	No	1 ÷ 8
Yes	No	168	No	1 ÷ 8
Yes	Yes	172	No	1 ÷ 8

Note: The weight will change after making each stump in order to guide how next stump is created.

INSAID

AdaBoost Working

- We will create a forest of stumps to predict if a person has heart disease or not.
- Then we will **estimate** the **Gini Index** of all the stumps and **choose** the stump with **lowest Gini** Index value.
- The stump with the lowest Gini Index will be the **first stump** in the forest.



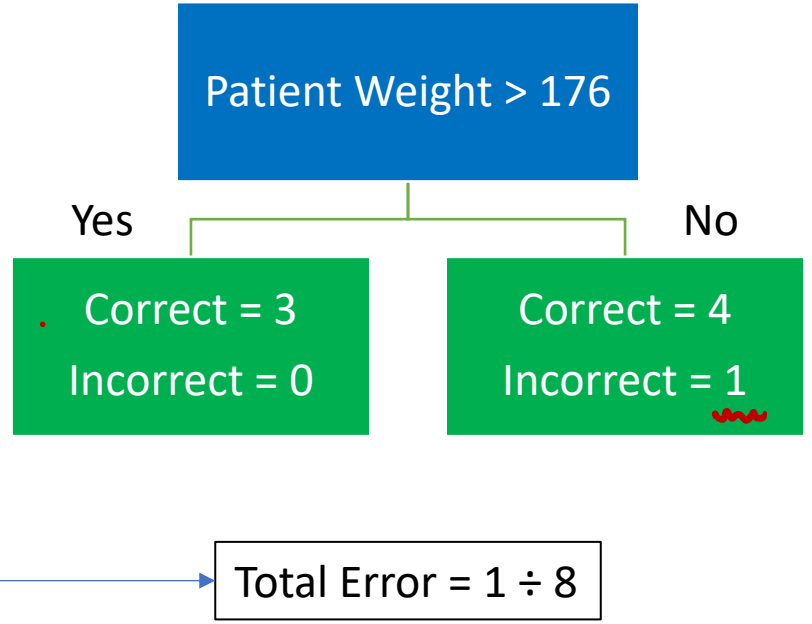
INSAID

AdaBoost Working

- Next, we will **determine** how much **authority** the chosen stump will have in the final classification.
- We will observe that how well the stump was able to classify the samples i.e. **Total Error**.
- The Total Error is the **sum of the weights** associated with the **incorrectly classified samples**.

X_1 X_2 X_3 y

Chest Pain	Blocked Arteries	Patient Weight	Heart Disease	Weight
Yes	Yes	205	Yes	1 ÷ 8
No	Yes	180	Yes	1 ÷ 8
Yes	No	210	Yes	1 ÷ 8
Yes	Yes	167	Yes	1 ÷ 8
No	Yes	156	No	1 ÷ 8
No	Yes	125	No	1 ÷ 8
Yes	No	168	No	1 ÷ 8
Yes	Yes	172	No	1 ÷ 8

