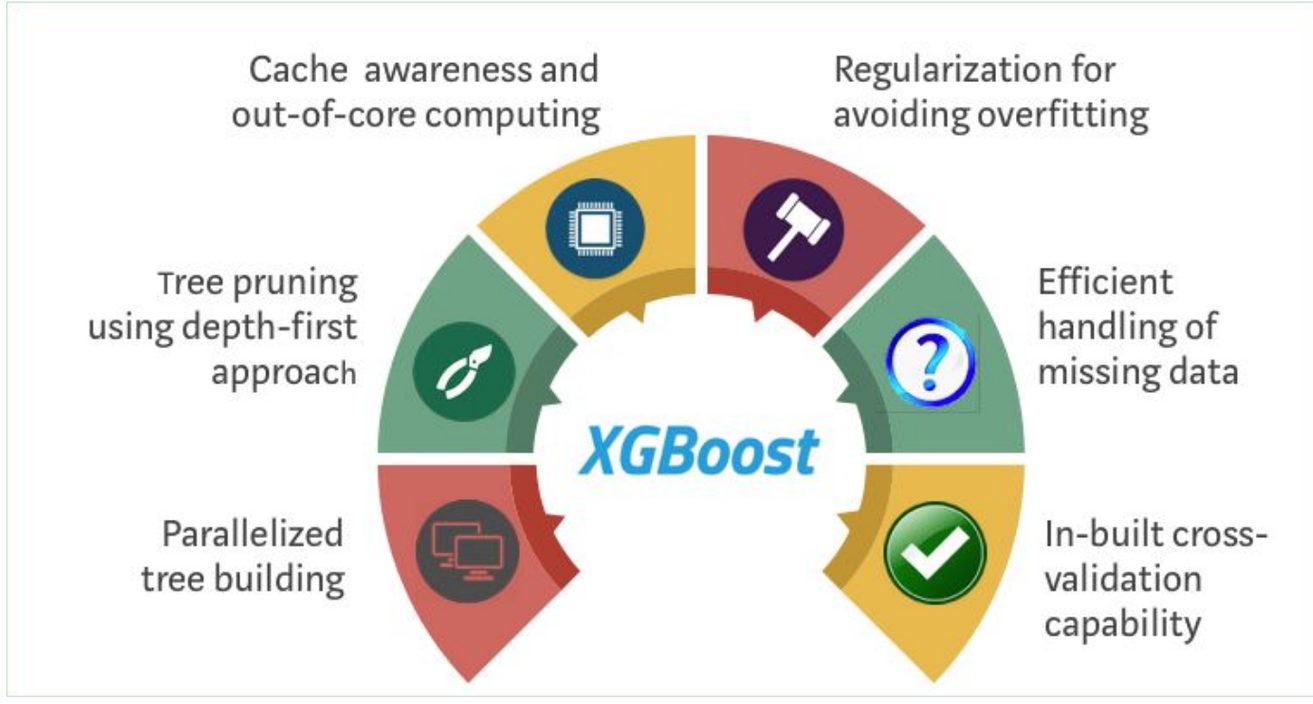
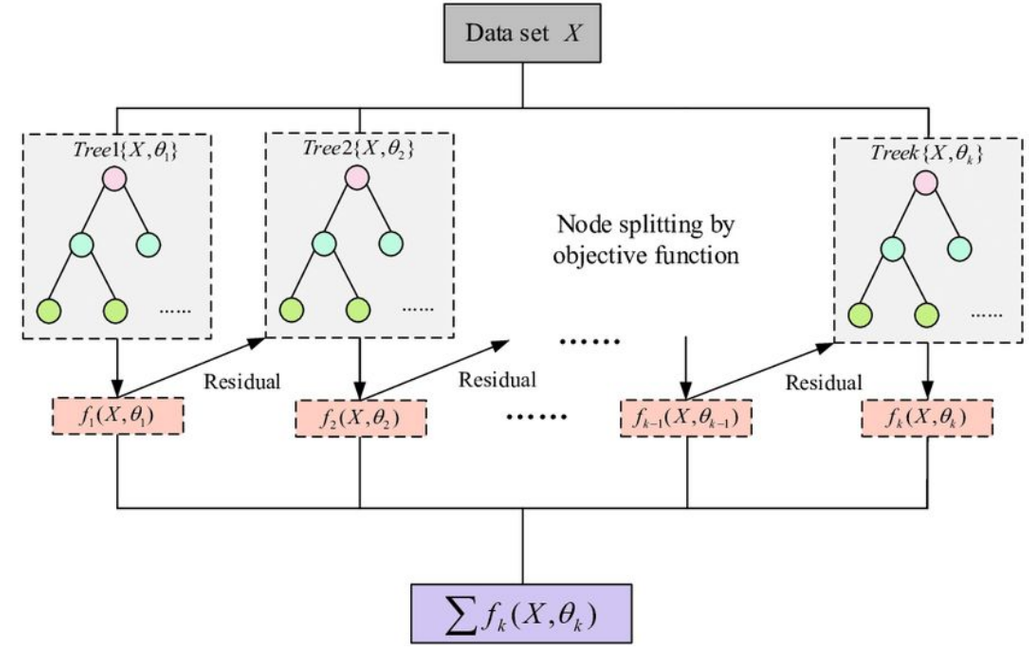
AI VIETNAM
All-in-One Course
(TA Session)

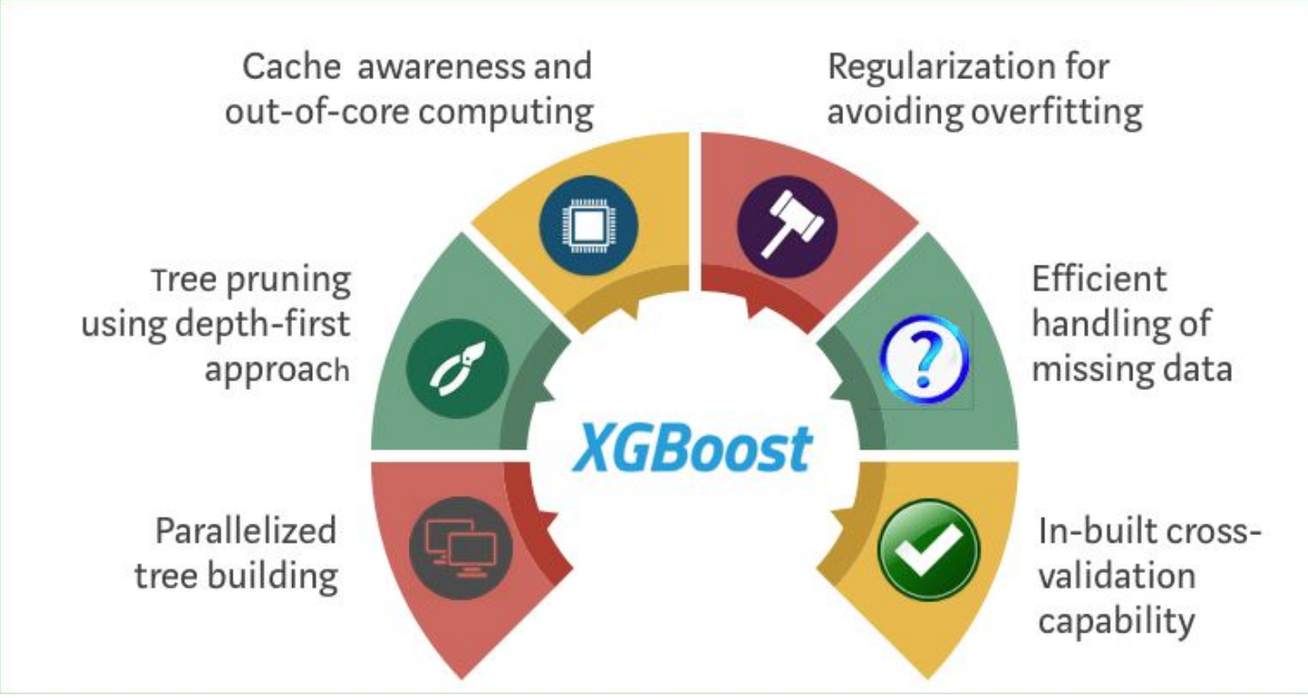
XGBoost

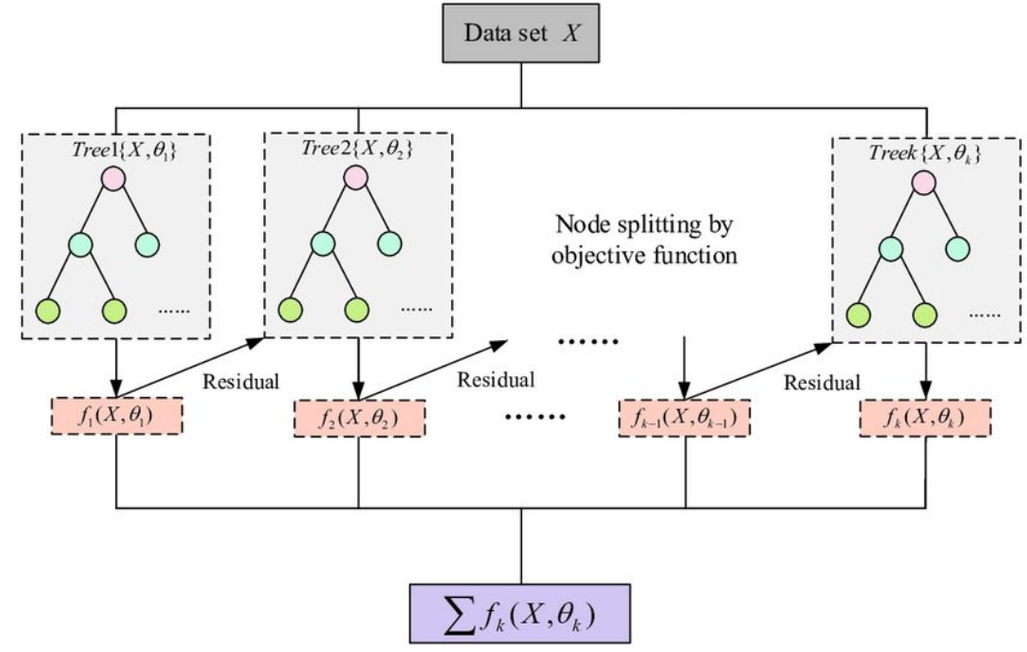




Nguyen Tho Anh Khoa - Nguyen Dang Nha

Objectives





- ✔ Introduce ensemble learning techniques.
- ✔ Describe decision tree structures and functions.
- ✔ Explain the concept and purpose of bagging.
- ✔ Describe the random forest algorithm.
- ✔ Discuss the boosting approach.
- ✔ Elucidate the gradient boosting process.
- ✔ Provide basics of XGBoost.
- ✔ Demonstrate applying XGBoost to simple data for regression and classification.
- ✔ Showcase practical implementation of XGBoost with library tools for regression and classification

AI VIETNAM
All-in-One Course
(TA Session)

Outline

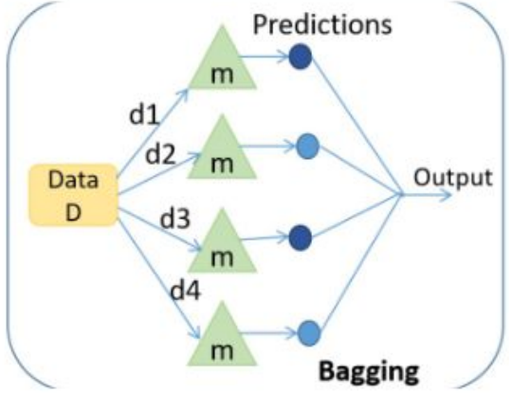
1. Review Ensemble Learning
2. Review Decision Tree
3. Review Bagging
4. Review Random Forest
5. Review Boosting
6. Review Gradient Boosting
7. Review XGBoost Basics
8. Problem 1-2
9. Problem 3-4
10. Applications

