AI VIET NAM
@aivietnam.edu.vn

LightGBM
A practical approach

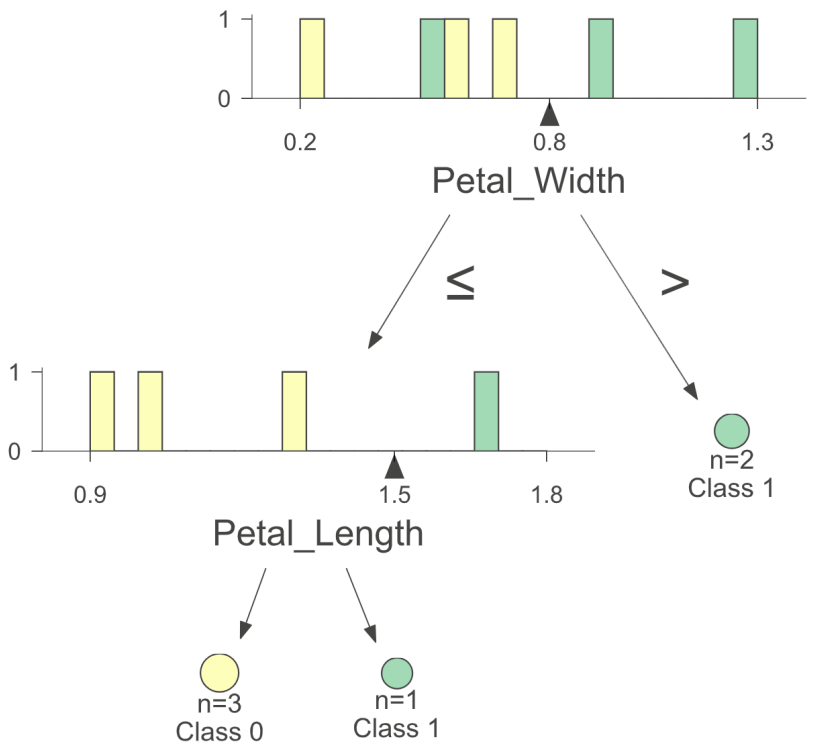
Quang-Vinh Dinh
Ph.D. in Computer Science

Year 2025 Data & Code Quiz & Feedback

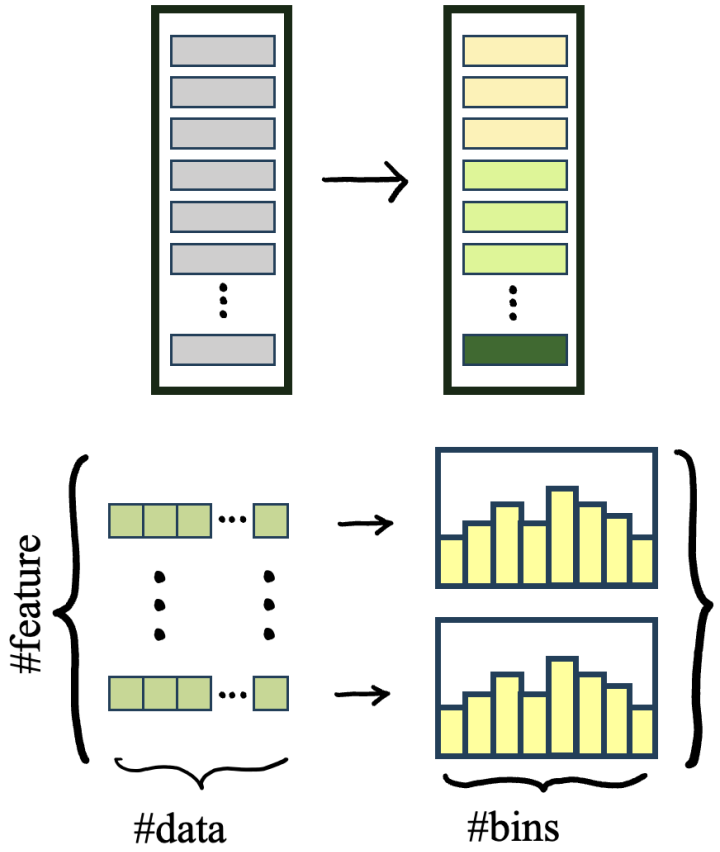
*This work is supported by STA Hào, STA Khôi, STA Linh, and STA Thịnh

Objectives

Dis. & Mov.



Improvements



Case Studies

Outline

SECTION 1

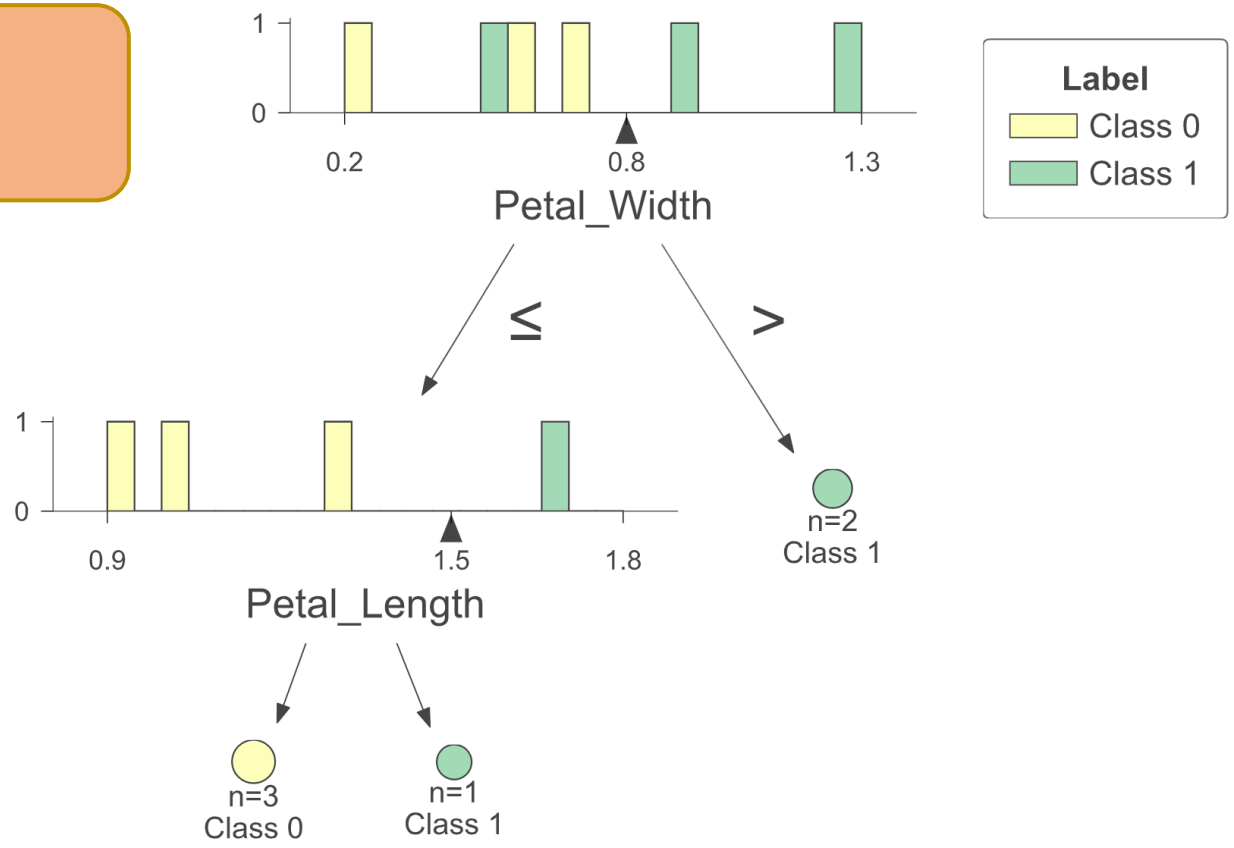
Discussion&Motivation

SECTION 2

Improvements

SECTION 3

Case Studies



Discussion

❖ Loss functions for regression

Mô hình	Hàm loss Regression	Ý nghĩa & Cải tiến
Decision Tree	$MSE = \frac{1}{n}\sum(y_i - \hat{y}_i)^2$	Dự đoán giá trị trung bình ở mỗi lá. Cắt theo tiêu chí giảm MSE nhiều nhất
Random Forest	MSE (từng cây), dự đoán trung bình	Giảm overfitting bằng **bagging** (lấy mẫu bootstrap + chọn random feature). Không thay đổi loss
AdaBoost	Adaboost Loss	Được dùng trong việc cập nhật các sample
Gradient Boosting	Bất kỳ hàm khả vi	Gradient Boosting coi bài toán là tối ưu hàm loss bất kỳ qua gradient descent (MSE)
XGBoost	Hàm loss khả vi + Regularization	Thêm L1, L2 để kiểm soát độ phức tạp của cây, giảm overfitting
LightGBM	Như XGBoost	Như XGBoost

Discussion

❖ Loss functions for classification

Mô hình	Hàm loss Classification	Ý nghĩa & Cải tiến
Decision Tree	Gini Index: $1 - \sum p_k^2$ hoặc Entropy: $-\sum p_k \log p_k$	Chọn split giảm impurity nhiều nhất.
Random Forest	Gini/Entropy	Giữ nguyên như Decision Tree, chỉ thay đổi sampling.
AdaBoost	Exponential Loss $\sum e^{-y_i f(x_i)}$	Trọng số tăng với các điểm phân loại sai.
Gradient Boosting	Log-loss: $-\sum [y_i \log p_i + (1 - y_i) \log(1 - p_i)]$	Sử dụng gradient của log-loss để cập nhật cây mới.
XGBoost	Log-loss + Regularization	Thêm penalty vào độ phức tạp của cây → cây đơn giản hơn, ít overfit.
LightGBM	Như XGBoost	Như XGBoost

Discussion

❖ Tree Building

Mô hình	Cách phát triển cây	Đặc điểm chính
Decision Tree	Top-down, Level-wise	Mỗi lần chọn split tốt nhất, phát triển toàn bộ cùng mức.
Random Forest	Nhiều cây độc lập, Level-wise	Bagging + Random features → giảm variance.
AdaBoost	Sequential Level-wise	Mỗi cây học trên lỗi của cây trước.
Gradient Boosting	Sequential Level-wise	Mỗi cây học trên residuals hoặc gradient của loss.
XGBoost	Sequential Level-wise + Regularization	Tối ưu thêm 2nd order Taylor + pruning theo gain.
LightGBM	Leaf-wise, Histogram, GOSS, and EFB	Tối ưu tốc độ & bộ nhớ

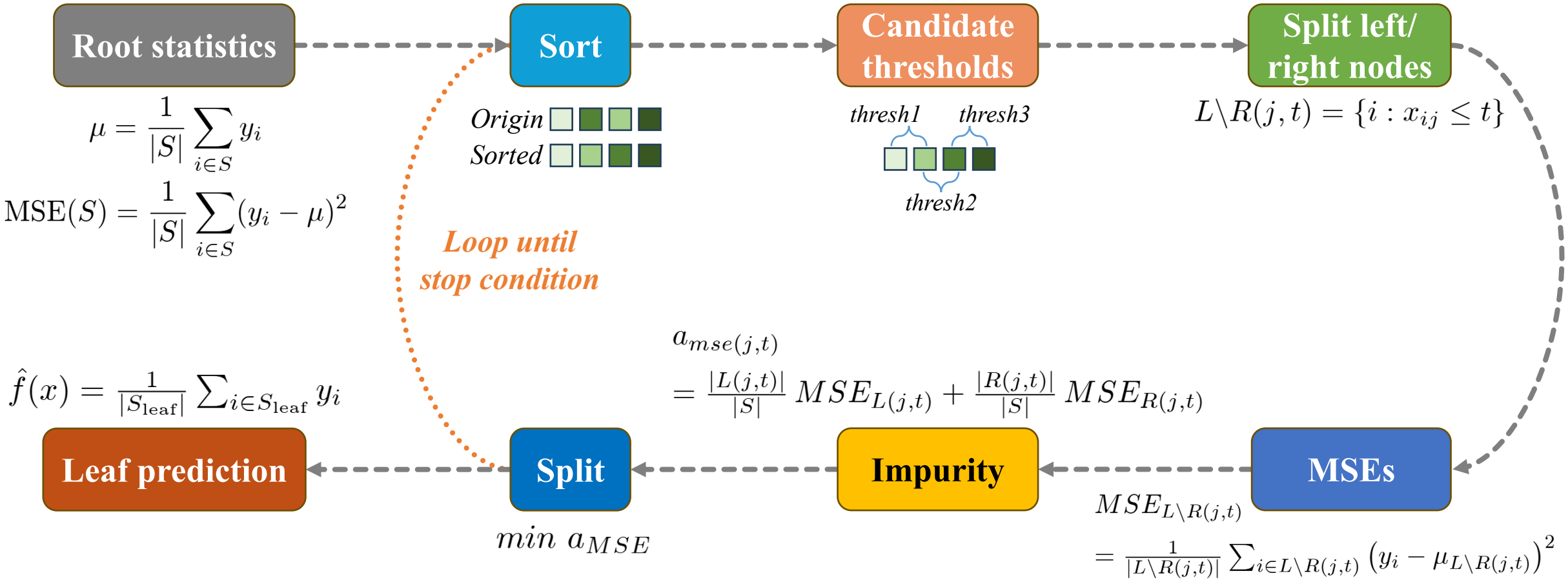
Discussion

❖ Summary

Giai đoạn	Regression Loss	Classification Loss	Cách phát triển cây	Điểm mới chính
Decision Tree	MSE	Gini/Entropy	Level-wise	Split tốt nhất theo impurity hoặc MSE
Random Forest	MSE	Gini/Entropy	Level-wise + Bagging	Giảm variance, random features
AdaBoost	MSE	Exponential Loss	Sequential Boosting	Tăng trọng số điểm sai
Gradient Boosting	Any diff. loss	Log-loss	Sequential Boosting	Boosting qua gradient
XGBoost	Any diff. loss + Reg.	Log-loss + Reg.	Level-wise + Reg.	Taylor 2nd order, pruning, regularization
LightGBM	Any diff. loss + Reg.	Log-loss + Reg.	Leaf-wise + Histogram + GOSS + EFD	Tăng tốc, leaf-wise growth, chọn mẫu quan trọng