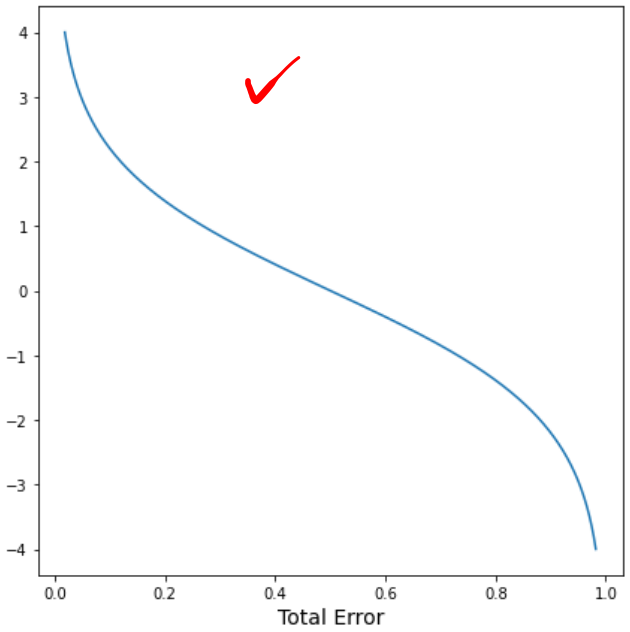


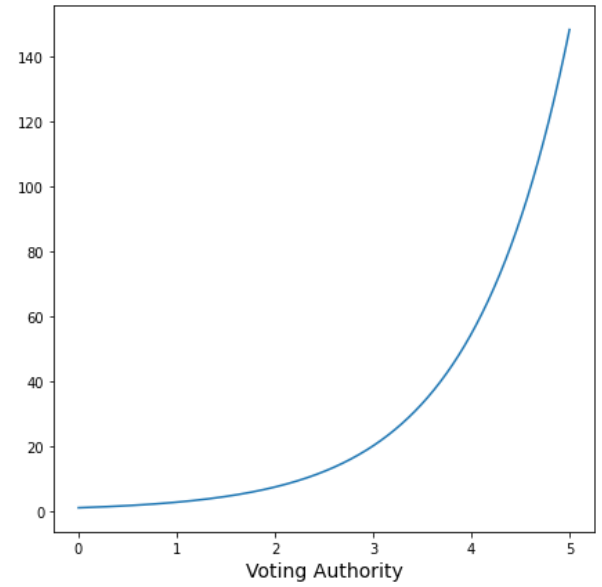
INSAID

AdaBoost Working

- **Total Error** will always be between **0**, for a **perfect stump**, and **1** for a **horrible stump**.
- We use Total Error to determine Voting Authority the stump has in the final classification using the following formula:

$$\text{Voting Authority} = \frac{1}{2}\log\left(\frac{1 - \text{Total Error}}{\text{Total Error}}\right)$$





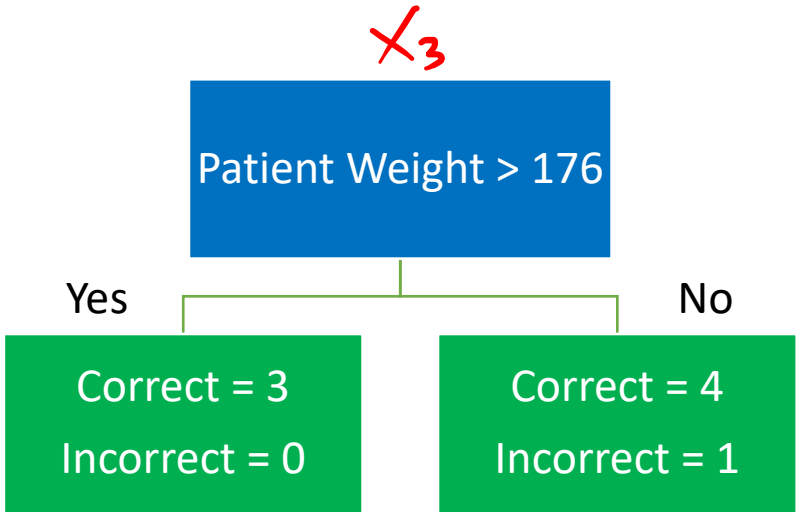
IF: Small Total Error, THEN: High Voting Authority

IF: Large Total Error, THEN: Low Voting Authority

INSAID

AdaBoost Working

- The Total Error in our case is 1 ÷ 8.



$$\text{Voting Authority} = \frac{1}{2}\log\left(\frac{1-(1\div 8)}{(1\div 8)}\right) = \frac{1}{2}\log(7) = 0.97$$