Review Ensemble Learning

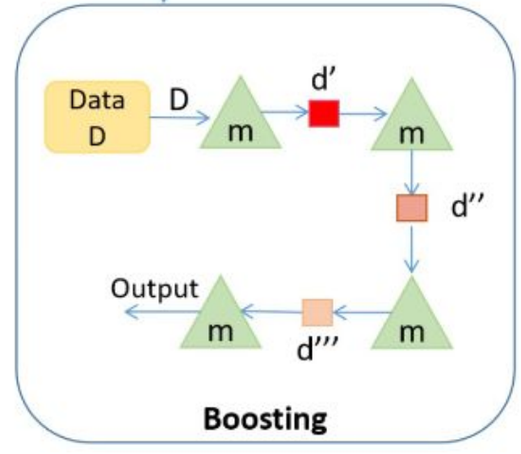
Thông dụng ở các cuộc thi về AI

Ensemple Learning

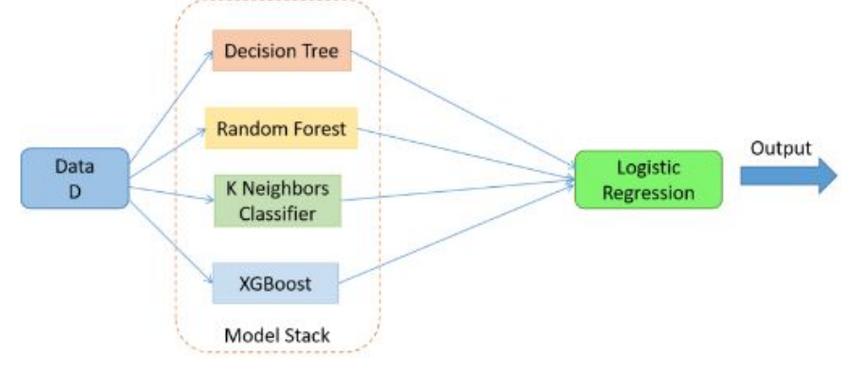
Bagging
homogeneous weak learners



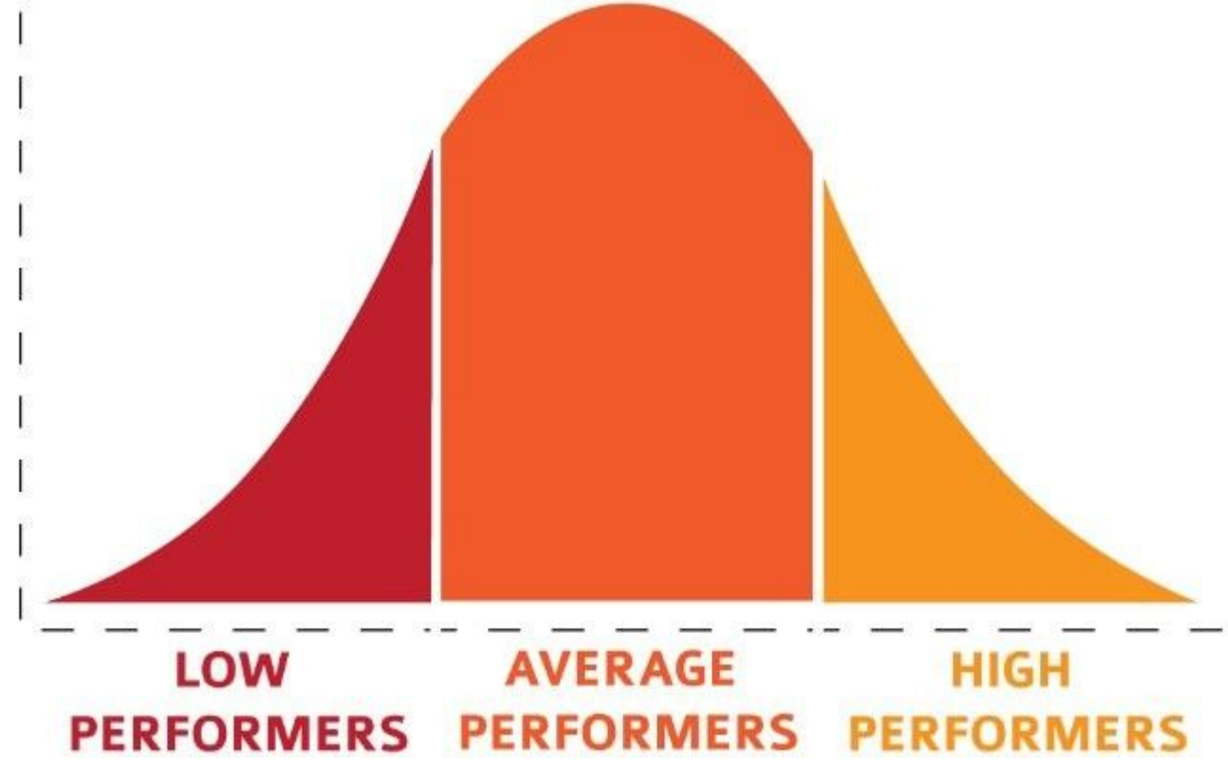
Boosting
homogeneous weak learners



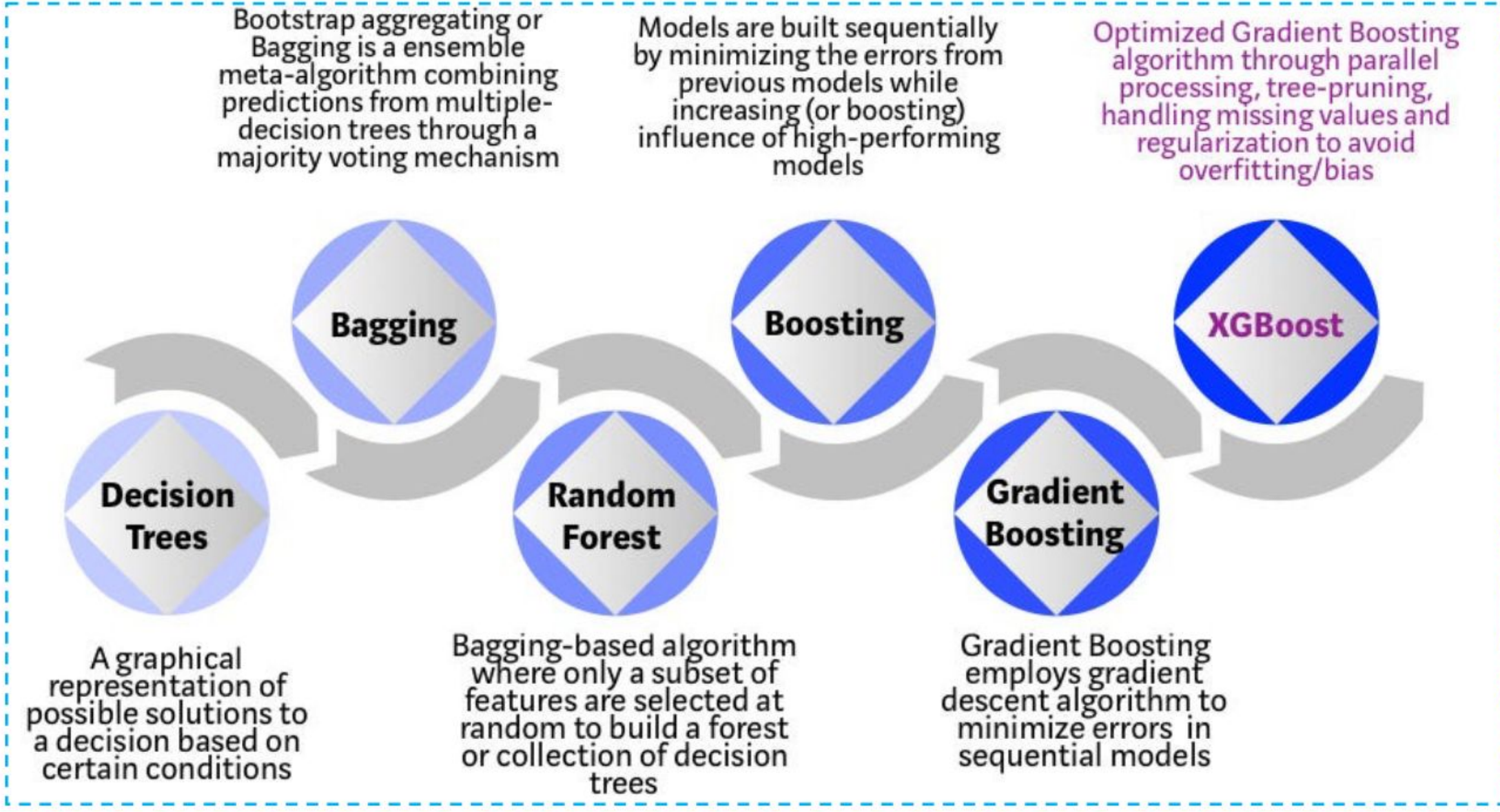
Stacking
Heterogeneous weak learners



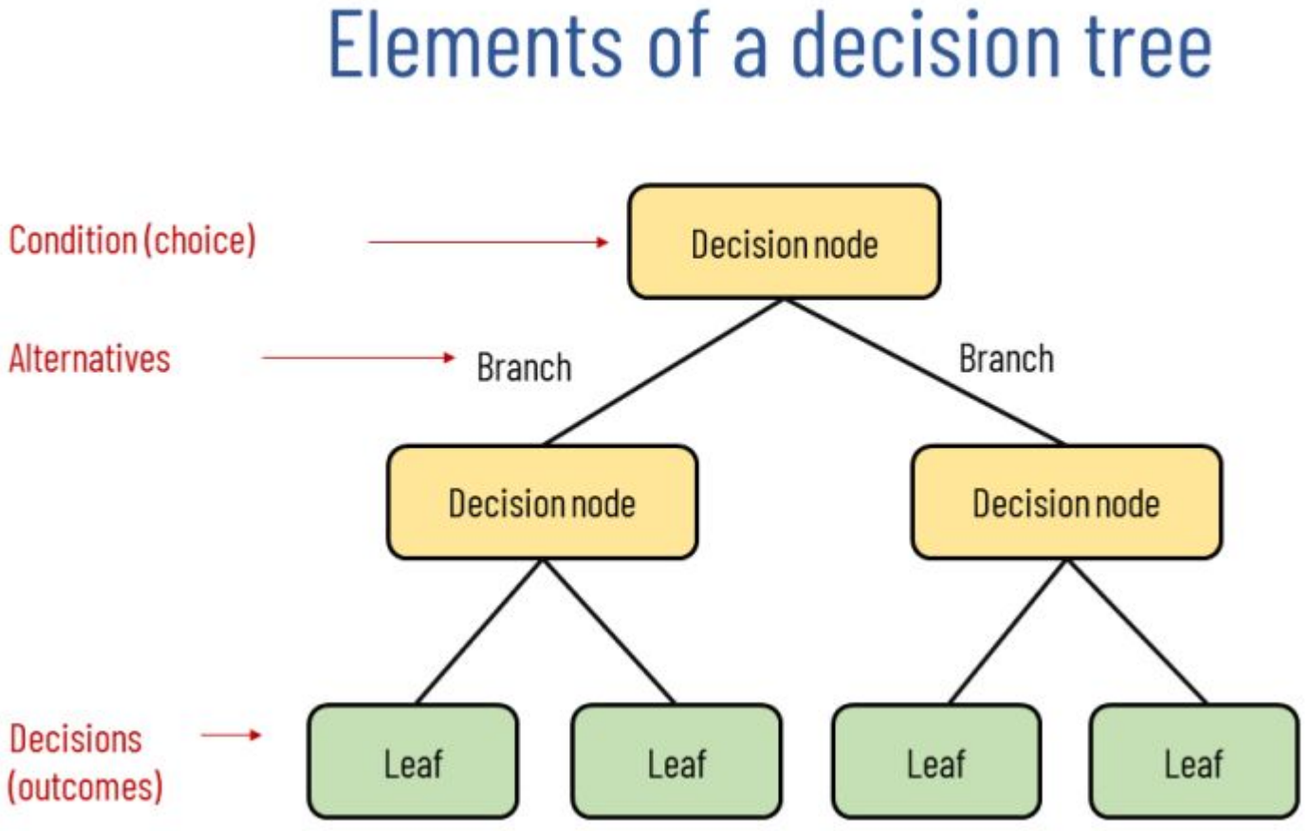
Review Ensemble Learning



Introduction



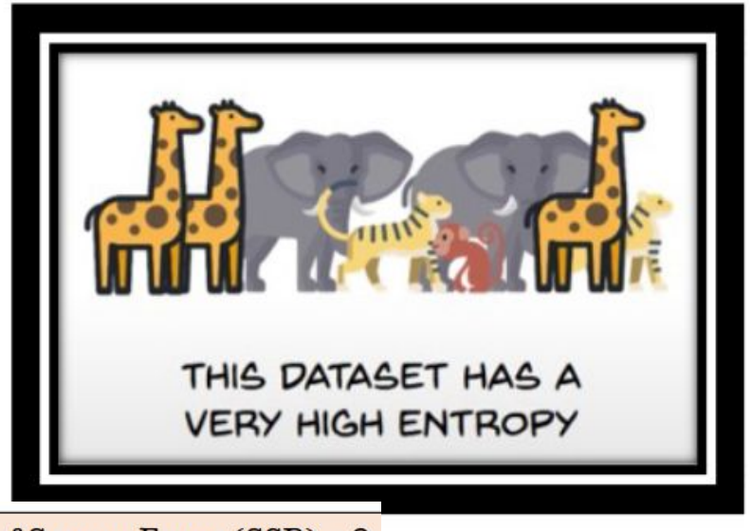
Review Decision Tree



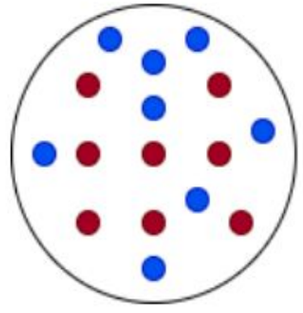
- ***Root Node**:* The initial node at the beginning of a decision tree, where the entire population or dataset starts dividing based on various features or conditions.
- ***Decision Nodes**:* Nodes resulting from the splitting of root nodes are known as decision nodes. These nodes represent intermediate decisions or conditions within the tree.
- ***Leaf Nodes**:* Nodes where further splitting is not possible, often indicating the final classification or outcome. Leaf nodes are also referred to as terminal nodes.
- ***Pruning***: The process of removing or cutting down specific nodes in a decision tree to prevent overfitting and simplify the model.
- ***Branch / Sub-Tree**:* A subsection of the entire decision tree is referred to as a branch or sub-tree. It represents a specific path of decisions and outcomes within the tree.
- ***Parent and Child Node**:* In a decision tree, a node that is divided into sub-nodes is known as a parent node, and the sub-nodes emerging from it are referred to as child nodes. The parent node represents a decision or condition, while the child nodes represent the potential outcomes or further decisions based on that condition

Review Decision Tree

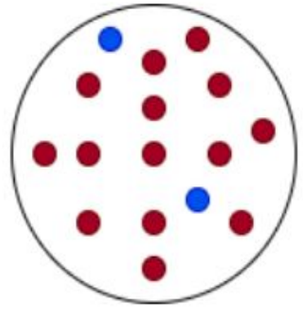
Entropy is an information theory metric that measures the impurity or uncertainty in a group of observations (dataset)



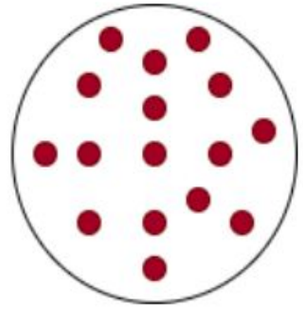
Very impure



Less Impure



Minimum Impurity



Sum of Square Error (SSR) = 0

$$\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2$$

number of samples: n; real value: Y_i; predicted value: $\hat{Y}_i$

sum of the errors of all samples

$$IG(Y, X) = E(Y) - E(Y|X)$$

$$Gain = E_{parent} - E_{children}$$

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

Này là gì vậy?
Nhìn rối quá

Review Decision Tree

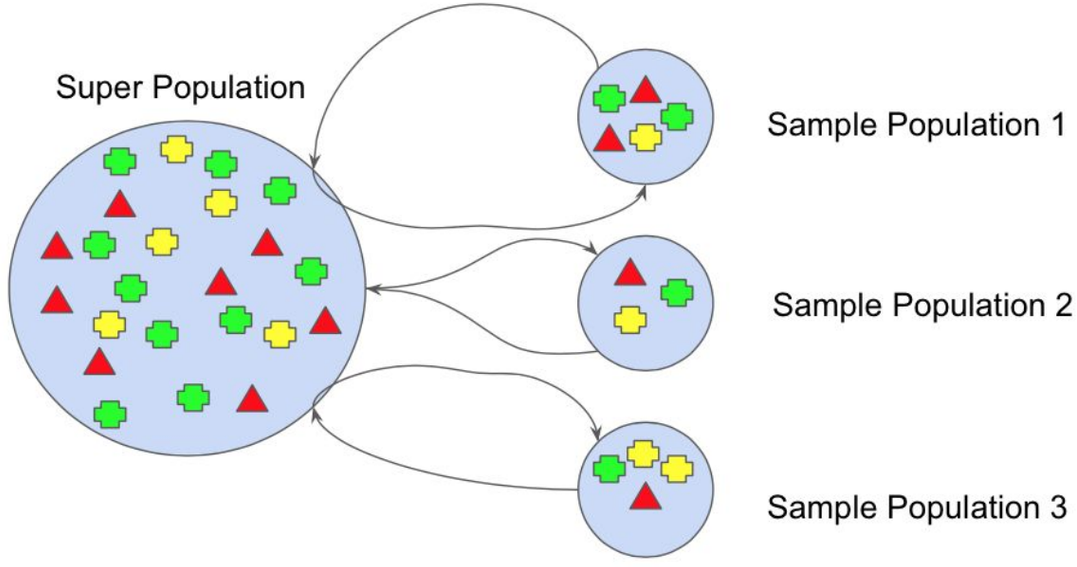
It was developed by statistician and sociologist Corrado Gini.

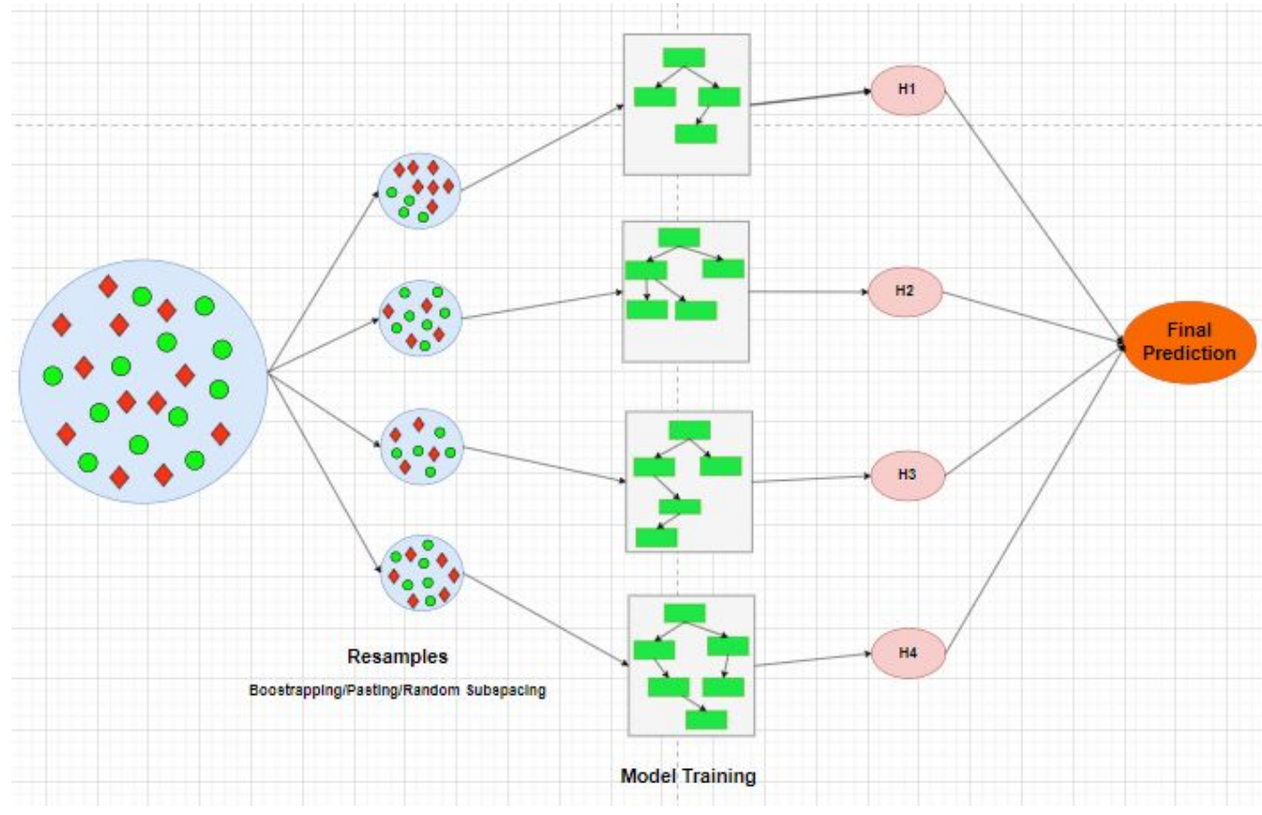
Consider a dataset D that contains samples from k classes. The probability of samples belonging to class i at a given node can be denoted as p_i. Then the Gini Impurity of D is defined as:

$$Gini(D) = 1 - \sum_{i=1}^{k} p_i^2$$

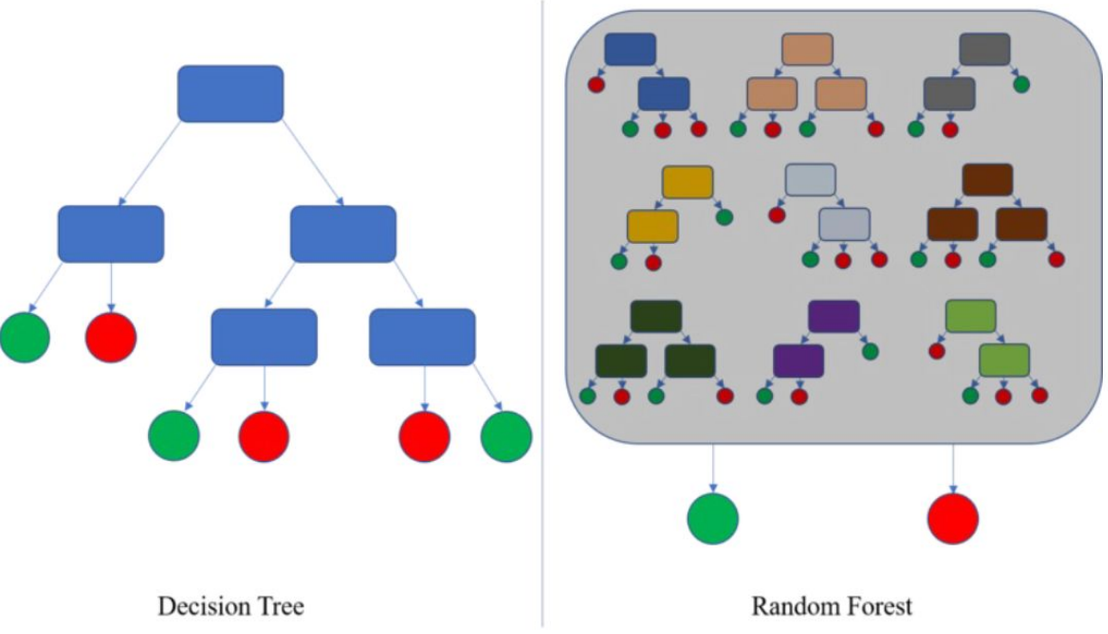
	Count		Probability		Gini Impurity
	n_1	n_2	p_1	p_2	$1 - p_1^2 - p_2^2$
Node A	0	10	0	1	$1 - 0^2 - 1^2 = 0$
Node B	3	7	0.3	0.7	$1 - 0.3^2 - 0.7^2 = 0.42$
Node C	5	5	0.5	0.5	$1 - 0.5^2 - 0.5^2 = 0.5$

