IEEE FCI - Helwan Student Branch

Decision Tree

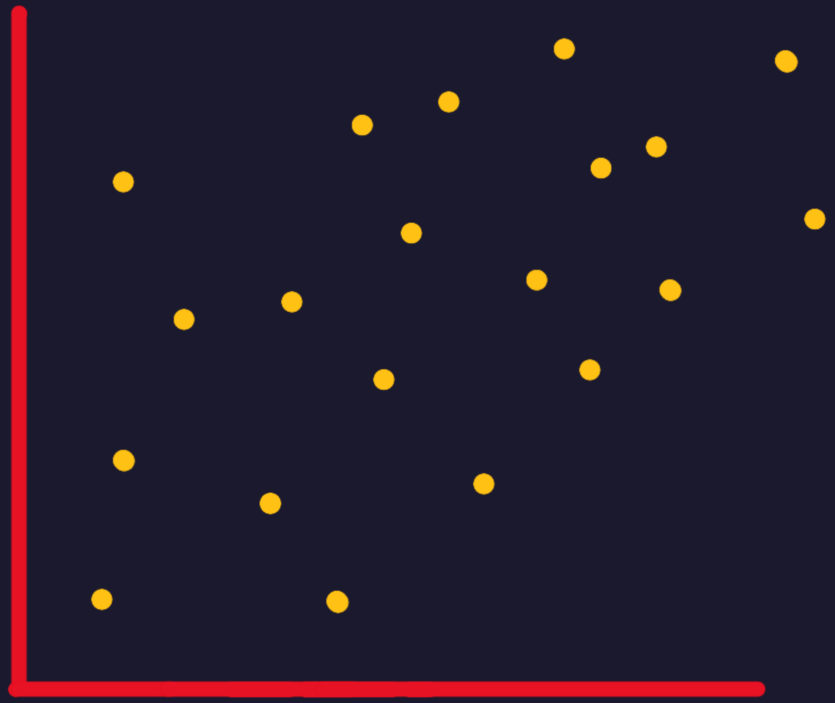
Hossam Ahmed

Ziad Waleed

Mario Mamdouh

Variance

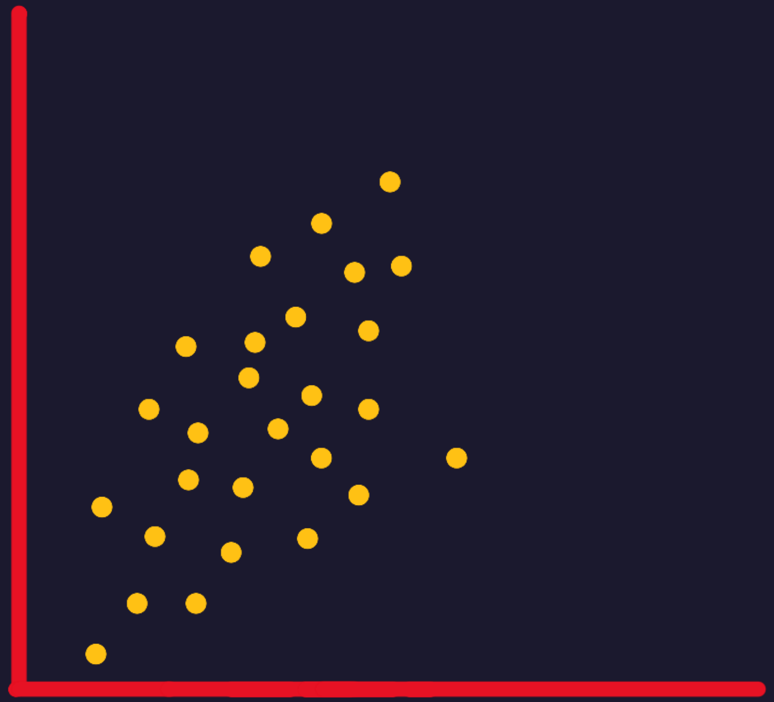
- Measurement of spread in the dataset
- مقياس لتباعد النقاط المختلفة من النقطة الوسط



High Variance

Variance

- Measurement of spread in the dataset
- مقياس لتباعد النقاط المختلفة من النقطة الوسط



Low Variance

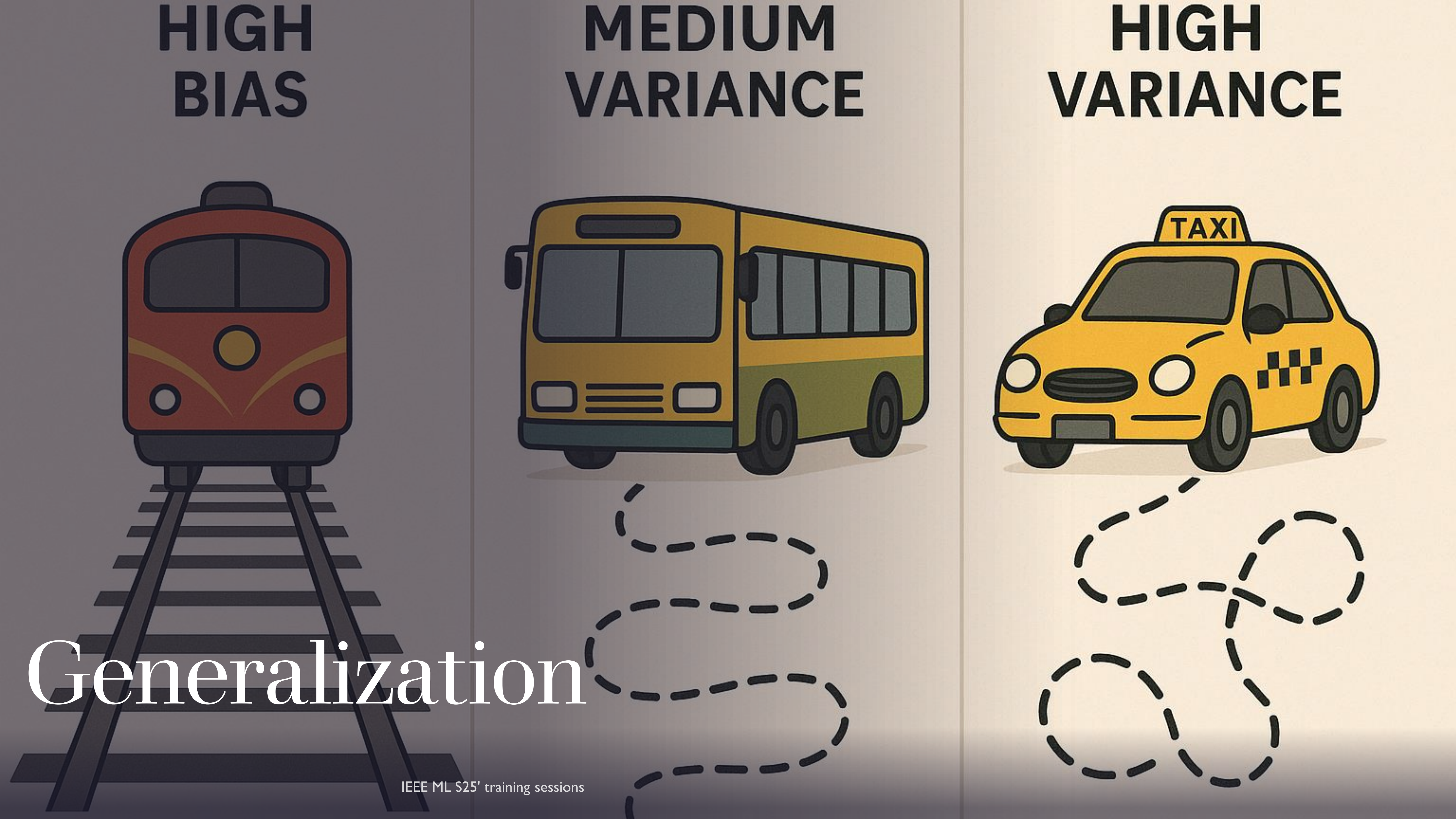
Bias Error

In statistics, bias refers to the tendency of a statistical estimator or method to consistently overestimate or underestimate a population parameter

tendency which causes differences between results and facts

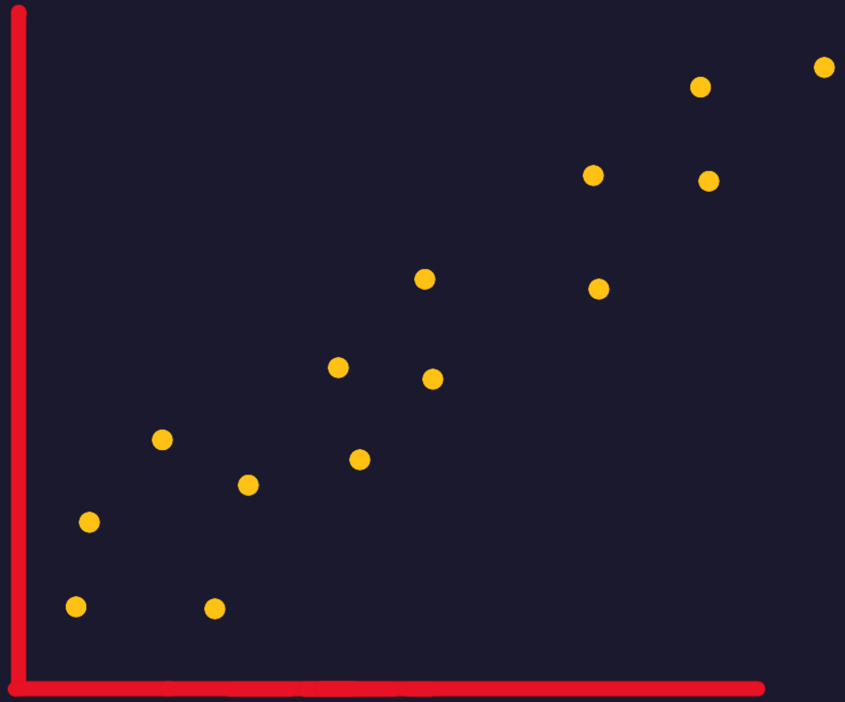
Bias in ML is a sort of mistake in which some aspects of a dataset are given more weight and/or representation than others

In ML bias inability to capture the underlying complexity of the data.