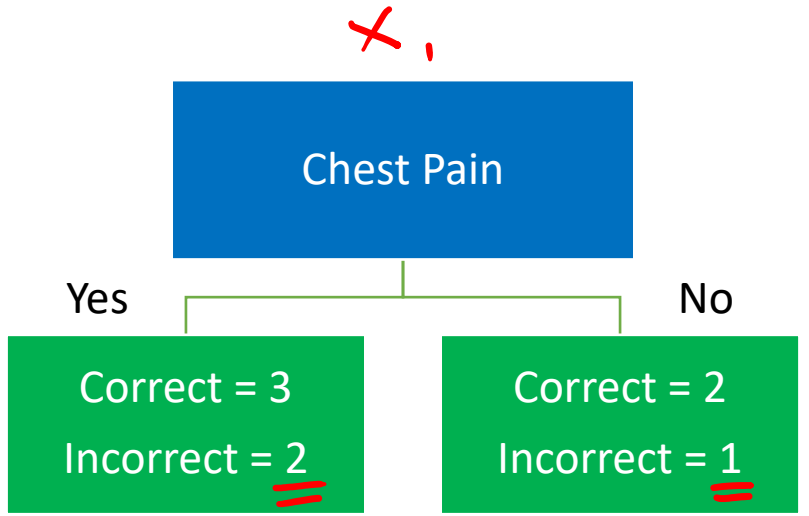
- Thus this stump will have 0.97 weightage as a voting authority in the final classification.

INSAID

AdaBoost Working

- Now let's say we had Chest Pain as our best stump.
- Chest Pain made 3 errors, so the Total Error in this case will be sum of weights for incorrectly classified samples.

Chest Pain	Blocked Arteries	Patient Weight	Heart Disease	Weight
Yes	Yes	205	Yes	1 ÷ 8
No	Yes	180	Yes	1 ÷ 8
Yes	No	210	Yes	1 ÷ 8
Yes	Yes	167	Yes	1 ÷ 8
No	Yes	156	No	1 ÷ 8
No	Yes	125	No	1 ÷ 8
Yes	No	168	No	1 ÷ 8
Yes	Yes	172	No	1 ÷ 8

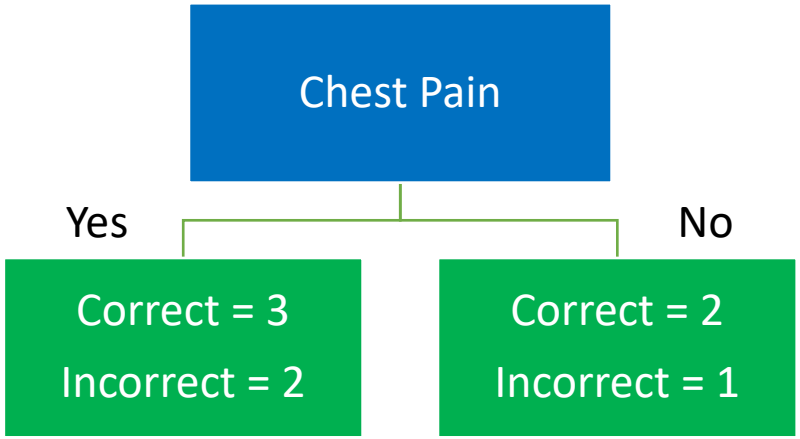


Total Error = (1 ÷ 8) + (1 ÷ 8) + (1 ÷ 8)
Total Error = 3 ÷ 8

INSAID

AdaBoost Working

- The Total Error in this case is 3 ÷ 8.



$$\text{Voting Authority} = \frac{1}{2}\log\left(\frac{1-(3\div 8)}{(3\div 8)}\right) = \frac{1}{2}\log(1.67) = 0.25$$