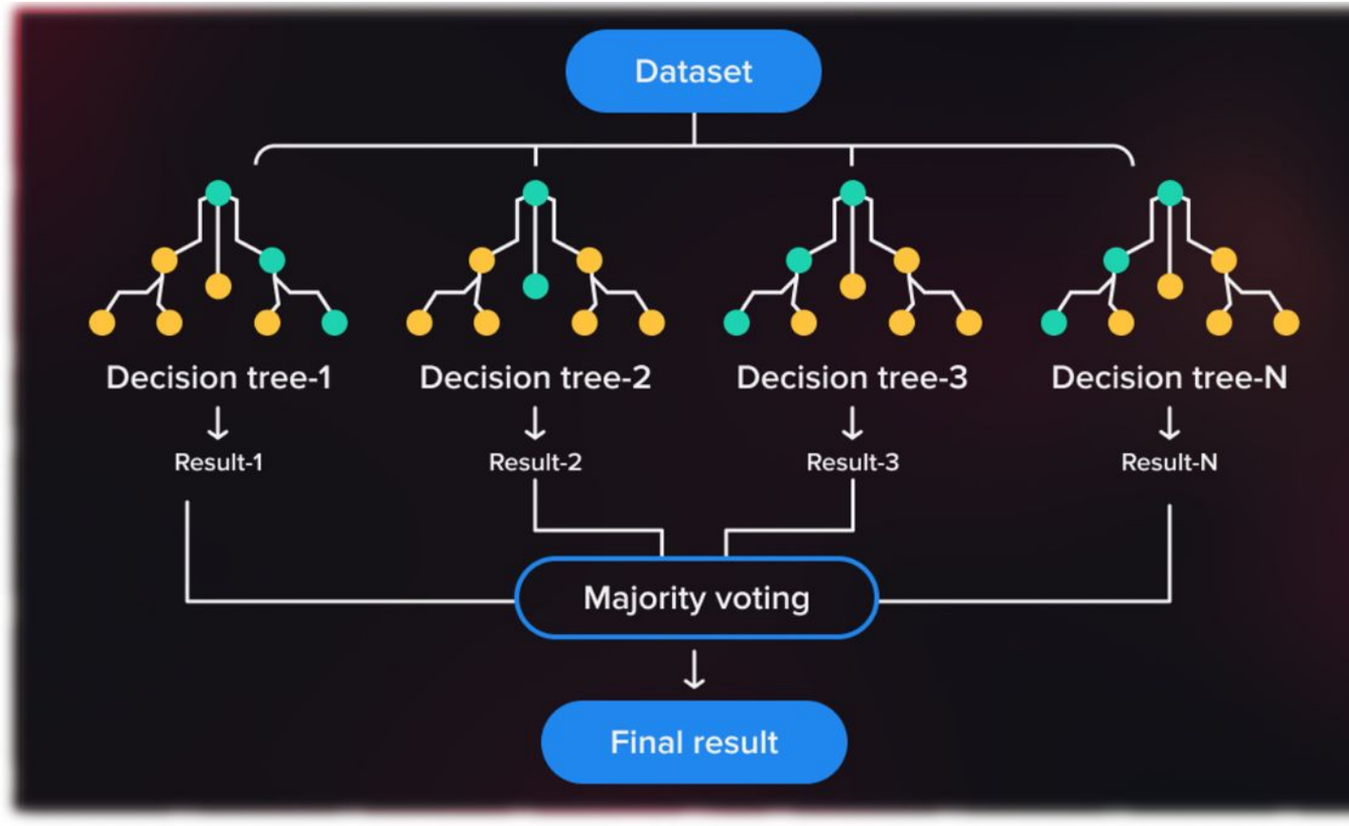
Review Bagging



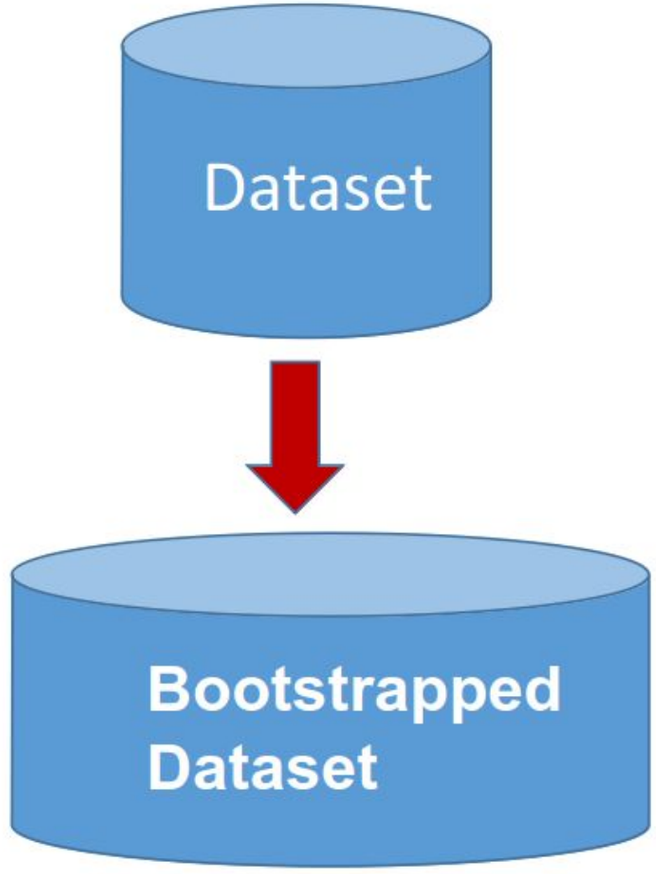


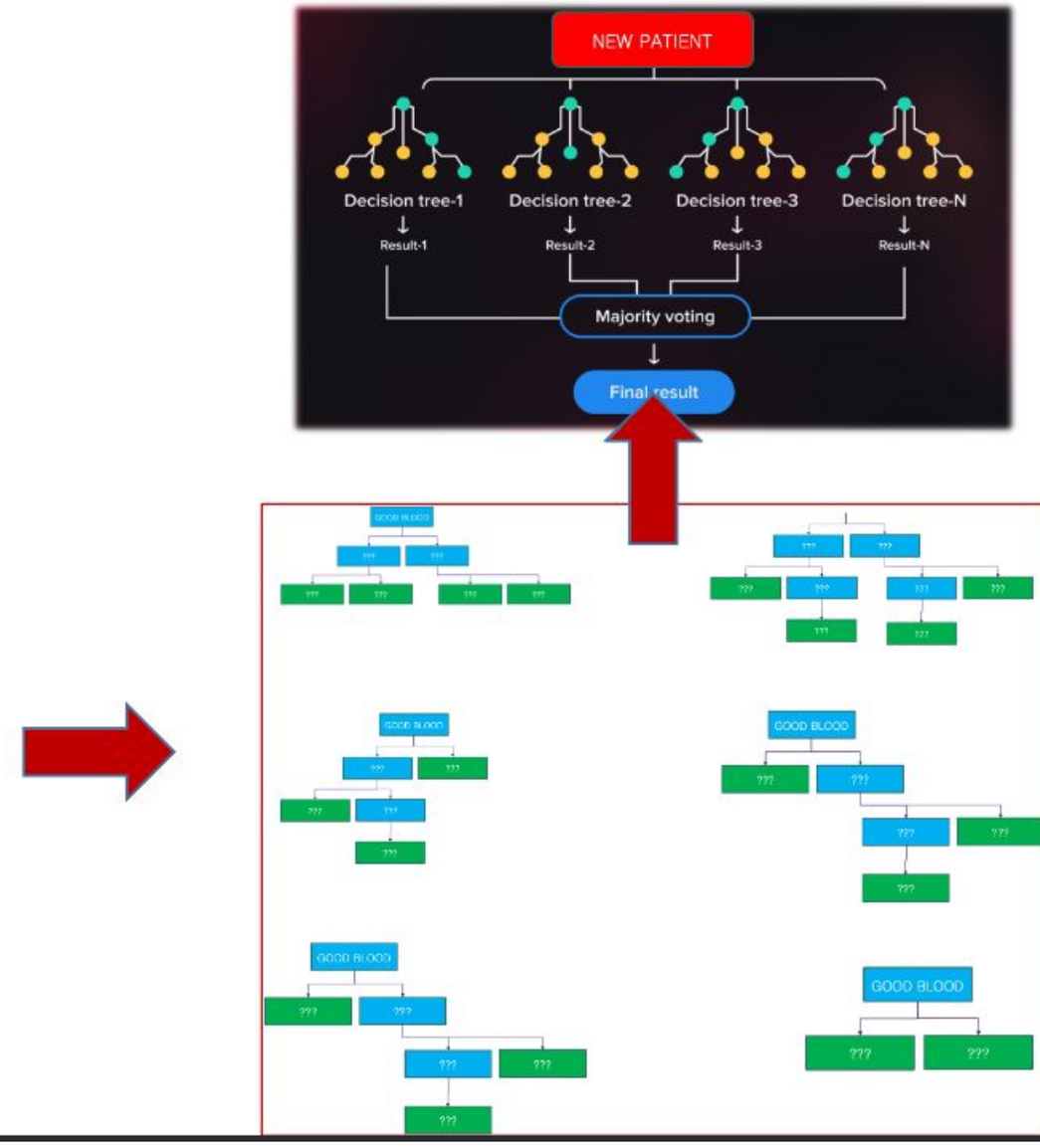
Review Random Forest



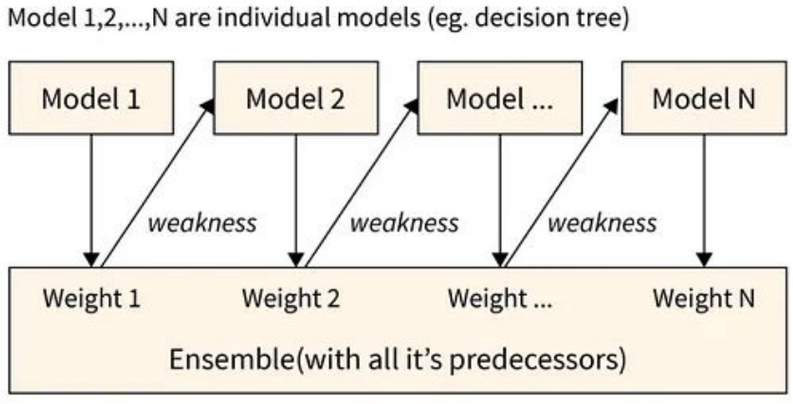


Review Random Forest

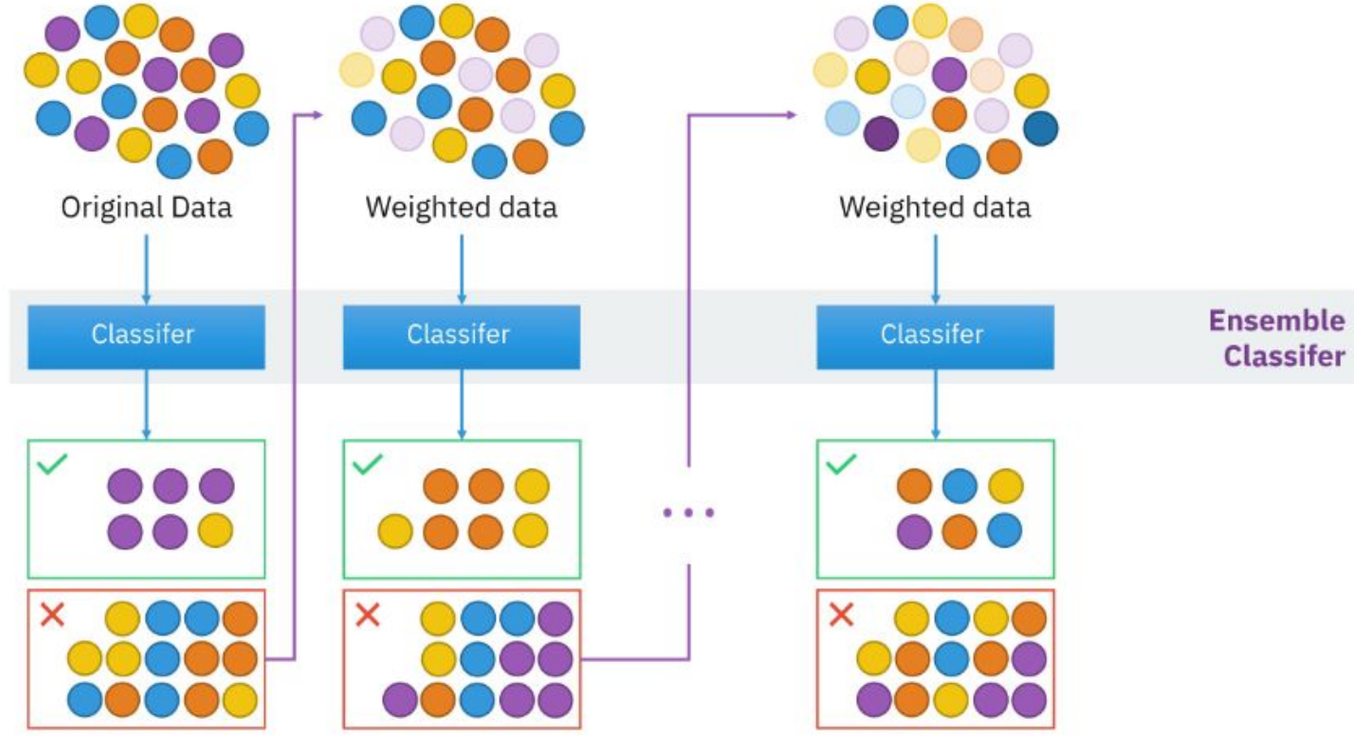




w Boosting



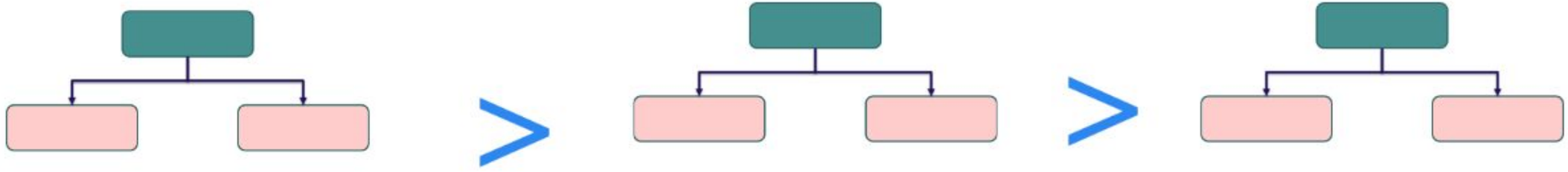
Boosting is an ensemble modelling, technique that attempts to build a strong classifier from the number of weak classifiers

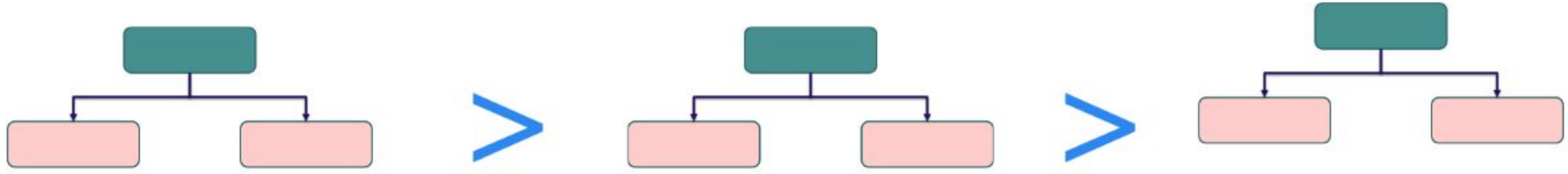


https://en.wikipedia.org/wiki/Boosting_%28machine_learning%29#/media/File:Ensemble_Boosting.svg

Review Boosting

- Stump are not equally weighted in the final decision.
- Stump that create more error will have less contribution in the final decision





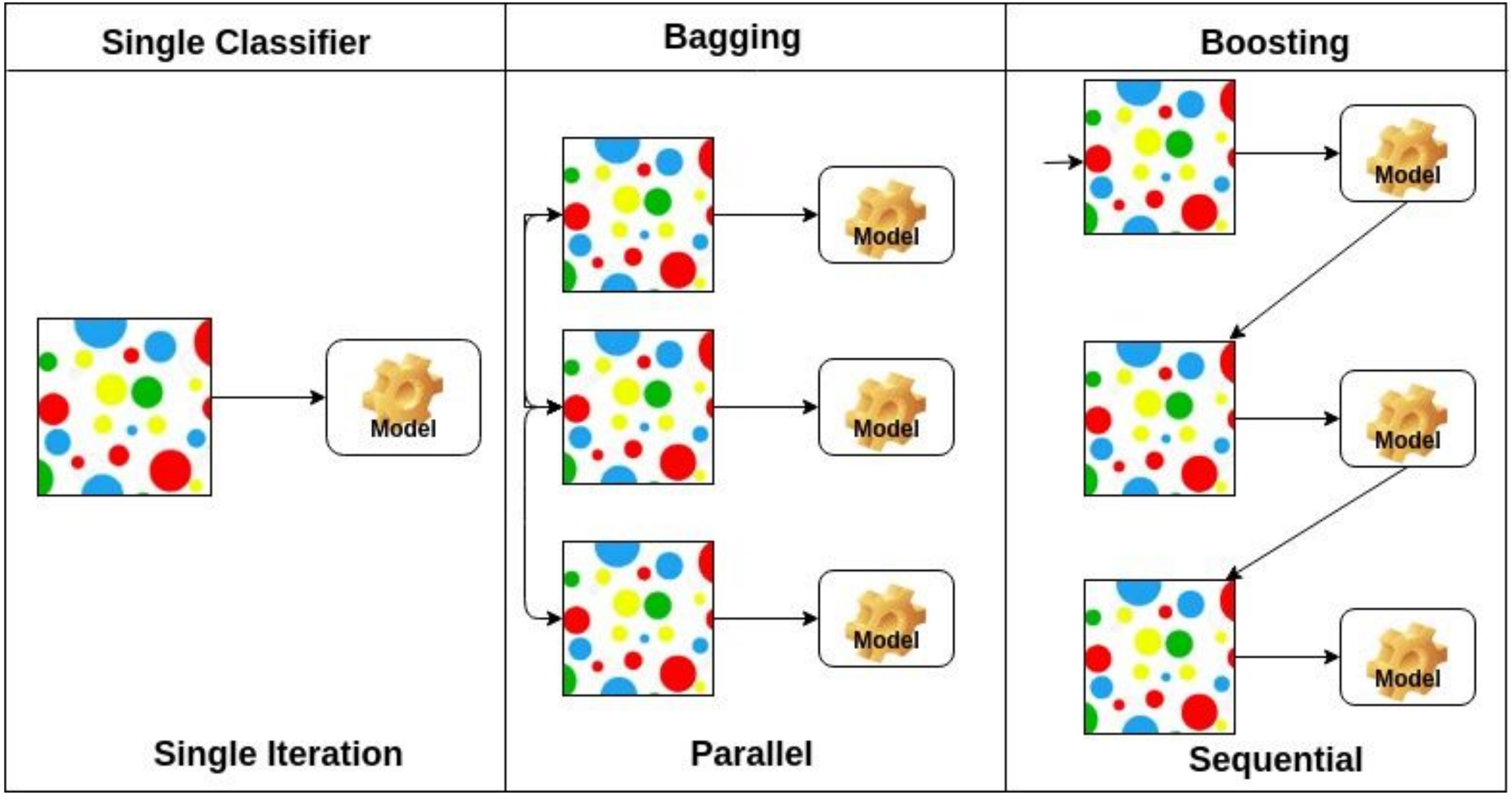
Review Boosting

$$H(x) = \sum_{t=1}^{T} \alpha_t h_t(x)$$

$$\alpha_t = \frac{1}{2} \log\left(\frac{1-\epsilon_t}{\epsilon_t}\right)$$

$$D_i^{t+1} = D_i^t \cdot \begin{cases} e^{\alpha_t}, \text{ if misclassified,} \\ e^{-\alpha_t}, \text{ if correctly classified.} \end{cases}$$

- T: total iterations
- h_t: weak learner
- α_t: weight

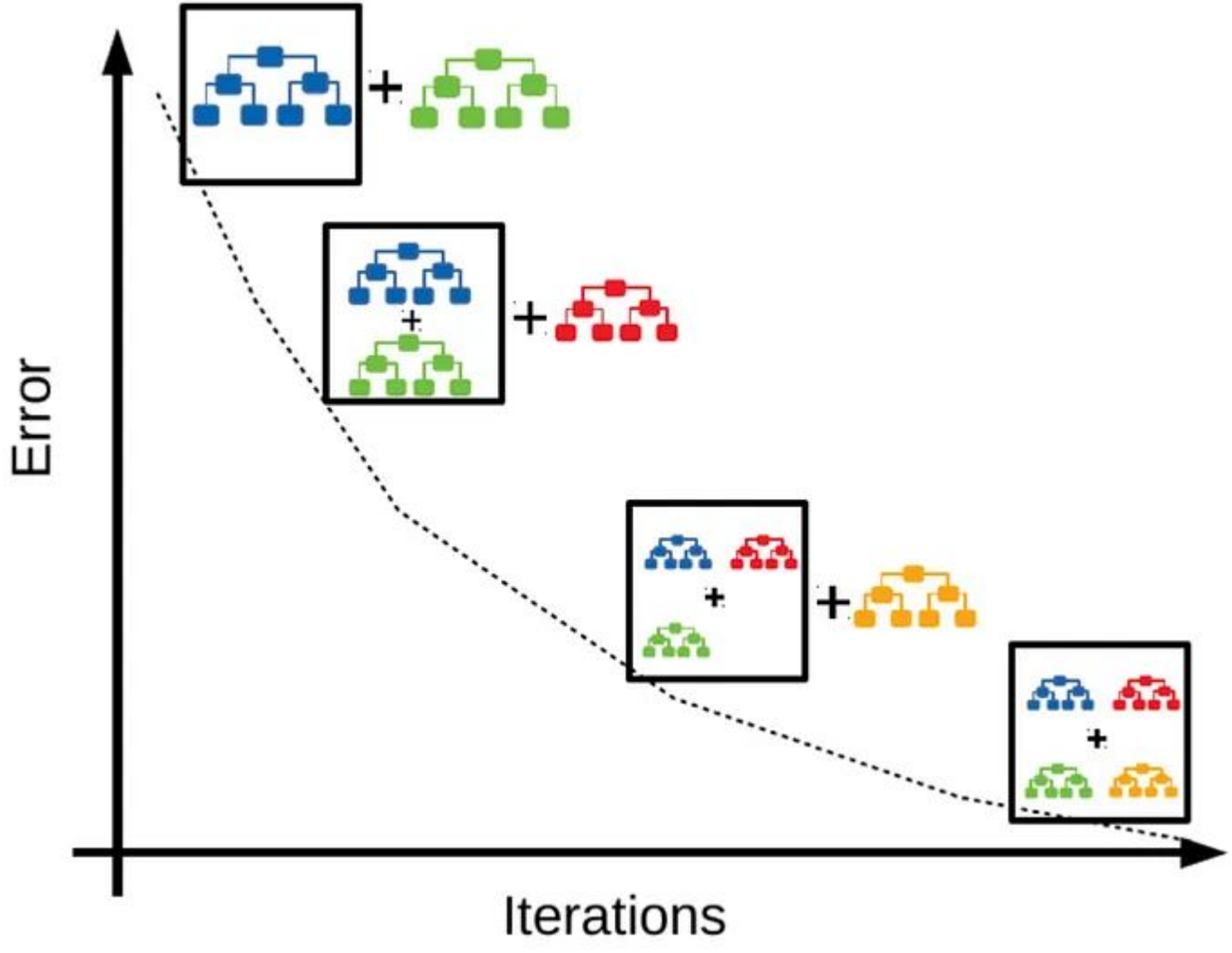

Single Classifier
Bagging
Boosting
Model
Model
Model
Model
Model
Model
Model
Model
Single Iteration
Parallel
Sequential

Quiz

Đặc điểm chính của boosting là gì?

A) Train các model theo thứ tự để sửa chữa các lỗi của các model trước đó.	B) Train các model song song để giảm thời gian tính toán.
C) Tập trung vào việc giảm bias của model bằng cách kết hợp các model yếu hơn.	D) Sử dụng cùng một tập dữ liệu mà không thay đổi cho mỗi model.

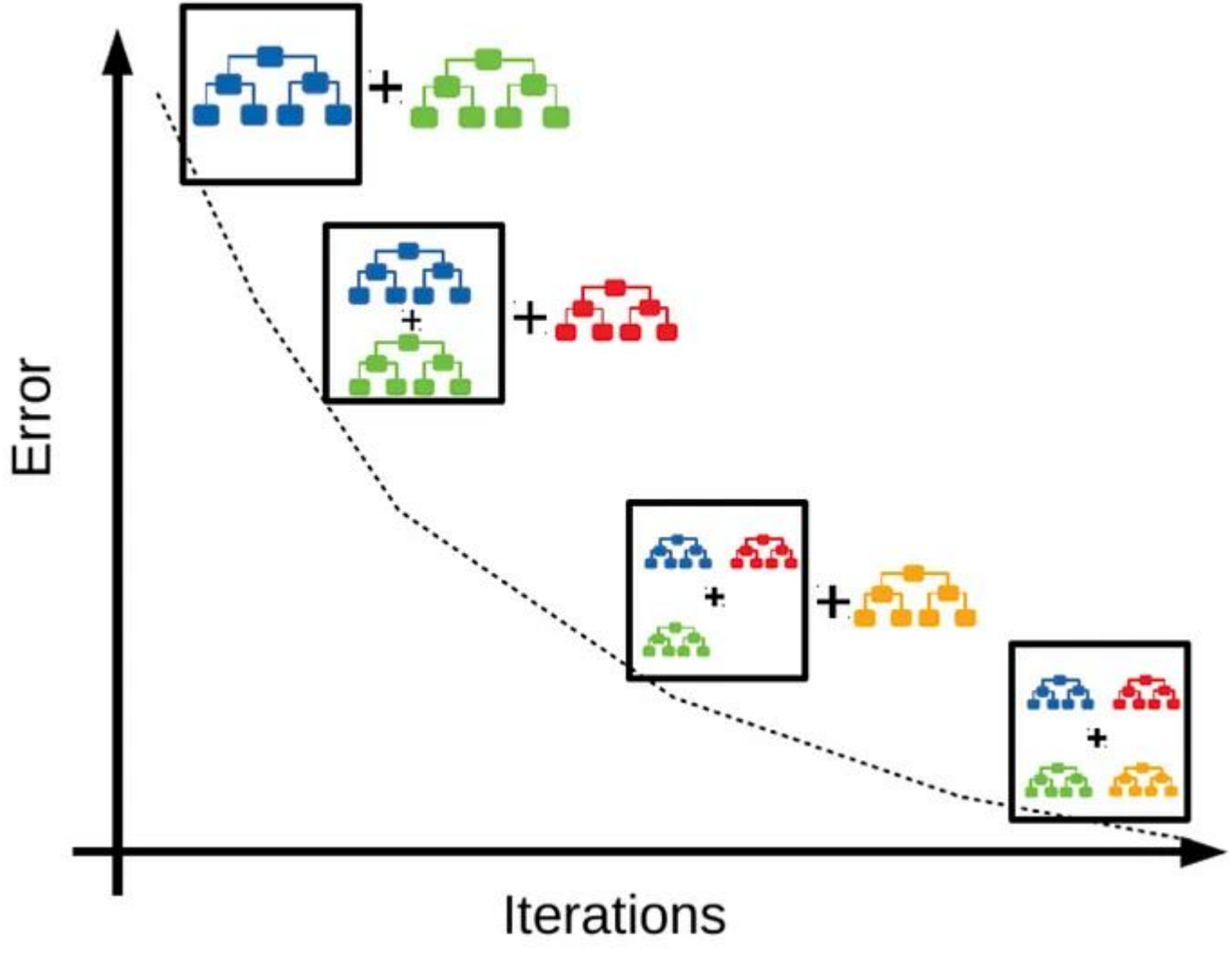
Review Gradient Boosting



Advantages	Disadvantages
High Predictive Accuracy	Computational Intensity
Feature Importance	Hyperparameter Tuning
Ensemble Learning	Black-Box Model

Gradient descent + boosting

Review Gradient Boosting

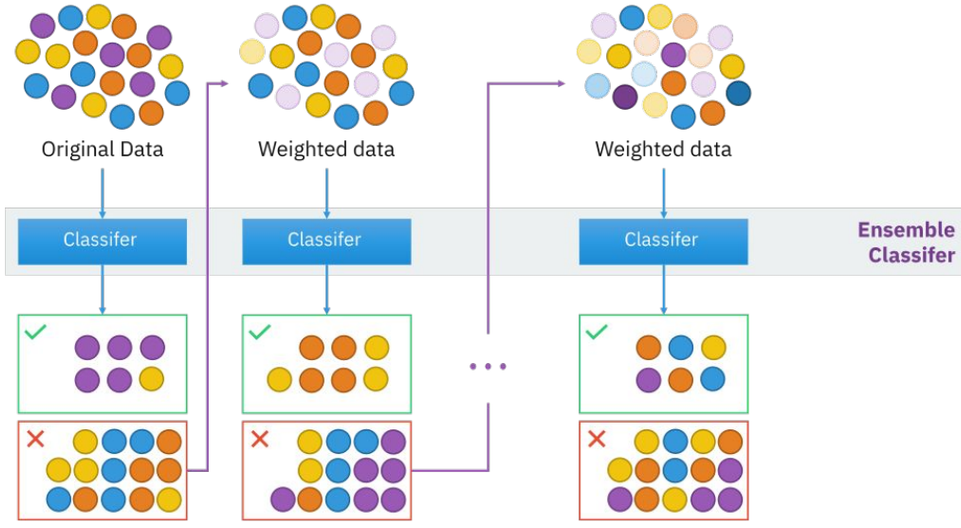


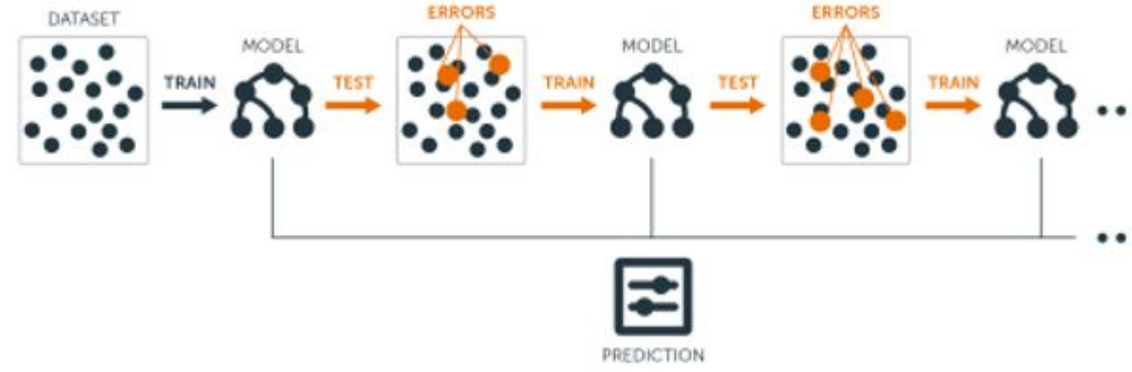
Advantages	Disadvantages
High Predictive Accuracy	Computational Intensity
Feature Importance	Hyperparameter Tuning
Ensemble Learning	Black-Box Model

$$\text{residual}(\text{true, predicted}) = \text{residual}(y, h(\boldsymbol{x})) = y - h(\boldsymbol{x})$$

$$f_{\text{loss}}(y, \boldsymbol{x}) = \tfrac{1}{2}(y - h(\boldsymbol{x}))^2.$$

$$\text{gradient}(\text{true, predicted}) = \frac{\partial f_{\text{loss}}}{\partial h}(y, h(\boldsymbol{x})) = -(y - h(\boldsymbol{x}))$$





Adaboost	Gradient Boost
An additive model where shortcomings of previous models are identified by high-weight data points.	An additive model where shortcomings of previous models are identified by the gradient.
The trees are usually grown as decision stumps.	The trees are grown to a greater depth usually ranging from 8 to 32 terminal nodes.
Each classifier has different weights assigned to the final prediction based on its performance.	All classifiers are weighed equally and their predictive capacity is restricted with learning rate to increase accuracy.
It gives weights to both classifiers and observations thus capturing maximum variance within data.	It builds trees on previous classifier’s residuals thus capturing variance in data.