Generalization

- Considering we have this data, and we want to train a 2 models



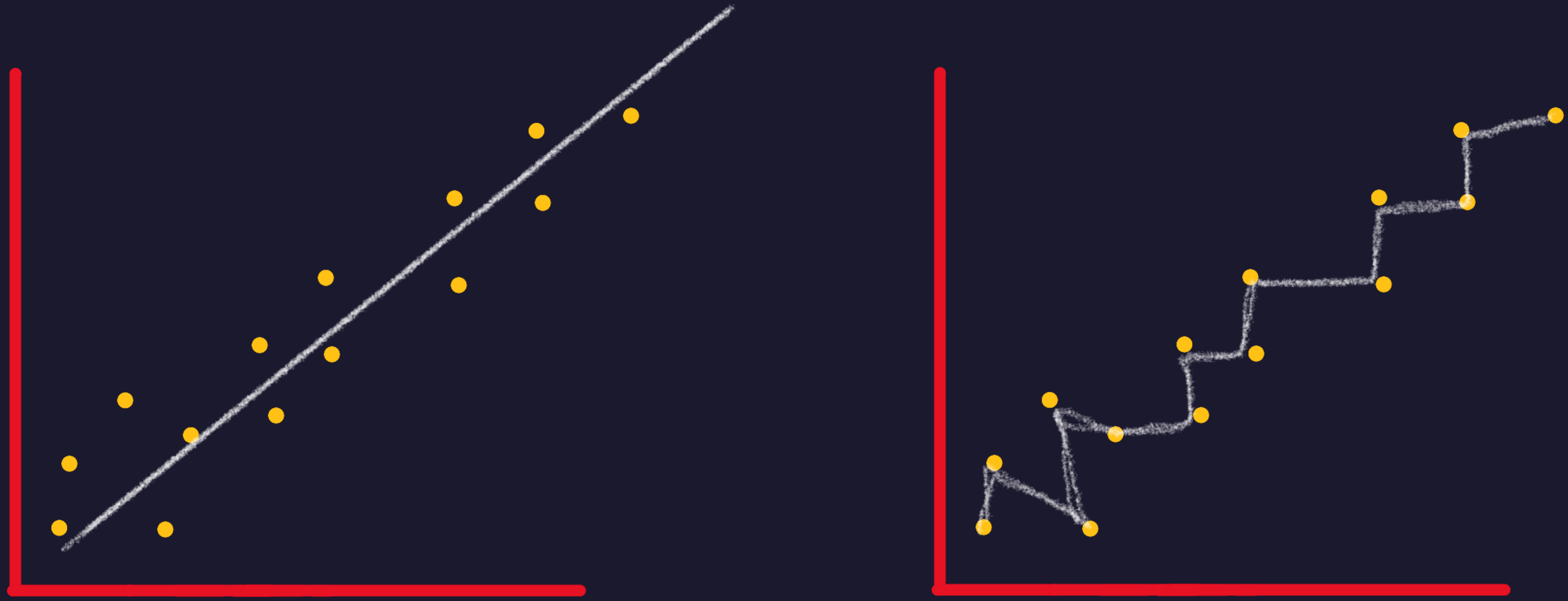
Models

- Linear
- Polynomial

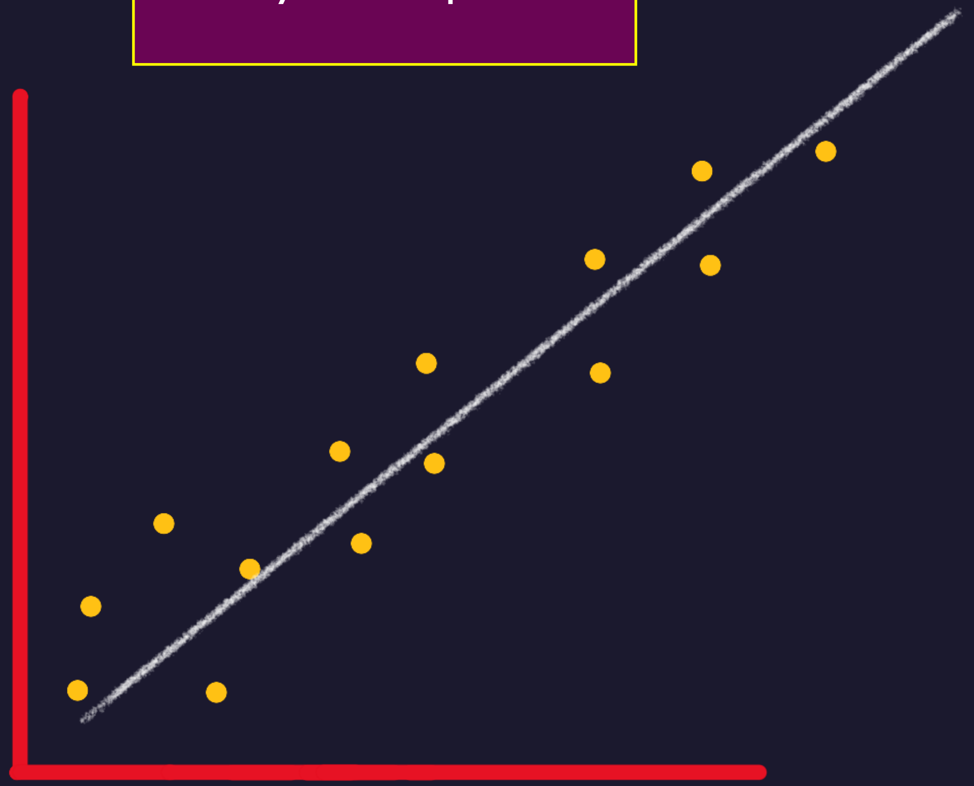
$$y = \beta_0 + \beta_1 x$$

$$y = \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 x^3 + \cdots + \beta_n x^n$$

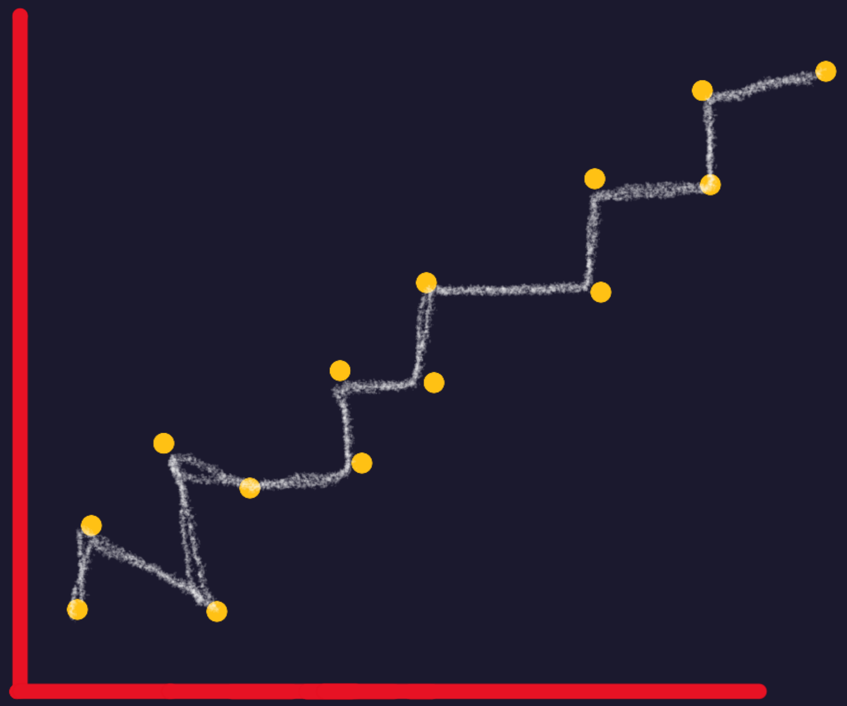
- Considering we have this data, and we want to train a 2 models



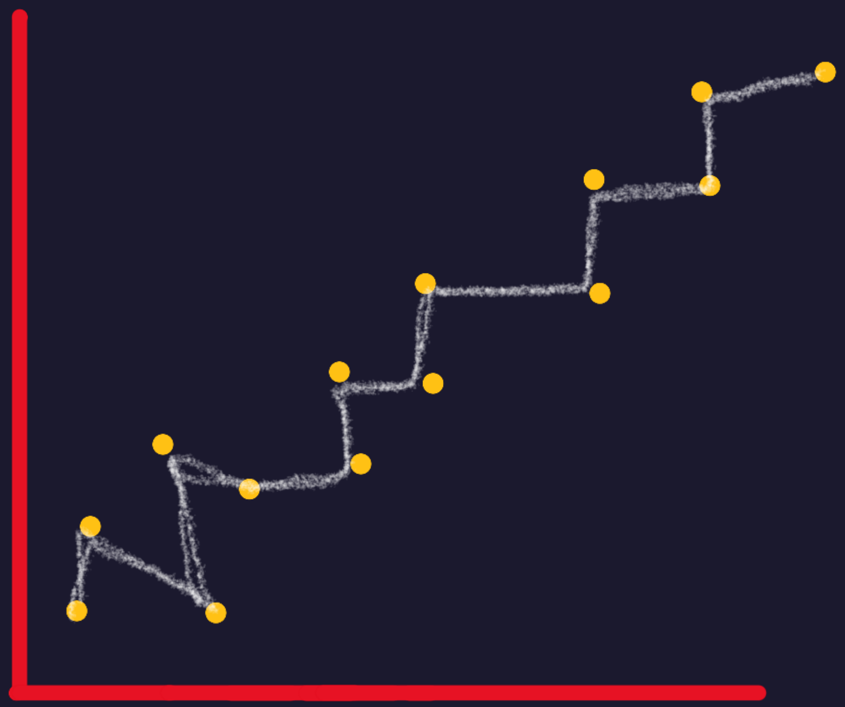
Only fit few points



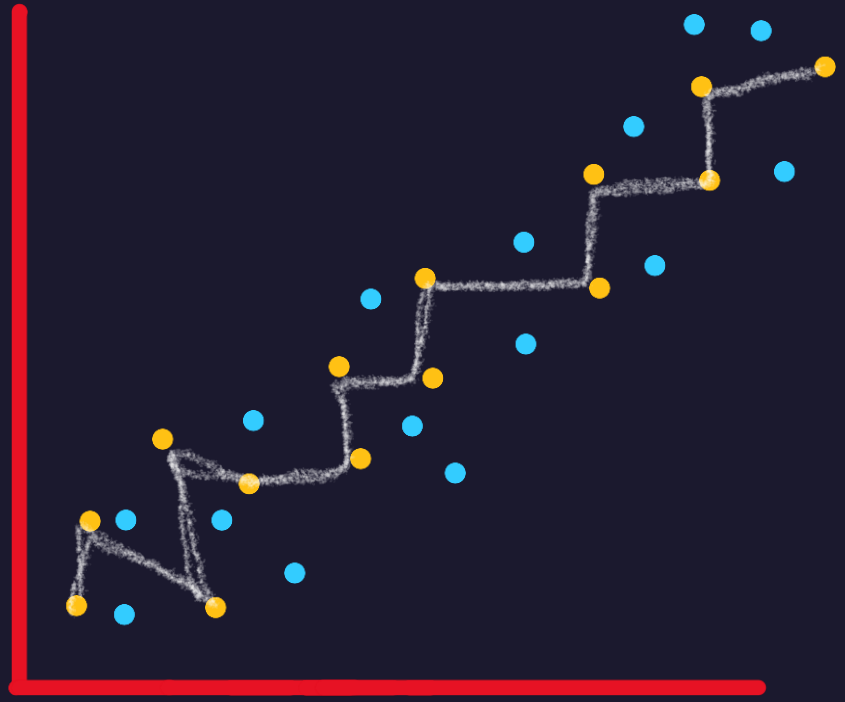
Perfect fit of the trained data 100%



- So, you decided now to deploy your perfect model on the real world



- So, a lot of data inputted to your model

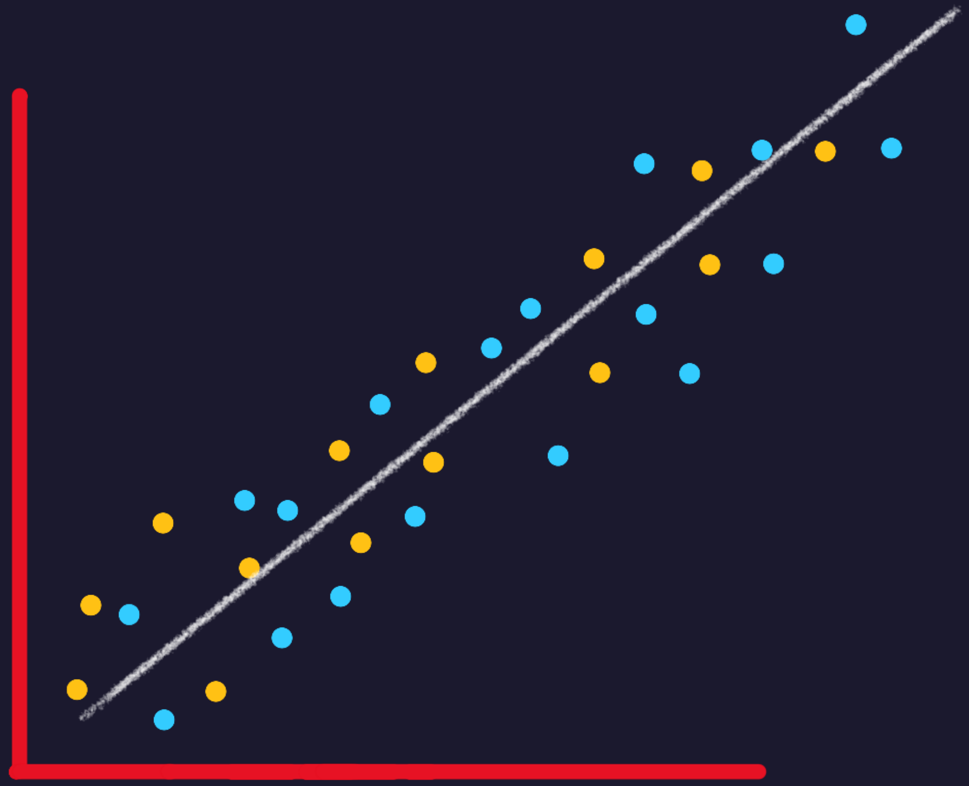


Your model is terrible on real data not near even

Seeking perfection in train set lead to **high variance** model

This called **overfitting**

- What about the linear Model



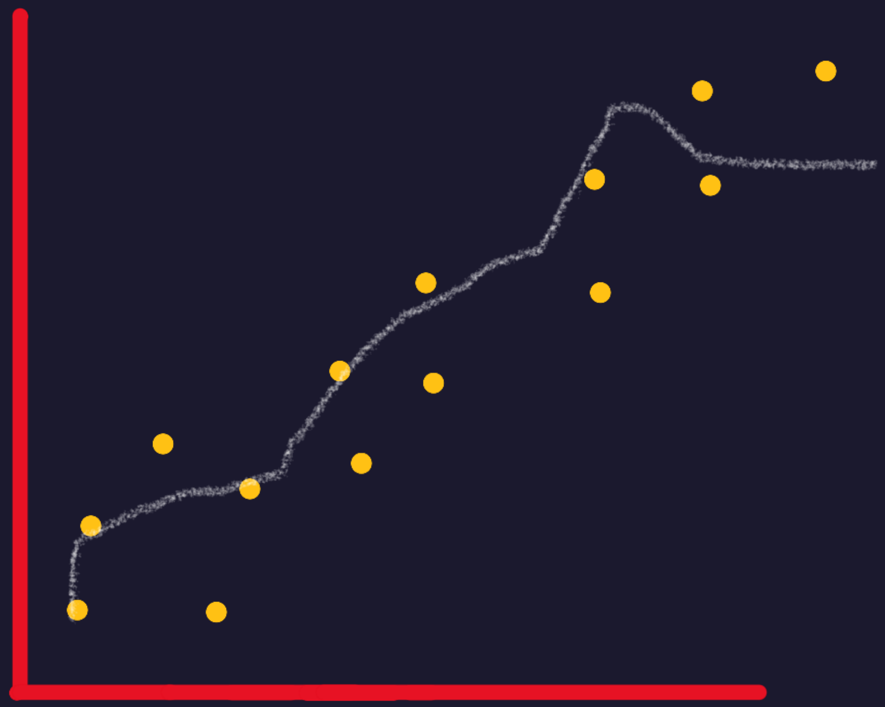
Terrible model but we the training error and testing error is not too far

If average error is 20% for each point you can say that new points can be + or – 20%

This model has a **high bias** following the line doesn't care about the data

This model **underfit** the data too simple to model it

- We need a mode that generalize
  - we shall tolerant some bias (error) in the training set
  - That mean making this model has less variance
  - It’s to simplify the model



We have errors in the training set
But we can have less error when dealing with new input