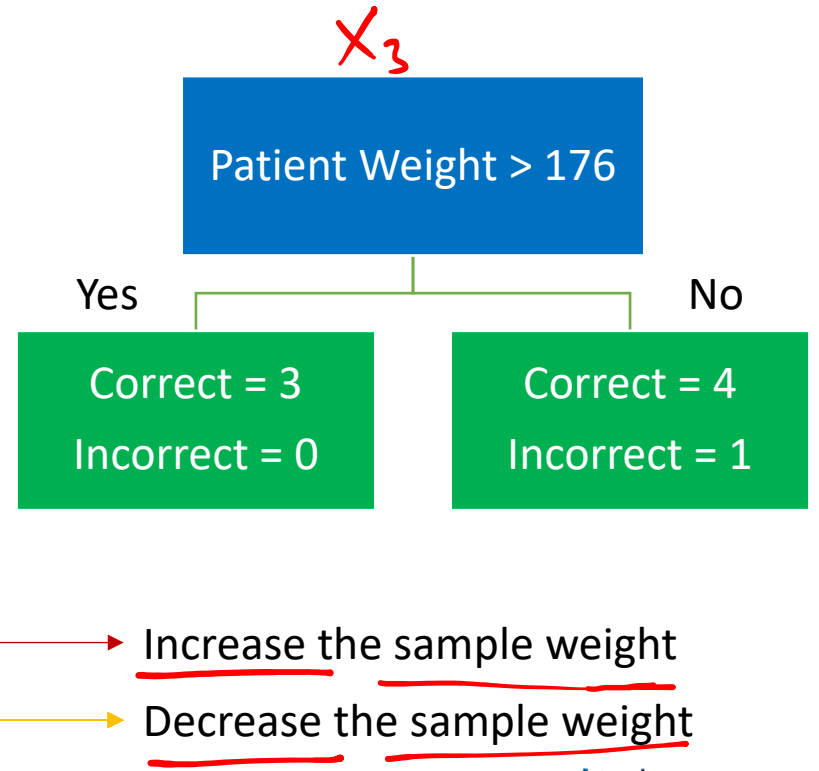
- Thus this stump will have **0.25** weightage as a **voting authority** in the final classification.
- Similarly, we can calculate stump and voting authority for blocked arteries feature.

INSAID

AdaBoost Working

- Now that we know that **Patient Weight** will be our **first stump** in forest, we need to **update** our **sample weights**.
- These weights will be taken into account for our next stump.

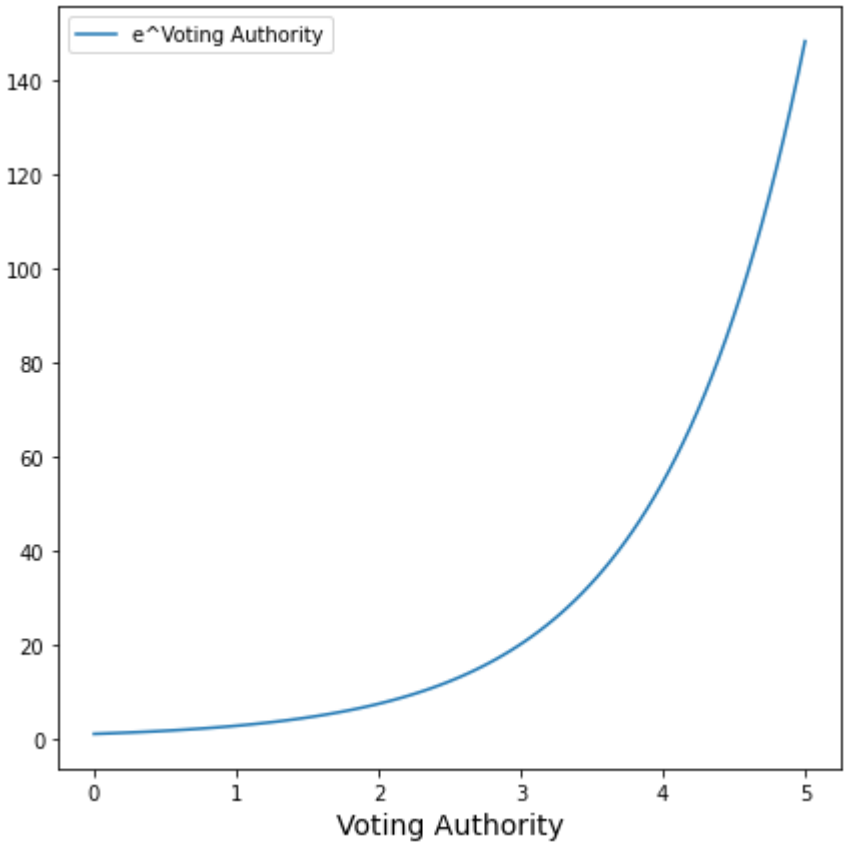
Chest Pain	Blocked Arteries	Patient Weight	Heart Disease	Weight
Yes	Yes	205	Yes	1 ÷ 8
No	Yes	180	Yes	1 ÷ 8
Yes	No	210	Yes	1 ÷ 8
Yes	Yes	167	Yes	1 ÷ 8
No	Yes	156	No	1 ÷ 8
No	Yes	125	No	1 ÷ 8
Yes	No	168	No	1 ÷ 8
Yes	Yes	172	No	1 ÷ 8