Review Gradient Boosting

Compare GD vs GB (loss)

$$\text{residual}(\text{true, predicted}) = \text{residual}(y, h(\boldsymbol{x})) = y - h(\boldsymbol{x})$$

$$f_{\text{loss}}(y, \boldsymbol{x}) = \tfrac{1}{2}(y - h(\boldsymbol{x}))^2.$$

$$\text{gradient}(\text{true, predicted}) = \frac{\partial f_{\text{loss}}}{\partial h}(y, h(\boldsymbol{x})) = -(y - h(\boldsymbol{x}))$$

Review Gradient Boosting

Compare GD vs GB (training)

Review XGBoost Basics

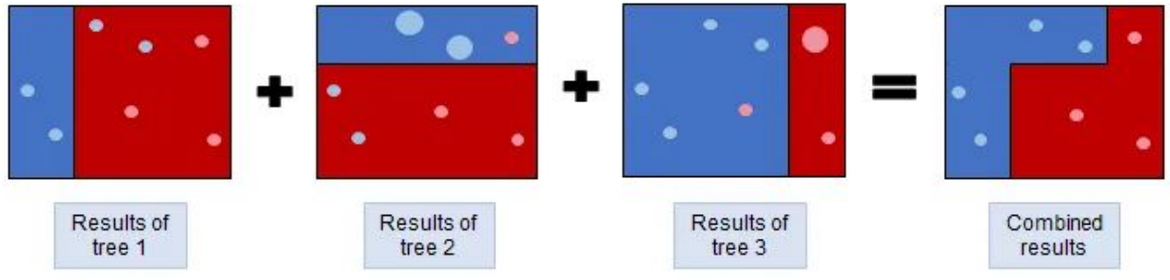
XGBoost:

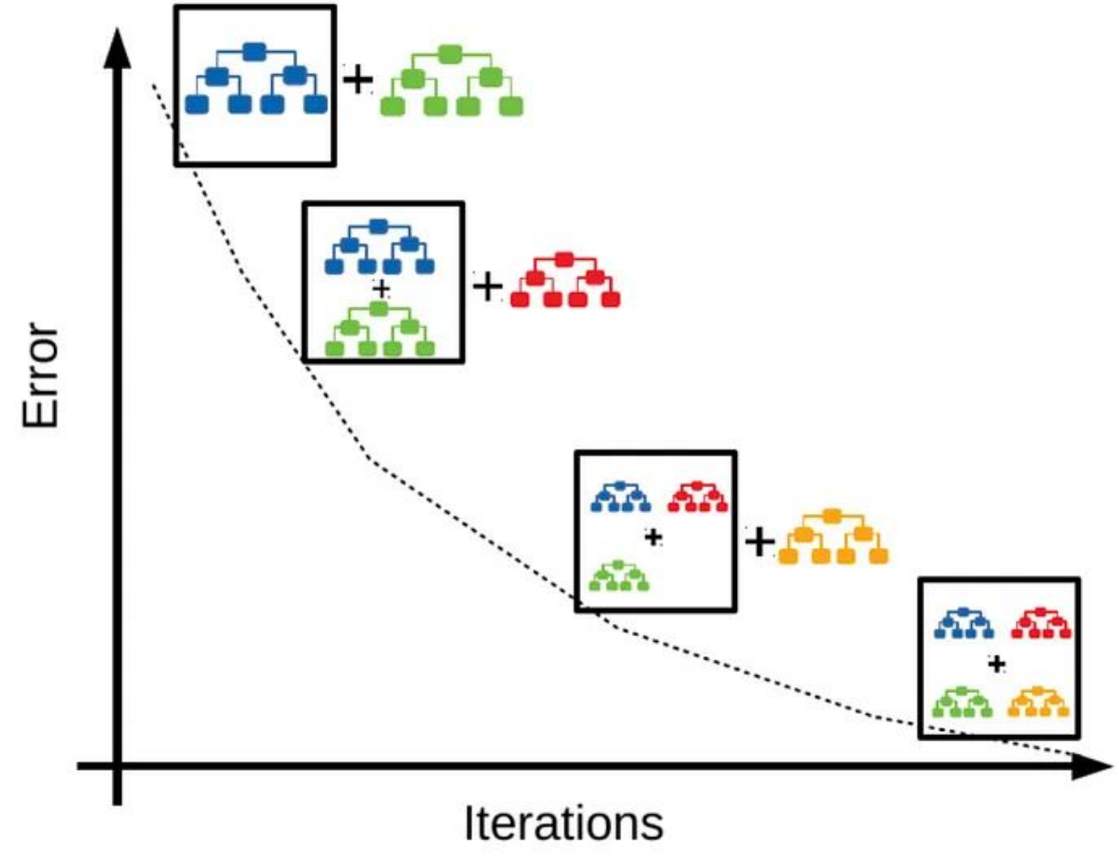
Taking the best parts of AdaBoost and random forests and adding additional features:

- **Sequential tree growing**
- Minimizing loss function **using gradient descent**
- **Parallel processing** to increase speed
- **Regularization** parameter

AdaBoost:

- Combining **weak learners** (decision trees)
- Assigning **weights to incorrect values**
- **Sequential tree growing** considering past mistakes





Review XGBoost Basics

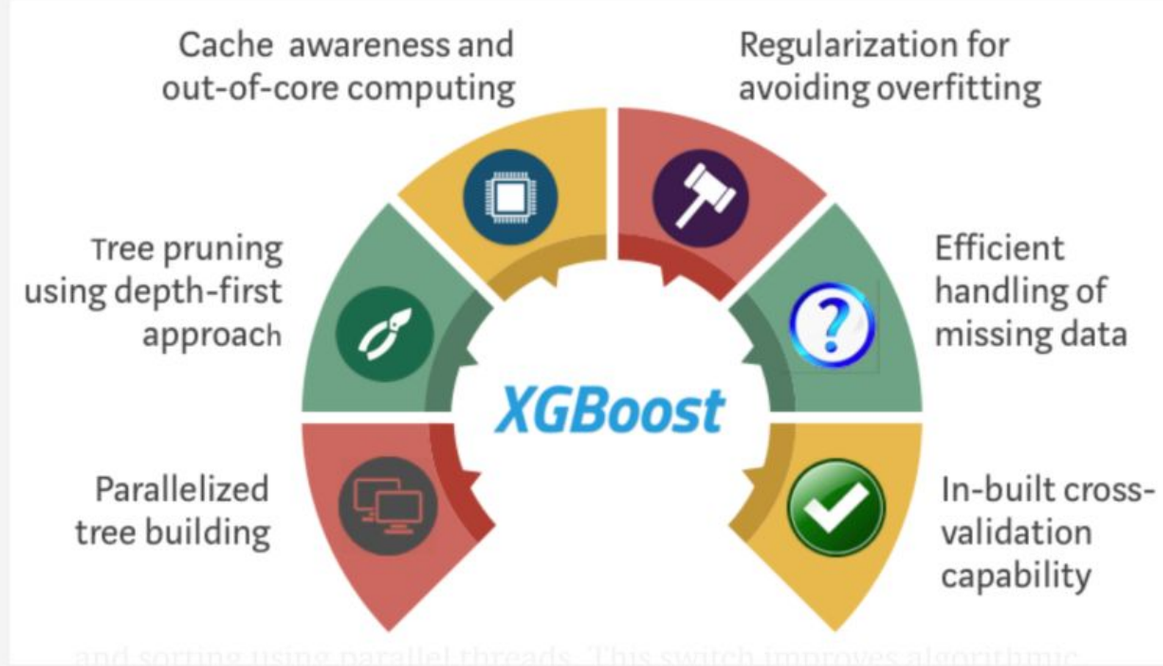
Why XGBoost algorithm is so good

Parallelization
- Utilize parallel processing, use multiple CPUs to execute the model
- A series of tree cannot build in parallization but build each tree can be parallelized

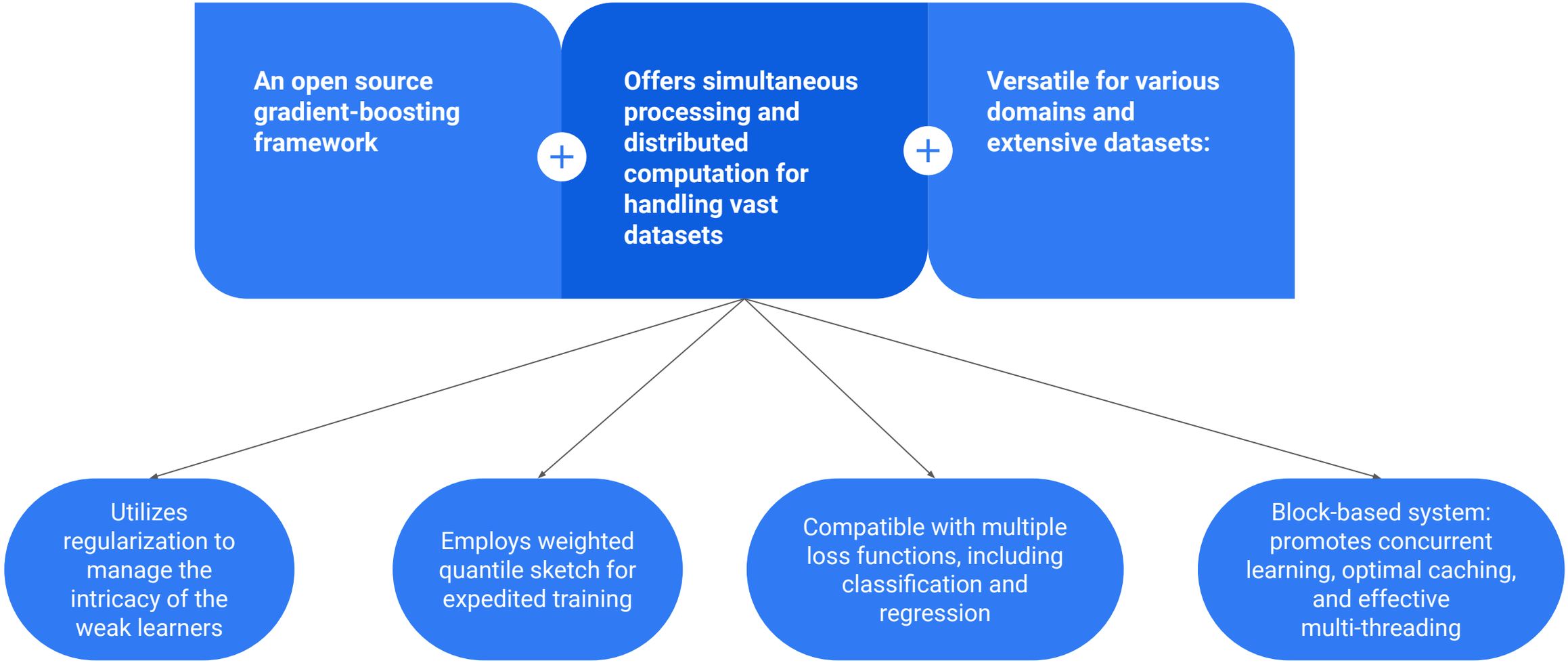
Regularization
- L1: lasso regularization, select features (force weights toward 0)
- L2: ridge regularization, smooth features (force small weights over param.)
- To avoid too complicated tree

Efficient tree pruning
- max depth to control tree grow
- prune the tree backwards and remove splits beyond which there is no positive gain.

.....



Review XGBoost Basics



Review XGBoost Basics

$$\alpha * R(x)^2 + \gamma * T + L(f(x), data)$$

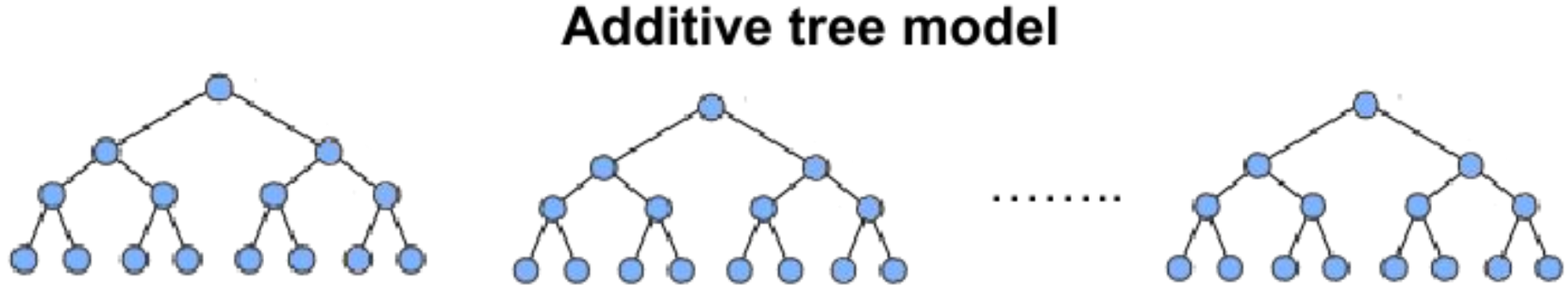
- α : Control degree of regularization
- $R(x)$: Regularization function
- γ : Control degree of regularization based on number of leaf nodes
- T: number of leaf nodes

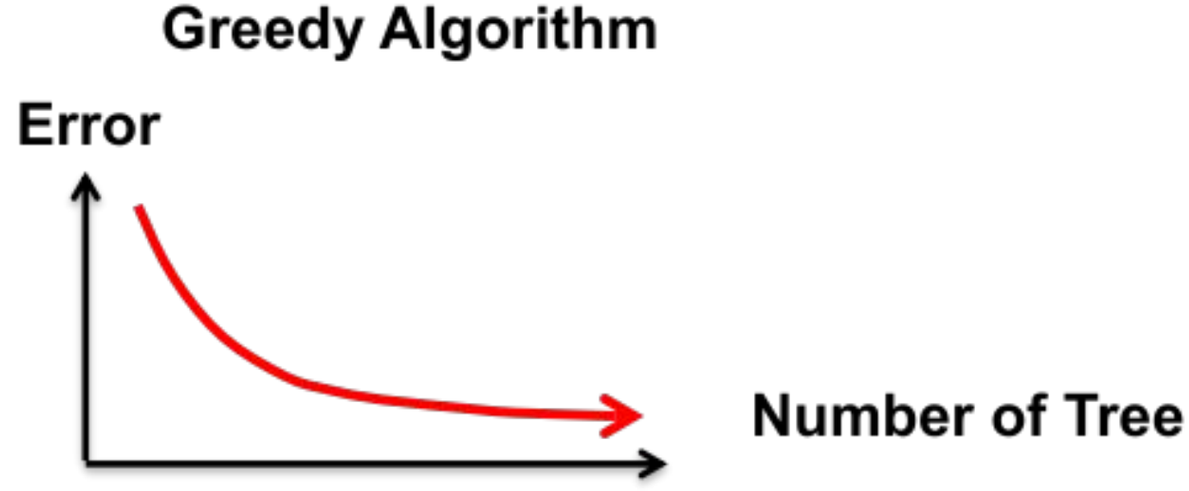
Review XGBoost Basics

Some of the important parameters that are used in XGBoost are as follows:

- `n_estimators/ntrees`: This specifies the number of trees to build. The default value is 50.
- `max_depth`: This specifies the maximum tree depth. The default value is 6. Higher values will make the model more complex and may lead to overfitting. Setting this value to 0 specifies no limit.
- `min_rows`: This specifies the minimum number of observations for a leaf. The default value is 1.
- `learn_rate`: This specifies the learning rate by which to shrink the feature weights. Shrinking feature weights after each boosting step makes the boosting process more conservative and prevents overfitting. The range is 0.0 to 1.0. The default value is 0.3.
- `sample_rate`: This specifies the row sampling ratio of the training instance (the *x axis*). For example, setting this value to 0.5 tells XGBoost to randomly collect half of the data instances to grow trees. The default value is 1 and the range is 0.0 to 1.0. Higher values may improve training accuracy.
- `col_sample_rate`: This specifies the column sampling rate (the *y axis*) for each split in each level. The default value is 1.0 and the range is from 0 to 1.0. Higher values may improve training accuracy.

Review XGBoost Basics





Quiz

Đâu là đặc điểm của XGBoost giúp giảm overfitting hiệu quả?

A) Tự động điều chỉnh tham số	B) Kỹ thuật cắt tỉa cây sau khi xây dựng cây
C) Sử dụng lượng dữ liệu nhỏ hơn trong quá trình đào tạo	D) Áp dụng mô hình tuyến tính

Regression

Step1: Khởi tạo giá trị f0 dự đoán của model bằng cách lấy trung bình của Y

Step2: Tính toán Similarity Score của root

- Similarity Score $= \frac{(Sum of Residuals)^2}{Number of Residuals + \lambda}$
- $Sum of Residuals$ là tổng của các giá trị trong Y - f0
- $Number of Residuals$ là số lượng sample

Step3: Có nhiều cách chọn điều kiện root, cơ bản nhất là lấy trung bình của 2 sample liên tiếp nhau. Sau đó tính Similarity Score cho các node trong nhánh trái và nhánh phải

Step4: Tính Gain cho từng điều kiện của root đã chọn ở trên và chọn ra Gain có giá trị lớn nhất

- Gain = Left Similarity Score + Right Similarity Score - Root Similarity Score

Step5 Tuỳ vào điều kiện độ sâu của tree mà ta sẽ thực hiện chia nhánh bằng cách lặp lại Step2 đến Step4. Sau đó ta đi tìm output cho root theo điều kiện có gain lớn nhất:

- Output $= \frac{Sum of Residuals}{Number of Residuals}$
- $Sum of Residuals$ là tổng của các giá trị trong Y - f0
- $Number of Residuals$ là số lượng sample

Step6: Dùng công thức sau $f0 + lr * Output$ để dự đoán kết quả cho toàn bộ training sample (thay thế cho f0) và tiếp tục thực hiện step 2 đến step 5 cho đến khi thoả mãn điều kiện dừng

Classification

Step1: Khởi tạo giá trị f0 dự đoán của model thông thường là 0.5 (xác suất của dự đoán)

Step2: Tính toán Similarity Score của root

- Similarity Score = $\frac{(SumofResiduals)^2}{\sum[PreviousProbability * (1 - PreviousProbability)] + \lambda}$
- $SumofResiduals$ là tổng của các giá trị trong Y (ở dạng xác suất) - f0
- $PreviousProbability$ là xác suất của model trước dự đoán ví dụ ở đây là 0.5

Step3: Có nhiều cách chọn điều kiện root, cơ bản nhất là lấy trung bình của 2 sample liền kề. Sau đó tính Similarity Score cho các node trong nhánh trái và nhánh phải

Step4: Tính Gain cho từng điều kiện của root đã chọn ở trên và chọn ra Gain có giá trị lớn nhất

- Gain = Left Similarity Score + Right Similarity Score - Root Similarity Score

Step5 Tuỳ vào điều kiện độ sâu của tree mà ta sẽ thực hiện chia nhánh bằng cách lặp lại Step2 đến Step4. Sau đó ta đi tìm output cho root theo điều kiện có gain lớn nhấ

- Output = $\frac{SumofResiduals}{\sum[PreviousProbability * (1 - PreviousProbability)]}$
- $SumofResiduals$ là tổng của các giá trị trong Y (ở dạng xác suất) - f0
- $PreviousProbability$ là xác suất của model trước dự đoán ví dụ ở đây là 0.5

Step6: Dùng công thức bên dưới để dự đoán kết quả (Probability) cho toàn bộ training sample (thay thế cho f0) và tiếp tục thực hiện step 2 đến step 5 cho đến khi thoả mãn điều kiện dừng

- Cần chọn nhánh phù hợp theo giá trị của X.
- Khi chọn được nhánh phù hợp ta tính Log prediction theo công thức
  - $LogPrediction = log(\frac{PreviousProbability}{1 - PreviousProbability}) + lr * Output$
  - log là Natural logarithm
- Xác suất dự đoán của các sample được tính theo công thức
  - $Probability = \frac{e^{log(LogPrediction)}}{1 + e^{log(LogPrediction)}}$

Problem 1-2

Problem 1: Regression

X	Y
23	50
24	70
26	80
27	85

Hình 1: X là input, Y là target

Build tree với điều kiện sau $\lambda = 0$, depth $= 1$, lr $= 0.3$

Step1: Khởi tạo giá trị f0 dự đoán của model bằng cách lấy trung bình của Y

Step2: Tính toán Similarity Score của root

- Similarity Score $= \frac{(Sum of Residuals)^2}{Number of Residuals + \lambda}$
- $Sum of Residuals$ là tổng của các giá trị trong Y - f0
- $Number of Residuals$ là số lượng sample

Step3: Chọn root theo từng yêu cầu sau và tính Similarity Score cho các node trong nhánh trái và nhánh phải

i. X < 23.5
ii. X < 25
iii. X < 26.5

Step4: Tính Gain cho từng case i, ii, iii ở trên và chọn ra Gain có giá trị lớn nhất

- Gain = Left Similarity Score + Right Similarity Score - Root Similarity Score

Step5 Sua khi chọn được model cho gain có giá trị lớn nhất ta tính giá Ouput cho từng node trong nhánh trái và phải theo công thức sau:

- Output $= \frac{Sum of Residuals}{Number of Residuals}$
- $Sum of Residuals$ là tổng của các giá trị trong Y - f0
- $Number of Residuals$ là số lượng sample

Step6: Bây giờ ta sẽ thực hiện dụ đoán kết quả khi x $= 25$ theo công thức bên dưới

- Cần chọn nhánh phù hợp theo giá trị của X.
- Khi chọn được nhánh phù hợp ta lấy giá trị Ouput của nhánh và thực hiện dự đoán theo công thức $f0 + lr * Output$

Regularization

Step0: lambda=0, depth=1, lr=0.3

Step1: Initialize model with a constant value

- f0 = mean(Y) = (50+70+80+85)/4 = 71.25

Step2: Compute Similarity Score

$$\text{Similarity Score} = \frac{\text{Sum of Residuals, Squared}}{\text{Number of Residuals} + \lambda}$$

SC = (-21.25 + -1.25 + 8.75 + 13.75)^2/(4) = 0

.