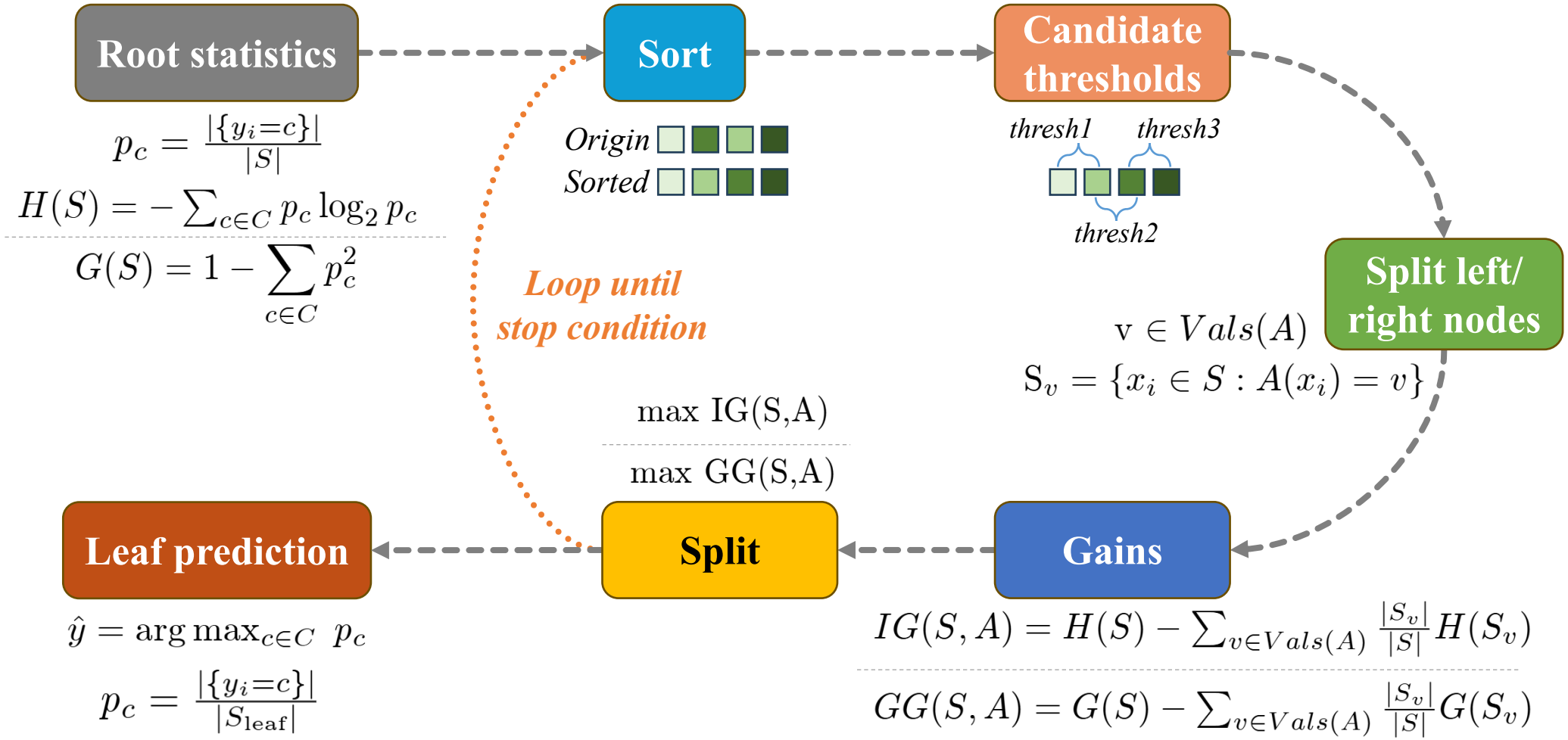
Regression Decision Tree

❖ Procedure



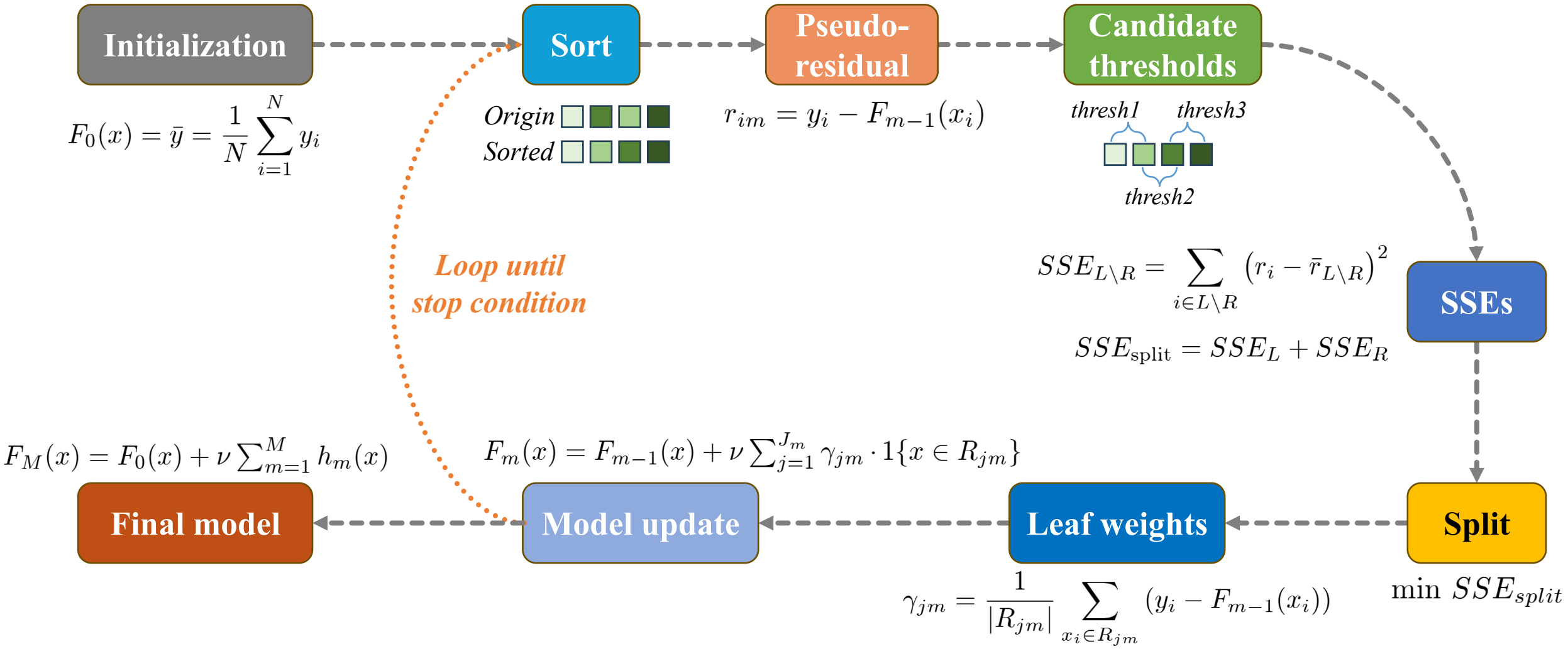
Classification Decision Tree

❖ Procedure



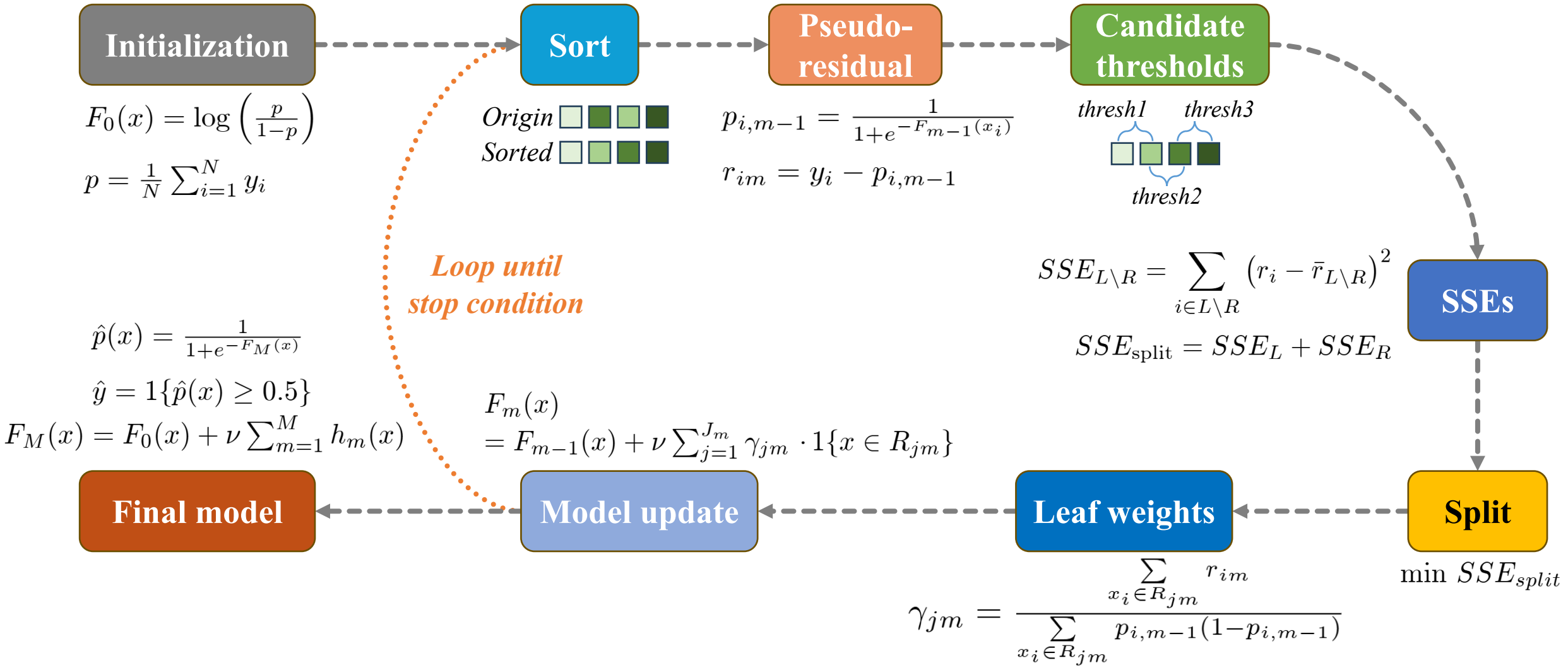
Regression Gradient Boosting

❖ Procedure



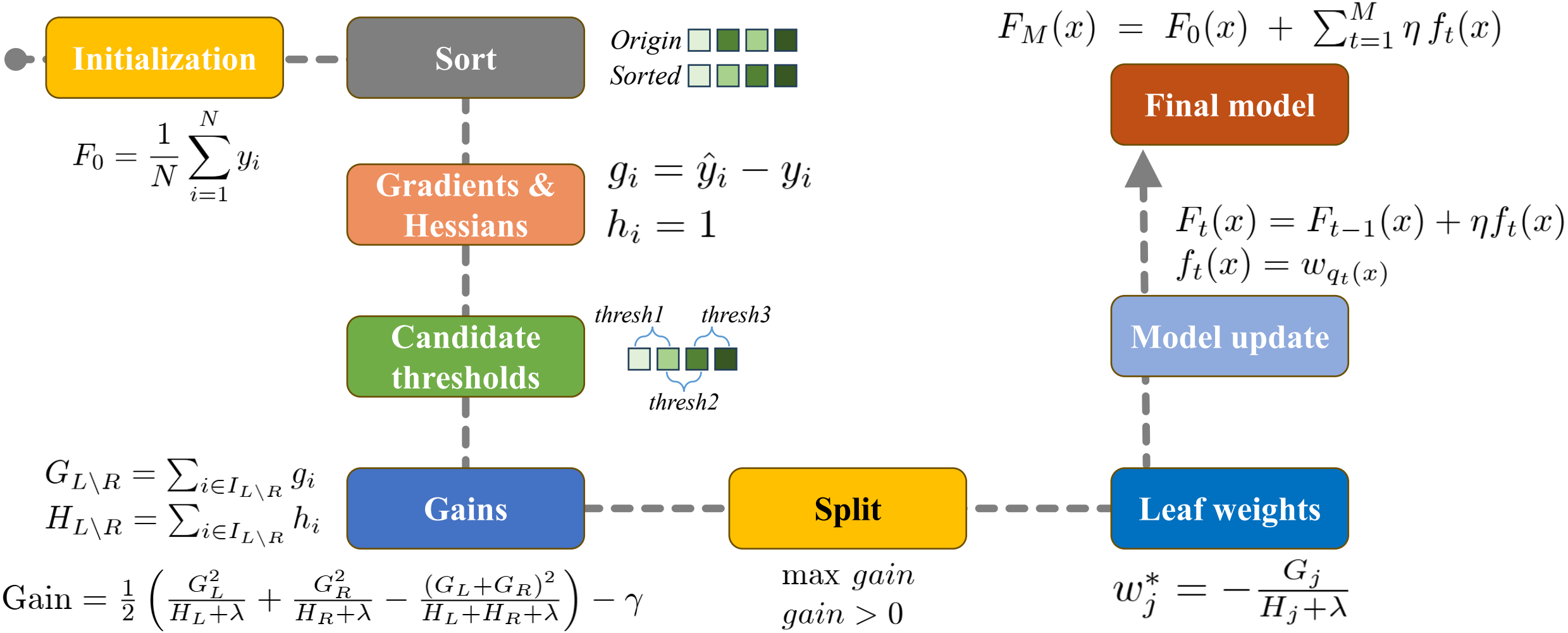
Classification Gradient Boosting

❖ Procedure



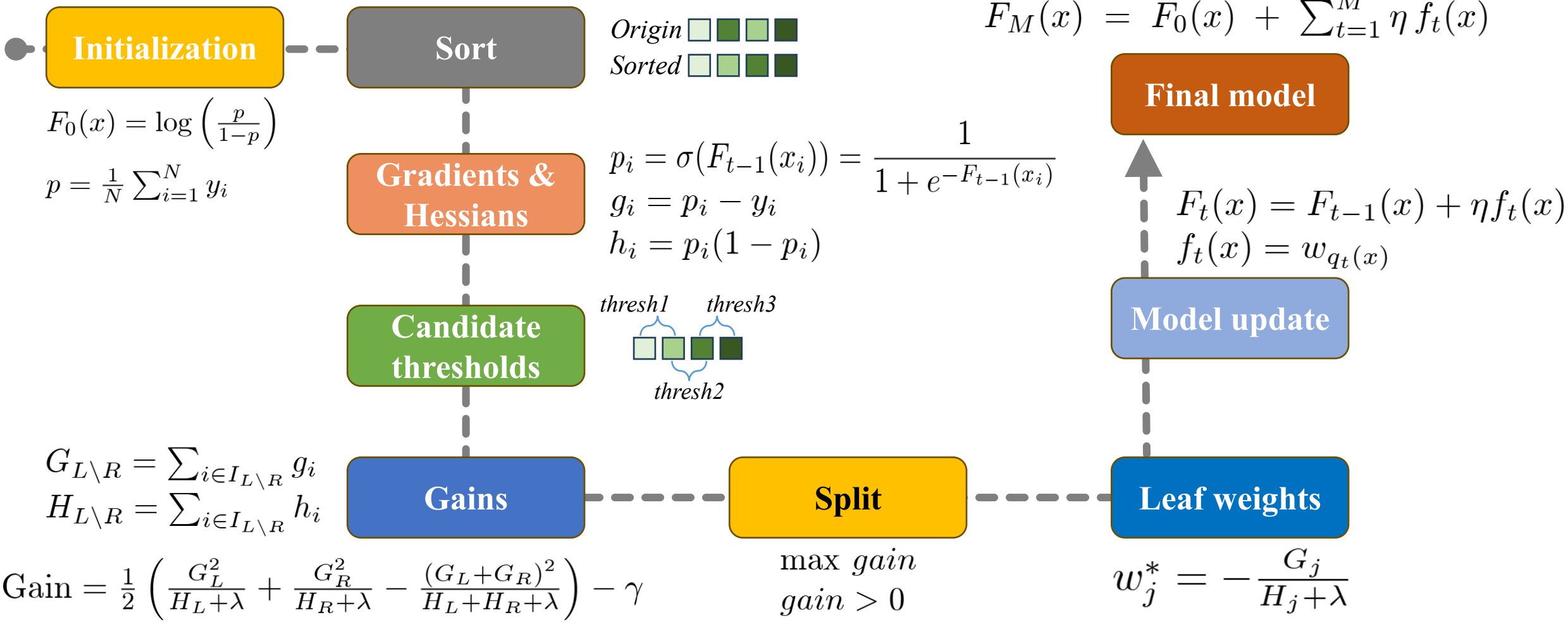
Regression XGBoost

❖ Procedure



Classification XGBoost

❖ Procedure



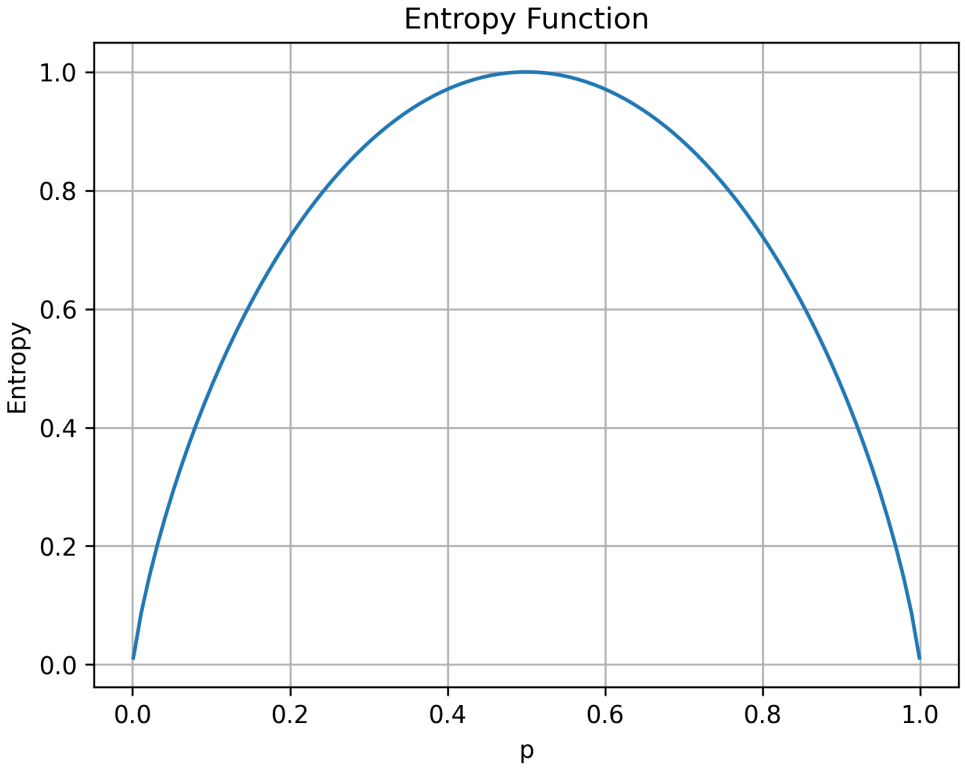
Loss Function

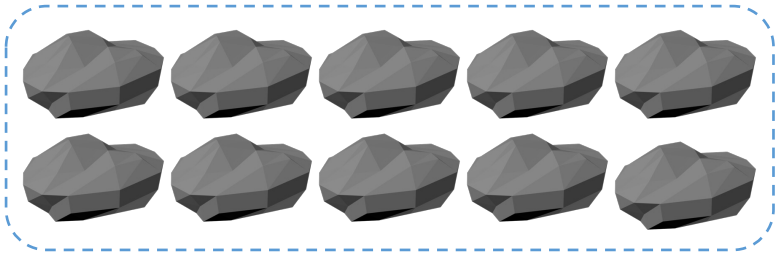
Review on Loss Functions

❖ Calculate Entropy

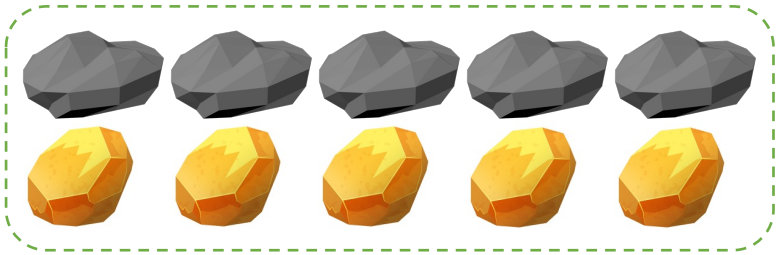
Entropy: Average of information

$$\mathrm{H}(X) := -\sum_{x \in X} p(x)\, log(p(x))$$





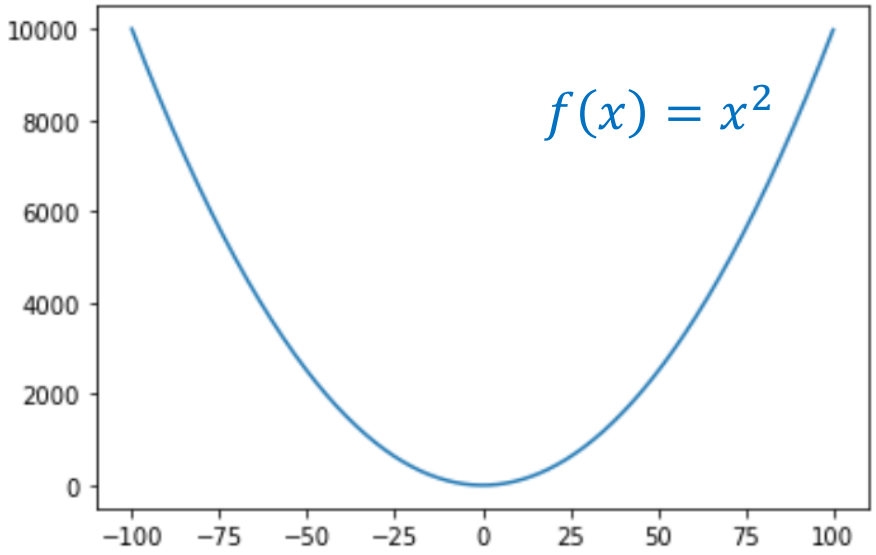
Entropy = ?