INSAID

AdaBoost Working

- The formula that is used to calculate new sample weight is as follows:



weight

(For Incorrectly Classified Samples)

$$\text{Sample Weight}_i = \text{Sample Weight}_{i-1} * e^{\text{Voting Authority}}$$

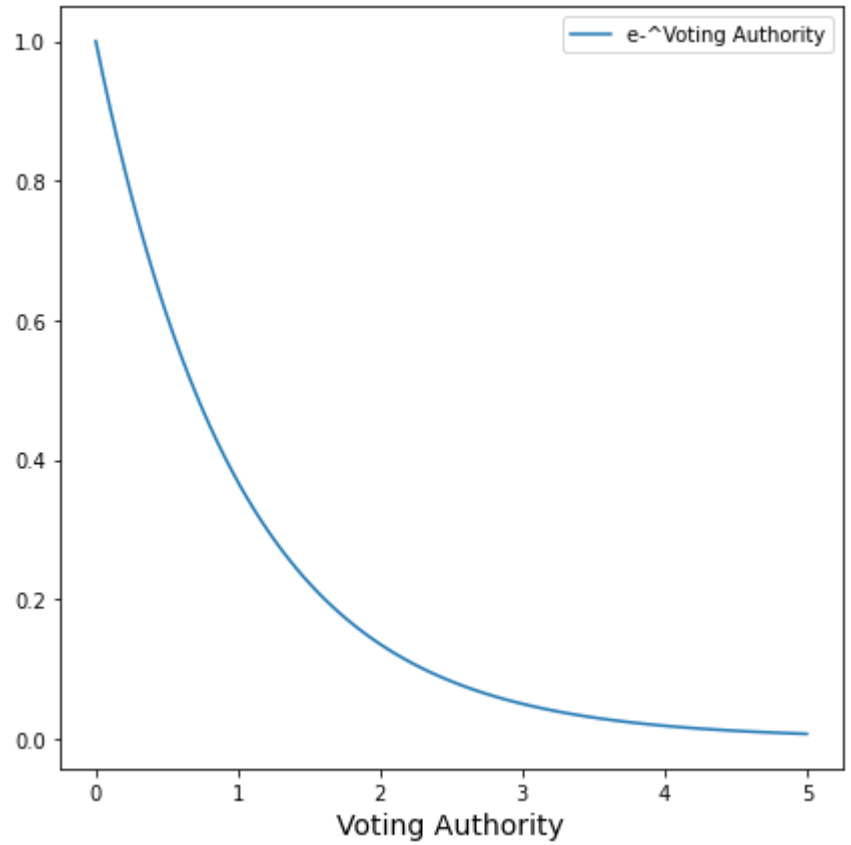
IF: High Voting Authority, THEN: Large Weight

IF: Low Voting Authority: THEN: Small Weight

INSAID

AdaBoost Working

- The formula that is used to calculate new sample weight is as follows:



Weight

(For Correctly Classified Samples)

$$\text{Sample Weight}_i = \text{Sample Weight}_{i-1} * e^{-\text{Voting Authority}}$$