X	Y	y-f0
23	50	-21.25
24	70	-1.25
26	80	8.75
27	85	13.75

Regularization

Step0: lambda=0, depth=1, lr=0.3

Step1: Initialize model with a constant value

- f0 = mean(Y) = (50+70+80+85)/4 = 71.25

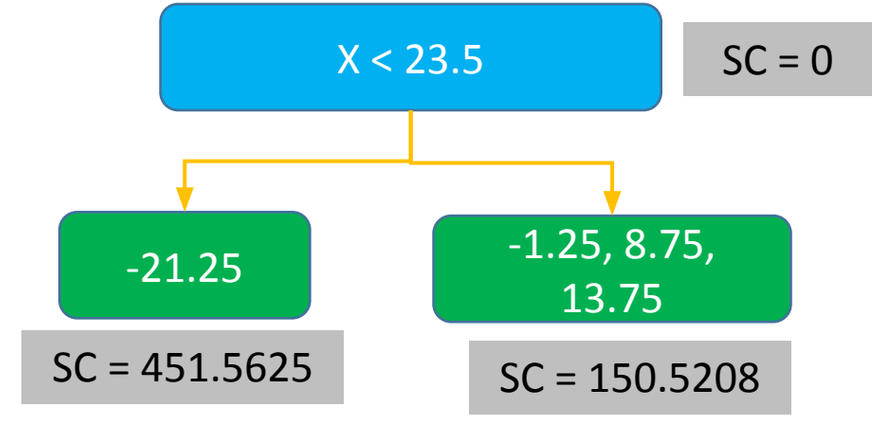
Step2: Compute Similarity Score

$$\text{Similarity Score} = \frac{\text{Sum of Residuals, Squared}}{\text{Number of Residuals} + \lambda}$$

SC = (-21.25 + -1.25 + 8.75 + 13.75)^2/(4) = 0

Step3: Take average of x0 and x1

- (23+24)/2 = 23.5



.

X	Y	y-f0
23	50	-21.25
24	70	-1.25
26	80	8.75
27	85	13.75

Regularization

Step0: lambda=0, depth=1, lr=0.3

Step1: Initialize model with a constant value

- f0 = mean(Y) = (50+70+80+85)/4 = 71.25

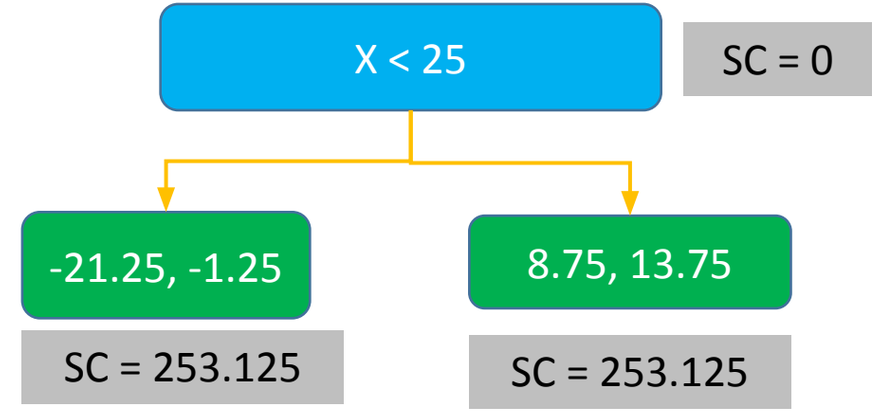
Step2: Compute Similarity Score

$$\text{Similarity Score} = \frac{\text{Sum of Residuals, Squared}}{\text{Number of Residuals} + \lambda}$$

SC = (-21.25 + -1.25 + 8.75 + 13.75)^2/(4) = 0

Step3: Take average of x0 and x1

- (24+26)/2 = 25



.

X	Y	y-f0
23	50	-21.25
24	70	-1.25
26	80	8.75
27	85	13.75

Regularization

Step0: lambda=0, depth=1

Step1: Initialize model with a constant value

- f0 = mean(Y) = (50+70+80+85)/4 = 71.25

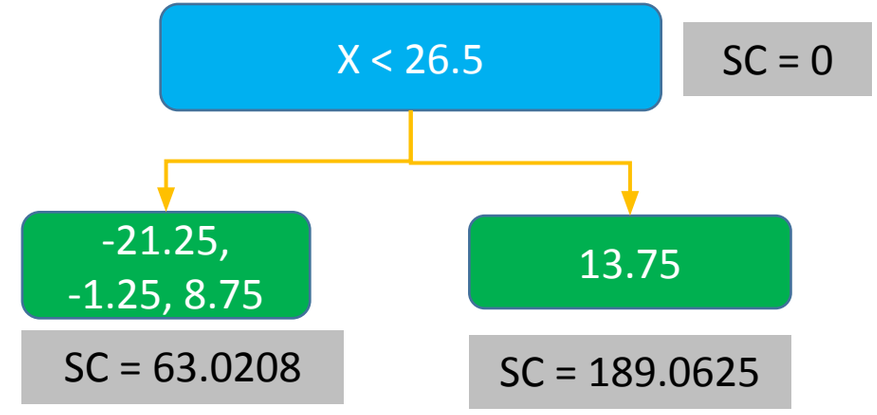
Step2: Compute Similarity Score

$$SC = \frac{\sum (output - predicted)^2}{m + \lambda}$$

SC = (-21.25 + -1.25 + 8.75 + 13.75)^2/(4) = 0

Step3: Take average of x0 and x1

- (26+27)/2 = 26.5



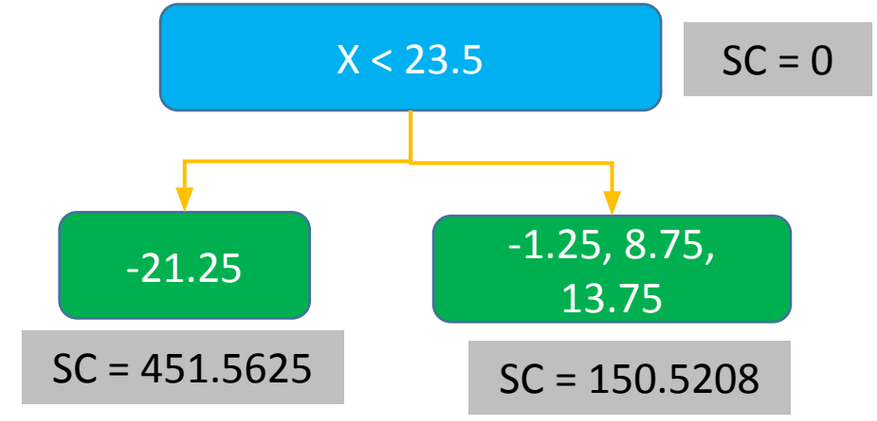
.

X	Y	y-f0
23	50	-21.25
24	70	-1.25
26	80	8.75
27	85	13.75

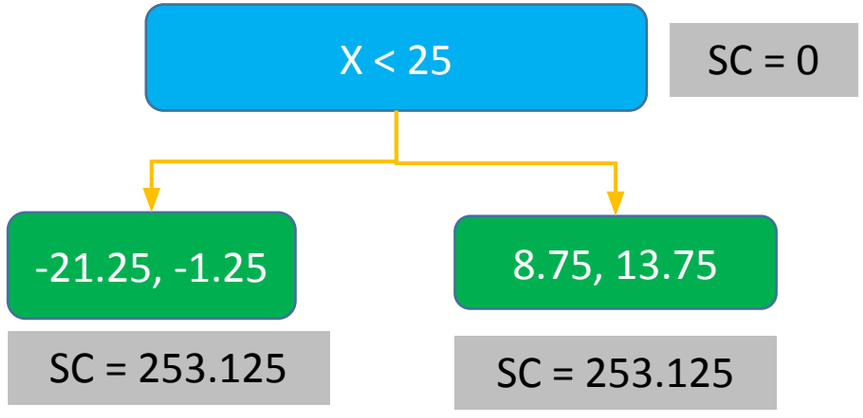
Regularization

Step4: Calculate the Gain.
Gain = Left SC + Right SC - Root SC

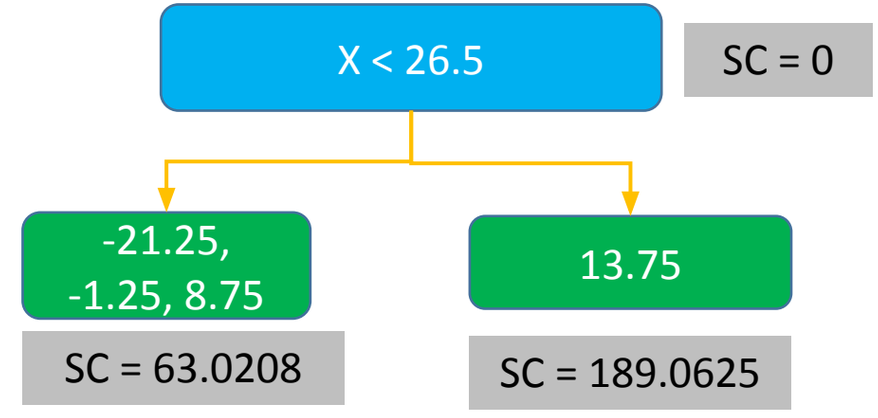
The Best Gain = 602.0833
Select X < 23.5



Gain = 451.5625 + 150.5208 - 0 = 602.0833



Gain = 253.125 + 253.125 - 0 = 506.25

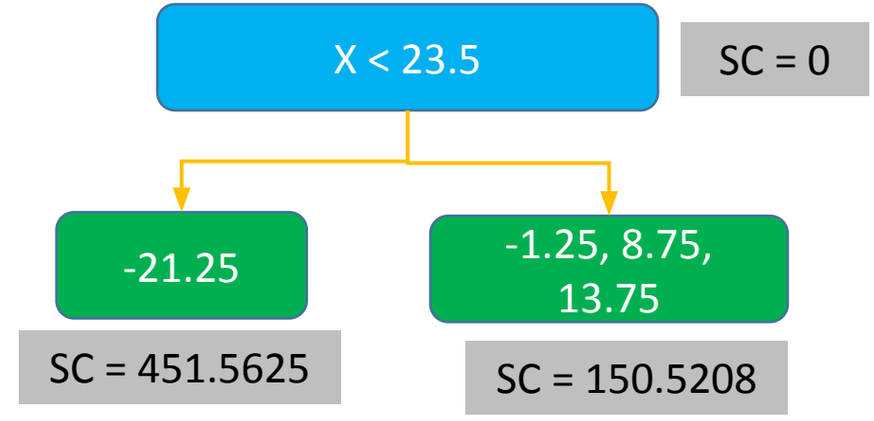


Gain = 63.0208 + 189.0625 - 0 = 252.0833

Regularization

Step 5: Calculate the Gain.
Gain = Left SC + Right SC - Root SC

The Best Gain = 602.0833
Select X < 23.5

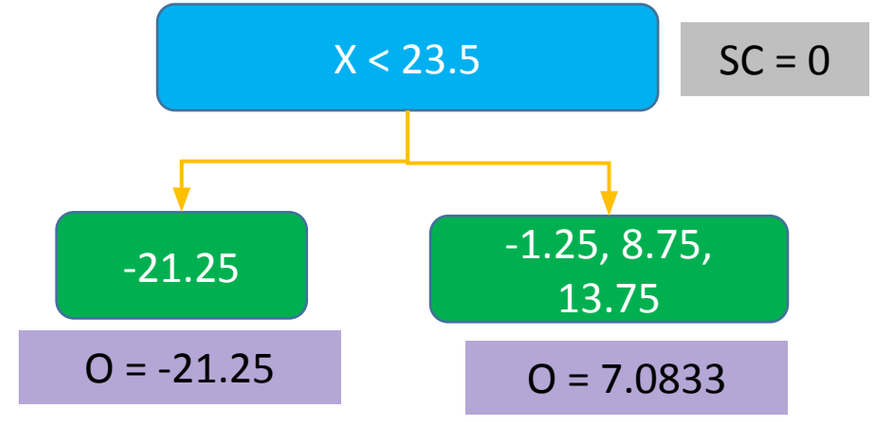


X	Y
23	50
24	70
25	?
26	80
27	85

Regularization

Step 5: Calculate the Gain.
Gain = Left SC + Right SC - Root SC

The Best Gain = 602.0833
Select X < 23.5



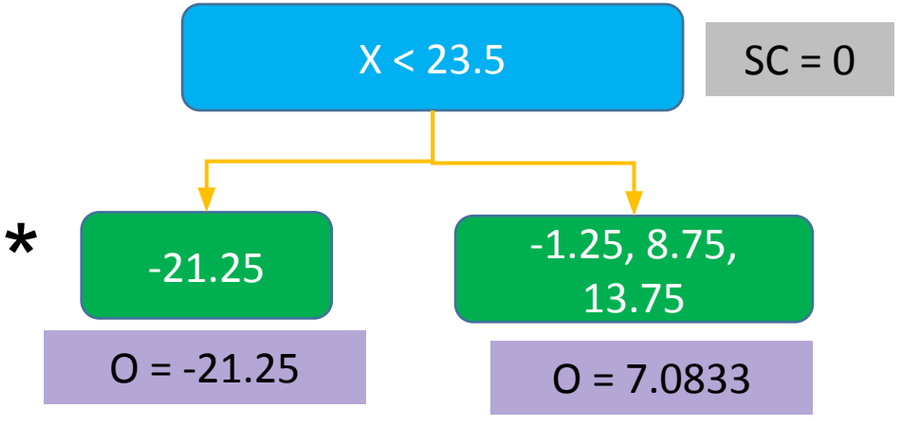
X	Y
23	50
24	70
25	?
26	80
27	85

Regularization

Step 5: Calculate the Gain.
Gain = Left SC + Right SC - Root SC

The Best Gain = 602.0833
Select X < 23.5

71.25 + LR *



f0 = mean(Y) = (50+70+80+85)/4 = 71.25
Lr = 0.3

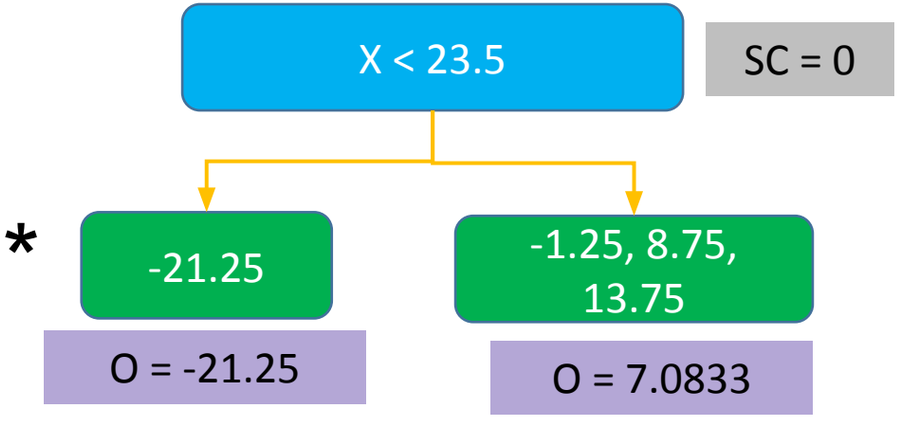
X	Y
23	50
24	70
25	73.375
26	80
27	85

Regularization

Step 5: Calculate the Gain.
Gain = Left SC + Right SC - Root SC

The Best Gain = 602.0833
Select X < 23.5

71.25 + LR *



f0 = mean(Y) = (50+70+80+85)/4 = 71.25
Lr = 0.3

X	Y	y-f0	yhat	y-yhat
23	50	-21.25	64.875	-14.875
24	70	-1.25	73.375	-3.375
26	80	8.75	73.375	6.625
27	85	13.75	73.375	11.625

Problem 1-2

Problem 2: Classification

X	Y
23	False
24	False
26	True
27	True

Hình 2: X là input, Y là target

Build tree với điều kiện sau $\lambda = 0$, depth $= 1$, lr $= 0.3$. Ta sẽ biến đổi giá trị của các sample trong Y thành xác suất, tương ứng False là 0 và True là 1

Step1: Khởi tạo giá trị f0 dự đoán của model là một hằng số 0.5 (xác suất của dự đoán)

Step2: Tính toán Similarity Score của root

- Similarity Score $= \frac{(SumofResiduals)^2}{\sum[PreviousProbability * (1 - PreviousProbability)] + \lambda}$
- $SumofResiduals$ là tổng của các giá trị trong Y (ở dạng xác suất) - f0
- $PreviousProbability$ là xác suất của model trước dự đoán ví dụ ở đây là 0.5

Step3: Chọn root theo từng yêu cầu sau và tính Similarity Score cho các node trong nhánh trái và nhánh phải

i. X < 23.5
ii. X < 25
iii. X < 26.5

Step4: Tính Gain cho từng case i, ii, iii ở trên và chọn ra Gain có giá trị lớn nhất

- Gain = Left Similarity Score + Right Similarity Score - Root Similarity Score

Step5 Sua khi chọn được model cho gain có giá trị lớn nhất ta tính giá Ouput cho từng node trong nhánh trái và phải theo công thức sau:

- Output $= \frac{SumofResiduals}{\sum[PreviousProbability * (1 - PreviousProbability)]}$
- $SumofResiduals$ là tổng của các giá trị trong Y (ở dạng xác suất) - f0
- $PreviousProbability$ là xác suất của model trước dự đoán ví dụ ở đây là 0.5

Step6: Bây giờ ta sẽ thực hiện dụ đoán kết quả khi x $= 25$ theo công thức bên dưới

- Cần chọn nhánh phù hợp theo giá trị của X.
- Khi chọn được nhánh phù hợp ta tính Log prediction theo công thức
  - $LogPrediction = log(\frac{PreviousProbability}{1 - PreviousProbability}) + lr * Output$
  - log là Natural logarithm
- Xác suất dự đoán của x $= 25$ được tính theo công thức
  - $Probability = \frac{e^{log(LogPrediction)}}{1 + e^{log(LogPrediction)}}$

Problem 1-2

Step0: lambda=0, depth=1, lr=0.3

Step1: Initialize model with a constant probability

- f0 = 0.5

Step2: Compute Similarity Score

$$\frac{(\sum \text{Residual}_i)^2}{\sum [\text{Previous Probability}_i \times (1 - \text{Previous Probability}_i)] + \lambda}$$

SC = (-0.5+-0.5+0.5+0.5)^2/(denominator) = 0

X	Y	Y-cat	y-f0
23	False	0	-0.5
24	False	0	-0.5
26	True	1	0.5
27	True	1	0.5

Problem 1-2

Step0: lambda=0, depth=1, lr=0.3

Step1: Initialize model with a constant probability

- f0 = 0.5

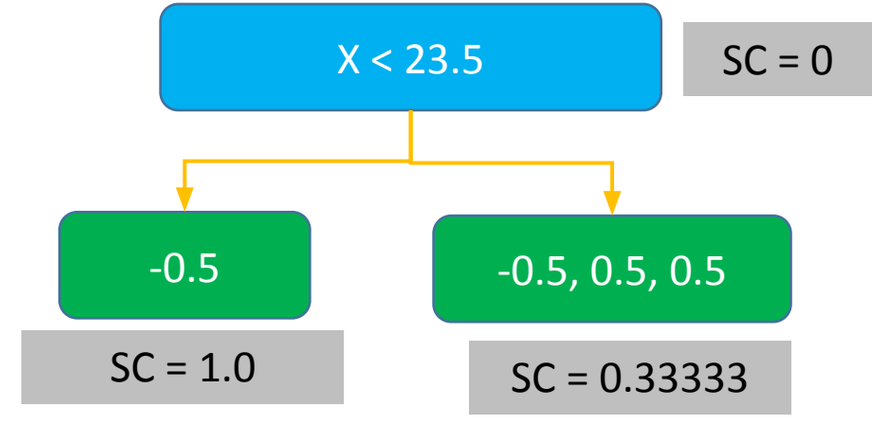
Step2: Compute Similarity Score

$$\frac{(\sum \text{Residual}_i)^2}{\sum [\text{Previous Probability}_i \times (1 - \text{Previous Probability}_i)] + \lambda}$$

SC = (-0.5+-0.5+0.5+0.5)^2/(denominator) = 0

Step3: Take average of x0 and x1

- (23+24)/2 = 23.5



X	Y	Y-cat	y-f0
23	False	0	-0.5
24	False	0	-0.5
26	True	1	0.5
27	True	1	0.5

Problem 1-2

Step0: lambda=0, depth=1, lr=0.3

Step1: Initialize model with a constant probability

- f0 = 0.5

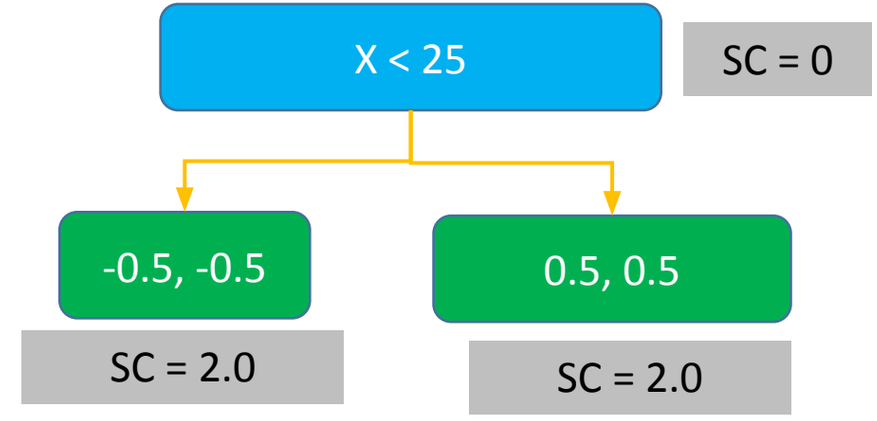
Step2: Compute Similarity Score

$$\frac{(\sum \text{Residual}_i)^2}{\sum [\text{Previous Probability}_i \times (1 - \text{Previous Probability}_i)] + \lambda}$$

SC = (-0.5+-0.5+0.5+0.5)^2/(denominator) = 0

Step3: Take average of x0 and x1

- (24+26)/2 = 25



X	Y	Y-cat	y-f0
23	False	0	-0.5
24	False	0	-0.5
26	True	1	0.5
27	True	1	0.5

Problem 1-2

Step0: lambda=0, depth=1, lr=0.3

Step1: Initialize model with a constant probability

- f0 = 0.5

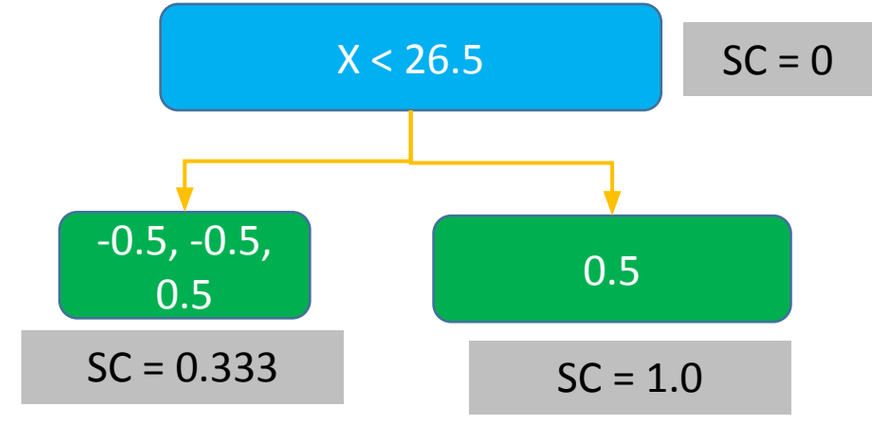
Step2: Compute Similarity Score

$$\frac{(\sum \text{Residual}_i)^2}{\sum [\text{Previous Probability}_i \times (1 - \text{Previous Probability}_i)] + \lambda}$$