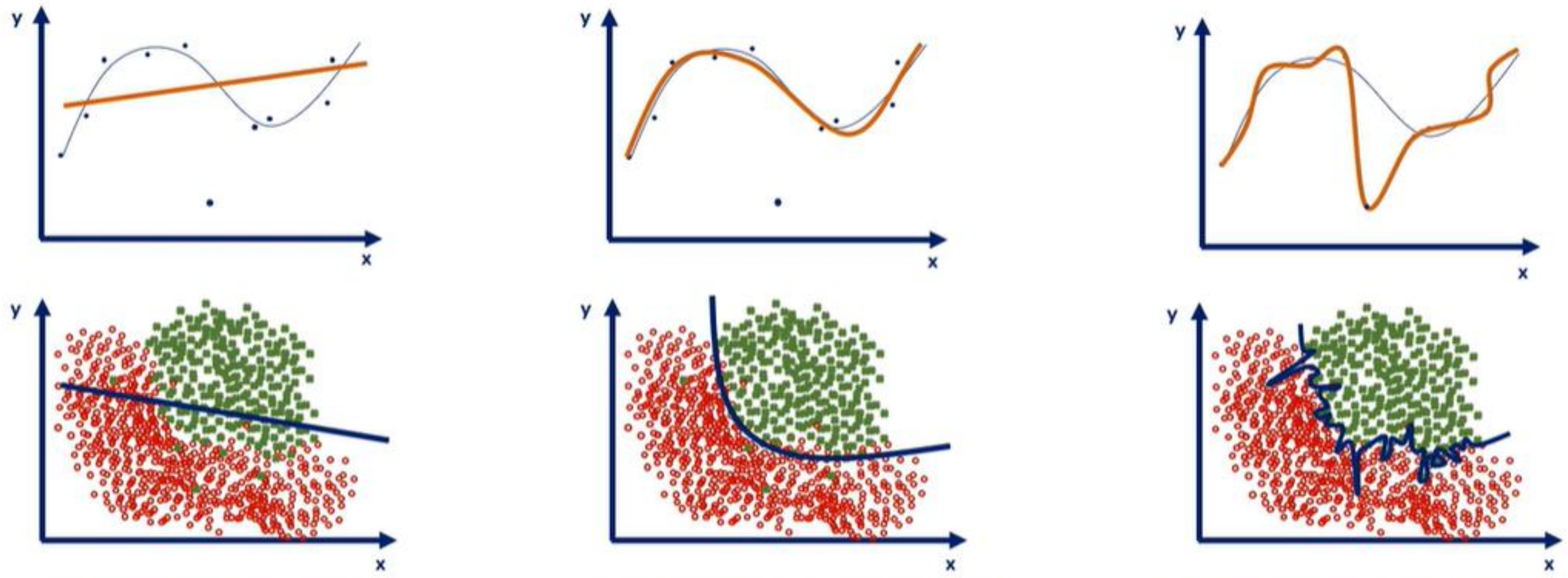
A Best fit trying to achieve balance between Bias(error, simplicity) and Variance (complexity)

It’s a tradeoff

An **underfitted** model

A **good** model

An **overfitted** model



Doesn't capture any logic

- High loss
- Low accuracy

Captures the underlying logic of the dataset

- Low loss
- High accuracy

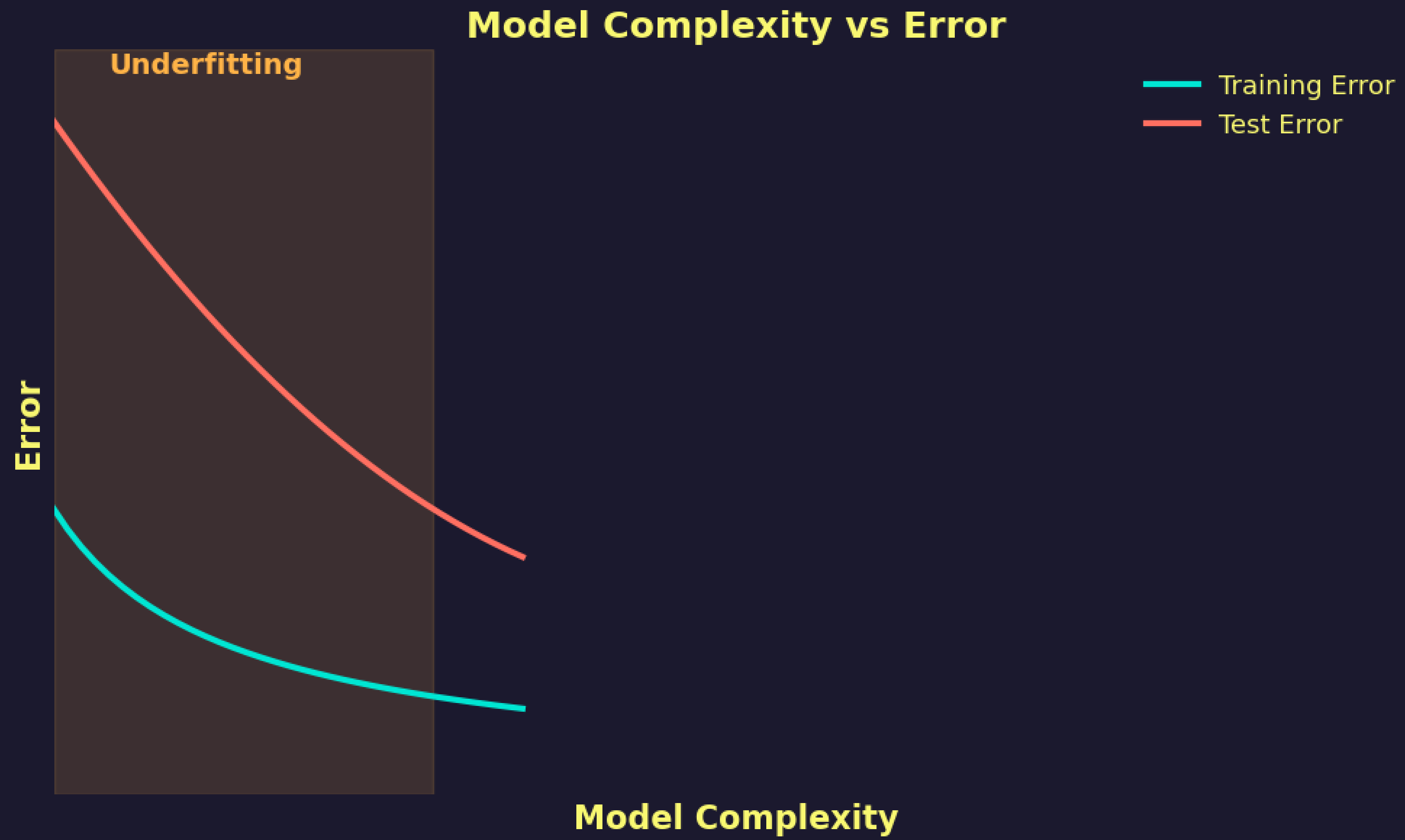
Captures all the noise, thus "missed the point"