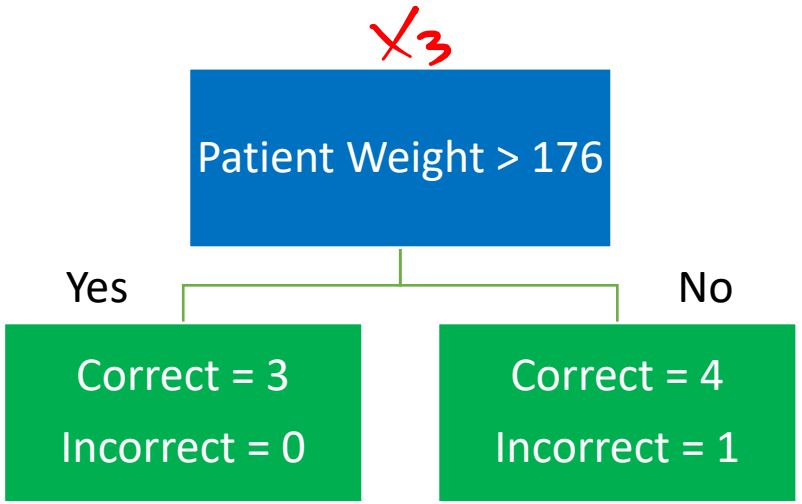
IF: High Voting Authority, THEN: Small Weight

IF: Low Voting Authority: THEN: Large Weight

INSAID

AdaBoost Working

- In our case, the stump made using Patient Weight have Voting Authority of 0.97.



$$\text{Voting Authority} = \frac{1}{2}\log\left(\frac{1-(1\div 8)}{(1\div 8)}\right) = \frac{1}{2}\log(7) = 0.97$$

Incorrect:

Sample Weight = (1 ÷ 8) * $e^{0.97}$ = (1 ÷ 8) * 2.64 = 0.33

Correct:

Sample Weight = (1 ÷ 8) * $e^{-0.97}$ = (1 ÷ 8) * 0.37 = 0.05

INSAID

AdaBoost Working

- Now we will **update** our **old weights** with the new weights and **normalize** them so that they adds up to 1.
- On taking **mean** of **New Weight**, we get **0.68**. We can divide the New Weight with this value to get Normalized Weight.

Chest Pain	Blocked Arteries	Patient Weight	Heart Disease	Weight (OLD)	New Weight (Updated)	Normalized Weight
Yes	Yes	205	Yes	1 ÷ 8	0.05	0.05 ÷ 0.68 = 0.07
No	Yes	180	Yes	1 ÷ 8	0.05	0.05 ÷ 0.68 = 0.07
Yes	No	210	Yes	1 ÷ 8	0.05	0.05 ÷ 0.68 = 0.07
Yes	Yes	167	Yes	1 ÷ 8	0.33	0.33 ÷ 0.68 = 0.49
No	Yes	156	No	1 ÷ 8	0.05	0.05 ÷ 0.68 = 0.07
No	Yes	125	No	1 ÷ 8	0.05	0.05 ÷ 0.68 = 0.07
Yes	No	168	No	1 ÷ 8	0.05	0.05 ÷ 0.68 = 0.07
Yes	Yes	172	No	1 ÷ 8	0.05	0.05 ÷ 0.68 = 0.07
				Mean New Weight	Σ= 0.68	Σ = 1

Σ = 1

INSAID

AdaBoost Working

n=8

- Now we can **replace** the **New Weight** feature with **Weight feature**, since those are what we will use for **next stump**.
- Next, we can **create new samples** out of this data that will have **same dimension** as the original one.

Chest Pain	Blocked Arteries	Patient Weight	Heart Disease	Weight (Updated)
Yes	Yes	205	Yes	0.07
No	Yes	180	Yes	0.07
Yes	No	210	Yes	0.07
Yes	Yes	167	Yes	0.49
No	Yes	156	No	0.07
No	Yes	125	No	0.07
Yes	No	168	No	0.07
Yes	Yes	172	No	0.07

repeat lesser

(repeats more)