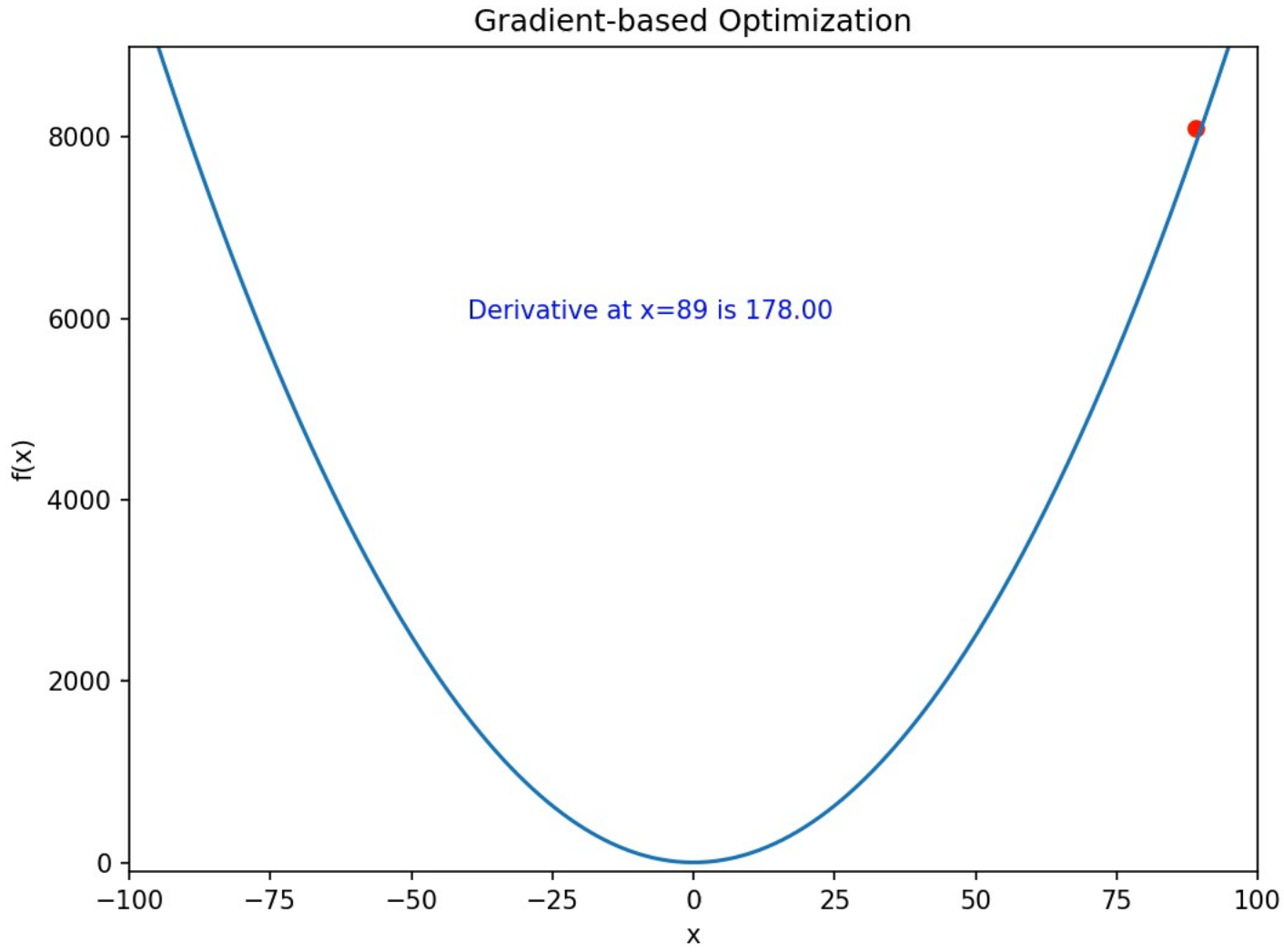
Entropy = ?

Understanding the GB Loss

❖ Square function

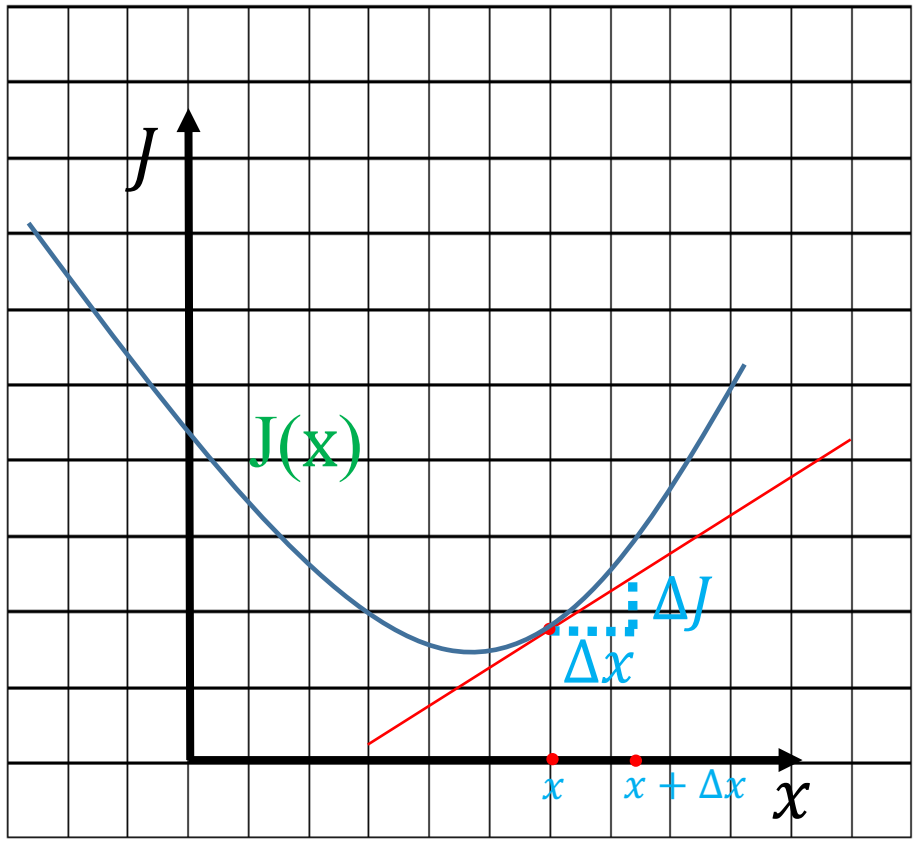


$$\frac{d}{dx} f(x) = 2x$$

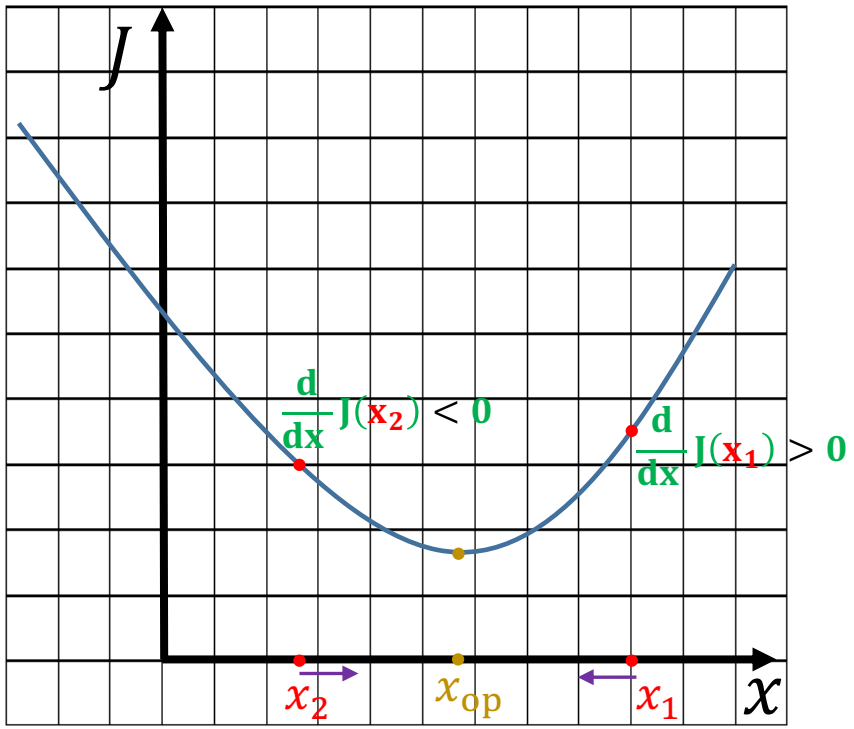


Understanding the GB Loss

❖ Gradient descent



$$\frac{d}{dx}J(x) = some\_function$$



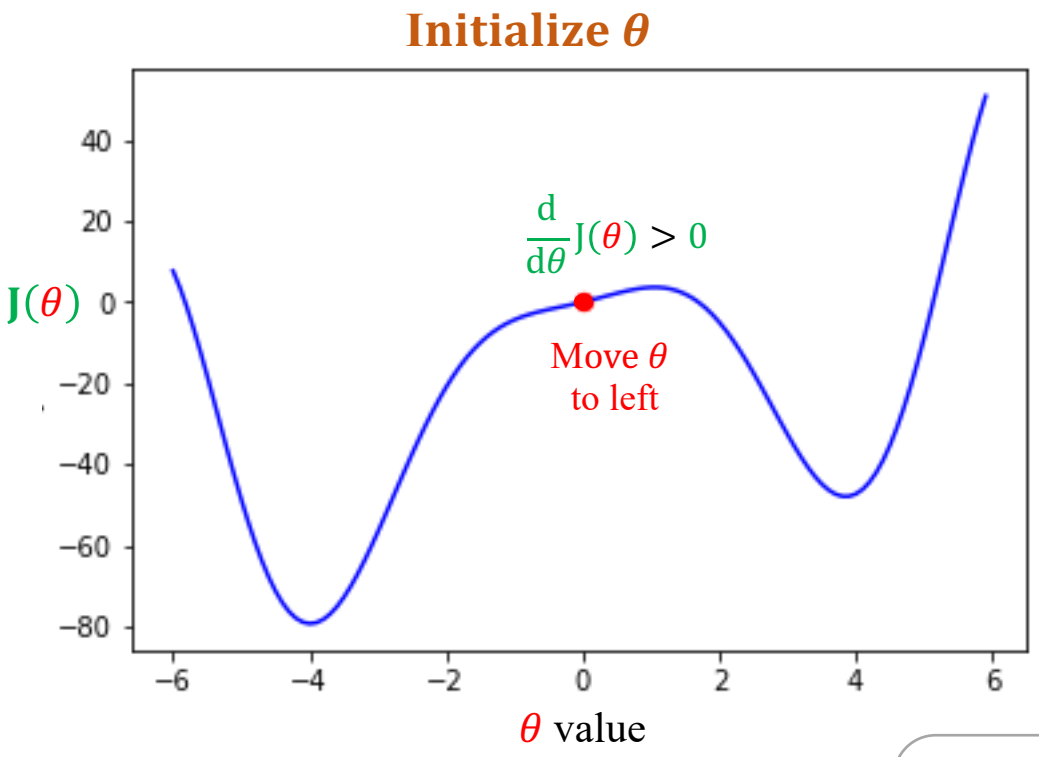
$$x_{new} = x_{old} - \eta \frac{d}{dx} J(x_{old})$$