- Low loss
- Low accuracy

Bias-variance tradeoff: The balance between underfitting and overfitting

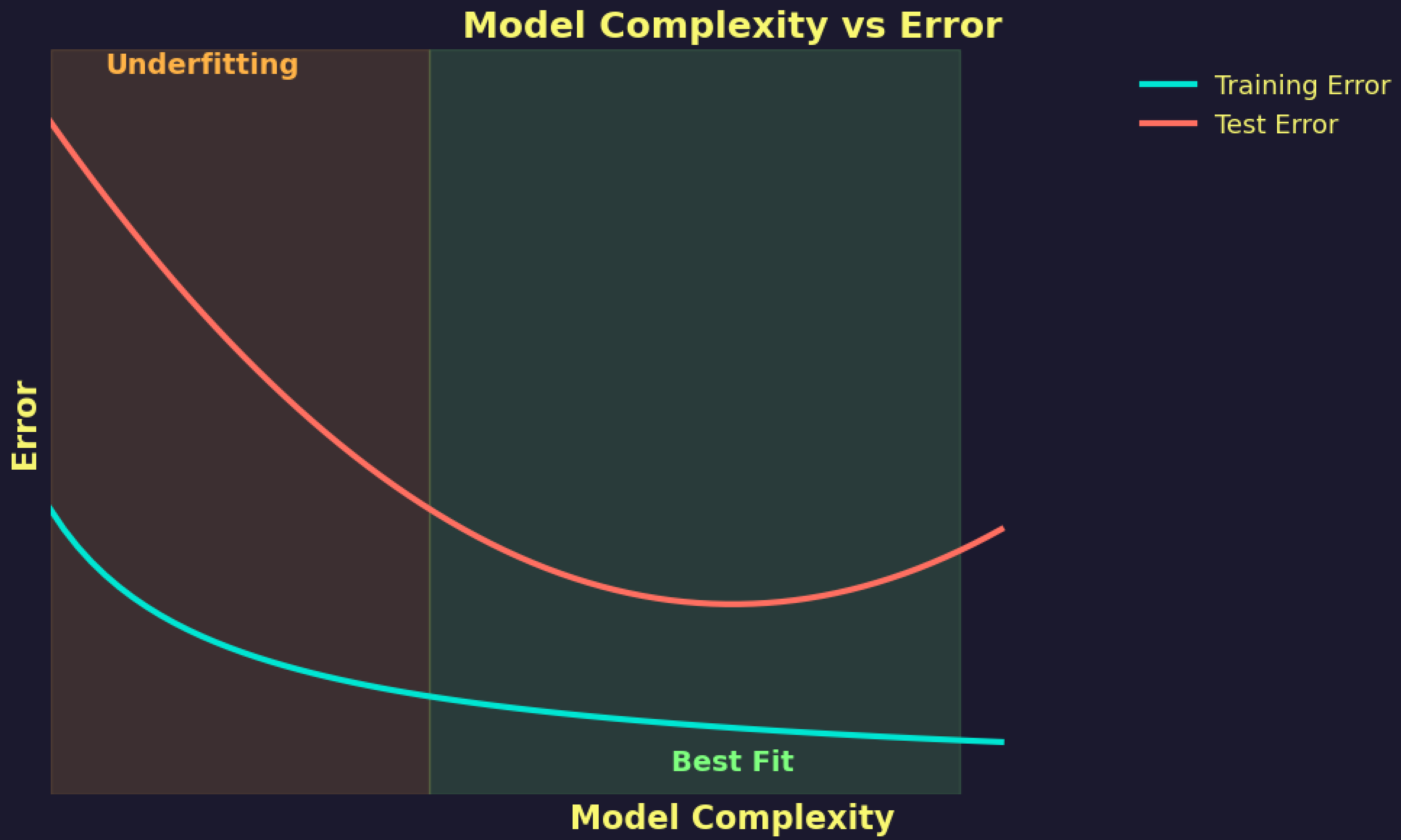
365√DataScience

Underfitting: Model too simple, high bias, poor performance



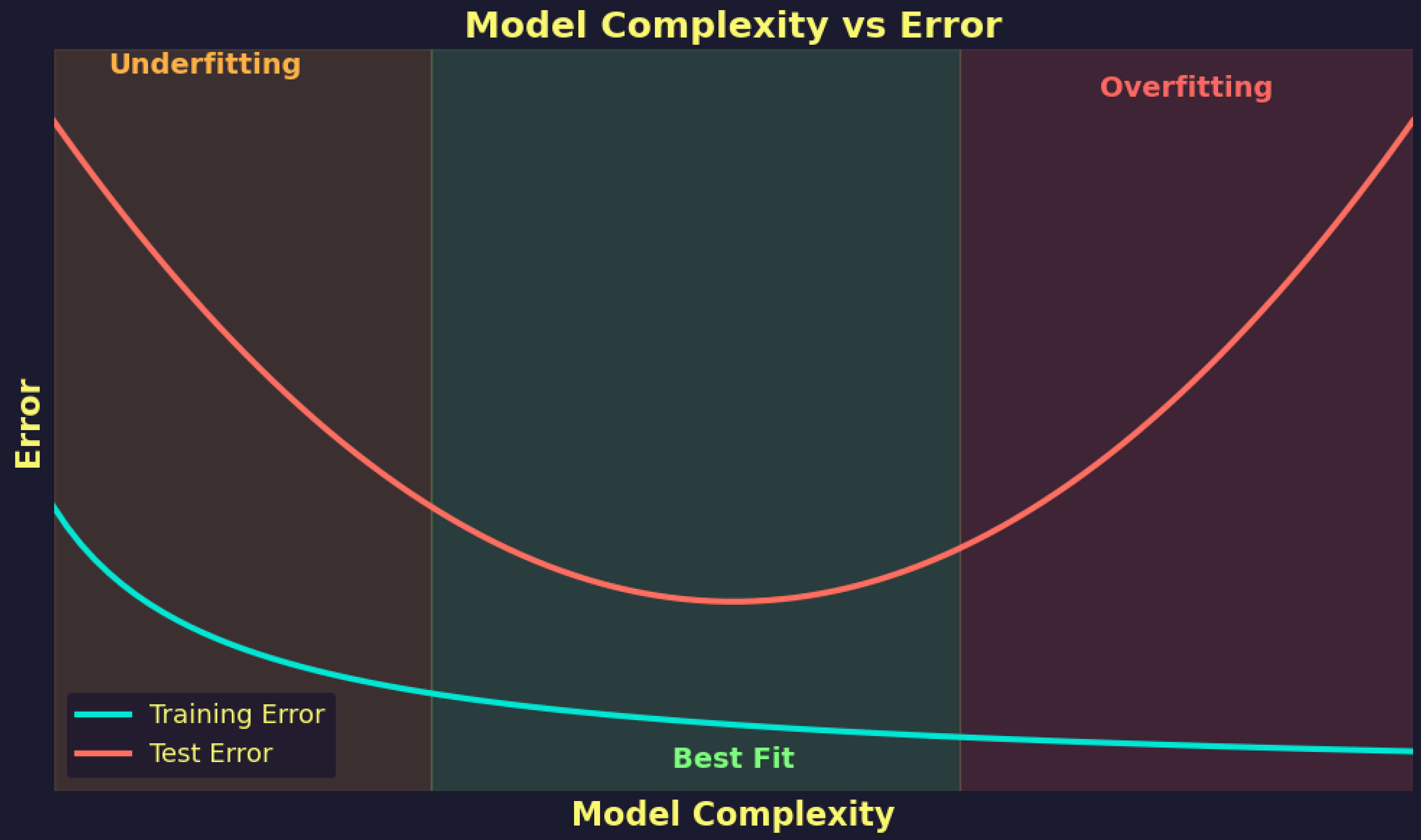
FCI - Helwan
Student Branch
IEEE

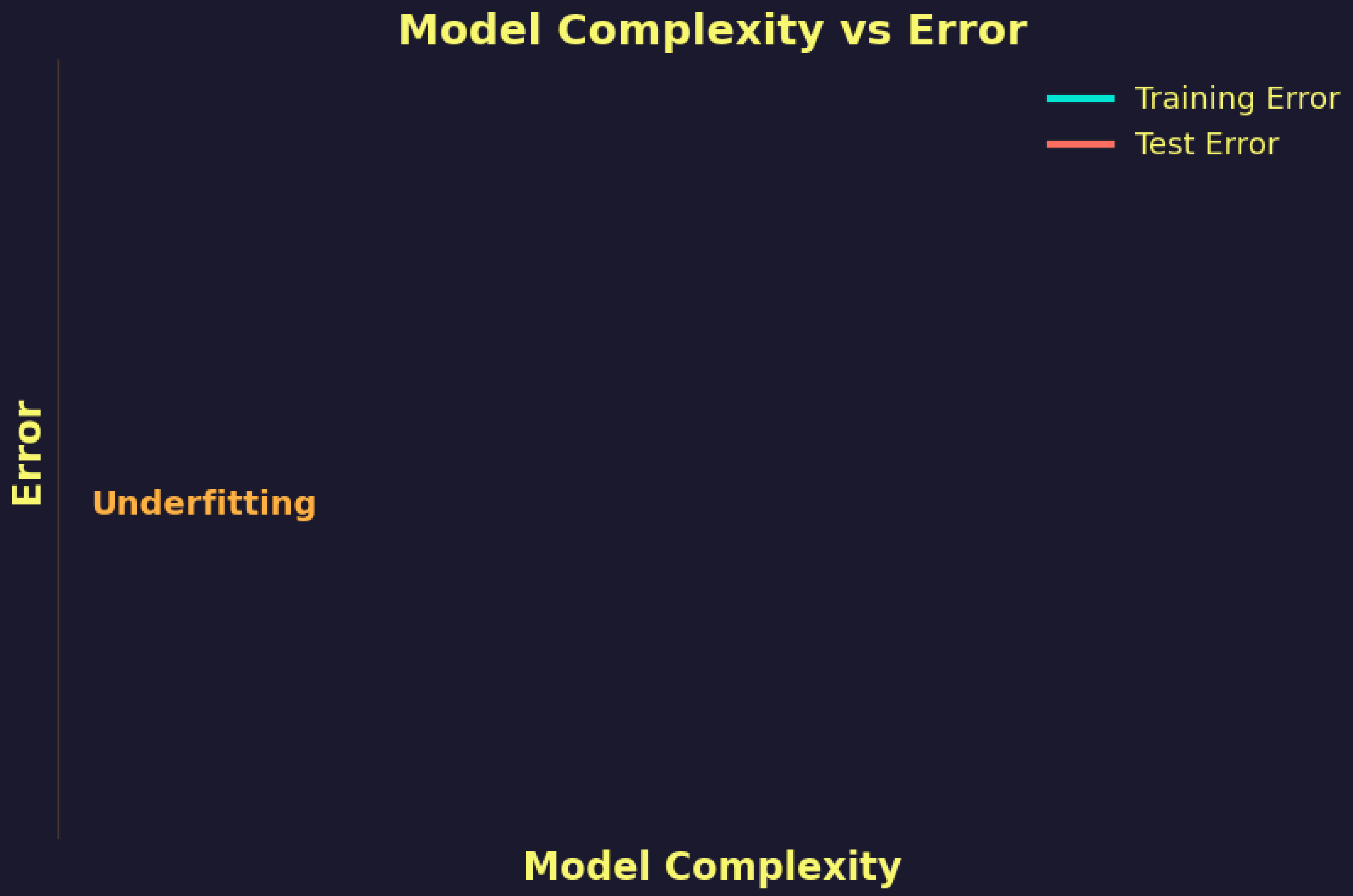
Best Fit: Optimal complexity, balanced bias and variance



FCI - Helwan Student Branch IEEE

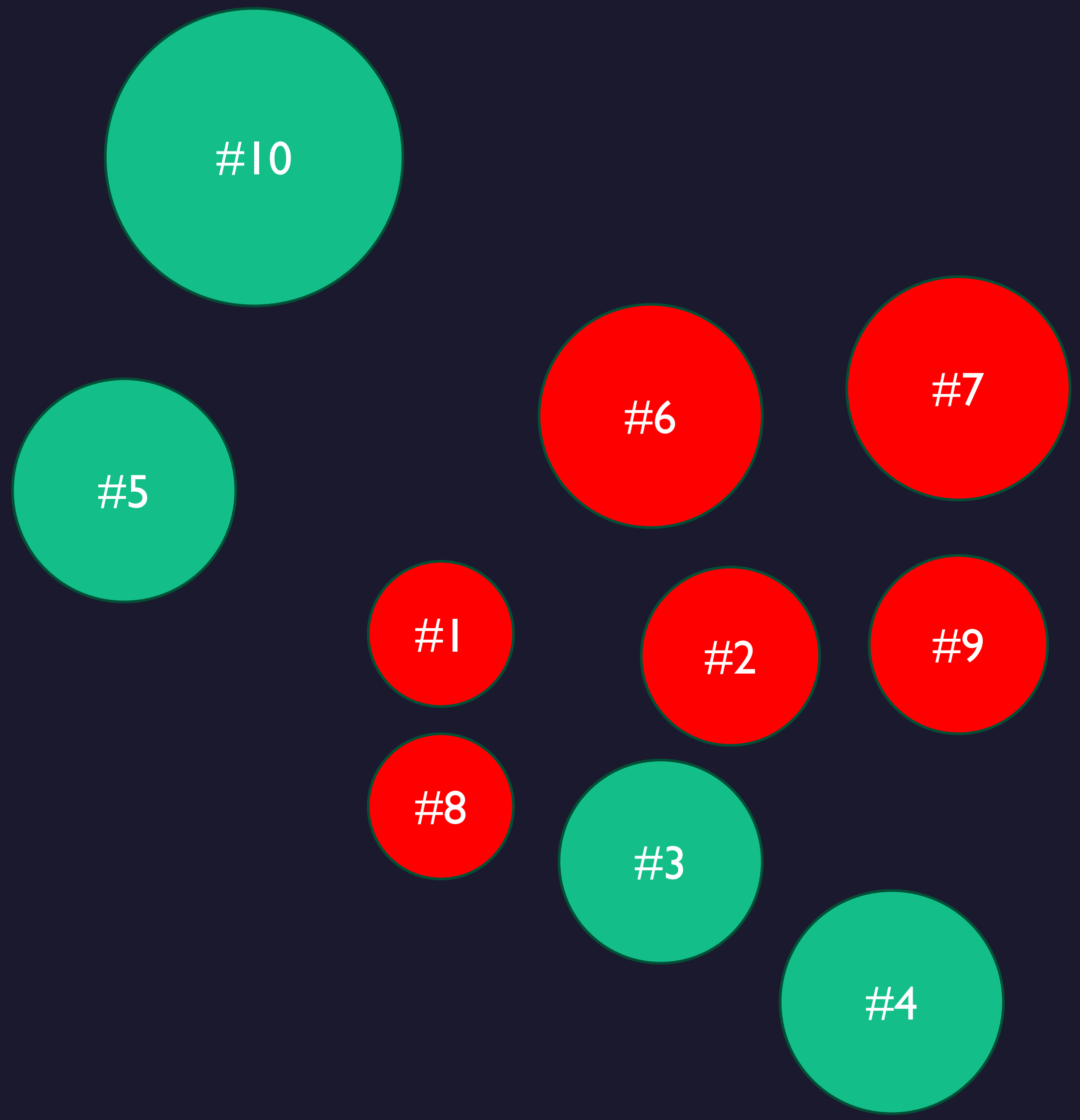
Overfitting: Model too complex, low bias, high variance, poor generalization




Model Complexity vs Error
Training Error
Test Error
Error
Underfitting
Model Complexity