SC = (-0.5+-0.5+0.5+0.5)^2/(denominator) = 0

Step3: Take average of x0 and x1

- (26+27)/2 = 26.5

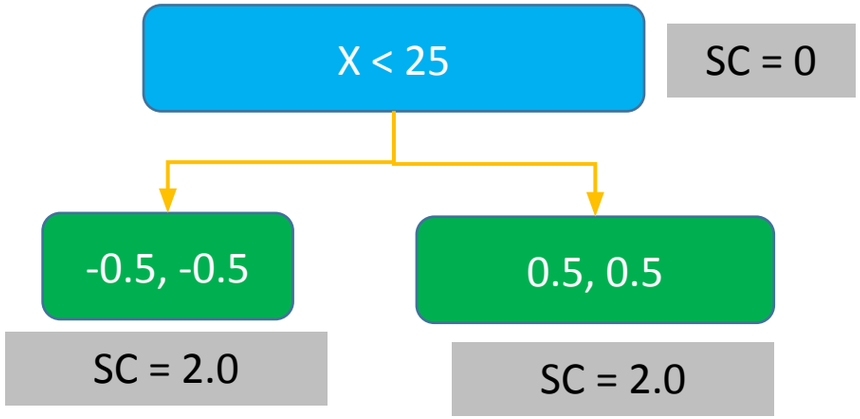


X	Y	Y-p	y-f0
23	False	0	-0.5
24	False	0	-0.5
26	True	1	0.5
27	True	1	0.5

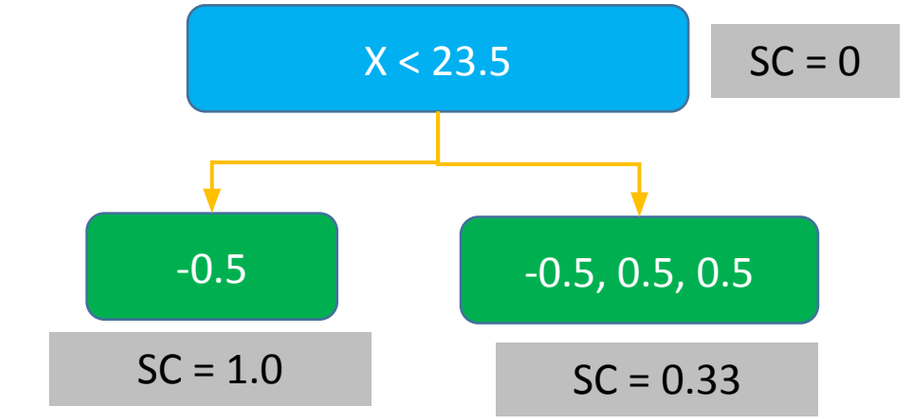
Problem 1-2

Step4: Calculate the Gain.
Gain = Left SC + Right SC - Root SC

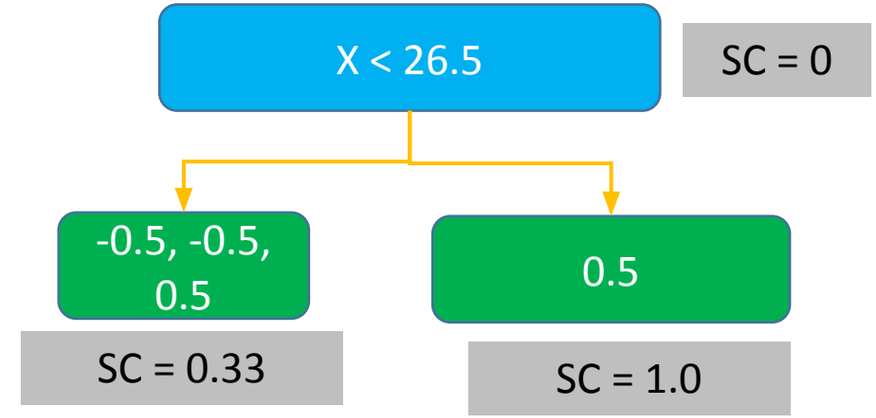
The Best Gain = 4
Select X < 25



Gain = 2.0 + 2.0 - 0 = 4



Gain = 1.0 + 0.33 - 0 = 1.33

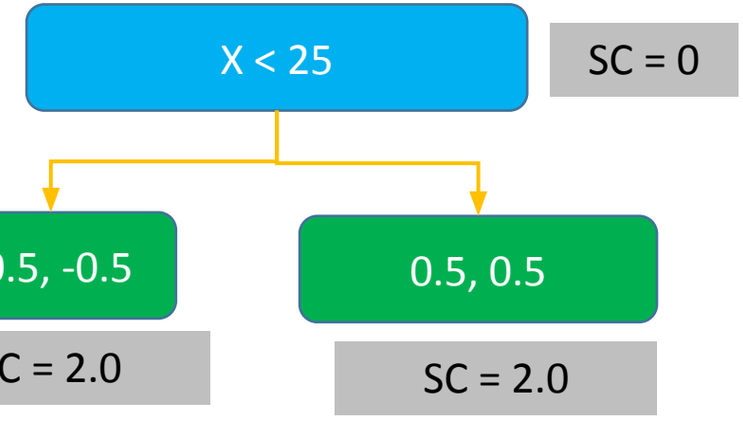


Gain = 0.33 + 1.0 - 0 = 1.33

Problem 1-2

Step 5: Calculate the Gain.
Gain = Left SC + Right SC - Root SC

The Best Gain = 4
Select X < 25



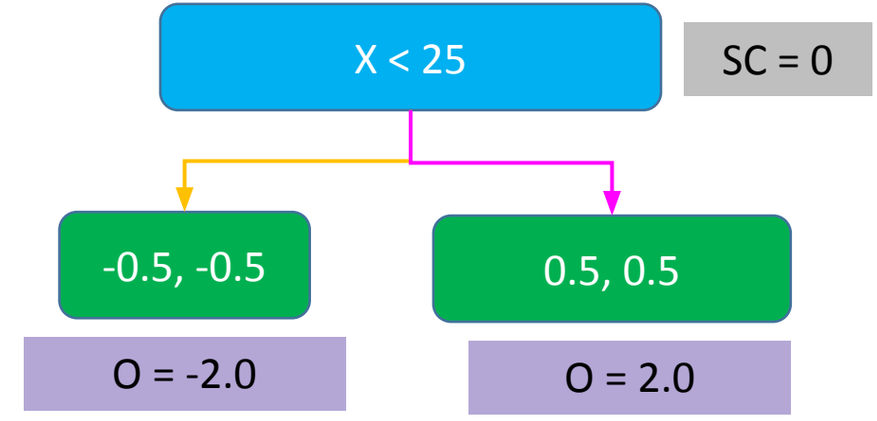
$$\frac{(\sum \text{Residual}_i)}{\sum [\text{Previous Probability}_i \times (1 - \text{Previous Probability}_i)] + \lambda}$$

X	Y-p
23	0
24	0
25.8	?
26	1
27	1

Problem 1-2

Step 5: Calculate the Gain.
Gain = Left SC + Right SC - Root SC

The Best Gain = 4
Select X < 25



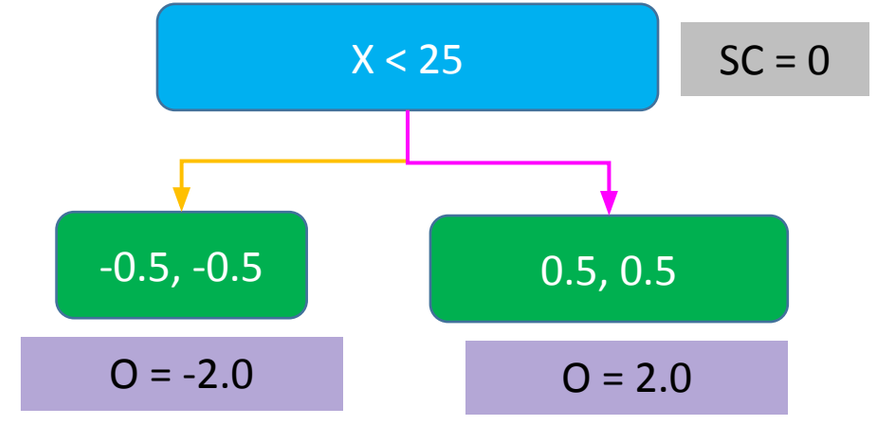
X	Y-p
23	0
24	0
25.8	?
26	1
27	1

Problem 1-2

Step 5: Calculate the Gain.
Gain = Left SC + Right SC - Root SC

The Best Gain = 4
Select X < 25

log(0.5/(1-0.5)) = 0



$$\frac{p}{1-p} = \text{odds}$$

$$\log\left(\frac{p}{1-p}\right) = \log(\text{odds})$$

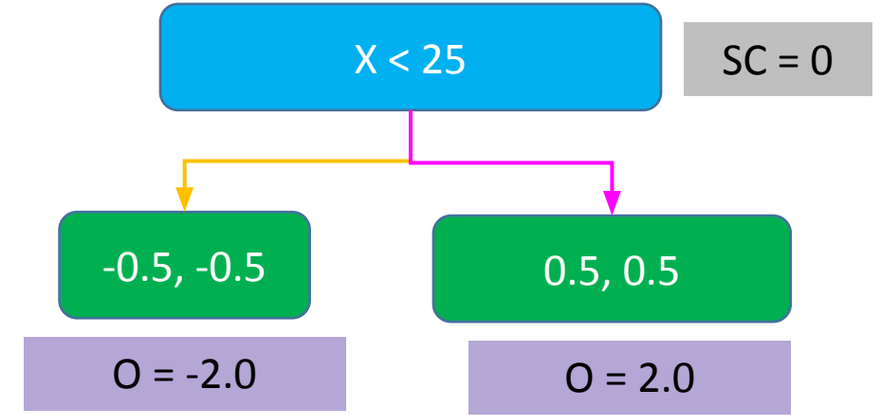
X	Y-p
23	0
24	0
25.8	?
26	1
27	1

Problem 1-2

Step 5: Calculate the Gain.
Gain = Left SC + Right SC - Root SC

The Best Gain = 4
Select X < 25

0 + LR *



$$\frac{p}{1-p} = \text{odds}$$

$$\log\left(\frac{p}{1-p}\right) = \log(\text{odds})$$

X	Y-p
23	0
24	0
25.8	?
26	1
27	1

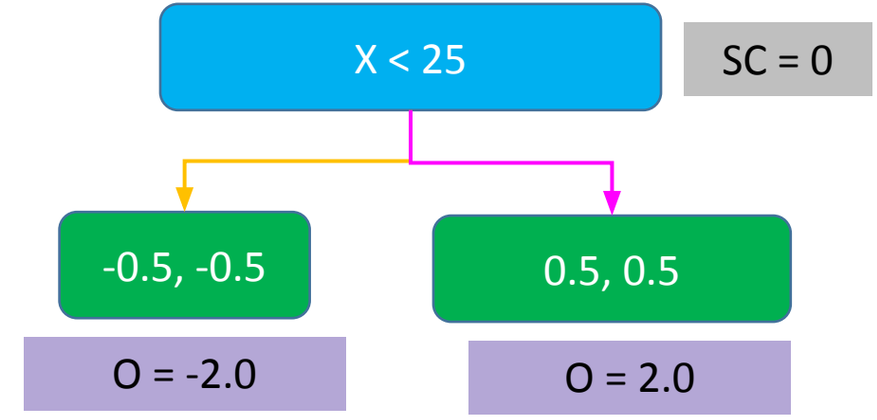
Log prediction = 0 + 0.3 * (2.0)
= 0.6

Problem 1-2

Step 5: Calculate the Gain.
Gain = Left SC + Right SC - Root SC

The Best Gain = 4
Select X < 25

0 + LR *



$$\frac{p}{1-p} = \text{odds}$$

$$\log\left(\frac{p}{1-p}\right) = \log(\text{odds})$$

X	Y
23	0
24	0
25.8	0.6457
26	1
27	1

Log prediction = 0 + 0.3 * (2.0)
= 0.6

$$\text{Probability} = \frac{e^{\log(\text{odds})}}{1 + e^{\log(\text{odds})}}$$

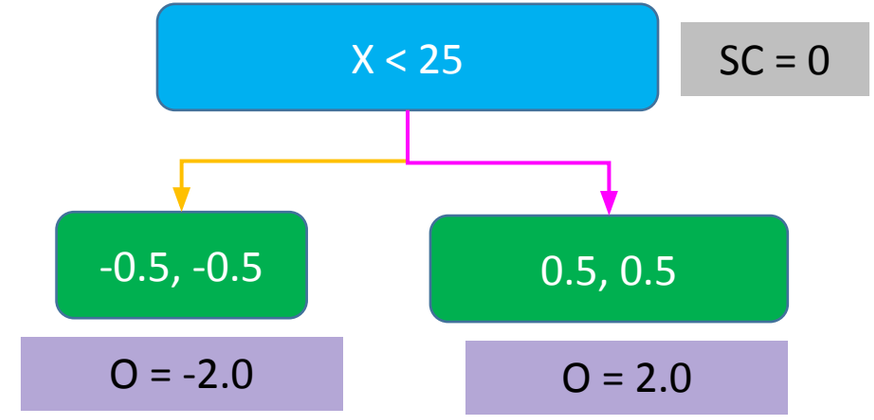
Probability = e^(0.6) / (1+e^(0.6)) = 0.6457

Problem 1-2

Step 5: Calculate the Gain.
Gain = Left SC + Right SC - Root SC

The Best Gain = 4
Select X < 25

0 + LR *



$$\frac{p}{1-p} = \text{odds}$$

$$\log\left(\frac{p}{1-p}\right) = \log(\text{odds})$$

X	Y-p	y-f0	yhat	y-yhat
23	0	-0.5	0.3543	-0.3543
24	0	-0.5	0.3543	-0.3543
26	1	0.5	0.6457	0.3543
27	1	0.5	0.6457	0.3543


QUIZ
QUIZ
QUIZ

Problem 3-4

Problem 3: Regression

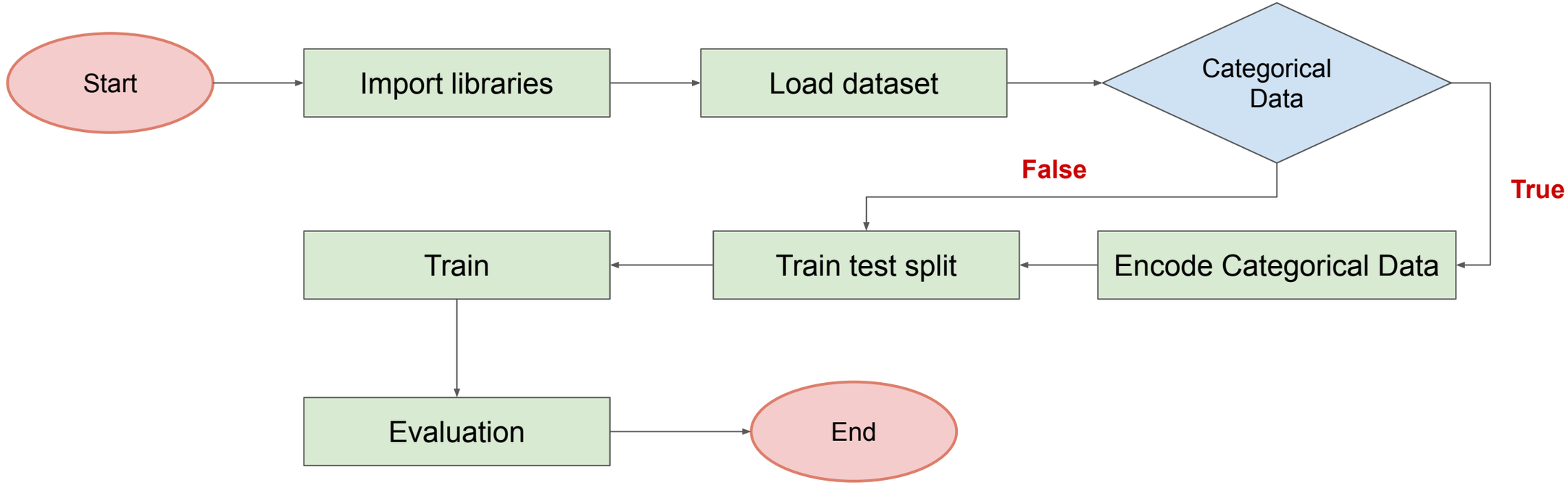
Hình 3: Forest fire prediction

	X	Y	month	day	FFMC	DMC	DC	ISI	temp	RH	wind	rain	area
0	7	5	mar	fri	4.468204	26.2	94.3	1.808289	8.2	51	6.7	False	0.000000
1	7	4	oct	tue	4.517431	35.4	669.1	2.041220	18.0	33	0.9	False	0.000000
2	7	4	oct	sat	4.517431	43.7	686.9	2.041220	14.6	33	1.3	False	0.000000
3	8	6	mar	fri	4.529368	33.3	77.5	2.302585	8.3	97	4.0	True	0.000000
4	8	6	mar	sun	4.503137	51.3	102.2	2.360854	11.4	99	1.8	False	0.000000
...	...	...	...	...	...	...	...	...	...	...	...	...	...
505	4	3	aug	sun	4.414010	56.7	665.6	1.064711	27.8	32	2.7	False	2.006871
506	2	4	aug	sun	4.414010	56.7	665.6	1.064711	21.9	71	5.8	False	4.012592
507	7	4	aug	sun	4.414010	56.7	665.6	1.064711	21.2	70	6.7	False	2.498152
508	1	4	aug	sat	4.558079	146.0	614.7	2.509599	25.6	42	4.0	False	0.000000
509	6	3	nov	tue	4.388257	3.0	106.7	0.741937	11.8	31	4.5	False	0.000000

- **Task**: Dự đoán diện tích rừng bị cháy (area)
- **Samples**: 510
- **Features**: 13, input 12, target: 1

Problem 3-4

Problem 3: Regression



Problem 3-4

Problem 3: Regression

- ['month', 'day', 'rain']
- Number of categories in month: 12
- Number of categories in day: 7
- Number of categories in rain: 2

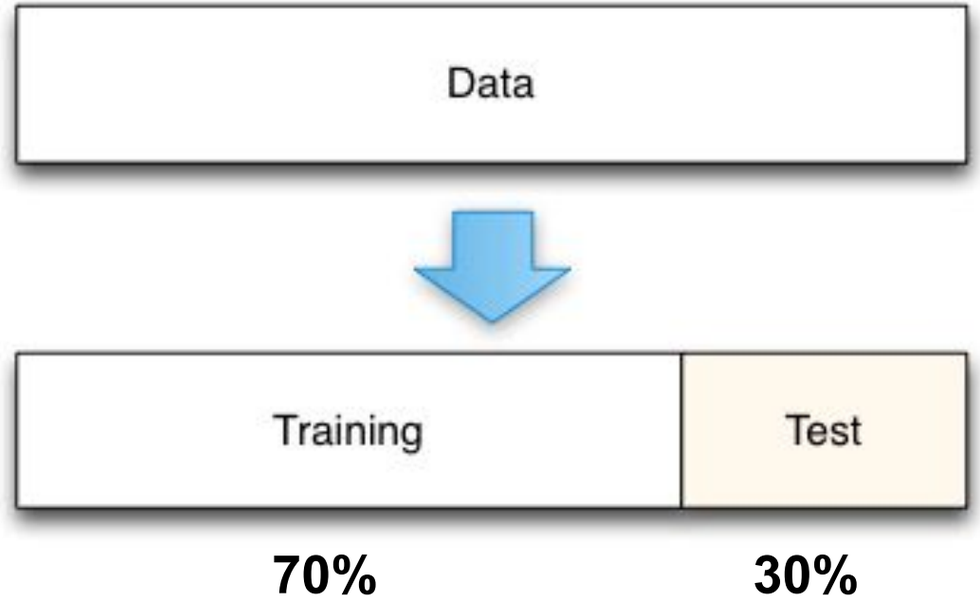
	X	Y	month	day	FFMC	DMC	DC	ISI	temp	RH	wind	rain	area
0	7	5	mar	fri	4.468204	26.2	94.3	1.808289	8.2	51	6.7	False	0.000000
1	7	4	oct	tue	4.517431	35.4	669.1	2.041220	18.0	33	0.9	False	0.000000
2	7	4	oct	sat	4.517431	43.7	686.9	2.041220	14.6	33	1.3	False	0.000000
3	8	6	mar	fri	4.529368	33.3	77.5	2.302585	8.3	97	4.0	True	0.000000
4	8	6	mar	sun	4.503137	51.3	102.2	2.360854	11.4	99	1.8	False	0.000000
...	...	...	...	...	...	...	...	...	...	...	...	...	...
505	4	3	aug	sun	4.414010	56.7	665.6	1.064711	27.8	32	2.7	False	2.006871
506	2	4	aug	sun	4.414010	56.7	665.6	1.064711	21.9	71	5.8	False	4.012592
507	7	4	aug	sun	4.414010	56.7	665.6	1.064711	21.2	70	6.7	False	2.498152
508	1	4	aug	sat	4.558079	146.0	614.7	2.509599	25.6	42	4.0	False	0.000000
509	6	3	nov	tue	4.388257	3.0	106.7	0.741937	11.8	31	4.5	False	0.000000

	X	Y	FFMC	DMC	DC	ISI	temp	RH	wind	area	month	day	rain
0	7	5	4.468204	26.2	94.3	1.808289	8.2	51	6.7	0.000000	7.0	0.0	0.0
1	7	4	4.517431	35.4	669.1	2.041220	18.0	33	0.9	0.000000	10.0	5.0	0.0
2	7	4	4.517431	43.7	686.9	2.041220	14.6	33	1.3	0.000000	10.0	2.0	0.0
3	8	6	4.529368	33.3	77.5	2.302585	8.3	97	4.0	0.000000	7.0	0.0	1.0
4	8	6	4.503137	51.3	102.2	2.360854	11.4	99	1.8	0.000000	7.0	3.0	0.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...
505	4	3	4.414010	56.7	665.6	1.064711	27.8	32	2.7	2.006871	1.0	3.0	0.0
506	2	4	4.414010	56.7	665.6	1.064711	21.9	71	5.8	4.012592	1.0	3.0	0.0
507	7	4	4.414010	56.7	665.6	1.064711	21.2	70	6.7	2.498152	1.0	3.0	0.0
508	1	4	4.558079	146.0	614.7	2.509599	25.6	42	4.0	0.000000	1.0	2.0	0.0
509	6	3	4.388257	3.0	106.7	0.741937	11.8	31	4.5	0.000000	9.0	5.0	0.0