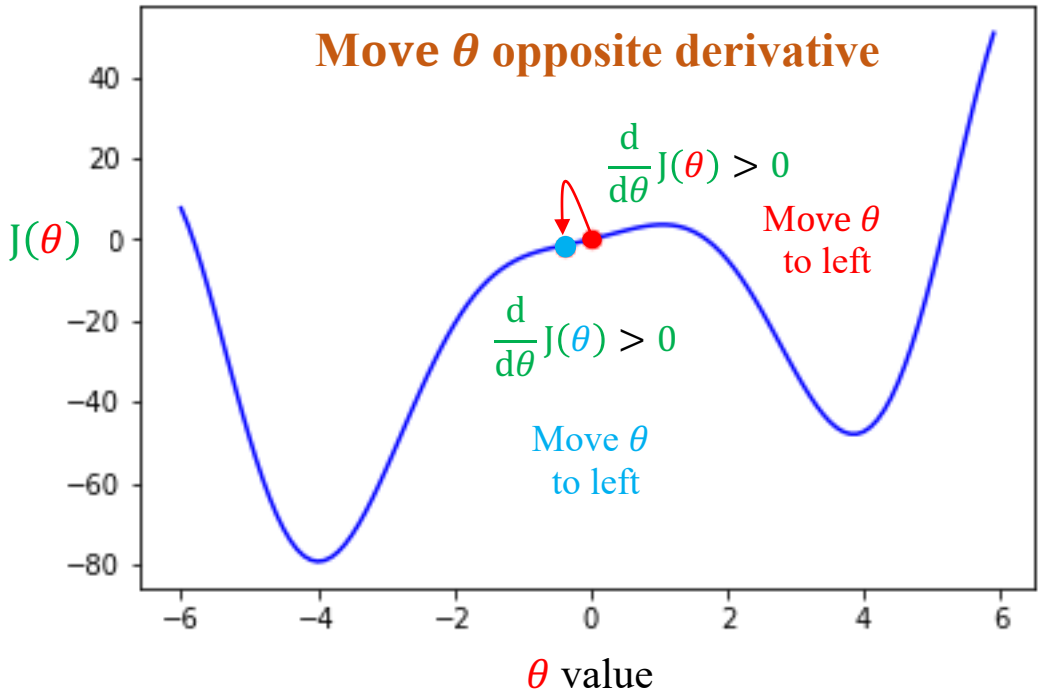
Derivative at x_{old}

learning rate

Understanding the GB Loss

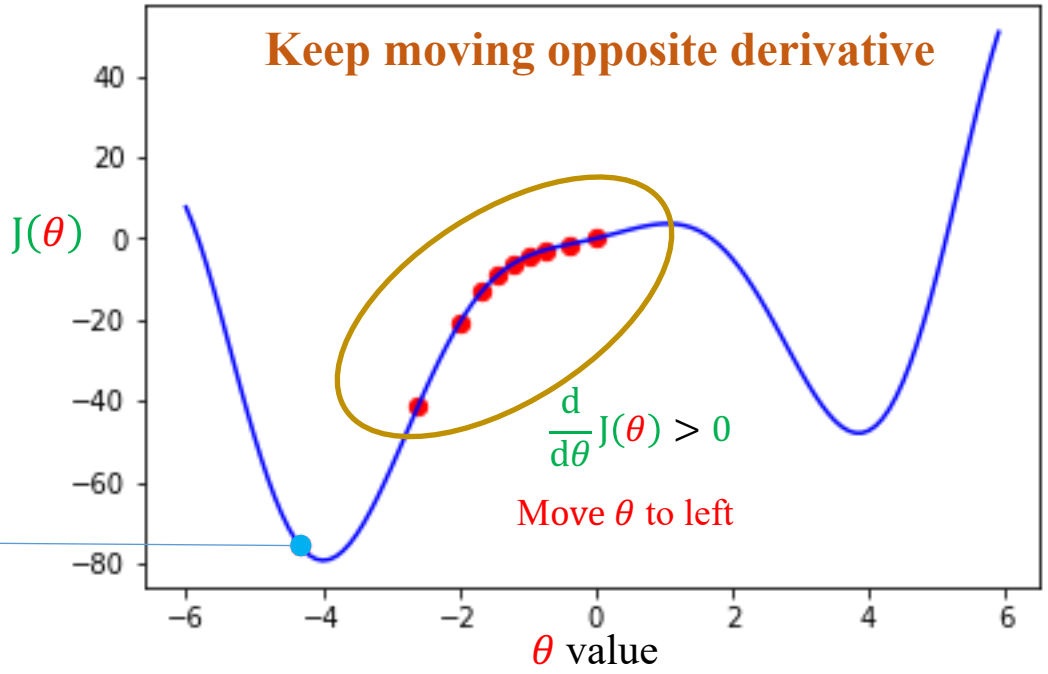
❖ Idea of gradient descent





$\frac{d}{d\theta}J(\theta) < 0$

Move θ to right



Discussion

Outline

SECTION 1

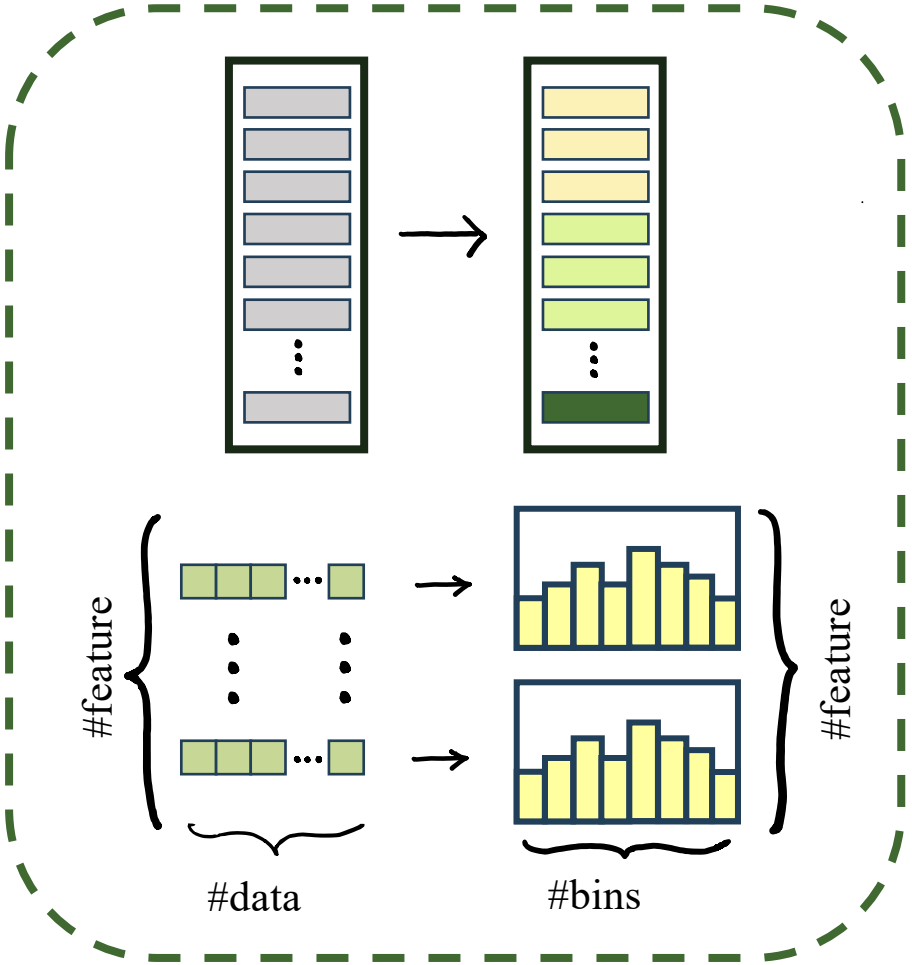
Discussion&Motivation

SECTION 2

Improvements

SECTION 3

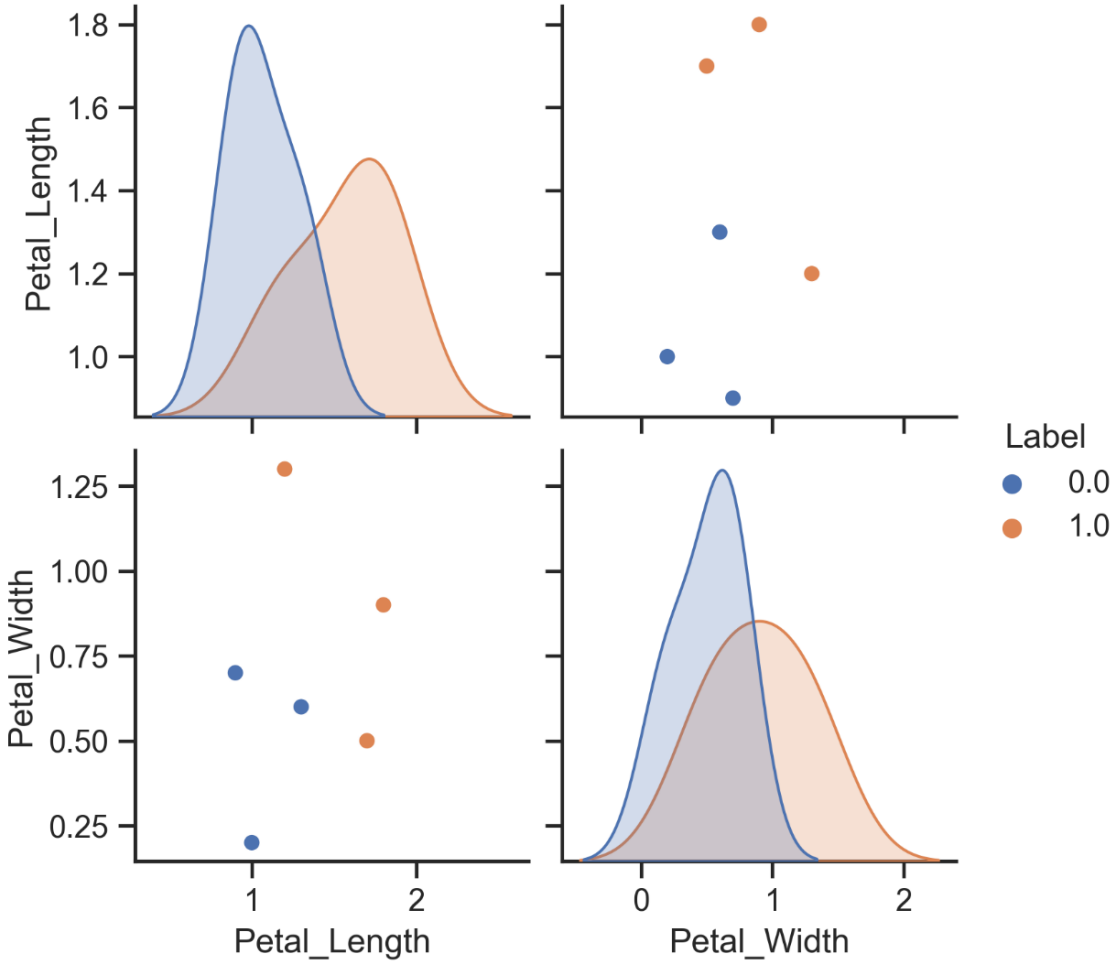
Case Studies

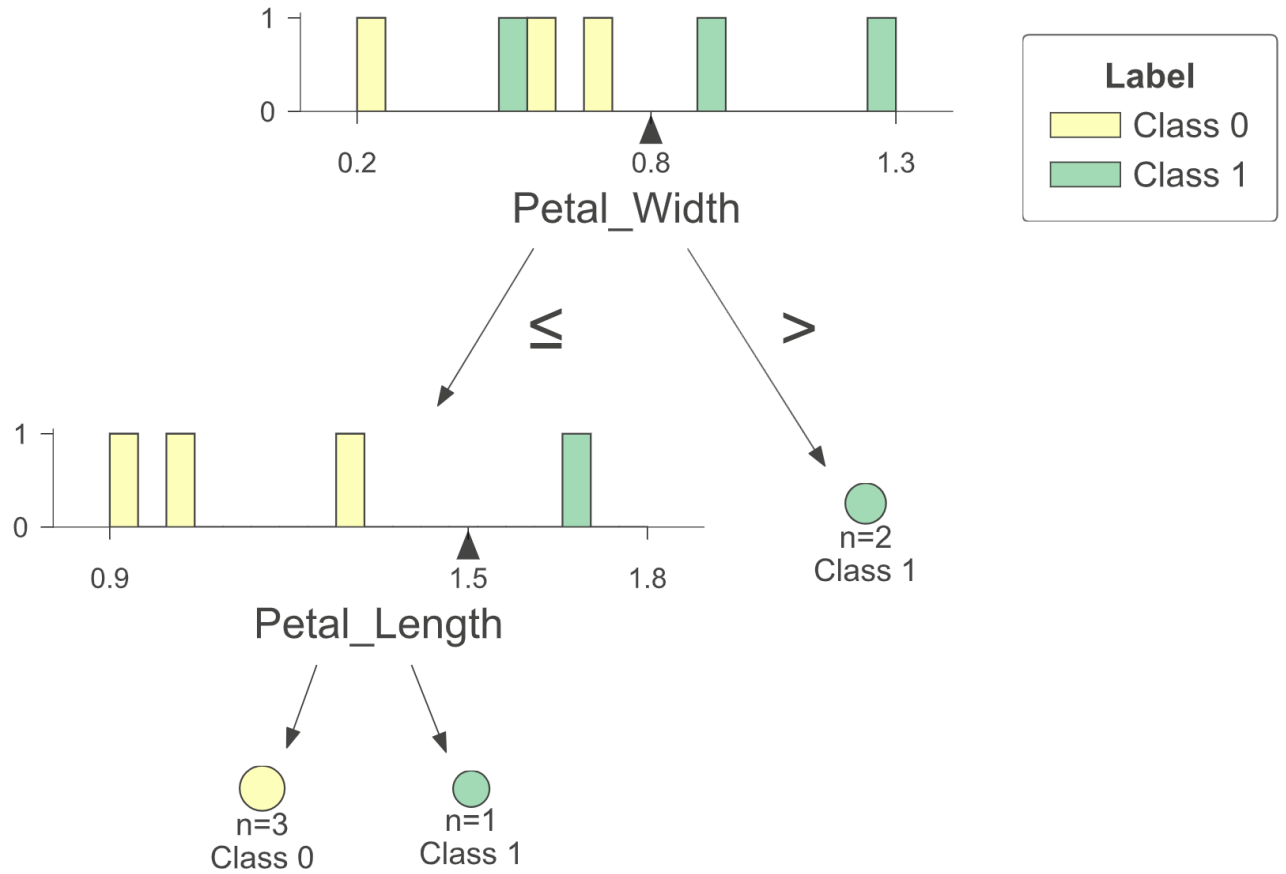


DT for Classification

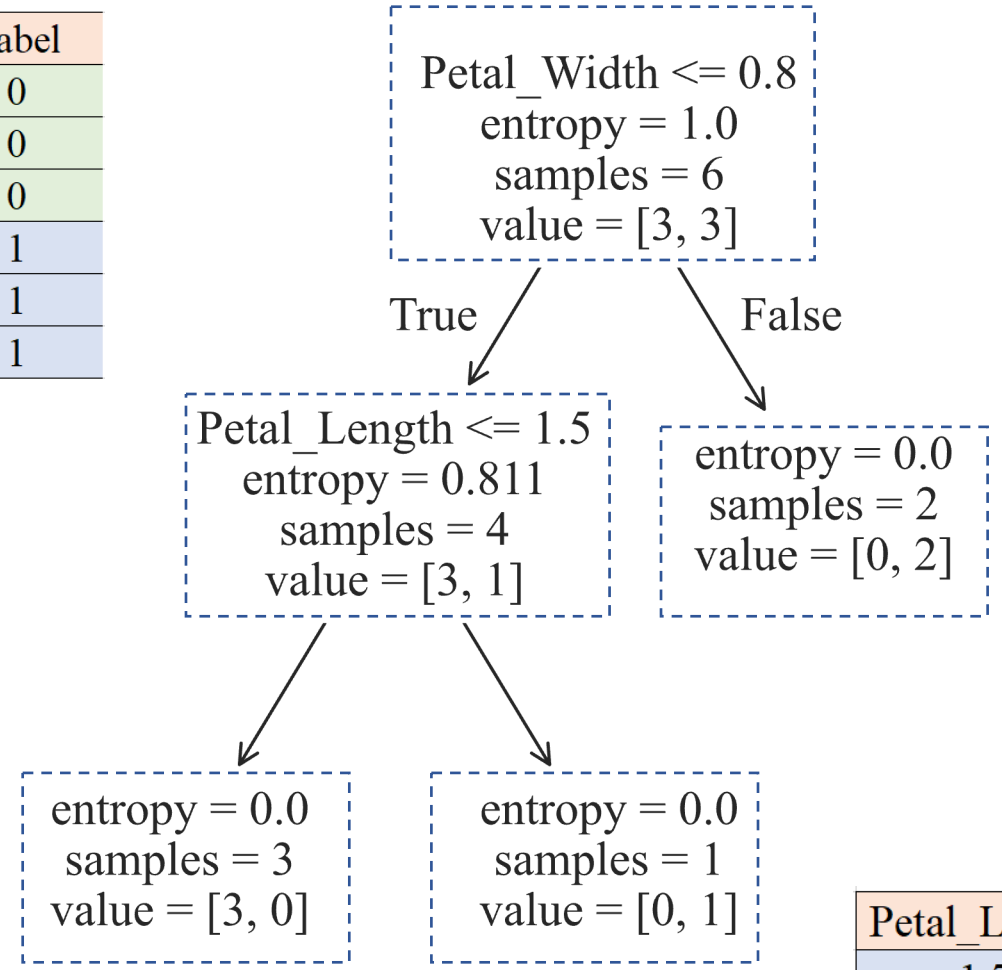
❖ Simple IRIS

Petal_Length	Petal_Width	Label
1	0.2	0
1.3	0.6	0
0.9	0.7	0
1.7	0.5	1
1.8	0.9	1
1.2	1.3	1





Petal_Length	Petal_Width	Label
1	0.2	0
1.3	0.6	0
0.9	0.7	0
1.7	0.5	1
1.8	0.9	1
1.2	1.3	1




```
|--- feature_1 <= 0.80
|   |--- feature_0 <= 1.50
|   |   |--- class: 0
|   |--- feature_0 >  1.50
|   |   |--- class: 1
|--- feature_1 >  0.80
|   |--- class: 1
```

Petal_Length	Petal_Width	Label
1	0.2	0
1.3	0.6	0
0.9	0.7	0
1.7	0.5	1

Petal_Length	Petal_Width	Label
1.8	0.9	1
1.2	1.3	1

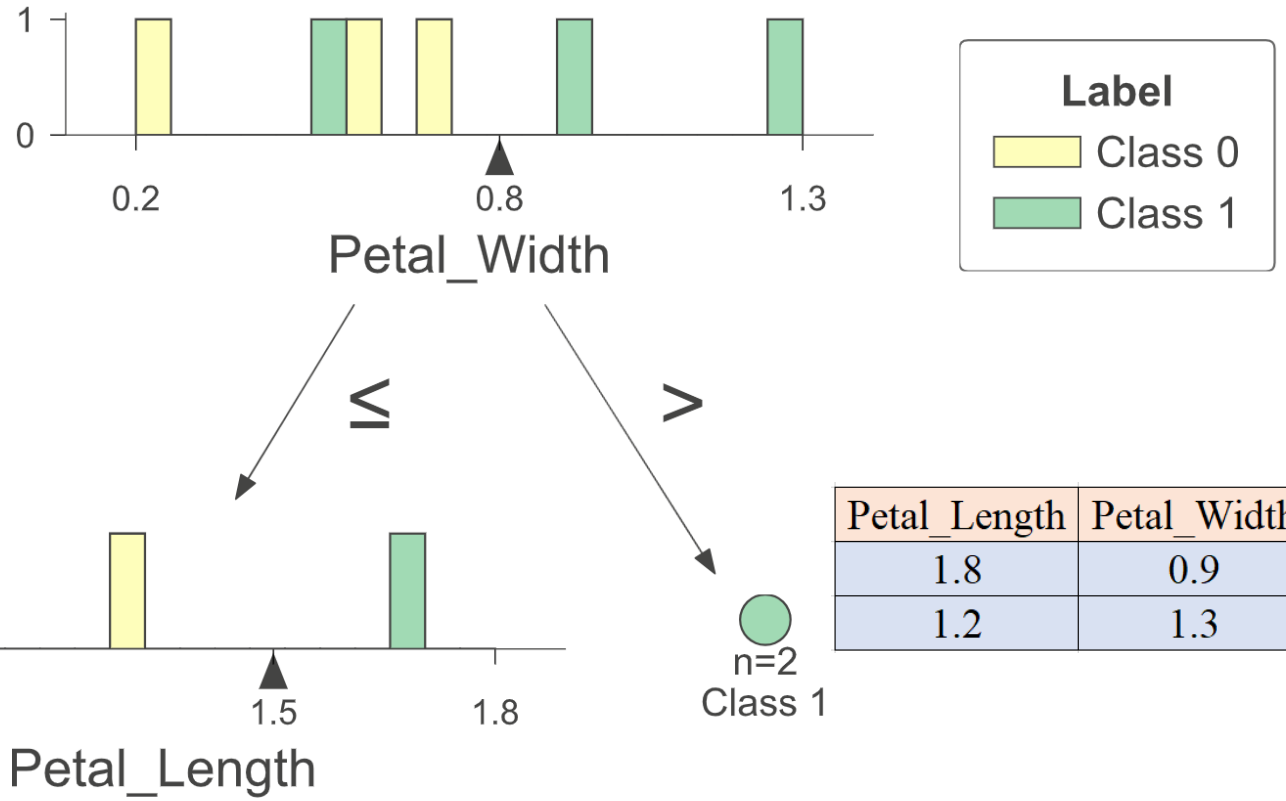
Petal_Length	Petal_Width	Label
1.7	0.5	1

Petal_Length	Petal_Width	Label
1	0.2	0
1.3	0.6	0
0.9	0.7	0

DT for Classification

Petal_Length	Petal_Width	Label
1	0.2	0
1.3	0.6	0
0.9	0.7	0
1.7	0.5	1
1.8	0.9	1
1.2	1.3	1