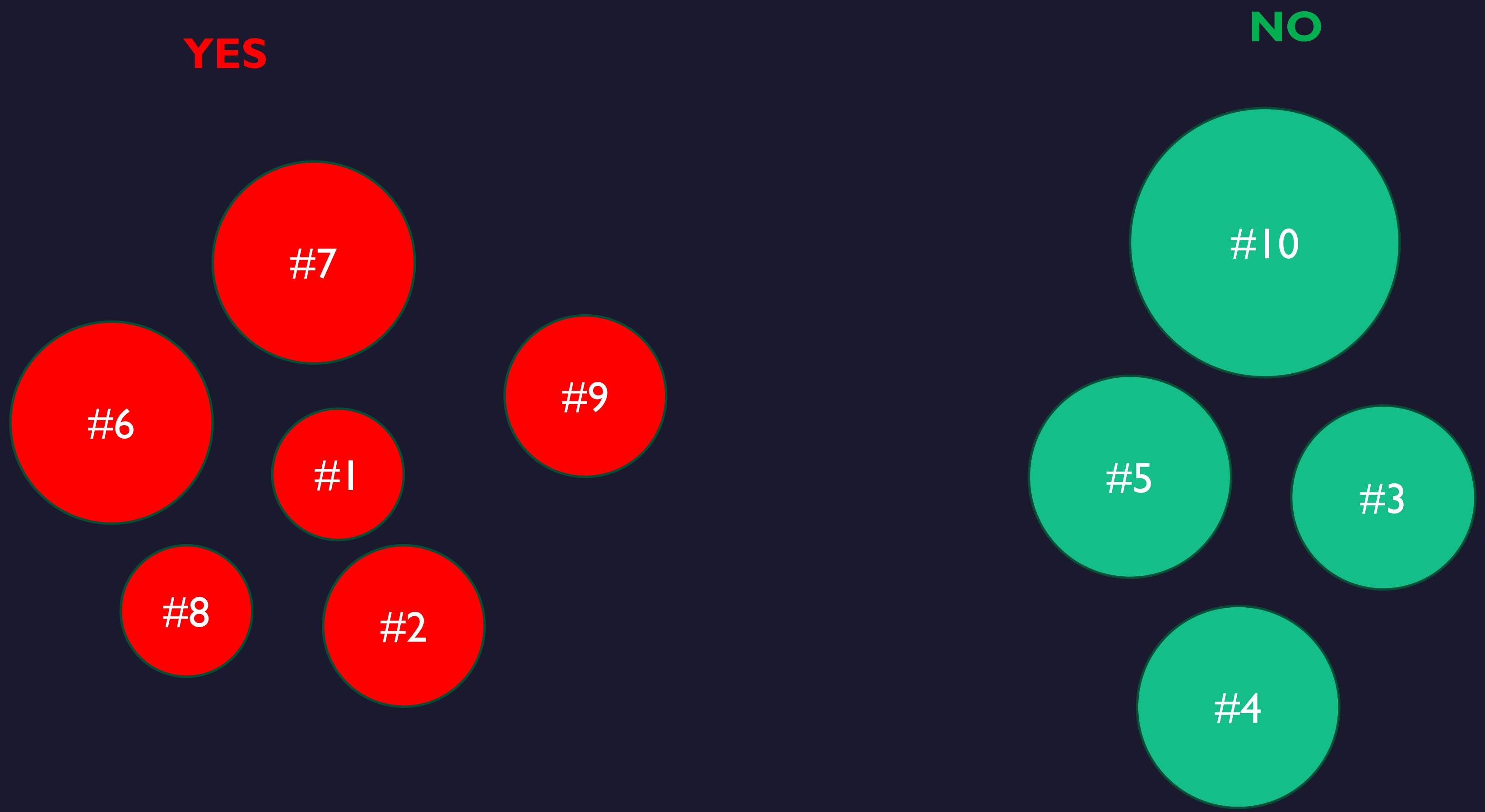
Classification Decision Tree

Circle#	color	r
1	R	1
2	R	3
3	G	4
4	G	5
5	G	6
6	R	7
7	R	7
8	R	1
9	R	3
10	G	8

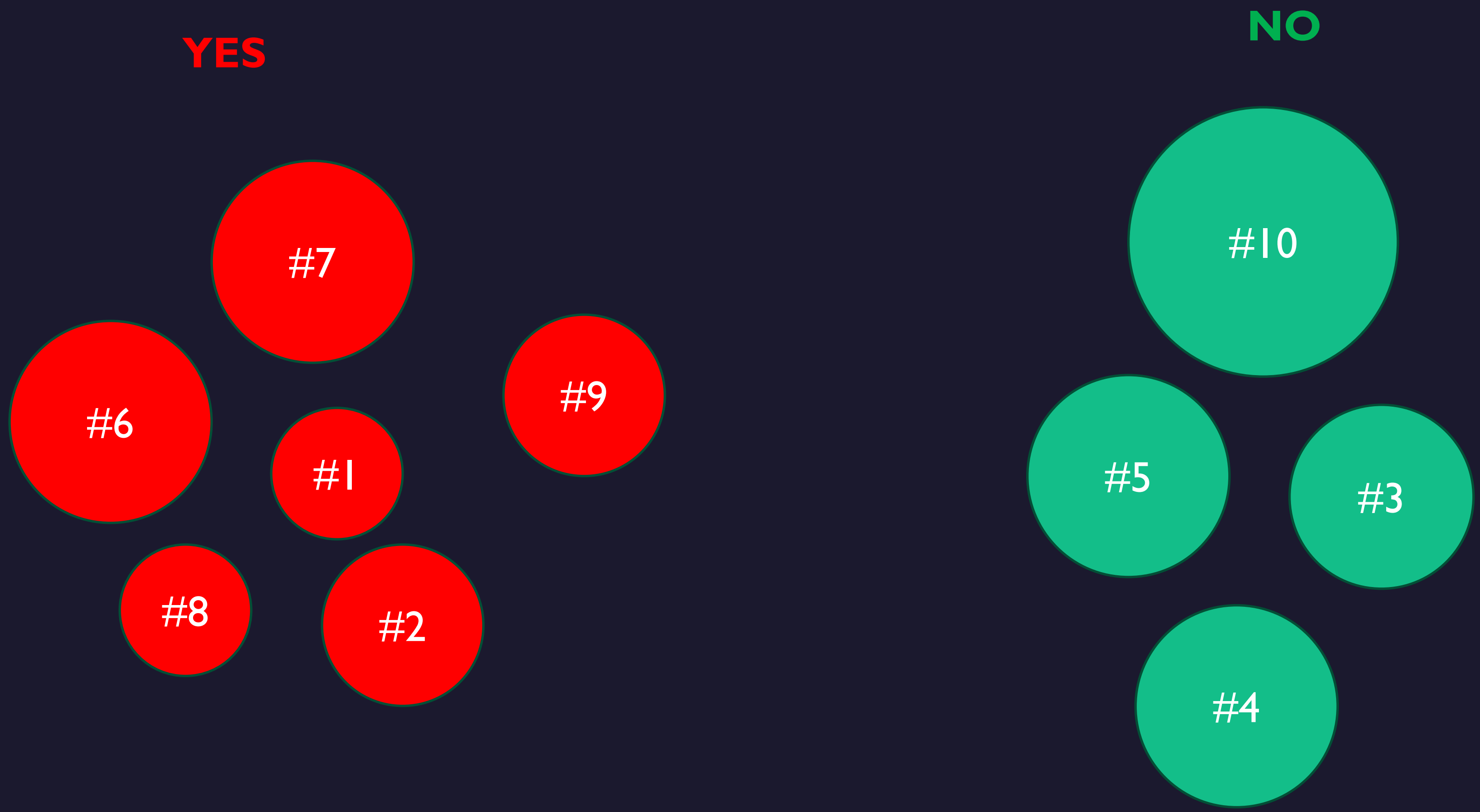


FCI - Helwan Student Branch

Is you color RED?



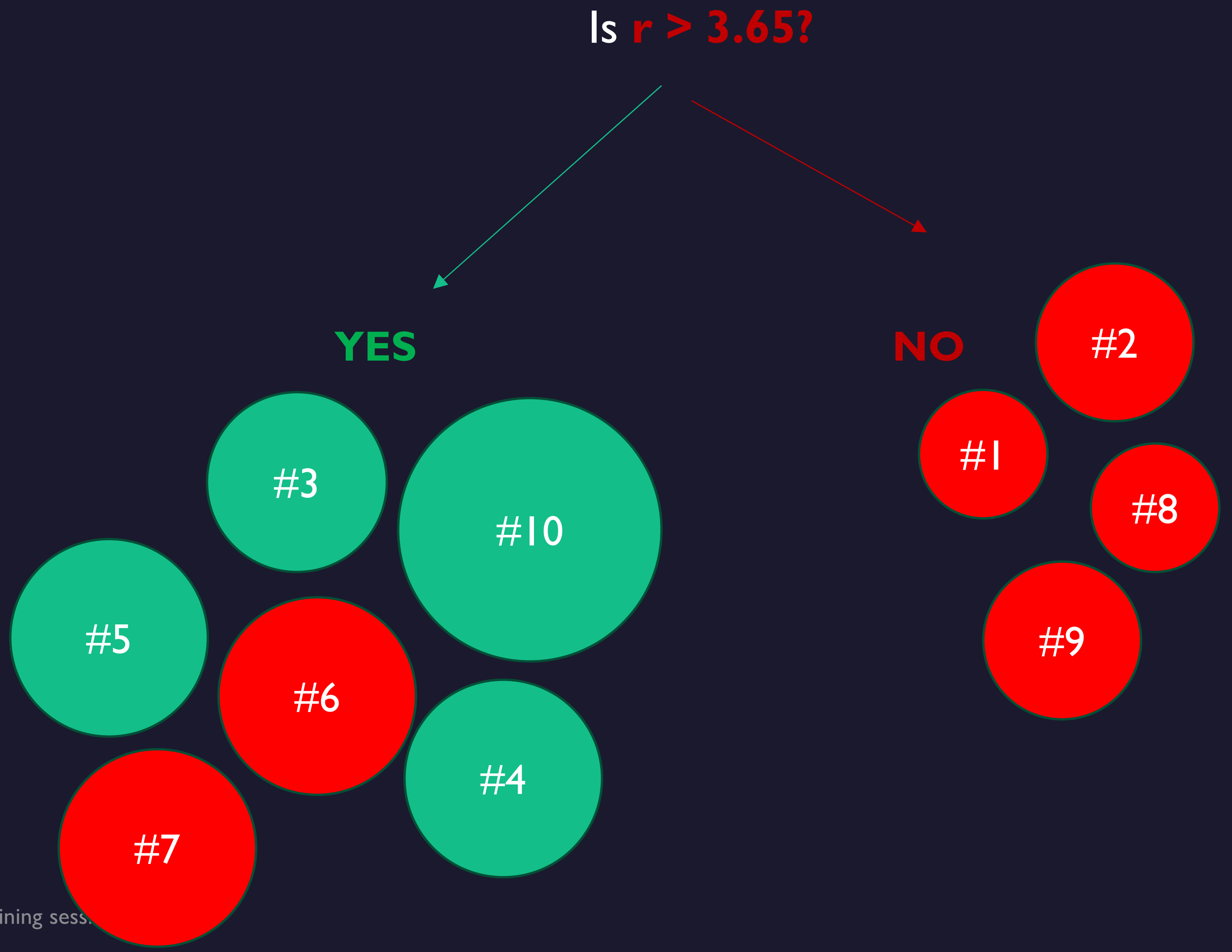
We were a little bit lucky 🍀 ? Choosing color was the best to separate the circles



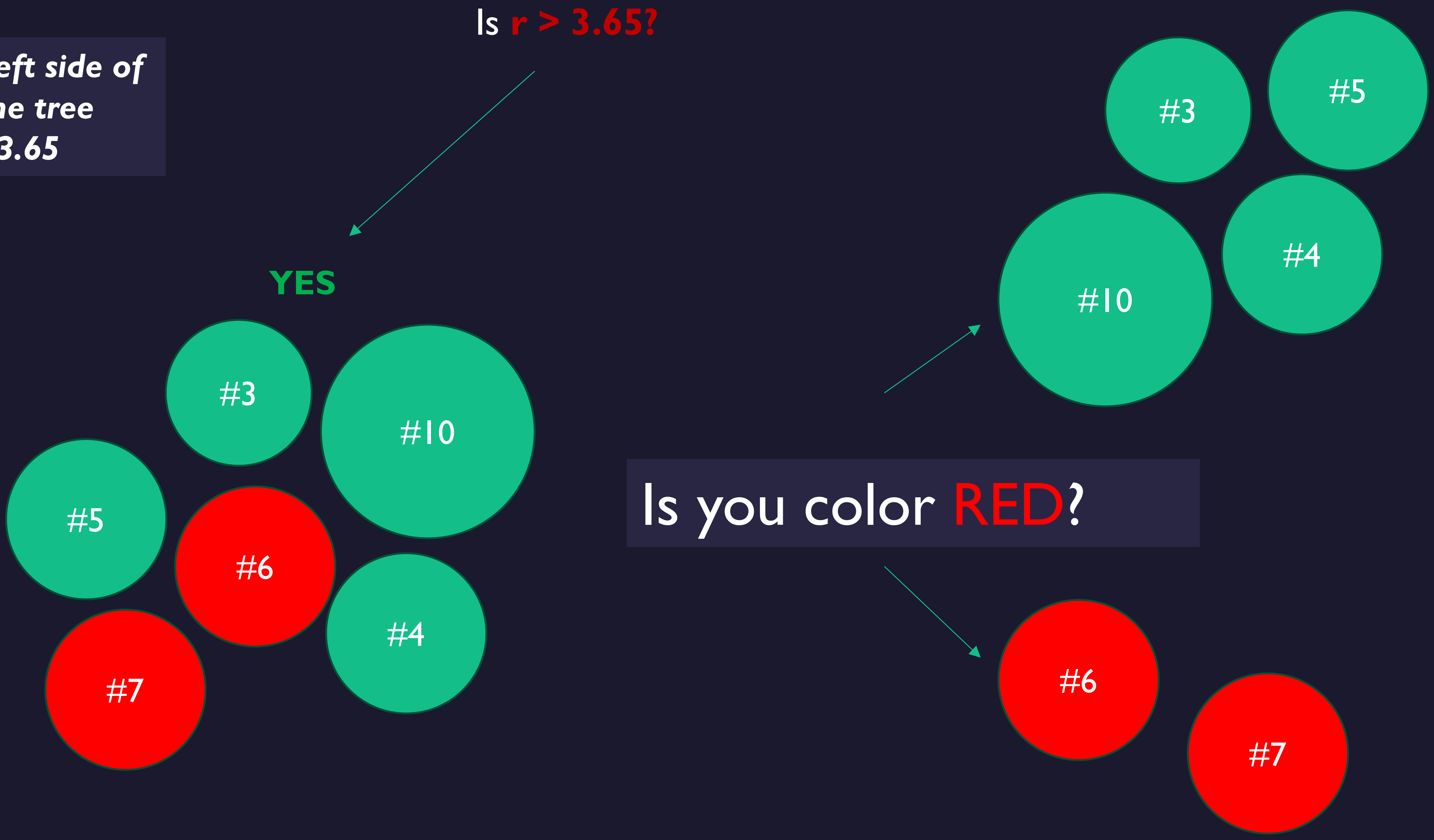
Circle#	color	r
1	R	1
2	R	3
3	G	4
4	G	5
5	G	6
6	R	7
7	R	7
8	R	1
9	R	3
10	G	8

What if we tried to split with r based on threshold 3.65

Circle#	color	r
1	R	1
2	R	3
3	G	4
4	G	5
5	G	6
6	R	7
7	R	7
8	R	1
9	R	3
10	G	8



Left side of the tree >3.65



FCI - Helwan Student Branch IEEE

Is r > 3.65?

right side of the tree <=3.65


Is you color RED?
R
G


Is r > 3.65?
R
Is you color RED?
G
R

From this simple example we see that the **splitting column** can **affect the tree size** , so selecting the **best feature to split** is the challenge in DTs

Also, we can notice that the best feature to split is the one that give us more **pure nodes**

We can see that here simply because we have 2 colors only but what if we got three colors or more classes in each feature

Entropy and **Gini index** two ways decide how to split , by selecting the split give us purest nodes as possible

DT features and notes

- Decision trees use a **top-down** approach called recursive binary splitting.
- Recursive binary splitting starts at the top of the tree and splits the predictor space into two branches.
- The algorithm is greedy because it selects the best split at each step, without considering future splits.
- The algorithm evaluates variables based on statistical criteria to choose the variable that performs best. {Entropy , Gini index}
- The predictor space is divided into two branches at each split, creating a hierarchical structure.
- The algorithm does not project forward to optimize the entire tree, but rather focuses on the current step.