repeat lesser

$\Sigma = 1$

INSAID

AdaBoost Working

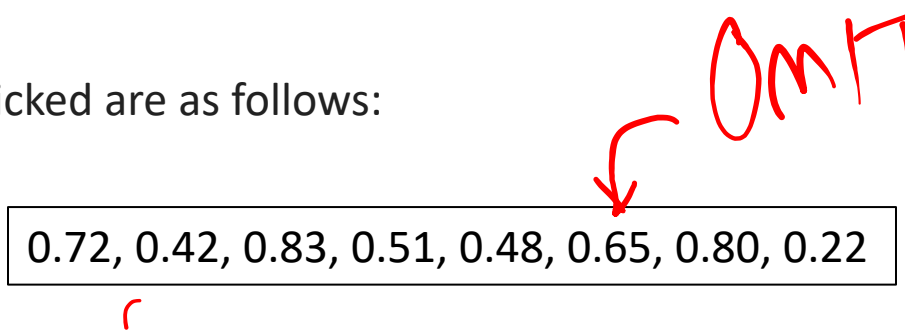
- We need to **pick up a random number** between 0 and 1.
- We will see where that number falls when we use **Weight** like a **distribution**.

Index	Chest Pain	Blocked Arteries	Patient Weight	Heart Disease	Weight	Distribution
1	Yes	Yes	205	Yes	0.07	0 – 0.07
2	No	Yes	180	Yes	0.07	0.07 – 0.14
3	Yes	No	210	Yes	0.07	0.14 – 0.21
4	Yes	Yes	167	Yes	0.49	0.21 – 0.70
5	No	Yes	156	No	0.07	0.70 – 0.77
6	No	Yes	125	No	0.07	0.77 – 0.84
7	Yes	No	168	No	0.07	0.84 – 0.91
8	Yes	Yes	172	No	0.07	0.91 – 0.98 ~ 1

INSAID

AdaBoost Working

- For example, imagine the number picked are as follows:



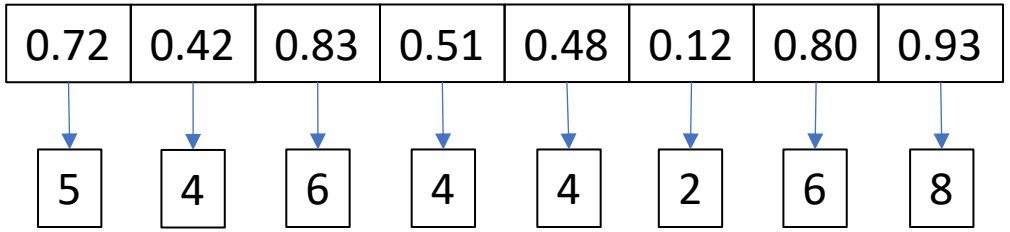
- Then we will **observe** under which **distribution** these **numbers falls** in.
- After getting the idea we can **choose** that **index row** and put it in **new empty dataset**.
- We will **repeat** this **over** the reset of the random values until the dataset have same dimension as the original one.

- **Note:** Some of the sample may repeat in the new dataset.

INSAID

AdaBoost Working

- From the table we can get new samples for a new dataset:



Index	Chest Pain	Blocked Arteries	Patient Weight	Heart Disease	Weight	Distribution
1	Yes	Yes	205	Yes	0.07	0 – 0.07
2	No	Yes	180	Yes	0.07	0.07 – 0.14
3	Yes	No	210	Yes	0.07	0.14 – 0.21
4	Yes	Yes	167	Yes	0.49	0.21 – 0.70
5	No	Yes	156	No	0.07	0.70 – 0.77
6	No	Yes	125	No	0.07	0.77 – 0.84
7	Yes	No	168	No	0.07	0.84 – 0.91
8	Yes	Yes	172	No	0.07	0.91 – 0.98 ~ 1

INSAID

AdaBoost Working

n=8

- The new dataset will be as follows:
- Then we can repeat the steps again that we did so far.

Adjusted Dataset.

Chest Pain	Blocked Arteries	Patient Weight	Heart Disease	Weight
No	Yes	156	No	1 ÷ 8
Yes	Yes	167	Yes	1 ÷ 8
No	Yes	125	No	1 ÷ 8
Yes	Yes	167	Yes	1 ÷ 8
Yes	Yes	167	Yes	1 ÷ 8
No	Yes	180	Yes	1 ÷ 8
No	Yes	125	No	1 ÷ 8
Yes	Yes	172	No	1 ÷ 8

- **Note:** We will set the weights to (1 ÷ Number of rows) as we have resampled the dataset.