Simple IRIS



Petal_Length	Petal_Width	Label
1	0.2	0
1.3	0.6	0
0.9	0.7	0
1.7	0.5	1

Petal_Length	Petal_Width	Label
1.8	0.9	1
1.2	1.3	1

Petal_Length	Petal_Width	Label
1	0.2	0
1.3	0.6	0
0.9	0.7	0

Petal_Length	Petal_Width	Label
1.7	0.5	1

DT for Classification

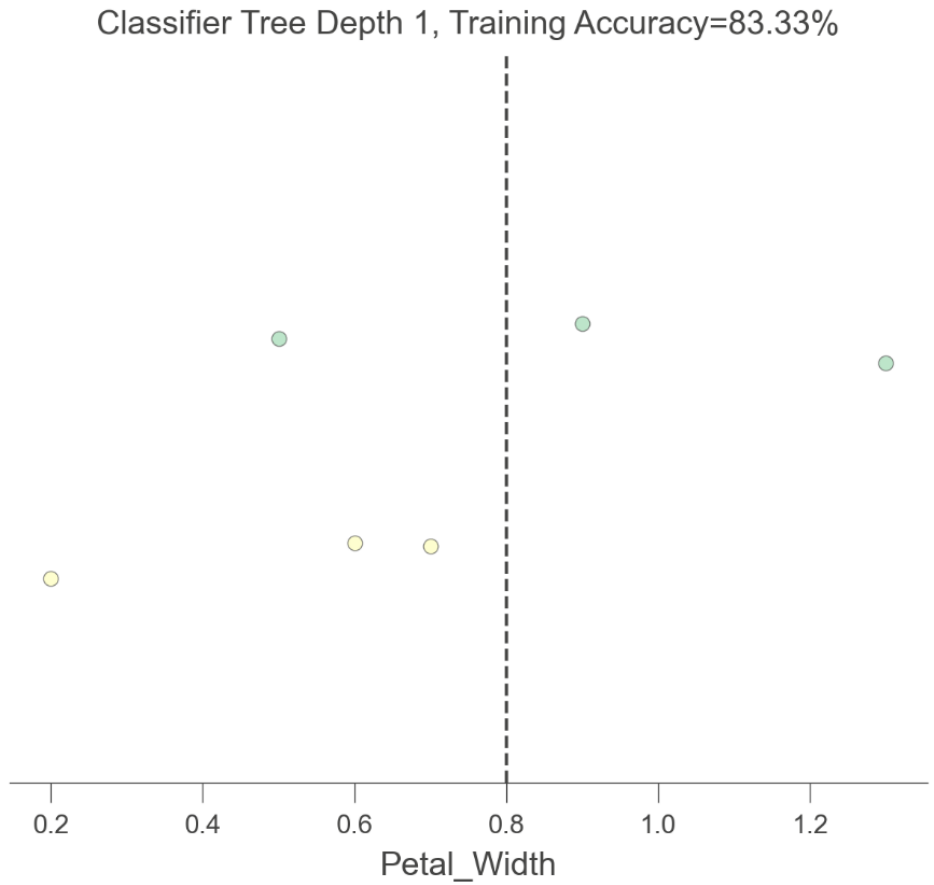
❖ Simple IRIS

Petal_Length	Petal_Width	Label
1	0.2	0
1.3	0.6	0
0.9	0.7	0
1.7	0.5	1
1.8	0.9	1
1.2	1.3	1

0.8

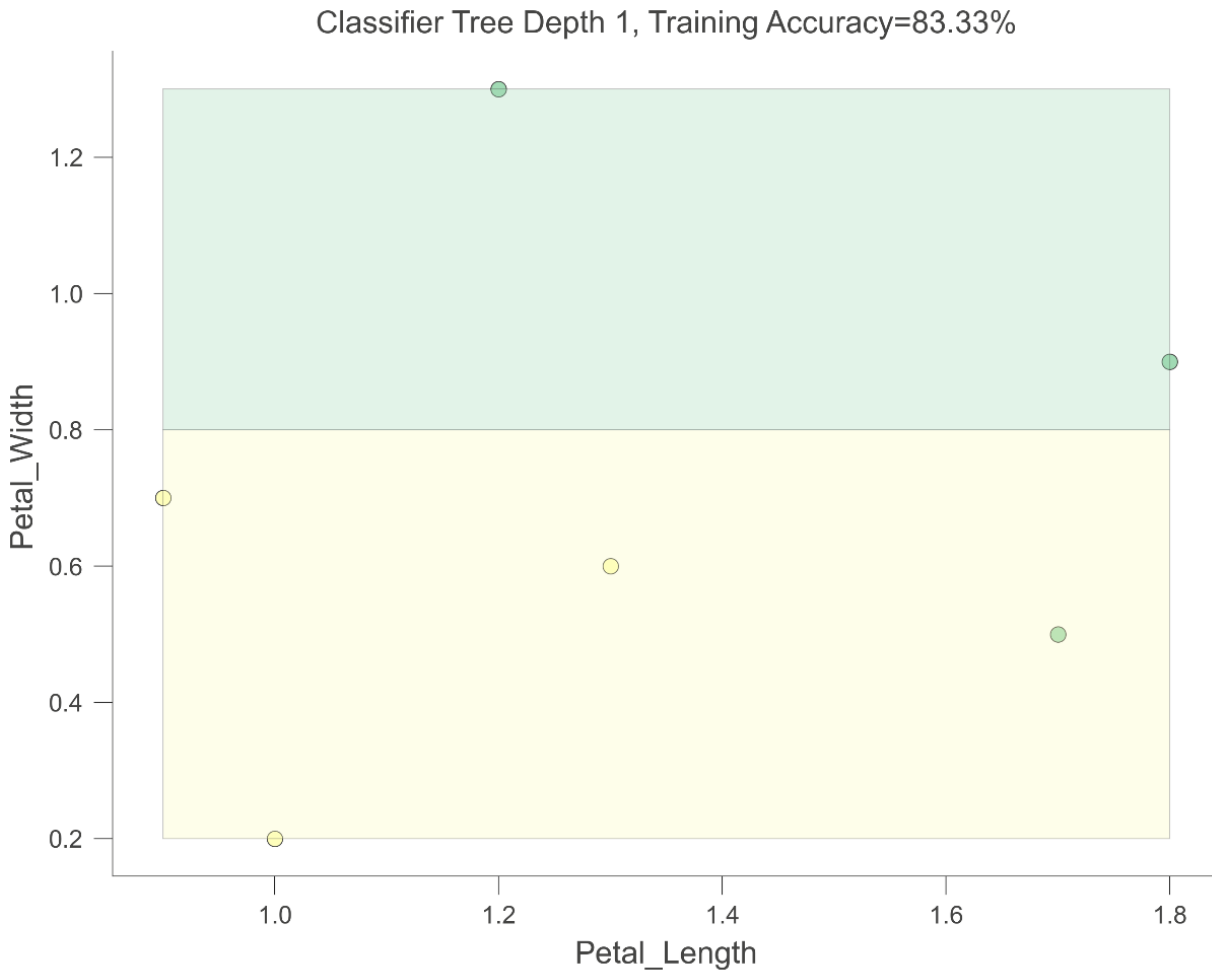
Petal_Length	Petal_Width	Label
1	0.2	0
1.3	0.6	0
0.9	0.7	0
1.7	0.5	1

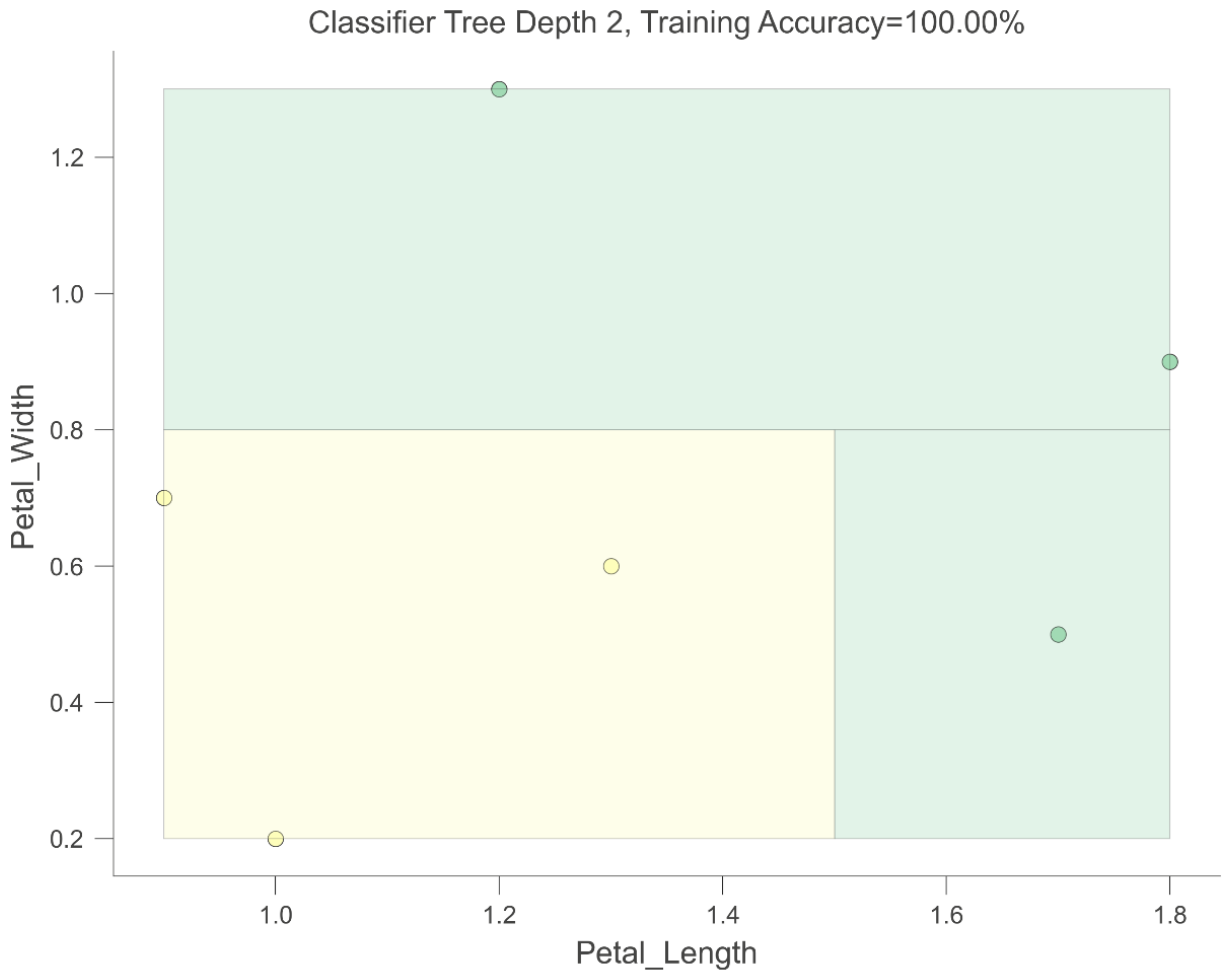
Petal_Length	Petal_Width	Label
1.8	0.9	1
1.2	1.3	1



DT for Classification

❖ Simple IRIS





❖ Example: Step-by-Step

Entropy:	Information Gain
$E(S) = -\sum_{c \in C} p_c log_2 p_c$	$IG(S,F) = E(S) - \sum_{f \in F} \frac{\lvert S_f \rvert}{\lvert S \rvert} E(S_f)$

Petal_Length	Petal_Width	Label
1	0.2	0
1.3	0.6	0
0.9	0.7	0
1.7	0.5	1
1.8	0.9	1
1.2	1.3	1

Root Entropy

$S = \{\text{Label 0: 3, Label 1: 3}\}$

$p_0 = \frac{3}{6} = 0.5, \qquad p_1 = \frac{3}{6} = 0.5$

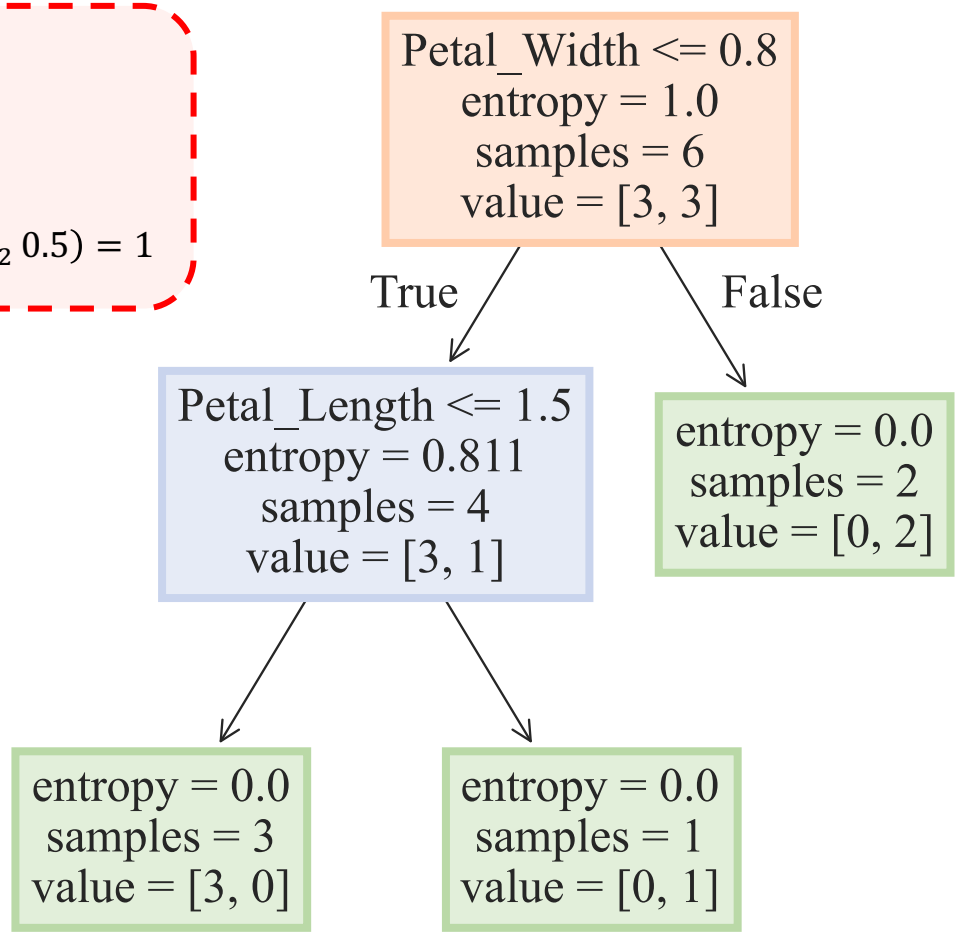
$E(S) = -(0.5 \log_2 0.5 + 0.5 \log_2 0.5) = 1$

Petal_Length	
0.9	0.95
1	1.10
1.2	1.25
1.3	1.50
1.7	1.75
1.8	

Ascending sorting

Petal_Width	
0.2	0.35
0.5	0.55
0.6	0.65
0.7	0.80
0.9	1.1
1.3	

Ascending sorting



❖ Example: Step-by-Step

Entropy:	Information Gain
$E(S) = -\sum_{c \in C} p_c log_2 p_c$	$IG(S,F) = E(S) - \sum_{f \in F} \frac{\lvert S_f \rvert}{\lvert S \rvert} E(S_f)$

Petal_Length	Label
0.9	0
1	0
1.2	1
1.3	0
1.7	1
1.8	1

0.95 | 1.10 | 1.25 | 1.50 | 1.75

Threshold (t)	E_{True} ($\leq$ t)	E_{False} ($>$ t)	E_{Split}	IG
0.95	0.0	0.97	0.81	0.19
1.10	0.0	0.81	0.54	**0.46**
1.25	0.92	0.92	0.92	0.08
1.50	0.81	0.0	0.54	**0.46**
1.75	0.97	0.0	0.81	0.19

$t \leq 0.95$

True (t≤0.95)

Petal_Length	Label
0.9	0

$S_{t \leq 0.95} = \{\text{Class 0: 1, Class 1: 0}\}$

$\Rightarrow p_0 = \frac{1}{1} = 1, p_1 = \frac{0}{1} = 0$

$E(S_{t \leq 0.95}) = -(1 \cdot \log_2 1 + 0 \cdot \log_2 0) = 0$

False (t>0.95)

Petal_Length	Label
1	0
1.2	1
1.3	0
1.7	1
1.8	1

$S_{t>0.95} = \{\text{Class 0: 2, Class 1: 3}\}$

$\Rightarrow p_0 = \frac{2}{5} = 0.4, p_1 = \frac{3}{5} = 0.6$

$E(S_{t>0.95}) = -\left(\frac{2}{5}\log_2\frac{2}{5} + \frac{3}{5}\log_2\frac{3}{5}\right) = 0.97$

$$E_{split} = \frac{1}{6} \cdot 0 + \frac{5}{6} \cdot 0.97 = 0.81$$

$$IG = 1 - 0.81 = 0.19$$

❖ Example: Step-by-Step

Entropy:	Information Gain
$E(S) = -\sum_{c \in C} p_c log_2 p_c$	$IG(S,F) = E(S) - \sum_{f \in F} \frac{\lvert S_f \rvert}{\lvert S \rvert} E(S_f)$

Petal_Width	Label
0.2	0
0.5	1
0.6	0
0.7	0
0.9	1
1.3	1

0.35 | 0.55 | 0.65 | 0.80 | 1.1

Threshold (t)	E_{True} ($\leq$ t)	E_{False} ($>$ t)	E_{Split}	IG
0.35	0.0	0.97	0.81	0.19
0.55	1.0	1.0	1.0	0.0
0.65	0.92	0.92	0.92	0.08
0.80	0.81	0.0	0.54	**0.46**
1.10	0.97	0.0	0.81	0.19

Choose threshold with highest IG

$t \leq 0.35$

True (t≤0.35)

Petal_Width	Label
0.2	0

$S_{t \leq 0.35} = \{\text{Class 0: 1, Class 1: 0}\}$

$\Rightarrow p_0 = \frac{1}{1} = 1, p_1 = \frac{0}{1} = 0$

$E(S_{t \leq 0.35}) = -(1 \cdot \log_2 1 + 0 \cdot \log_2 0) = 0$

False (t>0.35)