FCI - Helwan
Student Branch
IEEE

Gini index

- The **Gini Index measures the probability of misclassification** for a randomly chosen instance.
- **A lower Gini Index indicates a better split with a lower likelihood of misclassification.**
- The **Gini Index approach** focuses on **measuring impurity**

$$Gini = 1 - \sum_{i=1}^{n} P_i^2$$

- The Gini Index ranges **from 0 (highest purity) to 0.5 (random assignment of classes).**
- To calculate the Gini Index for a node, the Gini Index is **calculated for each sub node(#1)**, and then a **weighted average(#2)** is taken to determine the overall Gini Index for the node

FCI - Helwan Student Branch IEEE

Gini index

the root node : Student Background

#	Result	Other online courses	Background	Working
1	Pass	y	Math	NW
2	Fail	n	Math	W
3	Fail	y	Math	W
4	Pass	y	Cs	NW
5	Fail	n	Other	W
6	Fail	y	Other	W
7	Pass	y	Math	NW
8	Pass	y	Cs	NW
9	Pass	n	Math	W
10	Pass	n	Cs	W
11	Pass	y	Cs	W
12	Pass	n	Math	NW
13	Fail	y	Other	W
14	Fail	n	Other	NW
15	Fail	n	Math	W

1. **Calculate Gini index for each sub node**
   - **Math (total observation 7)**
     - **4 pass, 3 fail**
     - $1 - (P(pass|\ Math))^2 - (P(fail|Math))^2$
     - $1 - \left(\frac{4}{7}\right)^2 - \left(\frac{3}{7}\right)^2 = 0.48979$
   - **CS (total observation 4)**
     - **4 pass, 0 fail**
     - $1 - (P(pass|\ Cs))^2 - (P(fail|Cs))^2$
     - $1 - \left(\frac{4}{4}\right)^2 - \left(\frac{0}{4}\right)^2 = 0$
   - **Other (total observation 4)**
     - **0 pass, 4 fail**
     - $1 - (P(pass|\ Other))^2 - (P(fail|Other))^2$
     - $1 - \left(\frac{0}{4}\right)^2 - \left(\frac{4}{4}\right)^2 = 0$
2. $\text{Gini}_{background} = \frac{7}{15} \times 0.48979 + \frac{4}{15} \times 0 + \frac{4}{15} \times 0 = 0.22857$

Gini index

the root node :Working status

#	Result	Other online courses	Background	Working
1	Pass	y	Math	NW
2	Fail	n	Math	W
3	Fail	y	Math	W
4	Pass	y	Cs	NW
5	Fail	n	Other	W
6	Fail	y	Other	W
7	Pass	y	Math	NW
8	Pass	y	Cs	NW
9	Pass	n	Math	W
10	Pass	n	Cs	W
11	Pass	y	Cs	W
12	Pass	n	Math	NW
13	Fail	y	Other	W
14	Fail	n	Other	NW
15	Fail	n	Math	W

1. **Calculate Gini index for each sub node**
   - **Working (total observation 9)**
     - **6 pass, 3 fail**
     - $1 - (P(pass|\,Work))^2 - (P(fail|Work))^2$
     - $1 - \left(\frac{6}{9}\right)^2 - \left(\frac{3}{9}\right)^2 = 0.44444$
   - **Not Working (total observation 6)**
     - **5 pass, 1 fail**
     - $1 - (P(pass|\,NW))^2 - (P(fail|NW))^2$
     - $1 - \left(\frac{5}{6}\right)^2 - \left(\frac{1}{6}\right)^2 = 0.22777$
2. $\text{Gini}_{WorkingStatus} = \frac{9}{15} \times 0.44444 + \frac{6}{15} \times 0.22777 = 0.35772$

FCI - Helwan Student Branch IEEE

Gini index

the root node :Other online courses

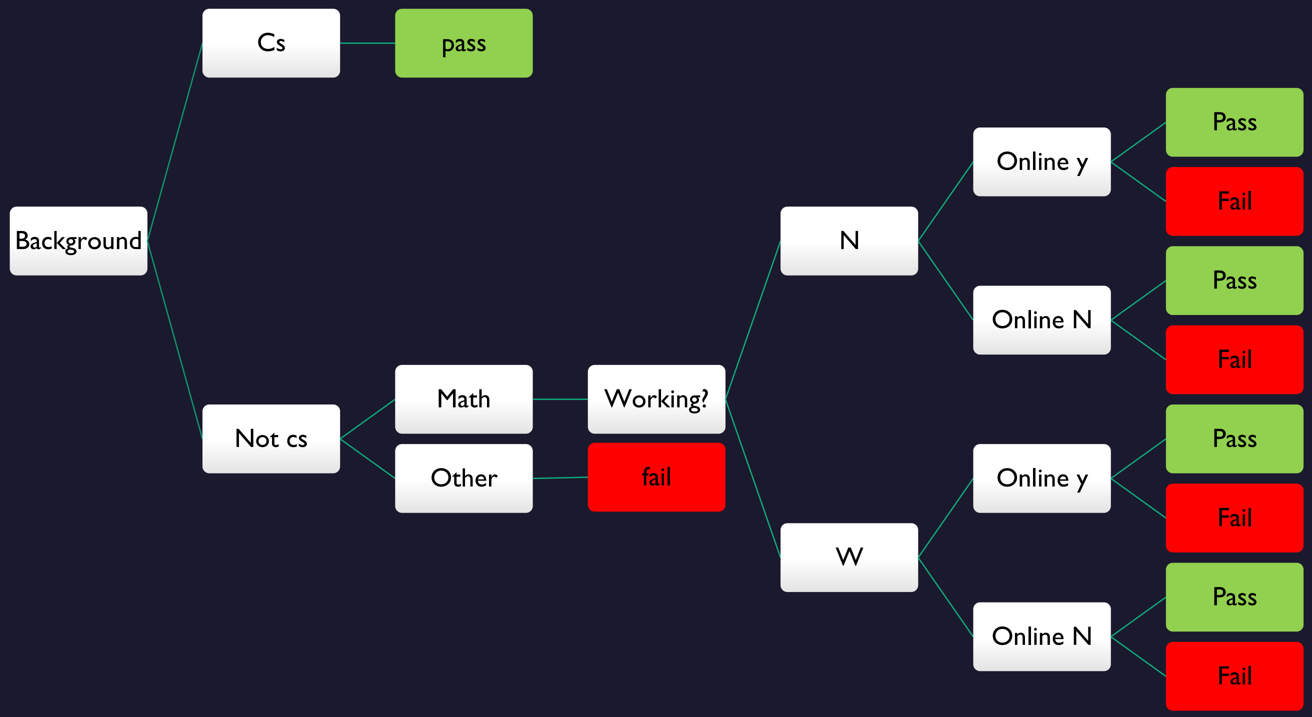
#	Result	Other online courses	Background	Working
1	Pass	y	Math	NW
2	Fail	n	Math	W
3	Fail	y	Math	W
4	Pass	y	Cs	NW
5	Fail	n	Other	W
6	Fail	y	Other	W
7	Pass	y	Math	NW
8	Pass	y	Cs	NW
9	Pass	n	Math	W
10	Pass	n	Cs	W
11	Pass	y	Cs	W
12	Pass	n	Math	NW
13	Fail	y	Other	W
14	Fail	n	Other	NW
15	Fail	n	Math	W

1. **Calculate Gini index for each sub node**
   - **Yes (total observation 8)**
     - **5 pass, 3 fail**
     - **1 -** $(P(pass|\, Yes))^2 - (P(fail|Yes))^2$
     - **1 –** $\left(\frac{5}{8}\right)^2 - \left(\frac{3}{8}\right)^2$ = 0.46875
   - **No (total observation 7)**
     - **3 pass, 4 fail**
     - **1 -** $(P(pass|\, No))^2 - (P(fail|No))^2$
     - **1 –** $\left(\frac{3}{7}\right)^2 - \left(\frac{5}{7}\right)^2$ = 0.48798
2. **Gini**$_{OnlineCourses}$ = $\frac{8}{15}\times 0.46875 + \frac{7}{15}\times 0.48798 = 0.47825$

Gini index

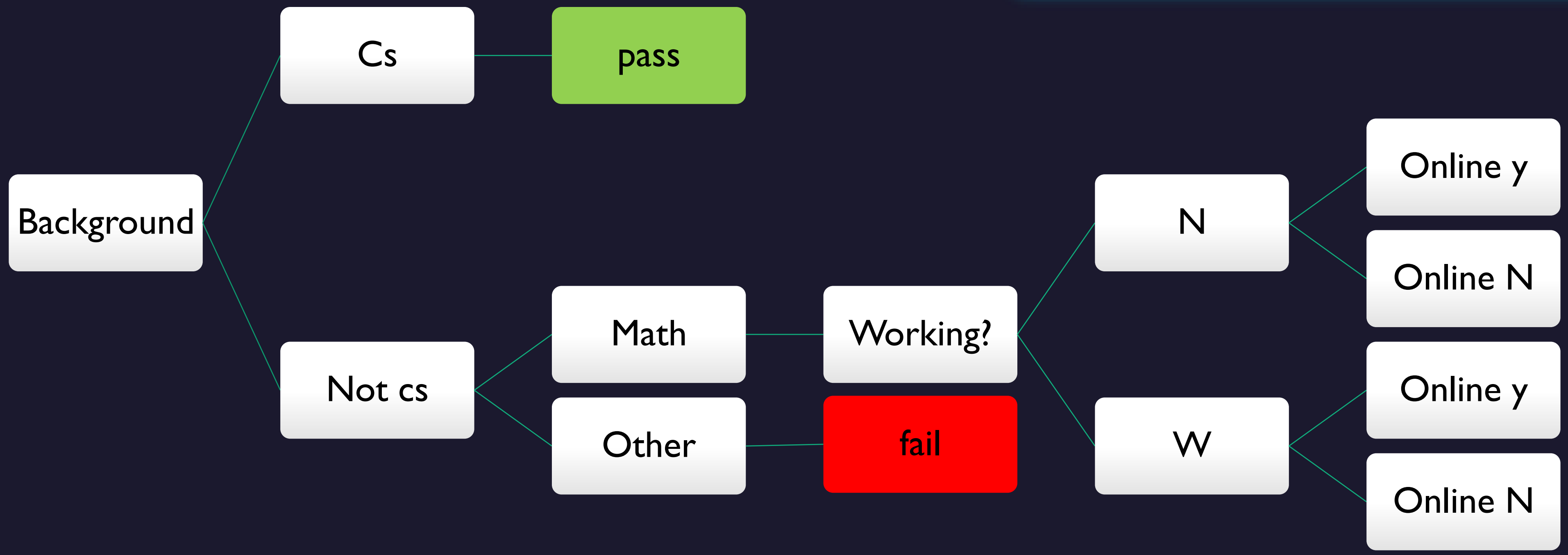
#	Result	Other online courses	Background	Working
1	Pass	y	Math	NW
2	Fail	n	Math	W
3	Fail	y	Math	W
4	Pass	y	Cs	NW
5	Fail	n	Other	W
6	Fail	y	Other	W
7	Pass	y	Math	NW
8	Pass	y	Cs	NW
9	Pass	n	Math	W
10	Pass	n	Cs	W
11	Pass	y	Cs	W
12	Pass	n	Math	NW
13	Fail	y	Other	W
14	Fail	n	Other	NW
15	Fail	n	Math	W

1. $\mathbf{Gini}_{OnlineCourses} = \frac{8}{15}\times 0.46875+\frac{7}{15}\times 0.48798= 0.47825$
2. $\mathbf{Gini}_{WorkingStatus} = \frac{9}{15}\times 0.44444+\frac{5}{15}\times 0.22777 = 0.35772$
3. $\mathbf{Gini}_{background} = \frac{7}{15}\times 0.48979+\frac{4}{15}\times 0+\frac{4}{15}\times 0 = 0.22857$



Gini index

We may not get all leaves pure single class pass/fail , maybe tree terminated with heterogeneous classes here we may classify based on the majority class in the leaf



FCI - Helwan Student Branch
IEEE

Gini index

A full decision tree is a tree that is grown until all the terminal nodes (i.e., the leaves) contain a **minimum number of observations,** or until all the observations in the training set belong to **the same class**. However, even if a decision tree is grown to its maximum depth, there may still be some leaves that are not pure if the algorithm determines that *further splitting of the data would not significantly improve the classification accuracy*

FCI - Helwan Student Branch

IEEE

Entropy

- ***In information theory, the entropy of a random variable is the average level of "information", "surprise", or "uncertainty" inherent to the variable's possible outcome***
- It is often associated with a state of disorder, randomness, or uncertainty.
- Entropy in Decision Trees is a measure of disorder or impurity in a node.
- **Nodes with a *more diverse composition have higher entropy* than nodes with a single category.**
- Entropy ranges from 0 to 1, with **0 representing minimum entropy (pure node)** and **1 representing maximum entropy (high disorder).**
- Entropy helps determine the homogeneity or purity of a node in Decision Trees.