Problem 3-4

Problem 3: Regression

- **Task**: Dự đoán diện tích rừng bị cháy (area)
- **Samples**: 510
- **Features**: 13, input 12, target: 1



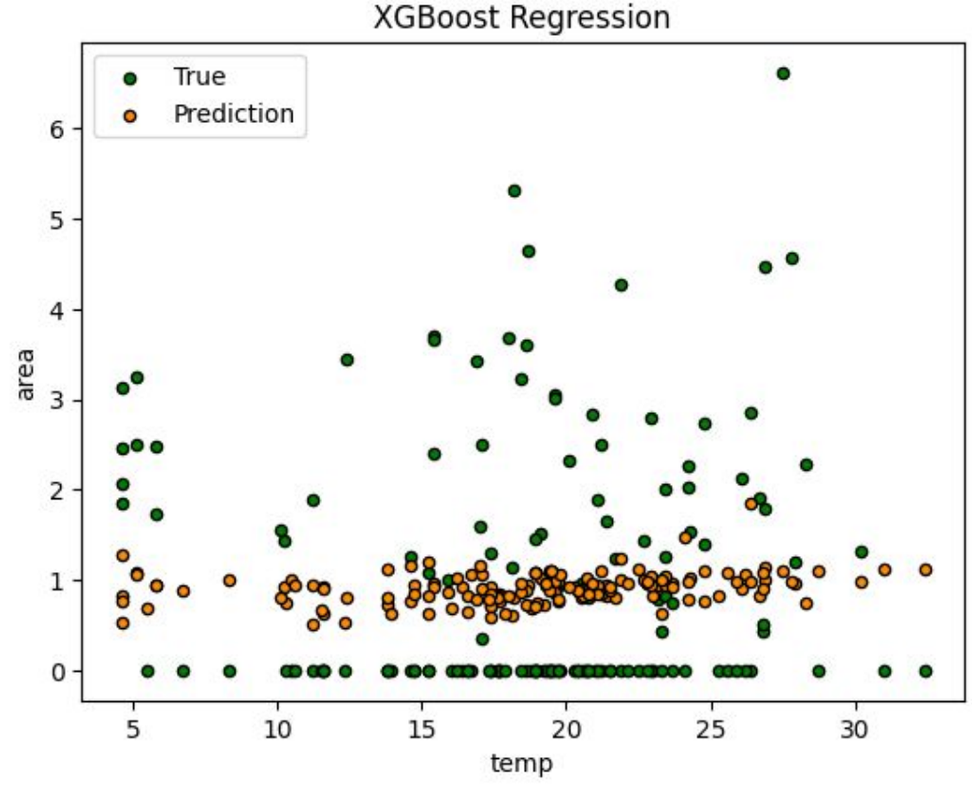
Number of training samples: 357
Number of val samples: 153

Problem 3-4

Problem 3: Regression

- **Build Model**: XGBoost
- **Evaluation**: MAE, MSE

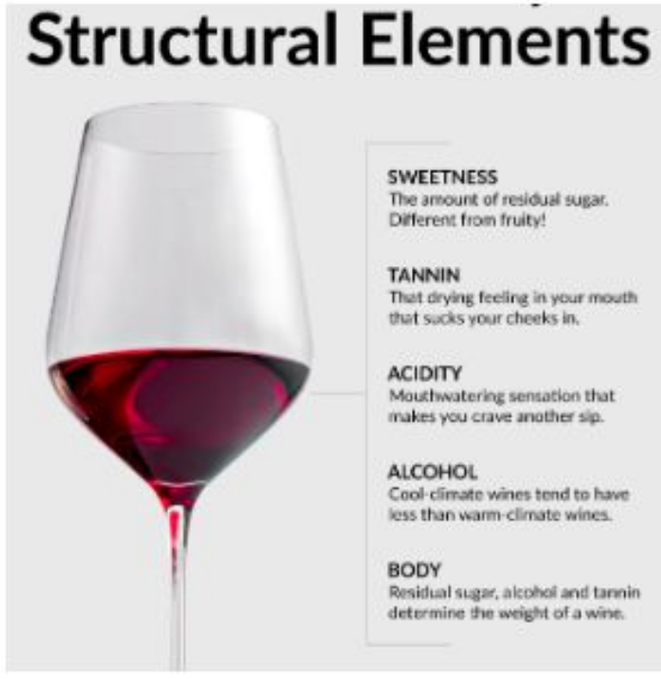
```
XGBRegressor
XGBRegressor(base_score=None, booster=None, callbacks=None,
             colsample_bylevel=None, colsample_bynode=None,
             colsample_bytree=None, early_stopping_rounds=None,
             enable_categorical=False, eval_metric=None, feature_types=None,
             gamma=None, gpu_id=None, grow_policy=None, importance_type=None,
             interaction_constraints=None, learning_rate=0.01, max_bin=None,
             max_cat_threshold=None, max_cat_to_onehot=None,
             max_delta_step=None, max_depth=3, max_leaves=None,
             min_child_weight=None, missing=nan, monotone_constraints=None,
             n_estimators=102, n_jobs=None, num_parallel_tree=None,
             predictor=None, random_state=None, ...)
```



Evaluation results on test set:
Mean Absolute Error: 1.0922937068201708
Mean Squared Error: 1.8814015748734487

Problem 3-4

Problem 4: Classification



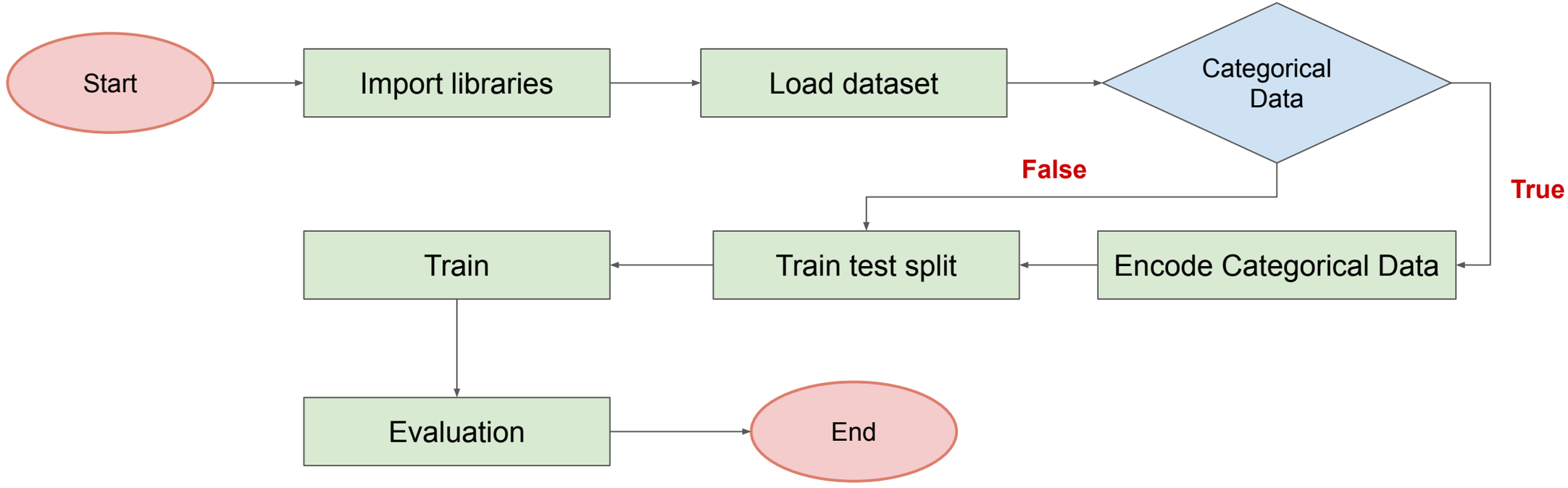
	alcohol	malic_acid	ash	alcalinity_of_ash	magnesium	total_phenols	flavanoids	nonflavanoid_phenols	proanthocyanins	color_intensity	hue	od280/od315_of_diluted_wines	proline	Target
0	14.23	1.71	2.43	15.6	127.0	2.80	3.06	0.28	2.29	5.64	1.04	3.92	1065.0	0
1	13.20	1.78	2.14	11.2	100.0	2.65	2.76	0.26	1.28	4.38	1.05	3.40	1050.0	0
2	13.16	2.36	2.67	18.6	101.0	2.80	3.24	0.30	2.81	5.68	1.03	3.17	1185.0	0
3	14.37	1.95	2.50	16.8	113.0	3.85	3.49	0.24	2.18	7.80	0.86	3.45	1480.0	0
4	13.24	2.59	2.87	21.0	118.0	2.80	2.69	0.39	1.82	4.32	1.04	2.93	735.0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
173	13.71	5.65	2.45	20.5	95.0	1.68	0.61	0.52	1.06	7.70	0.64	1.74	740.0	2
174	13.40	3.91	2.48	23.0	102.0	1.80	0.75	0.43	1.41	7.30	0.70	1.56	750.0	2
175	13.27	4.28	2.26	20.0	120.0	1.59	0.69	0.43	1.35	10.20	0.59	1.56	835.0	2
176	13.17	2.59	2.37	20.0	120.0	1.65	0.68	0.53	1.46	9.30	0.60	1.62	840.0	2
177	14.13	4.10	2.74	24.5	96.0	2.05	0.76	0.56	1.35	9.20	0.61	1.60	560.0	2

178 rows × 14 columns

- **Task**: Phân loại chất lượng rượu vang (Target)
- **Samples**: 178
- **Features**: 14, input 13, target: 1

Problem 3-4

Problem 4: Classification



Problem 3-4

Problem 4: Classification

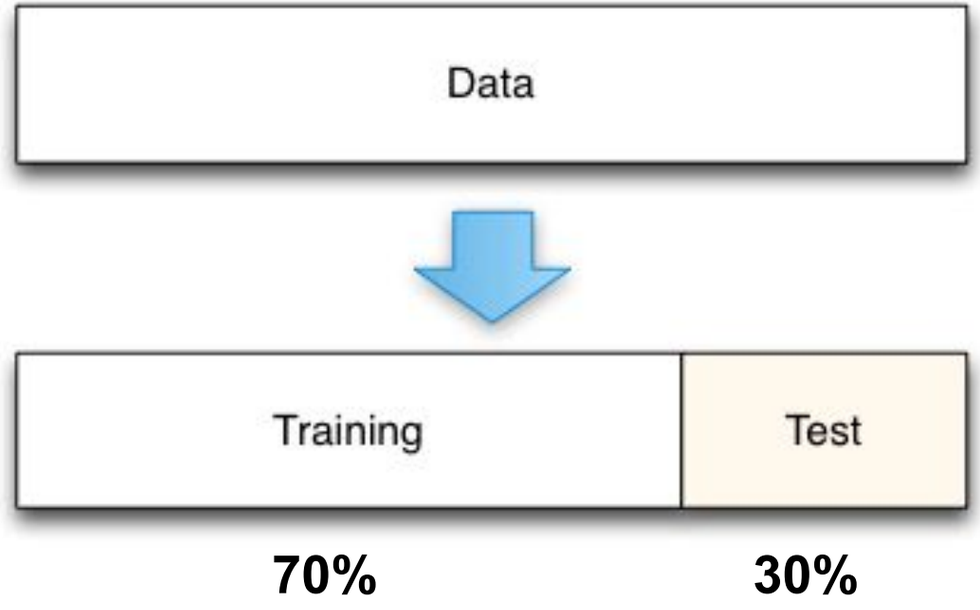
- Không có Categorical feature

	alcohol	malic_acid	ash	alcalinity_of_ash	magnesium	total_phenols	flavanoids	nonflavanoid_phenols	proanthocyanins	color_intensity	hue	od280/od315_of_diluted_wines	proline	Target
0	14.23	1.71	2.43	15.6	127.0	2.80	3.06	0.28	2.29	5.64	1.04	3.92	1065.0	0
1	13.20	1.78	2.14	11.2	100.0	2.65	2.76	0.26	1.28	4.38	1.05	3.40	1050.0	0
2	13.16	2.36	2.67	18.6	101.0	2.80	3.24	0.30	2.81	5.68	1.03	3.17	1185.0	0
3	14.37	1.95	2.50	16.8	113.0	3.85	3.49	0.24	2.18	7.80	0.86	3.45	1480.0	0
4	13.24	2.59	2.87	21.0	118.0	2.80	2.69	0.39	1.82	4.32	1.04	2.93	735.0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
173	13.71	5.65	2.45	20.5	95.0	1.68	0.61	0.52	1.06	7.70	0.64	1.74	740.0	2
174	13.40	3.91	2.48	23.0	102.0	1.80	0.75	0.43	1.41	7.30	0.70	1.56	750.0	2
175	13.27	4.28	2.26	20.0	120.0	1.59	0.69	0.43	1.35	10.20	0.59	1.56	835.0	2
176	13.17	2.59	2.37	20.0	120.0	1.65	0.68	0.53	1.46	9.30	0.60	1.62	840.0	2
177	14.13	4.10	2.74	24.5	96.0	2.05	0.76	0.56	1.35	9.20	0.61	1.60	560.0	2

Problem 3-4

Problem 4: Classification

- **Task**: Phân loại chất lượng rượu vang (Target)
- **Samples**: 178
- **Features**: 14, input 13, target: 1



Number of training samples: 124
Number of val samples: 54

Problem 3-4

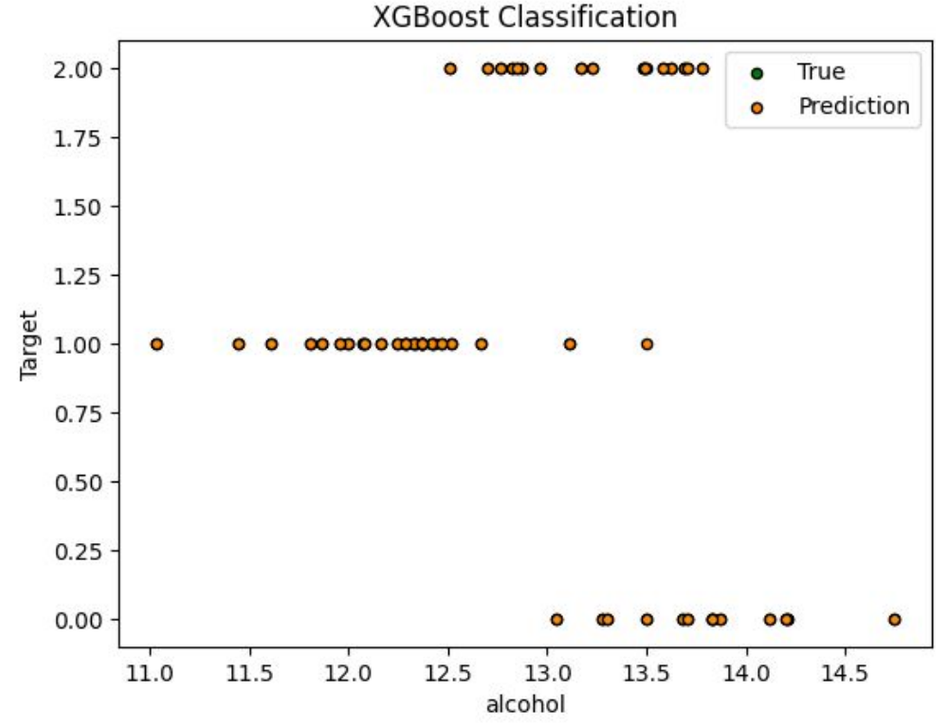
Problem 4: Classification

- **Build Model**: XGBoost
- **Evaluation**: Accuracy

Train ACC: 1.0
Test ACC: 0.9814814814814815

XGBClassifier

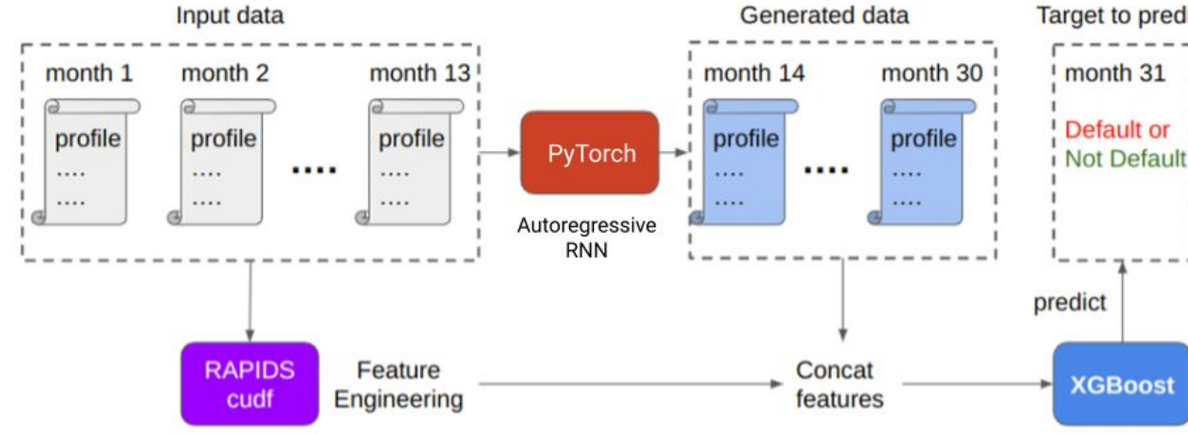
```
XGBClassifier(base_score=None, booster=None, callbacks=None,
              colsample_bylevel=None, colsample_bynode=None,
              colsample_bytree=None, early_stopping_rounds=None,
              enable_categorical=False, eval_metric=None, feature_types=None,
              gamma=None, gpu_id=None, grow_policy=None, importance_type=None,
              interaction_constraints=None, learning_rate=None, max_bin=None,
              max_cat_threshold=None, max_cat_to_onehot=None,
              max_delta_step=None, max_depth=None, max_leaves=None,
              min_child_weight=None, missing=nan, monotone_constraints=None,
              n_estimators=100, n_jobs=None, num_parallel_tree=None,
              objective='multi:softprob', predictor=None, ...)
```



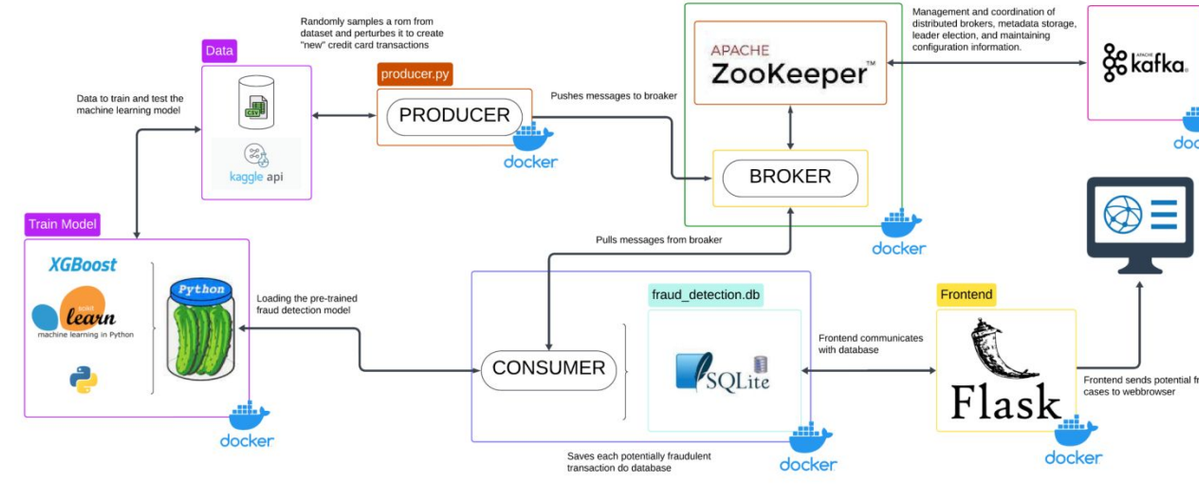
AI VIETNAM
All-in-One Course
(TA Session)

XGBoost in Finance

Forecasting Stock Prices



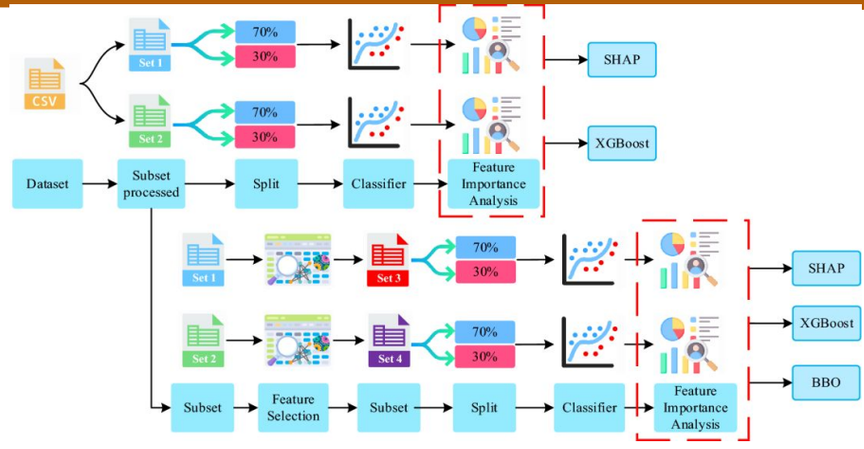
Predicting Credit Defaults



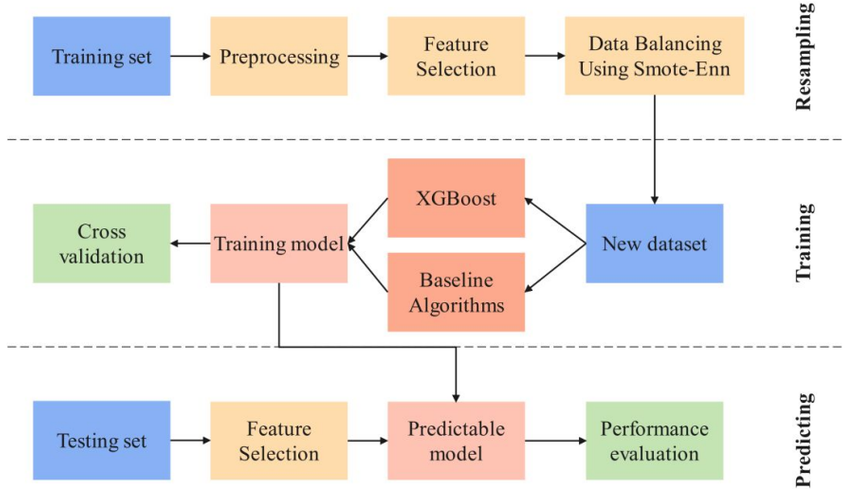
Credit Card Fraud Detection

AI VIETNAM
All-in-One Course
(TA Session)