Petal_Width	Label
0.5	1
0.6	0
0.7	0
0.9	1
1.3	1

$S_{t>0.35} = \{\text{Class 0: 2, Class 1: 3}\}$

$\Rightarrow p_0 = \frac{2}{5} = 0.4, p_1 = \frac{3}{5} = 0.6$

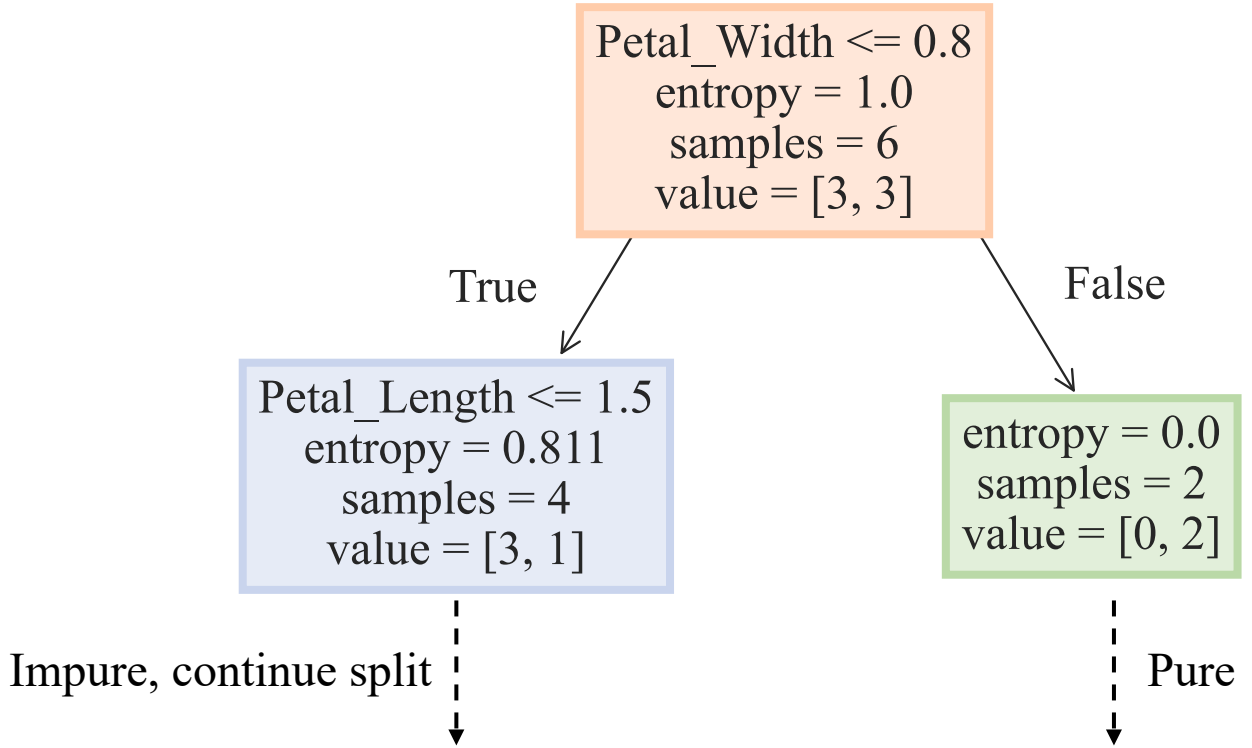
$E(S_{t>0.35}) = -\left(\frac{2}{5}\log_2\frac{2}{5} + \frac{3}{5}\log_2\frac{3}{5}\right) = 0.97$

$$E_{split} = \frac{1}{6} \cdot 0 + \frac{5}{6} \cdot 0.97 = 0.81$$

$$IG = 1 - 0.81 = 0.19$$

❖ Example: Step-by-Step

Petal_Length	Petal_Width	Label
1	0.2	0
1.3	0.6	0
0.9	0.7	0
1.7	0.5	1
1.8	0.9	1
1.2	1.3	1



$S = \{\text{Label 0: 3, Label 1: 1}\}$

$$p_0 = \frac{3}{4} = 0.75, \qquad p_1 = \frac{1}{4} = 0.25$$

$$E(S) = -(0.75 \log_2 0.75 + 0.25 \log_2 0.25) = 0.81$$

Petal_Length	Petal_Width	Label
1	0.2	0
1.3	0.6	0
0.9	0.7	0
1.7	0.5	1

Petal_Length	Petal_Width	Label
1.8	0.9	1
1.2	1.3	1

Root Entropy

❖ Example: Step-by-Step

Entropy:	Information Gain
$E(S) = -\sum_{c \in C} p_c log_2 p_c$	$IG(S,F) = E(S) - \sum_{f \in F} \frac{\lvert S_f \rvert}{\lvert S \rvert} E(S_f)$

Ascending sorting

	Petal_Width	Label
0.35	0.2	0
0.55	0.5	1
0.65	0.6	0
	0.7	0

$E(S) = 0.81$

Threshold (t)	E_{True} ($\leq$ t)	E_{False} ($>$ t)	E_{Split}	IG
0.35	0.0	0.92	0.69	0.12
0.55	1.0	0.0	0.50	**0.31**
0.65	0.92	0.0	0.69	0.12

$t \leq 0.35$

True (t≤0.35)

Petal_Width	Label
0.2	0

$S_{t \leq 0.35}$ = {Class 0: 1, Class 1: 0}

$\Rightarrow p_0 = \frac{1}{1} = 1, p_1 = \frac{0}{1} = 0$

$E(S_{t \leq 0.35}) = -(1 \cdot \log_2 1 + 0 \cdot \log_2 0) = 0$

False (t>0.35)

Petal_Width	Label
0.5	1
0.6	0
0.7	0

$S_{t>0.35}$ = {Class 0: 2, Class 1: 1}

$\Rightarrow p_0 = \frac{2}{3}, p_1 = \frac{1}{3}$

$E(S_{t>0.35}) = -\left(\frac{2}{3}\log_2\frac{2}{3} + \frac{1}{3}\log_2\frac{1}{3}\right) = 0.92$

$$E_{split} = \frac{1}{4} \cdot 0 + \frac{3}{4} \cdot 0.92 = 0.69$$

$$IG = 0.81 - 0.69 = 0.12$$

❖ Example: Step-by-Step

Entropy:	Information Gain
$E(S) = -\sum_{c \in C} p_c log_2 p_c$	$IG(S,F) = E(S) - \sum_{f \in F} \frac{\lvert S_f \rvert}{\lvert S \rvert} E(S_f)$

Ascending sorting

	Petal_Length	Label
0.95	0.9	0
1.15	1	0
1.30	1.3	0
1.50	1.7	1

$E(S) = 0.81$

Threshold (t)	E_{True} ($\leq$ t)	E_{False} ($>$ t)	E_{Split}	IG
0.95	0.0	0.918	0.69	0.12
1.15	0.0	1.0	0.50	0.31
1.50	0.0	0.0	0.69	**0.81**

Choose threshold with highest IG for splitting

$t \leq 0.95$

True (t≤0.95)

Petal_Width	Label
0.9	0

$S_{t \leq 0.95}$ = {Class 0: 1, Class 1: 0}

$\Rightarrow p_0 = \frac{1}{1} = 1, p_1 = \frac{0}{1} = 0$

$E(S_{t \leq 0.95}) = -(1 \cdot \log_2 1 + 0 \cdot \log_2 0) = 0$

False (t>0.95)

Petal_Width	Label
1	0
1.3	0
1.7	1

$S_{t>0.95}$ = {Class 0: 2, Class 1: 1}

$\Rightarrow p_0 = \frac{2}{3}, p_1 = \frac{1}{3}$

$E(S_{t>0.95}) = -\left(\frac{2}{3}\log_2\frac{2}{3} + \frac{1}{3}\log_2\frac{1}{3}\right) = 0.92$

$$E_{split} = \frac{1}{4} \cdot 0 + \frac{3}{4} \cdot 0.92 = 0.69$$

$$IG = 0.81 - 0.69 = 0.12$$

❖ Example: Step-by-Step

	Petal_Width	Label
0.35	0.2	0
0.55	0.5	1
0.65	0.6	0
0.80	0.7	0
1.1	0.9	1
	1.3	1

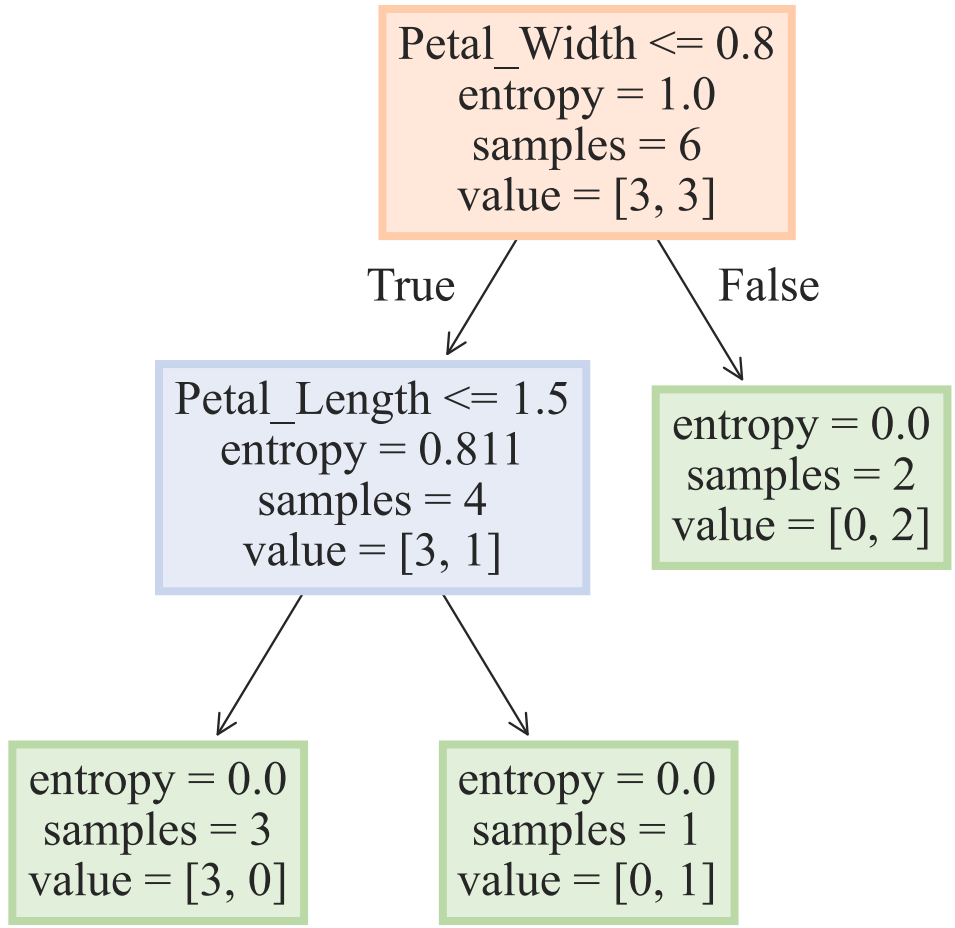
$E(S) = 1.0$

Threshold (t)	E_{Split}	IG
0.35	0.81	0.19
0.55	1.0	0.0
0.65	0.92	0.08
0.80	0.54	**0.46**
1.10	0.81	0.19

	Petal_Length	Label
0.95	0.9	0
1.15	1	0
1.50	1.3	0
	1.7	1

$E(S) = 0.81$

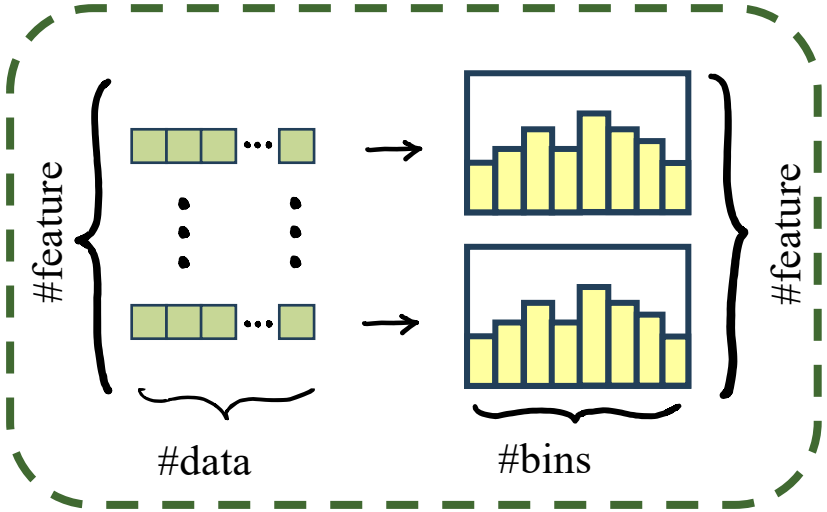
Threshold (t)	E_{Split}	IG
0.95	0.69	0.12
1.15	0.50	0.31
1.50	0.69	**0.81**



Histogram-based Threshold

When we have a large number of samples, so ...

Histogram-based Algorithm



Petal_Length	Petal_Width	Label
1	0.2	0
1.3	0.6	0
0.9	0.7	0
1.7	0.5	1
1.8	0.9	1
1.2	1.3	1

Petal_Length	
0.9	0.95
1	1.10
1.2	1.25
1.3	1.50
1.7	1.75
1.8	

Ascending sorting

Petal_Width	
0.2	0.35
0.5	0.55
0.6	0.65
0.7	0.80
0.9	1.1
1.3	

Ascending sorting

❖ Histogram-based idea

Instead of testing all midpoints between consecutive values (too many),

1. Choose the number of bins B

2. Compute the bin width: $w = \frac{max - min}{B}$

3. Create bin edges and only evaluate splitting thresholds at those edges: $e_k = min + \text{kw},\ k = 0,1,\dots,B$

In vector form: $Edges = [min, min + w, min + 2w, \dots, min + (B-1)w, max]$

4. Evaluate splits only at the internal edges, use thresholds $t \in \{min + w, \dots, min + (B-1)w\}$

Note: exclude $t = min$, $t = max$ and empty bin ⟶ create empty branch, invalid split.

1. Choose the number of bins B (here we use $B = 3$ for each feature): $B = 3$

Petal_Width	
0.2	Min
0.5	
0.6	
0.7	
0.9	
1.3	Max

2. $w = \frac{max - min}{B} = \frac{1.3 - 0.2}{3} = 0.367$

3. $Edges = [min, min + w, min + 2w, max]$

$= [0.2, 0.2 + 0.367, 0.2 + 2 \times 0.367, 1.3]$

$= [0.2, 0.567, 0.933, 1.3]$

min, internal edges, max

- Bin1: [0.2, 0.567] = {0.2 , 0.5}
- Bin2: (0.567, 0.933] = {0.6, 0.7, 0.9}
- Bin3: (0.933, 1.3] = {1.3}

Petal_Length	
0.9	Min
1.0	
1.2	
1.3	
1.7	
1.8	Max

2. $w = \frac{max - min}{B} = \frac{1.8 - 0.9}{3} = 0.3$

3. $Edges = [0.9, 1.2, 1.5, 1.8]$

min, internal edges, max

- Bin1: [0.9, 1.2] = {0.9 , 1.0 , 1.2}
- Bin2: (1.2, 1.5] = {1.3}
- Bin3: (1.5, 1.8] = {1.7, 1.8}

4. Evaluate splits only at the internal edges, use thresholds $t \in \{min + w, \dots, min + (B-1)w\}$

Note: exclude $t = min$, $t = max$ and empty bin ⟶ create empty branch, invalid split.

Entropy:	Information Gain
$E(S) = -\sum_{c \in C} p_c log_2 p_c$	$IG(S,F) = E(S) - \sum_{f \in F} \frac{\lvert S_f \rvert}{\lvert S \rvert} E(S_f)$

Petal_Width	
0.2	Min
0.5	0.567
0.6	
0.7	
0.9	0.933
1.3	Max

$$Edges = [0.2, 0.567, 0.933, 1.3]$$

Min — Internal edges — Max

⇒ Evaluate thresholds $t \in \{0.567, 0.933\}$

- Bin1: [0.2, 0.567] = {0.2 , 0.5}
- Bin2: (0.567, 0.933] = {0.6, 0.7, 0.9}
- Bin3: (0.933, 1.3] = {1.3}

Threshold	E_{True} (≤ t)	E_{False}(> t)	E_{Split}	IG
0.567	1.0	1.0	1.0	0.0
0.933	0.971	0.0	0.81	**0.19**

$t \leq 0.567$

True (t≤0.567)

Petal_Width	Label
0.2	0
0.5	1

$S_{t \leq 0.567}$ = {Class 0: 1, Class 1: 1}

$\Rightarrow p_0 = \frac{1}{2} = 0.5, p_1 = \frac{1}{2} = 0.5$

$E(S_{t \leq 0.567}) = -(0.5 \cdot \log_2 0.5 + 0.5 \cdot \log_2 0.5) = 1$

False (t>0.567)

Petal_Width	Label
0.6	0
0.7	0
0.9	1
1.3	1

$S_{t>0.567}$ = {Class 0: 2, Class 1: 2}

$\Rightarrow p_0 = \frac{2}{4} = 0.5 , p_1 = \frac{2}{4} = 0.5$

$E(S_{t>0.567}) = -(0.5 \cdot \log_2 0.5 + 0.5 \cdot \log_2 0.5) = 1$

$$E_{split} = \frac{2}{6} \cdot 1 + \frac{4}{6} \cdot 1 = 1$$

$$IG = 1 - 1 = 0$$

4. Evaluate splits only at the internal edges, use thresholds $t \in \{min + w, \dots, min + (B-1)w\}$

Note: exclude $t = min$, $t = max$ and empty bin ⟶ create empty branch, invalid split.