FCI - Helwan Student Branch IEEE

Entropy

$$E = -\sum_{i=1}^{n} P_i \log_2(P_i)$$

- **Information Gain** measures the amount of information a feature provides for predicting the target variable.

$$Information\ Gain = Entropy_{\ parent} - Entropy_{\ children}$$

FCI - Helwan Student Branch
IEEE

Entropy

$$P(pass) = 8/15 = 0.5333$$
$$P(fail) = 7/15 = 0.4666 P(fail|student\ background) = 7/15 = 0.4666$$

1. calculate the Entropy

$$E = -\sum_{i=1}^{2} P_i \times \log_2 P_i =$$
$$-[0.5333 \times \log_2 0.533 + 0.4666 \times \log_2 0.4666] = 0.99679$$

Entropy of the column parent
P(target = T/F)

you have 8 pass, 7 fail

you have 9 students Work and 6 not working

$P(pass|working) = [P(pass) \cap P(work)]/P(work) = 3/9$

كام واحد شغال ونجح على عدد ال شغالين كلهم

$P(fial|working) = 6/9$

$P(pass|Not\ working) = 5/6$

$P(fail|Not\ working) = 1/6$

Entropy for the column classes (childrens)
P(target|column class)
class := eg. true/false...

$$E_{working} = -[\frac{3}{9} \times \log_2(3/9) + \frac{6}{9} \times \log_2(6/9)] = 0.918295$$

$$E_{Not\ Working} = -[5/6 \times \log_2(5/6) + 1/6 \times \log_2(1/6)] = 0.65002$$

Average Entropy of the classes of the column

$$E_{working-status} = [\frac{working}{15} \times Entropy_{working} + \frac{not\ working}{15} \times Entropy_{Not\ working}]$$

$$E_{working-status} = [\frac{9}{15} \times 0.918295 + \frac{6}{15} \times 0.65002] = 0.8109$$

calculate information gain

$$InformationGain = Entropy_{Parent} — Entropy_{child} = 0.99679–0.8109 = 0.18589$$

$$Information\ Gain = E_{parent} - AvgE_{child}$$

Do for each column and select to split with higher info gain

	Entropy Node	Average Entropy	Information Gain
Parent	0.9968		
working	0.9183	0.8110	0.1858
Not_work	0.6500		
Bkgrd_Ma	0.9852	0.4598	0.5370
Bkgrd_CS	0.0000		
Bkgrd_oth	0.0000		
online_co	0.9544	0.9688	0.0280
online_no	0.9852		

Gini-index *Vs* Entropy

Aspect	Entropy	Gini Index
Formula	$-\sum P_i \log_2 P_i$	$1-\sum P_i^2$
Range	0 to 1	0 to 0.5
Sensitivity	More sensitive to class distribution	Less sensitive to class distribution handle imbalanced labels better
Computation Speed	Slower (logarithms)	Faster (squared probabilities)
Tree Depth	Can create deeper trees	Tends to create shallower trees
When to Use	Precise splits & information gain	Faster computation & simpler trees

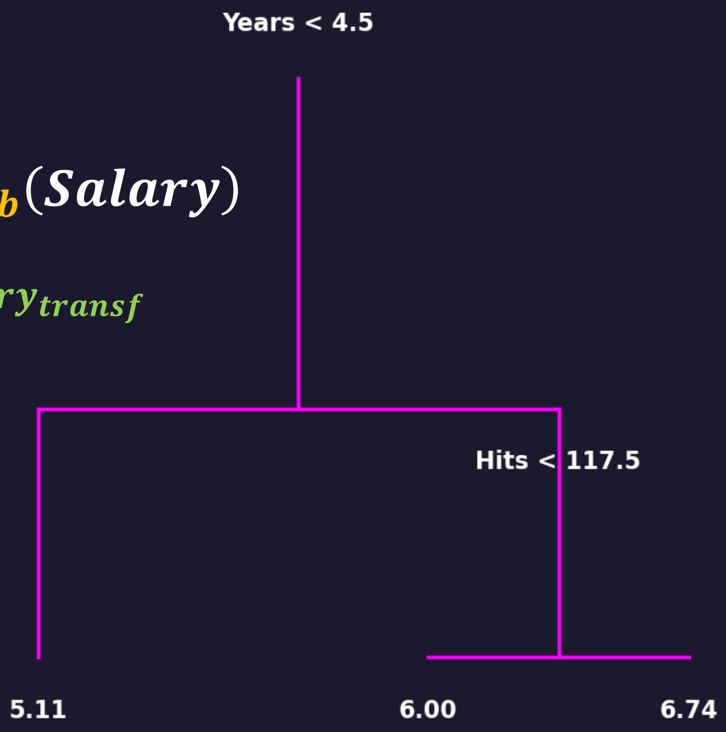
Regression DTs

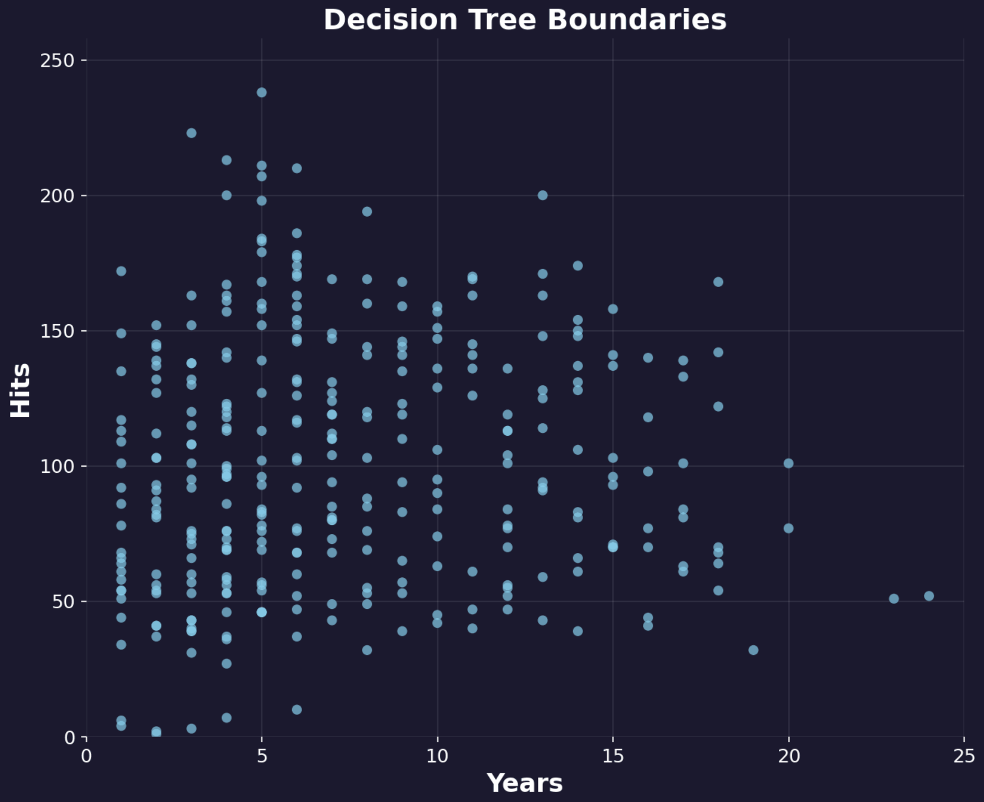
The goal is to predict a baseball player's **log-transformed salary (y)** based on:

1. Number of **years** (x_1) they've played in the major leagues
2. Number of **hits** (x_2) they made in the previous year