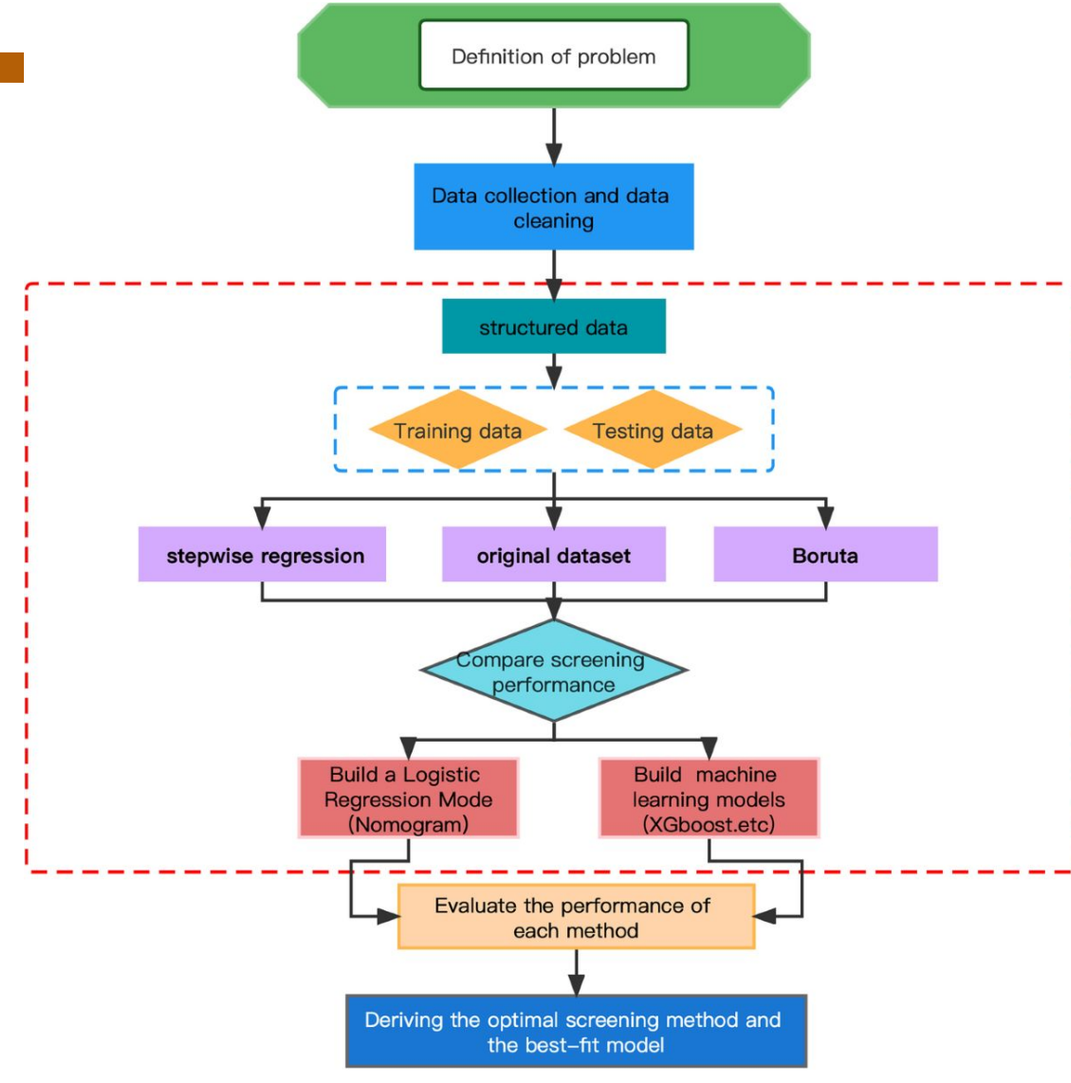
XGBoost in Healthcare



Detection of the chronic kidney disease



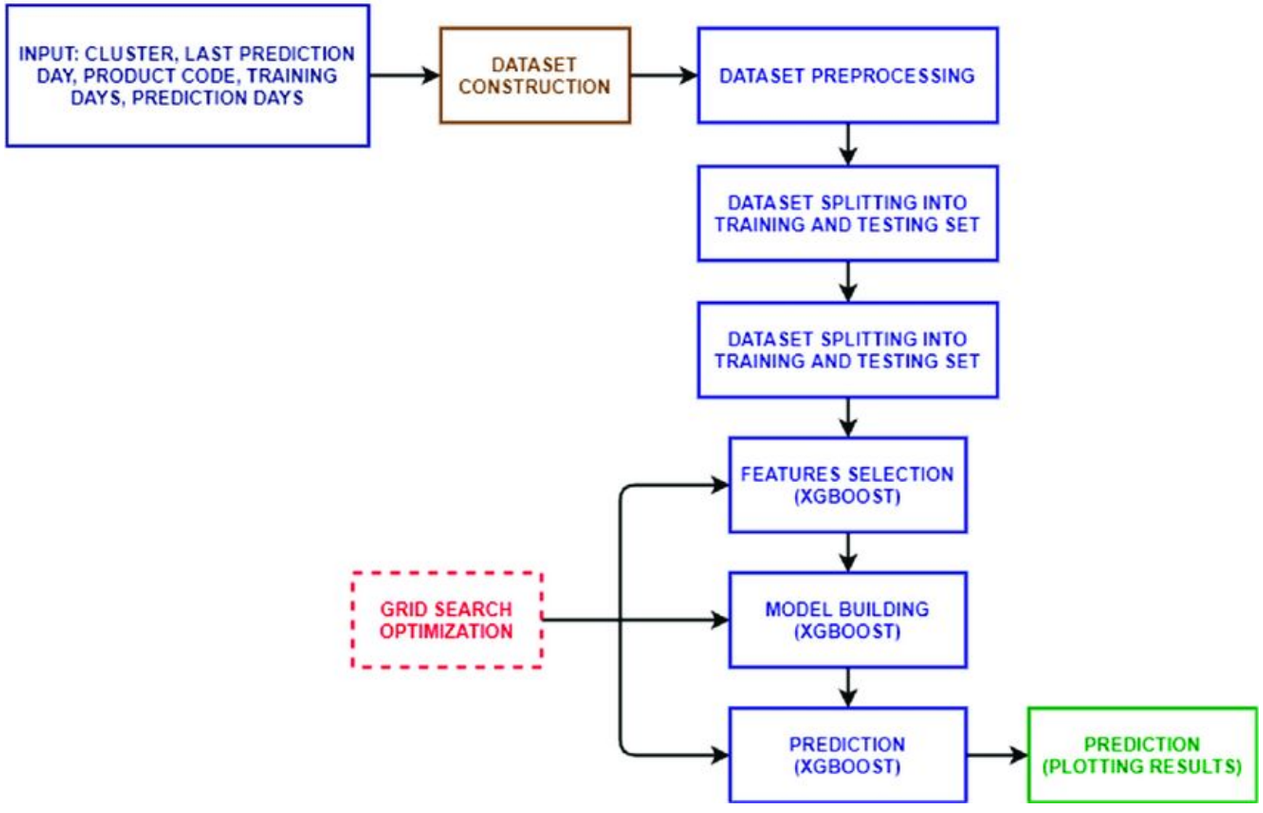
A Heart Disease Prediction

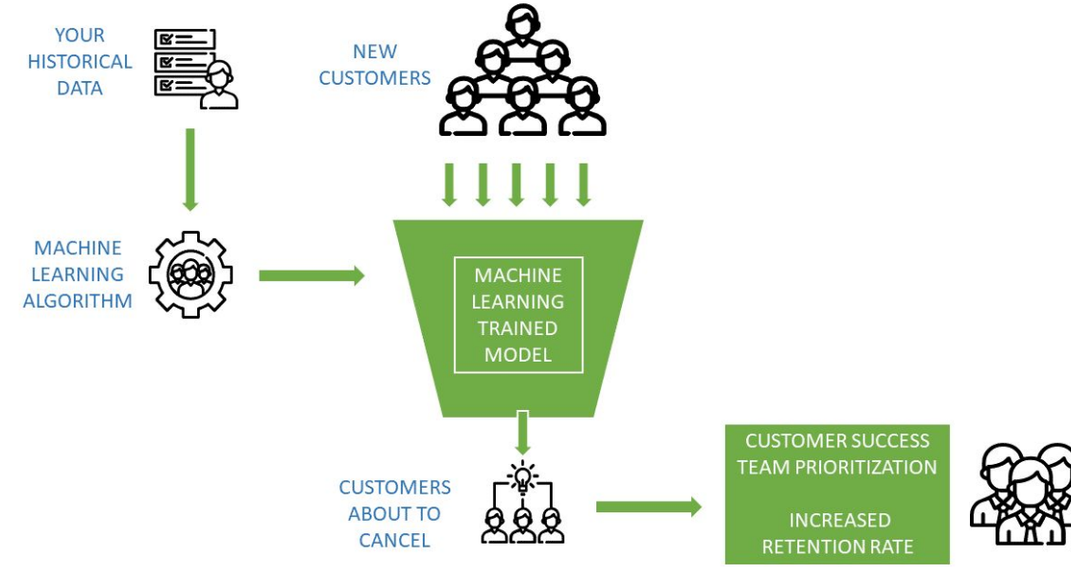


Early lung cancer prediction

AI VIETNAM
All-in-One Course
(TA Session)

XGBoost in Marketing and Customer Analytics



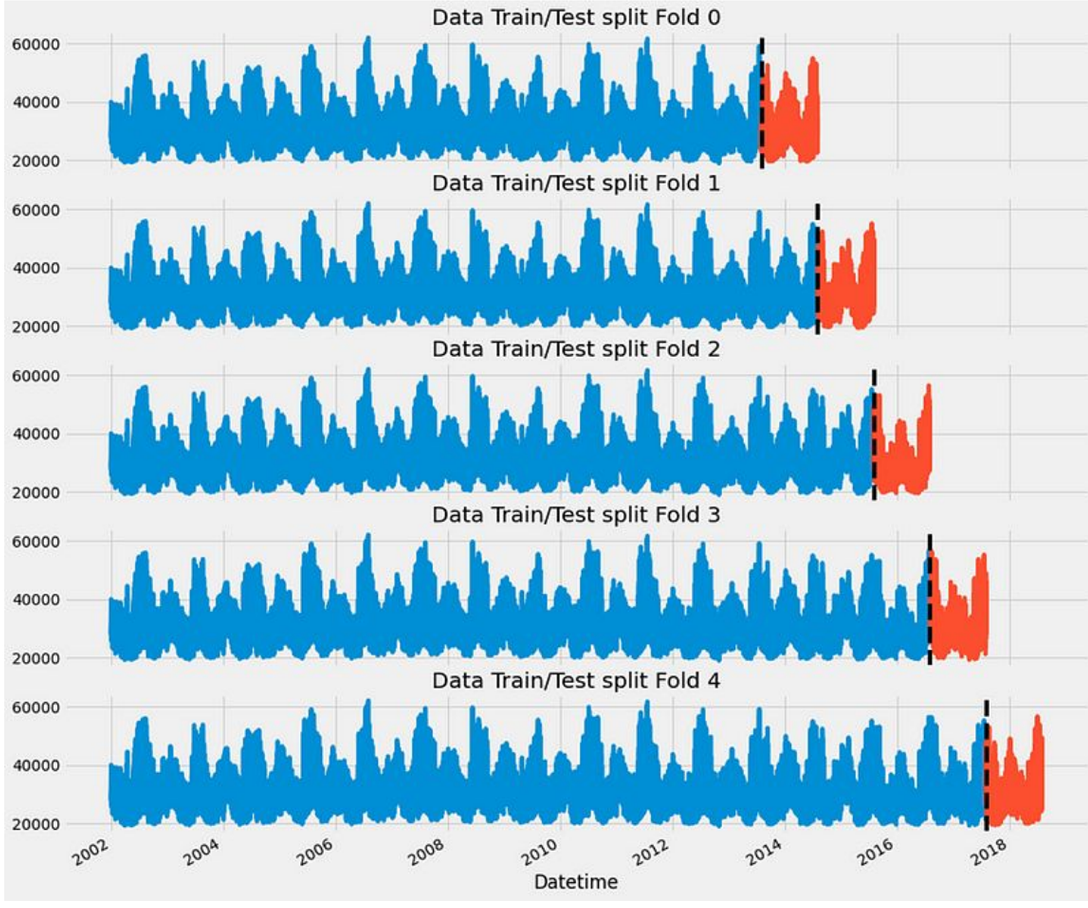


Cluster sales forecasting

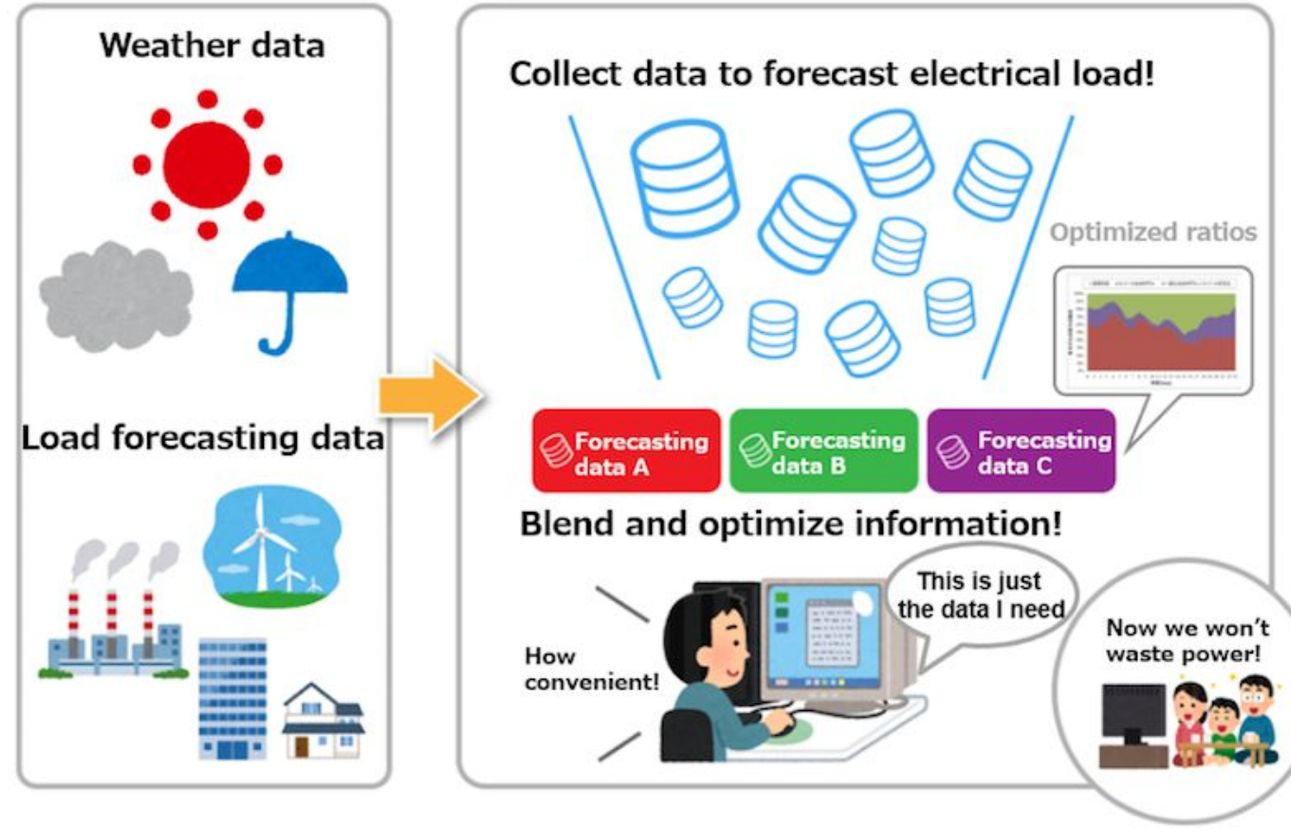
Churn Prediction

AI VIETNAM
All-in-One Course
(TA Session)

XGBoost in Time Series Data



Power Consumption Forecasting



Weather Prediction

XGBoost in AI

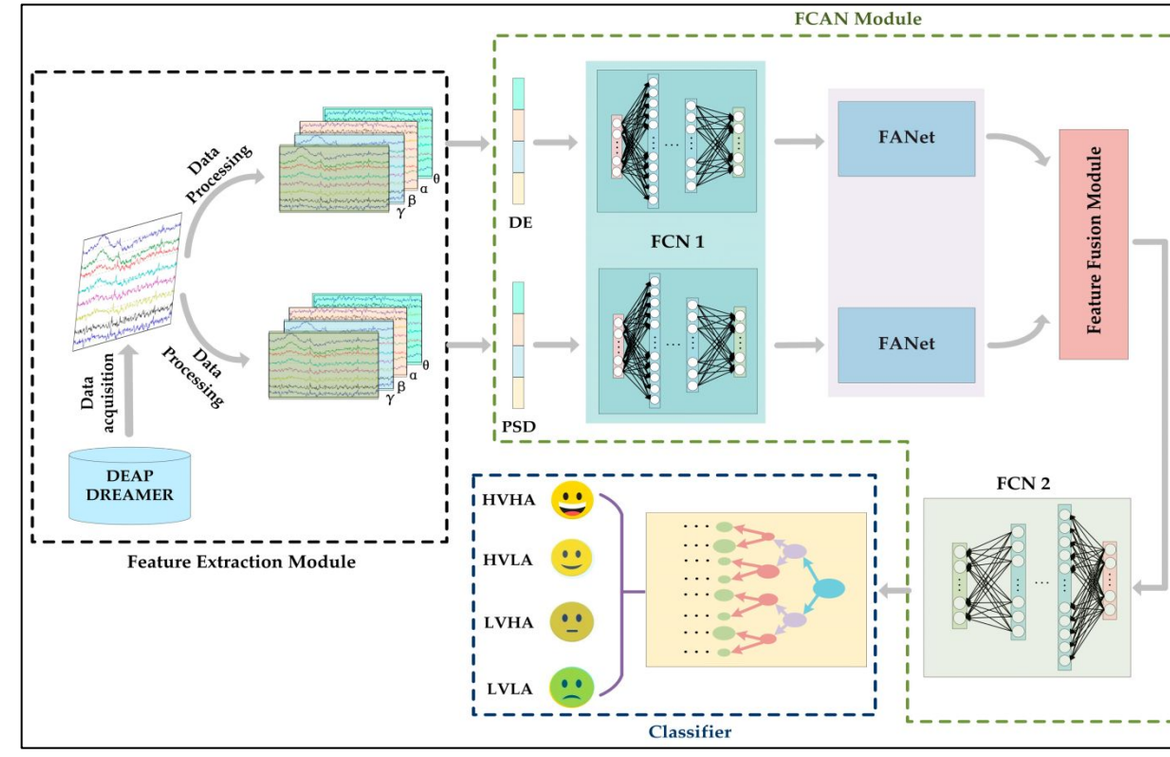
TIANCHI

NUMERAI

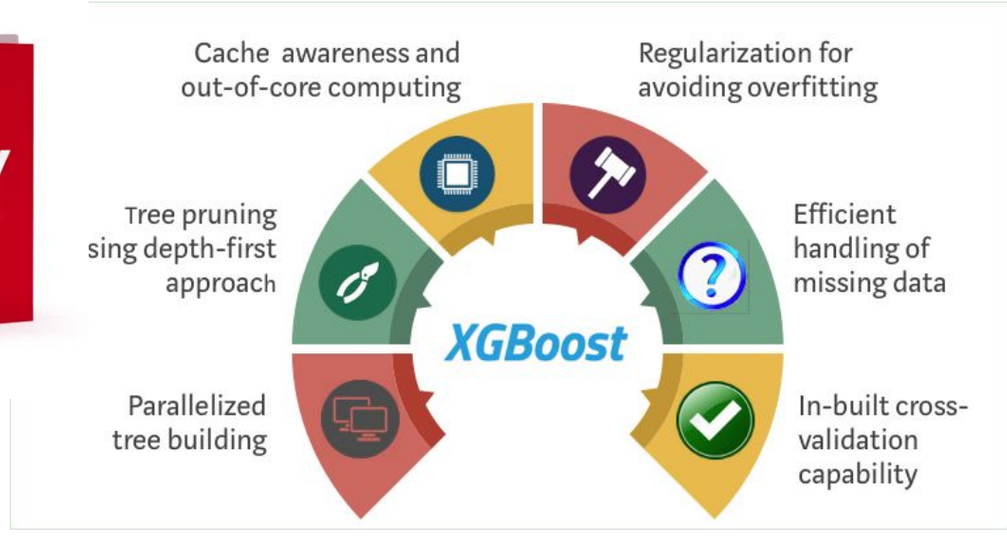
DRIVENDATA

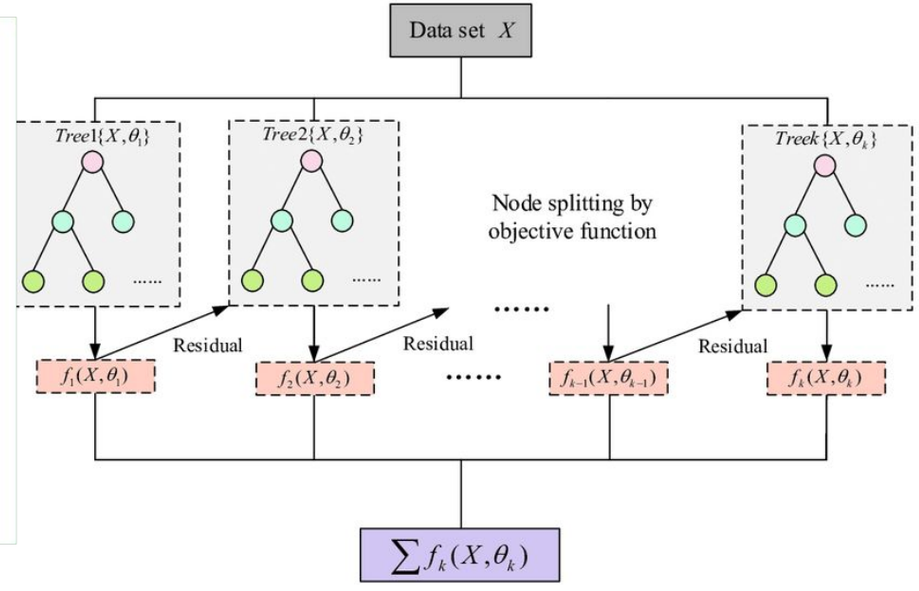
kaggle

Data Science Competitions



Combine with Deep Learning Algorithms





- ✔ Recap ensemble learning principles.
- ✔ Summarize decision tree functionality.
- ✔ Outline bagging benefits.
- ✔ Highlight random forest enhancements.
- ✔ Review boosting and sequential improvements.
- ✔ Condense gradient boosting key aspects.
- ✔ Reiterate XGBoost fundamentals.
- ✔ Review manual XGBoost applications on simple problems.
- ✔ Discuss practical XGBoost library use in regression and classification tasks.