INSAID

AdaBoost Working

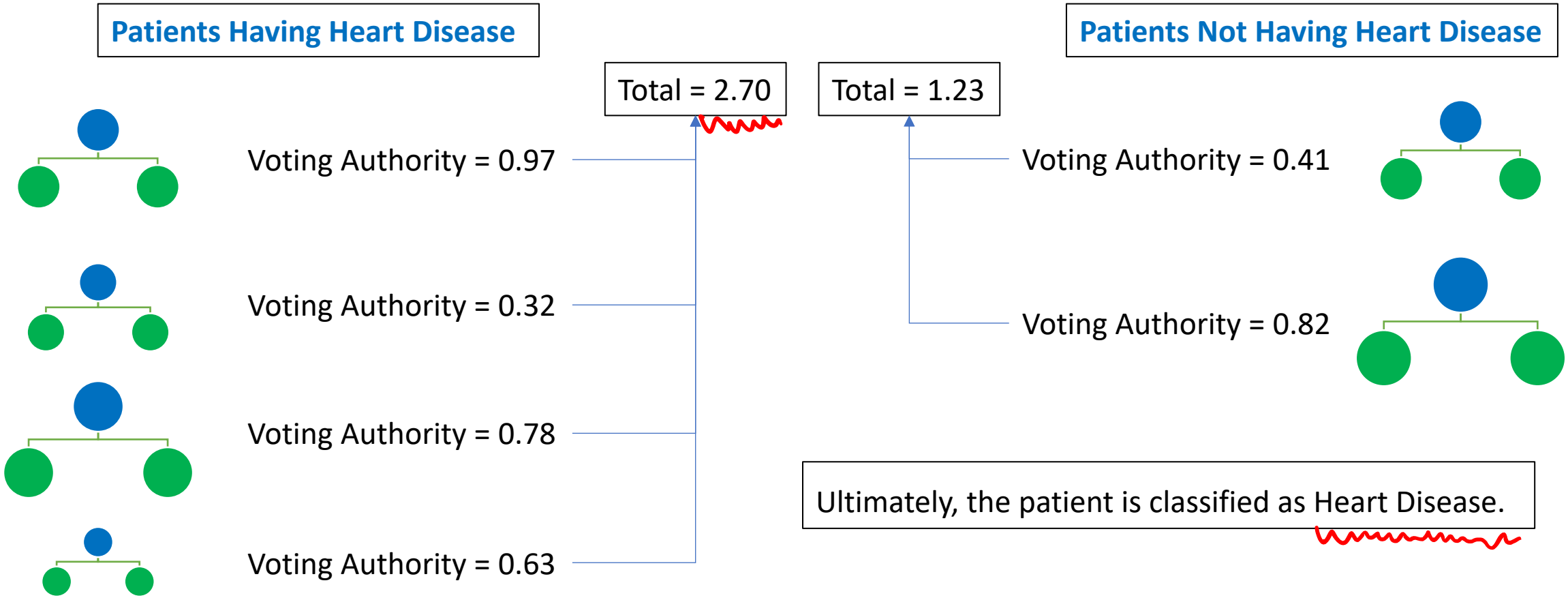
- This next build **stump** will have **more emphasis** on the **incorrectly classified samples** in the last iteration.
- These samples will be treated as a block, creating a **large penalty** for being misclassified.

Chest Pain	Blocked Arteries	Patient Weight	Heart Disease	Weight
No	Yes	156	No	1 ÷ 8
Yes	Yes	167	Yes	1 ÷ 8
No	Yes	125	No	1 ÷ 8
Yes	Yes	167	Yes	1 ÷ 8
Yes	Yes	167	Yes	1 ÷ 8
No	Yes	180	Yes	1 ÷ 8
No	Yes	125	No	1 ÷ 8
Yes	Yes	172	No	1 ÷ 8

INSAID

AdaBoost Working

- At last we will accumulate all the stumps that defines patient having heart disease or not.



INSAID

INSAID