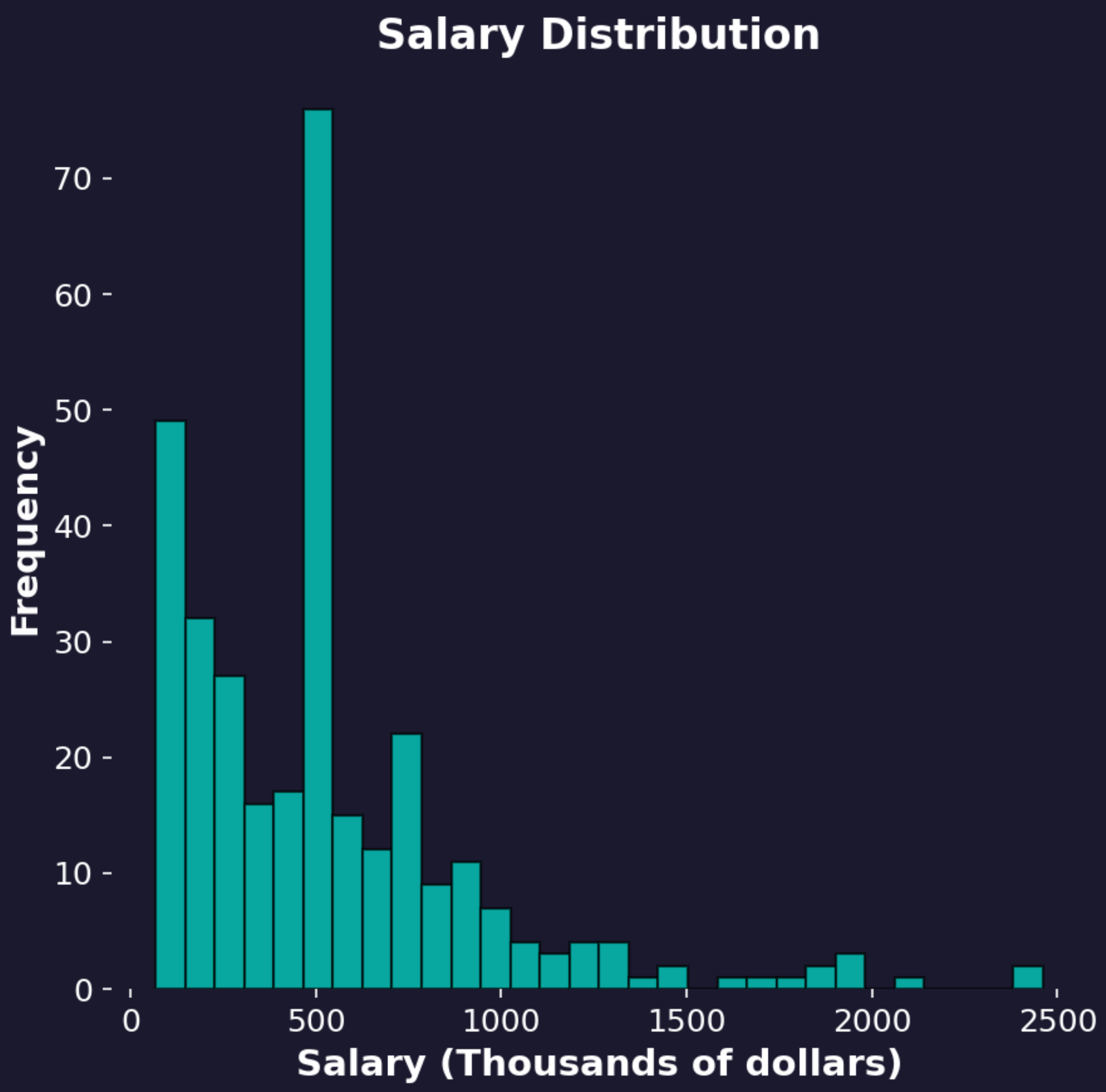
$$Salary_{transf} = log_b(Salary)$$

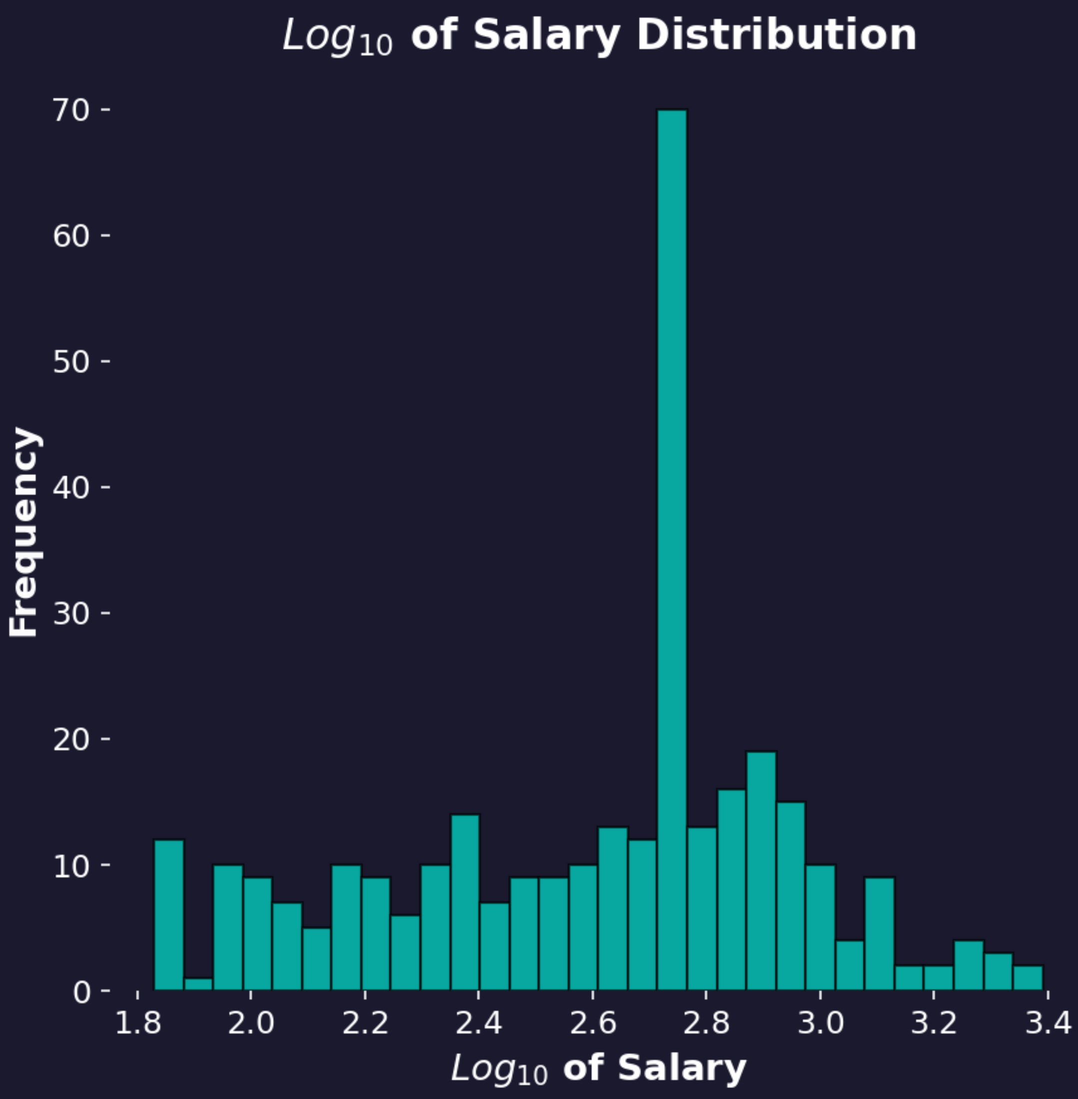
$$Salary_{orig} = b^{Salary_{transf}}$$





Regression DTs





FCI - Helwan Student Branch

IEEE

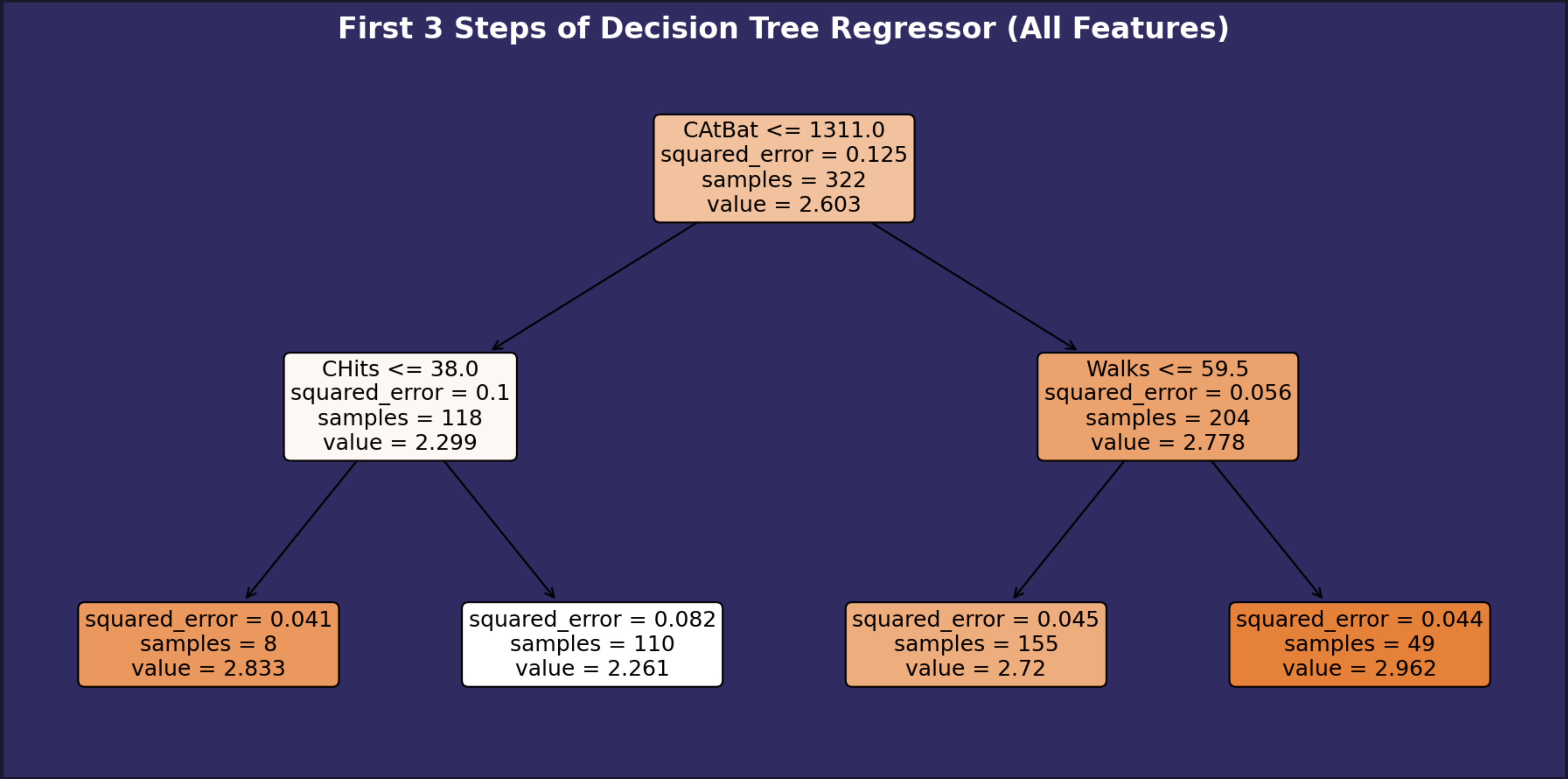
Building the Regression tree

A **regression DT** predicts **continuous** values (e.g., house prices, salaries) by **partitioning** the feature space into regions and **assigning a constant value** (typically the mean) to each region

1. Start with the full dataset at the root node
2. Select splits based on splitter strategy (minimizing total MSE/RSS) :
   - "best" : For all features, evaluate all possible thresholds (midpoints)
   - "random" : Try random feature(s) and random thresholds
3. Split the dataset into left (≤ threshold) and right (> threshold) subsets (binary splitting)
4. Repeat steps 2-3 **recursively** for each child node
5. Stop splitting when :
   - Max depth, min samples, or min impurity decrease is reached, or the node is pure
6. Assign leaf prediction as the **mean target value** in that node
7. **Predict by traversing** the tree based on input features and **returning the leaf value**

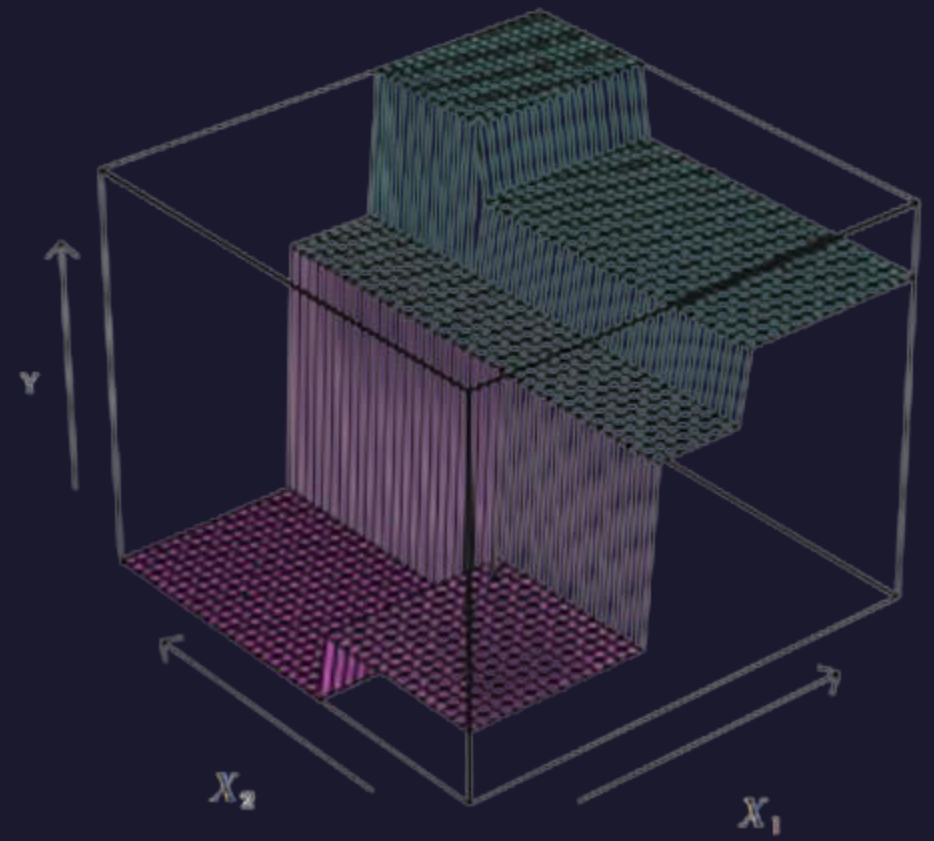
FCI - Helwan Student Branch
IEEE

Building the Regression tree



FCI - Helwan Student Branch
IEEE

Tree in 3 predictive variable space



FCI - Helwan Student Branch IEEE

Tree pruning

Tree pruning

Pruning is the process of removing non-essential branches from a decision tree to reduce overfitting, improve generalization, and enhance model interpretability.

Aspect	**Pre-Pruning (Early Stopping)**	**Post-Pruning (Cost-Complexity Pruning)**
Timing	During tree growth	After full tree is built
Method	Stops splits based on fixed criteria (e.g., depth, samples)	Removes branches using a complexity penalty (α)
Bias–Variance Tradeoff	May increase bias, reduce variance	More balanced; guided by validation
Risk	Underfitting if too strict	Lower risk; overfitting is corrected after training
Selection Basis	User-defined thresholds (conditions)	Validation error (cross-validation or hold-out set)
Efficiency	Faster training	More computation required

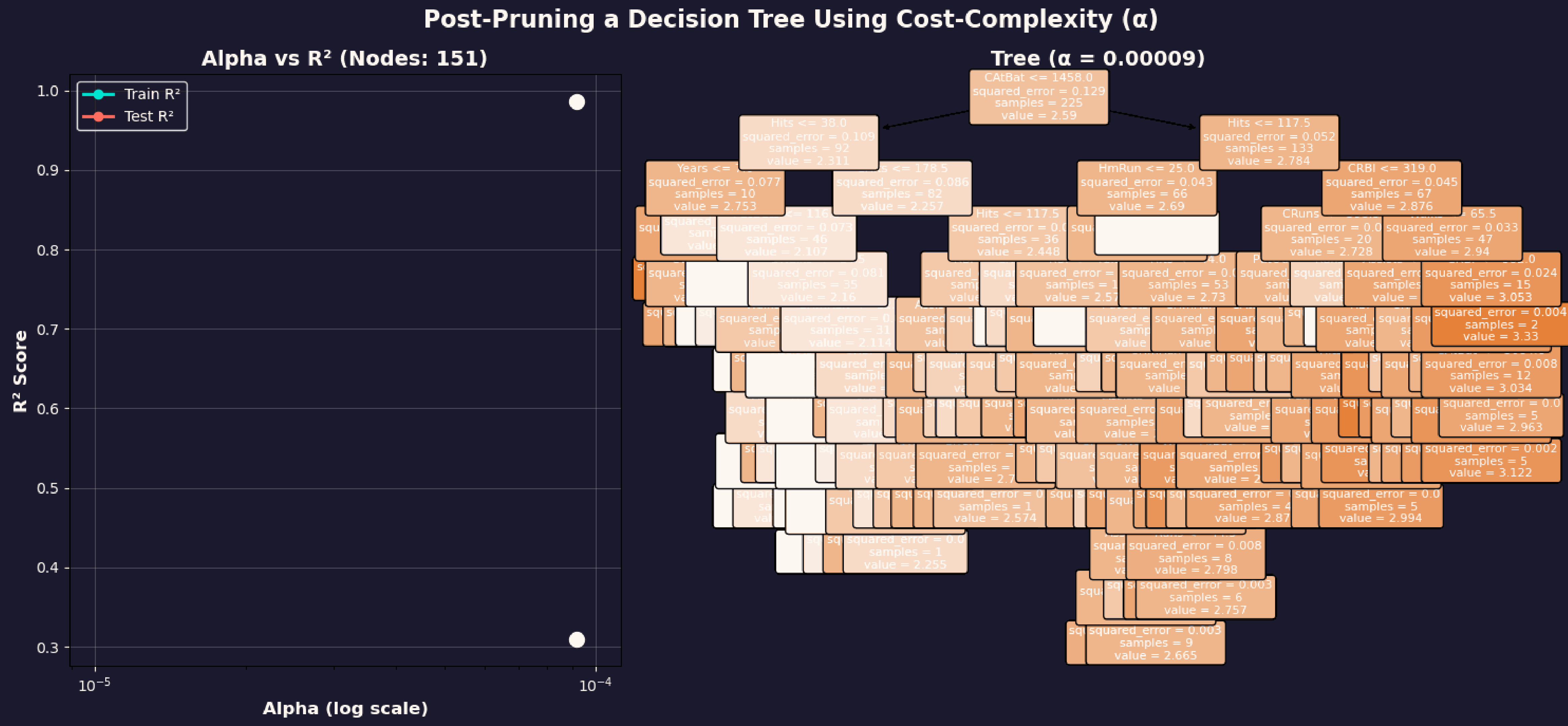
Post-pruning (CCP)

Instead of limiting tree growth prematurely (pre-pruning), CCP grows a **full tree T_0** and then prunes it back, enabling better fitting and more informed complexity control

- Select a subtree $T \subseteq T_0$ that minimize the test error, by balancing accuracy and complexity
- For a giving $\boldsymbol{\alpha} \geq 0, CCP$ minimize $R_\alpha(T) = R(T) + \boldsymbol{\alpha}\ |\boldsymbol{T}|$
  - Where $R(T)$ is the training error and $|\boldsymbol{T}|$ (penalty) is the number of terminal nodes
- Behavior of $\boldsymbol{\alpha}$
  - $\alpha = 0$ selects the full tree
  - As $\boldsymbol{\alpha}$ increases, simpler tree are favored
  - Produce a **nested sequence** of subtrees $\boldsymbol{T_0} \supset \boldsymbol{T_1} \supset \boldsymbol{T_2} \supset \cdots \supset \boldsymbol{T_k}$ from the full tree T_0 to the root-only tree
- Use cross-validation or a validation set to choose the subtree with the lowest estimated test error

FCI - Helwan Student Branch

Post-pruning (CCP)



See 👀

- http://www.r2d3.us/visual-intro-to-machine-learning-part-1/
- http://www.r2d3.us/visual-intro-to-machine-learning-part-2/
- https://mlu-explain.github.io/decision-tree/

References

- http://www.r2d3.us/visual-intro-to-machine-learning-part-1/
- http://www.r2d3.us/visual-intro-to-machine-learning-part-2/
- https://hossam-ahmed.notion.site/8-Tree-based-Model-c3a1186914ed40f7b4298dd5493f3fb3?pvs=4
- https://hossam-ahmed.notion.site/Entropy-Information-Gain-Gini-Index-1fbaf1424fe8495f91b68d11a071a930?pvs=4
- https://en.wikipedia.org/wiki/Decision_tree_learning
- https://quantdare.com/decision-trees-gini-vs-entropy/
- https://www.kaggle.com/datasets/floser/hitters [Hitters Dataset]
- Book: statistical learning