Entropy:	Information Gain
$E(S) = -\sum_{c \in C} p_c log_2 p_c$	$IG(S,F) = E(S) - \sum_{f \in F} \frac{\lvert S_f \rvert}{\lvert S \rvert} E(S_f)$

Petal_Length	
0.9 Min	
1.0	
1.2	1.2
1.3	1.5
1.7	
1.8 Max	

$$Edges = [0.9, 1.2, 1.5, 1.8]$$

Min — Internal edges — Max

⇒ Evaluate thresholds $t \in \{1.2, 1.5\}$

- Bin1: [0.9, 1.2] = {0.9 , 1.0 , 1.2}
- Bin2: (1.2, 1.5] = {1.3}
- Bin3: (1.5, 1.8] = {1.7, 1.8}

Threshold	E_{True} ($\leq$ t)	E_{False} ($>$ t)	E_{Split}	IG
1.2	0.92	0.92	0.92	0.08
1.5	0.81	0.0	0.54	**0.46**

Choose threshold with highest IG for spliting

$t \leq 1.2$

True (t$\leq$ 1.2)

Petal_Length	Label
0.9	0
1.0	0
1.2	1

$S_{t \leq 1.2} = \{\text{Class 0: 2, Class 1: 1}\}$

$\Rightarrow p_0 = \frac{2}{3}, p_1 = \frac{1}{3}$

$$E(S_{t \leq 1.2}) = -\left(\frac{2}{3} \cdot \log_2 \frac{2}{3} + \frac{1}{3} \cdot \log_2 \frac{1}{3}\right) = 0.92$$

False (t> 1.2)

Petal_Length	Label
1.3	0
1.7	1
1.8	1

$S_{t > 1.2} = \{\text{Class 0: 1, Class 1: 2}\}$

$\Rightarrow p_0 = \frac{1}{3}, p_1 = \frac{2}{3}$

$$E(S_{t > 1.2}) = -\left(\frac{1}{3} \cdot \log_2 \frac{1}{3} + \frac{2}{3} \cdot \log_2 \frac{2}{3}\right) = 0.92$$

$$E_{split} = \frac{3}{6} \cdot 0.92 + \frac{3}{6} \cdot 0.92 = 0.92$$

$$IG = 1 - 0.92 = 0.08$$

❖ Example: Step-by-Step

Petal_Length	Petal_Width	Label
1	0.2	0
1.3	0.6	0
0.9	0.7	0
1.7	0.5	1
1.8	0.9	1
1.2	1.3	1

Petal_Width (t)	E_{Split}	IG
0.567	1.0	0.0
0.933	0.81	**0.19**

Petal_Length (t)	E_{Split}	IG
1.2	0.92	0.08
1.5	0.54	**0.46**

Choose threshold with highest IG for splitting

Petal_Length<=1.5
entropy = 1.0
samples = 6
value = [3, 3]

True

False

Petal_Width<=0.93
entropy = 0.811
samples = 4
value = [3, 1]

entropy = 0.0
samples = 2
value = [0, 2]

Pure

Impure, continue split

$$S = \{\text{Label 0: } 3, \text{Label 1: } 1\}$$

$$p_0 = \frac{3}{4} = 0.75, \qquad p_1 = \frac{1}{4} = 0.25$$

$$E(S) = -(0.75 \log_2 0.75 + 0.25 \log_2 0.25) = 0.81$$

Root Entropy

Petal_Length	Petal_Width	Label
1	0.2	0
1.3	0.6	0
0.9	0.7	0
1.2	1.3	1

samples with Petal_Length <=1.5

Note: exclude $t = min$, $t = max$ and empty bin ⟶ create empty branch, invalid split.

Entropy:	Information Gain
$E(S) = -\sum_{c \in C} p_c log_2 p_c$	$IG(S,F) = E(S) - \sum_{f \in F} \frac{\lvert S_f \rvert}{\lvert S \rvert} E(S_f)$

Petal_Length
0.9
1.0
1.2
1.3

1.2

$E(S) = 0.81$

- Bin1: [0.9, 1.2] = {0.9 , 1.0, 1.2}
- Bin2: (1.2, 1.5] = {1.3}
- Bin3: (1.5, 1.8] ⟶ **empty bin (exclude)**

⇒ Evaluate thresholds $t \in \{1.2\}$

Threshold	E_{True} (≤ t)	E_{False}(> t)	E_{Split}	IG
1.2	0.92	0	0.69	0.12

$t \leq 1.2$

True (t≤ 1.2)

Petal_Length	Label
0.9	0
1.0	0
1.2	1

$S_{t \leq 1.2} = \{\text{Class 0: 2, Class 1: 1}\}$

$\Rightarrow p_0 = \frac{2}{3}, p_1 = \frac{1}{3}$

$E(S_{t \leq 1.2}) = -\left(\frac{2}{3} \cdot \log_2 \frac{2}{3} + \frac{1}{3} \cdot \log_2 \frac{1}{3}\right) = 0.92$

False (t> 1.2)

Petal_Length	Label
1.3	0

$S_{t>1.2} = \{\text{Class 0: 1, Class 1: 0}\}$

$E(S_{t>1.2}) = 0$

$$E_{split} = \frac{3}{4} \cdot 0.92 + \frac{1}{4} \cdot 0 = 0.69$$

$$IG = 0.81 - 0.69 = 0.12$$

Note: exclude $t = min$, $t = max$ and empty bin ⟶ create empty branch, invalid split.

Petal_Width
0.2
0.6
0.7
1.3

0.567

0.933

$E(S) = 0.81$

- Bin1: [0.2, 0.567] = {0.2 }
- Bin2: (0.567, 0.933] = {0.6, 0.7}
- Bin3: (0.933, 1.3] = {1.3}

⇒ Evaluate thresholds $t \in \{0.567, 0.933\}$

Entropy:	Information Gain
$E(S) = -\sum_{c \in C} p_c log_2 p_c$	$IG(S,F) = E(S) - \sum_{f \in F} \frac{\lvert S_f \rvert}{\lvert S \rvert} E(S_f)$

Threshold	E_{True} ($\leq$ t)	E_{False} ($>$ t)	E_{Split}	IG
0.567	0	0.92	0.69	0.12
0.933	0	0	0	**0.81**

Choose threshold with highest IG for splitting

$t \leq 0.567$

True (t≤0.567)

Petal_Width	Label
0.2	0

$S_{t \leq 0.567} = \{\text{Class 0: 1, Class 1: 0}\}$

$E(S_{t \leq 0.567}) = 0$

False (t>0.567)

Petal_Width	Label
0.6	0
0.7	0
1.3	1

$S_{t>0.567} = \{\text{Class 0: 2, Class 1: 1}\}$

$\Rightarrow p_0 = \frac{2}{3}, p_1 = \frac{1}{3}$

$E(S_{t>0.567}) = -\left(\frac{2}{3} \cdot \log_2 \frac{2}{3} + \frac{1}{3} \cdot \log_2 \frac{1}{3}\right) = 0.92$

$$E_{split} = \frac{1}{4} \cdot 0 + \frac{3}{4} \cdot 0.92 = 0.69$$

$$IG = 0.81 - 0.69 = 0.12$$

❖ Example: Step-by-Step

Petal_Length	Petal_Width	Label
1	0.2	0
1.3	0.6	0
0.9	0.7	0
1.2	1.3	1

$E(S) = 0.81$

Petal_Length
0.9
1.0
1.2
1.3

1.2

Ascending sorting

Petal_Length (t)	E_{Split}	IG
1.2	0.69	0.12

Petal_Width
0.2
0.6
0.7
1.3

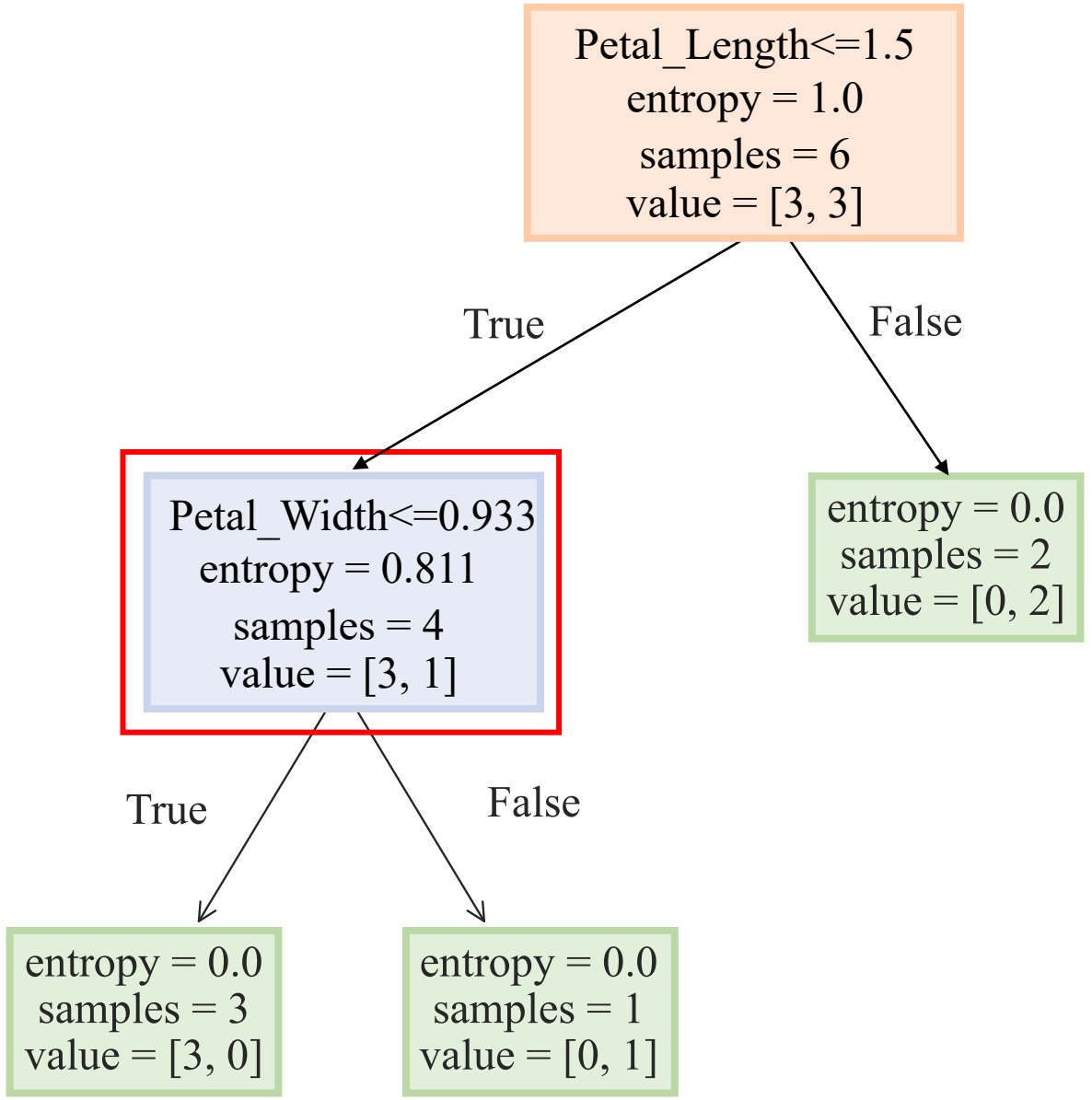
0.567

0.933

Ascending sorting

Petal_Width (t)	E_{Split}	IG
0.567	0.69	0.12
0.933	0	**0.81**

Choose threshold with highest IG for splitting



❖ Example: Step-by-Step

Petal_Length	Label
0.9	0
1.0	0
1.2	1
1.3	0
1.7	1
1.8	1

1.2

1.5

$E(S) = 1.0$

Petal_Length (t)	E_{Split}	IG
1.2	0.92	0.08
1.5	0.54	**0.46**

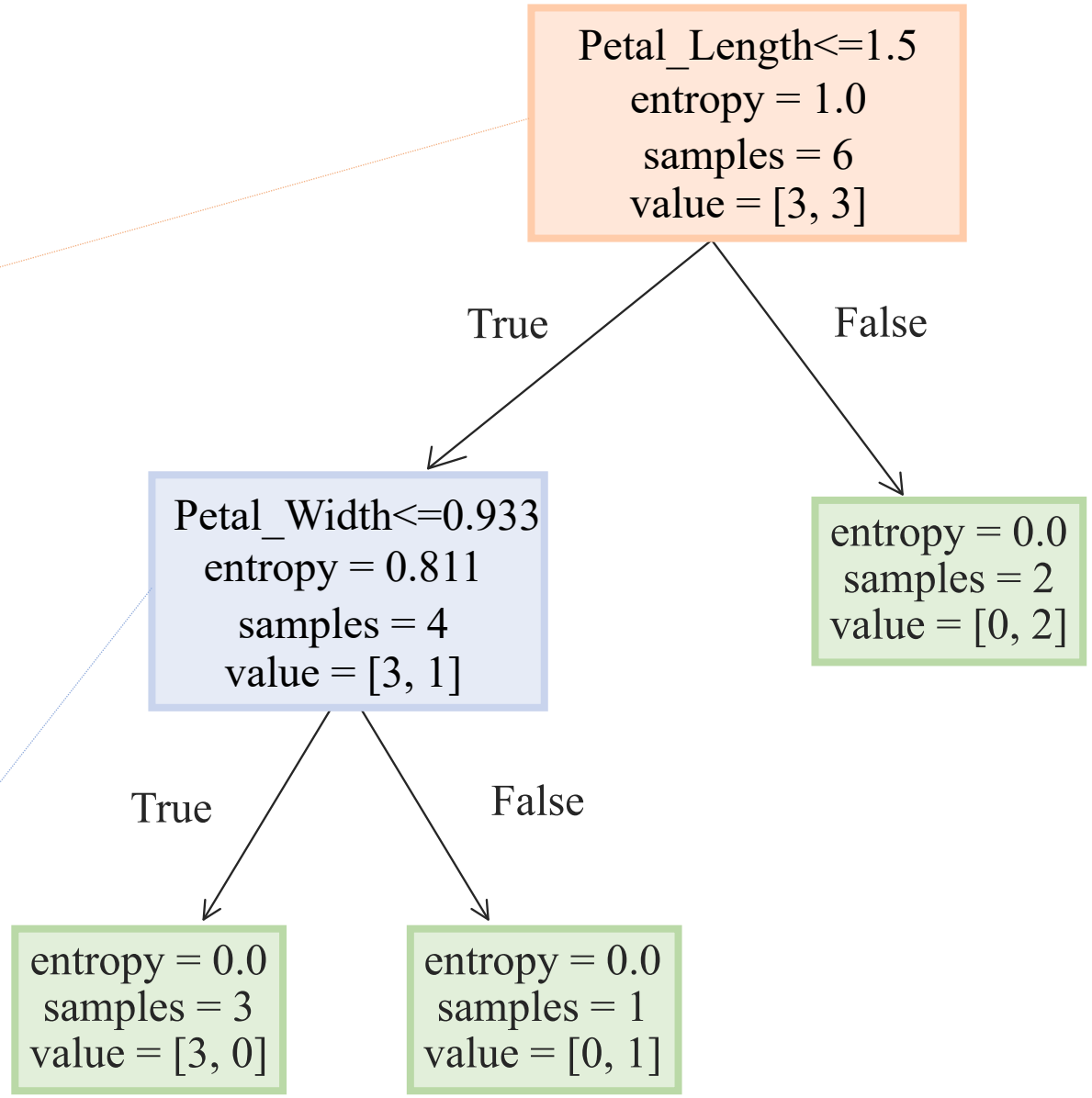
Petal_Width	Label
0.2	0
0.6	0
0.7	0
1.3	1

0.567

0.933

$E(S) = 0.81$

Petal_Width (t)	E_{Split}	IG
0.567	0.69	0.12
0.933	0	**0.81**



Histogram-based Threshold

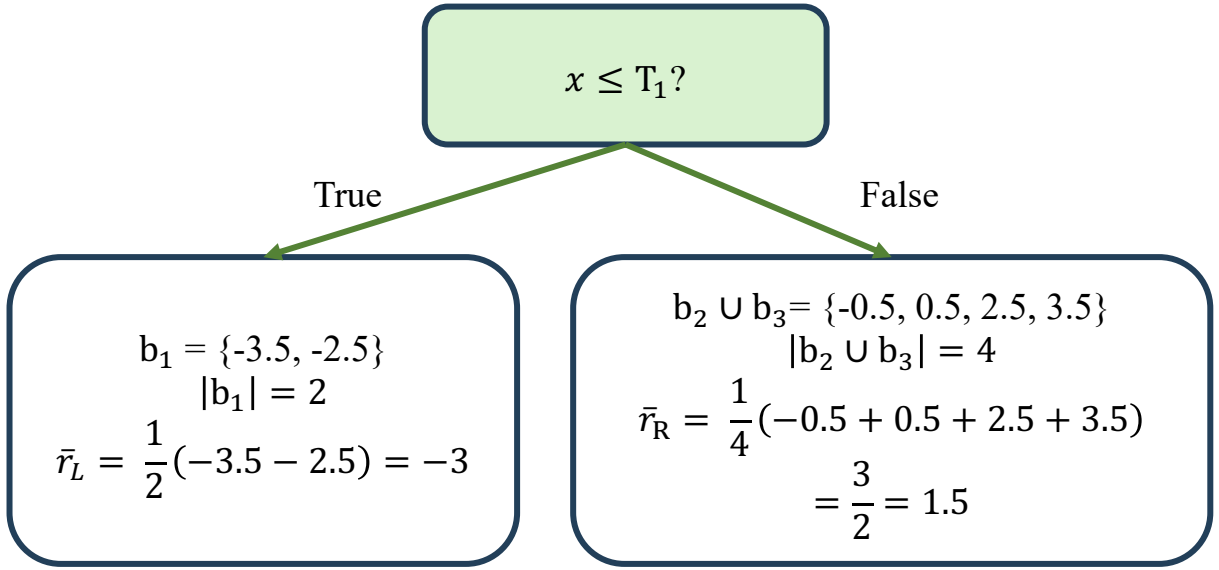
❖ Another example (gradient boosting)

	x	y_{reg}	r_{i1}
	1	2	-3.5
	2	3	-2.5
T_1	3	5	-0.5
	4	6	0.5
T_2	5	8	2.5
	6	9	3.5

$F_0(x) = 5.5$
Then for the 1st boosting round:

Compute pseudo-residuals:
$r_{\text{im}} = [-3.5, -2.5, 0.5, -0.5, 2.5, 3.5]$

Choose the best split threshold



Histogram-based Threshold

❖ Another example (gradient boosting)

	x	y_{reg}	r_{i1}
	1	2	-3.5
	2	3	-2.5
T_1	3	5	-0.5
	4	6	0.5
T_2	5	8	2.5
	6	9	3.5

$$r_{b1} = \frac{-3.5 - 2.5}{2} = -3$$

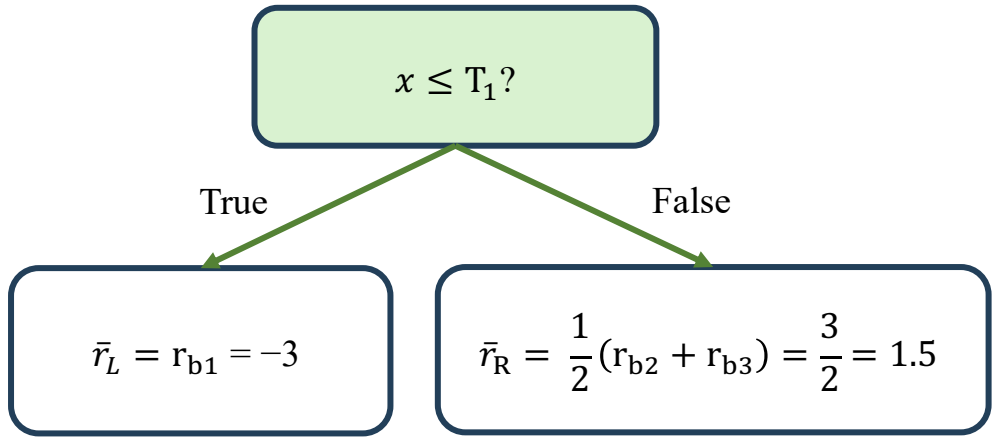
$$r_{b2} = \frac{-0.5 + 0.5}{2} = 0$$

$$r_{b3} = \frac{3.5 + 2.5}{2} = 3$$

$F_0(x) = 5.5$
Then for the 1st boosting round:

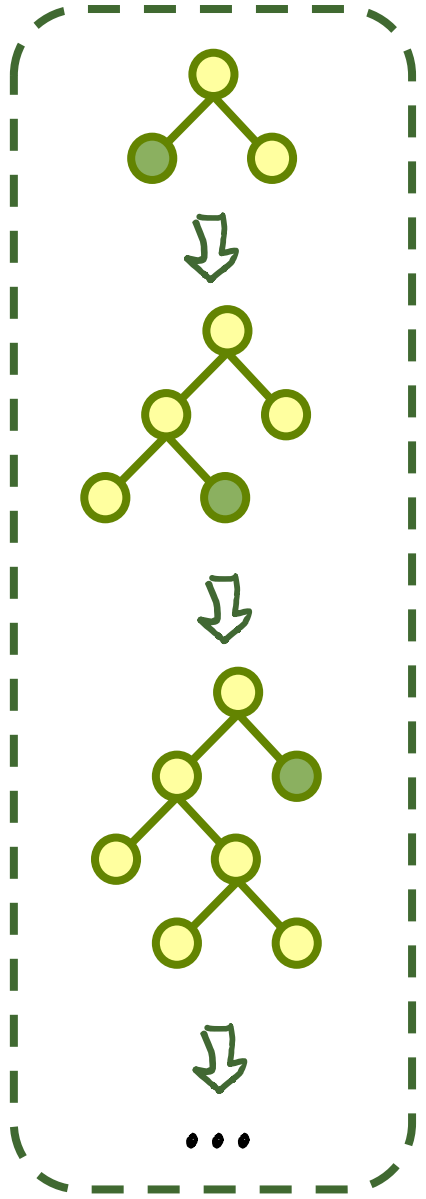
Compute pseudo-residuals:
$r_{bj} = [-3, 0, 3]$

Choose the best split threshold



Leaf-wise Growth

Mở rộng lá có tiềm năng giảm loss lớn nhất bất kể nằm ở mức nào



Level-wise Growth

❖ Salary prediction

Experience	Salary
1	5
1.5	7
2	9
2.5	10
3	60
3.5	64
4	55
4.5	61
5	66
5.5	83
6	93
6.5	91
7	98
7.5	101

When Experience = 5.3,

Salary = ?

Experience	Salary
1	5
1.5	7
2	9
2.5	10
3	60
3.5	64
4	55
4.5	61
5	66
5.5	83
6	93
6.5	91
7	98
7.5	101

$$\mu = \frac{1}{|S|}\sum_i S_i = 57.36$$

$$mse = \frac{1}{|S|}\sum_i (S_i - \mu)^2 = 1187.09$$

Experience	Salary
1	5

$$\mu_L = \frac{1}{|L|}\sum_i L_i = 5$$

$$mse_L = \frac{1}{|L|}\sum_i (L_i - \mu)^2 = 0$$

Experience	Salary
1.5	7
2	9
2.5	10
3	60
3.5	64
4	55
4.5	61
5	66
5.5	83
6	93
6.5	91
7	98
7.5	101

$$a_{mse} = \frac{|L|}{|S|}mse_L + \frac{|R|}{|S|}mse_R$$

$$= \frac{1}{14} * 0 + \frac{13}{14} * 1051.31$$

$$= 976.22$$

$$\mu_R = \frac{1}{|R|}\sum_i R_i = 61.38$$

$$mse_R = \frac{1}{|R|}\sum_i (R_i - \mu)^2 = 1015.31$$

Experience	Salary
1	5
1.5	7
2	9
2.5	10
3	60
3.5	64
4	55
4.5	61
5	66
5.5	83
6	93
6.5	91
7	98
7.5	101

$$\mu = \frac{1}{|S|}\sum_i S_i = 57.36$$

$$mse = \frac{1}{|S|}\sum_i (S_i - \mu)^2 = 1187.09$$

Experience	Salary
1	5
1.5	7
2	9
2.5	10

$$\mu_L = \frac{1}{|L|}\sum_i L_i = \frac{31}{4} = 7.75$$

$$mse_L = \frac{1}{|L|}\sum_i (L_i - \mu)^2 = 3.69$$

Experience	Salary
3	60
3.5	64
4	55
4.5	61
5	66
5.5	83
6	93
6.5	91
7	98
7.5	101

$$a_{mse} = \frac{|L|}{|S|}mse_L + \frac{|R|}{|S|}mse_R$$

$$= \frac{4}{14} * 3.69 + \frac{10}{14} * 282.36$$

$$= 202.74$$

$$\mu_R = \frac{1}{|R|}\sum_i R_i = 77.2$$

$$mse_R = \frac{1}{|R|}\sum_i (R_i - \mu)^2 = 282.36$$

Experience	Salary
1	5
1.5	7
2	9
2.5	10
3	60
3.5	64
4	55
4.5	61
5	66
5.5	83
6	93
6.5	91
7	98
7.5	101

$a_{mse} = 976.22$

$a_{mse} = 747.49$

$a_{mse} = 495.49$

$a_{mse} = 202.74$

$a_{mse} = 335.26$

$a_{mse} = 441.77$

$a_{mse} = 438.67$

$a_{mse} = 451.87$

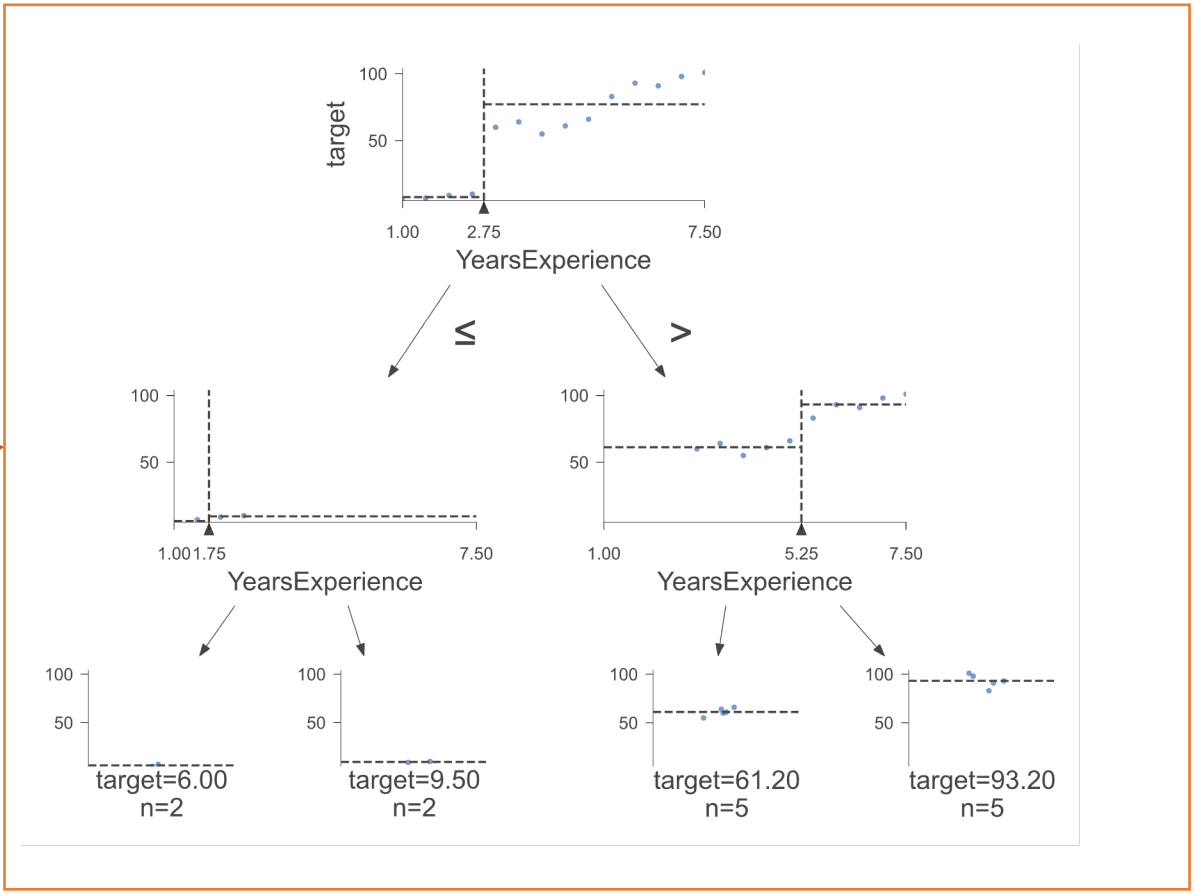
$a_{mse} = 473.36$

$a_{mse} = 597.48$

$a_{mse} = 765.66$

$a_{mse} = 891.08$

$a_{mse} = 1040.57$



Experience	Salary
1	5
1.5	7
2	9
2.5	10

Experience	Salary
3	60
3.5	64
4	55
4.5	61
5	66
5.5	83
6	93
6.5	91
7	98
7.5	101

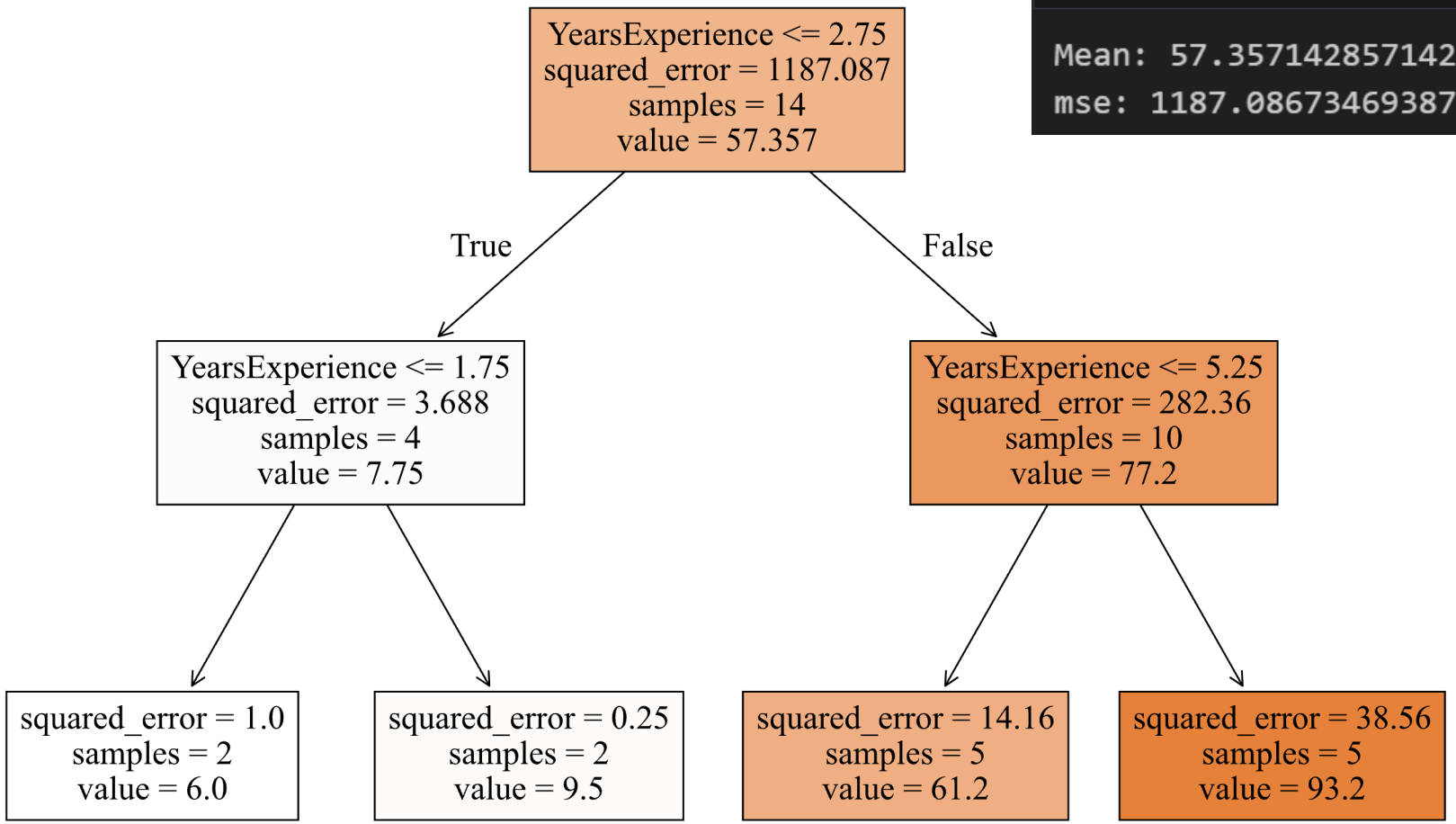
Level-wise Growth

❖ Salary prediction

Experience	Salary
1	5
1.5	7
2	9
2.5	10
3	60
3.5	64
4	55
4.5	61
5	66
5.5	83
6	93
6.5	91
7	98
7.5	101

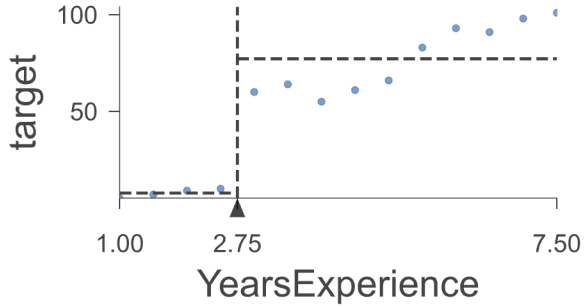
```
y_mean = y.mean()
print('Mean:', y_mean)

diff = (y - y_mean)**2
mse = diff.sum()/14
print('mse:', mse)
```

```
Mean: 57.357142857142854
mse: 1187.0867346938774
```



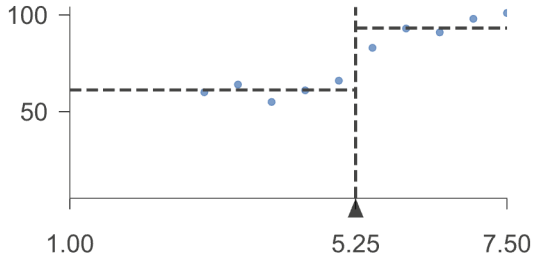
Level-wise Growth

Experience	Salary
1	5
1.5	7
2	9
2.5	10
3	60
3.5	64
4	55
4.5	61
5	66
5.5	83
6	93
6.5	91
7	98
7.5	101

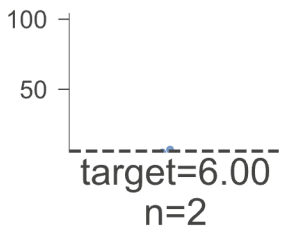


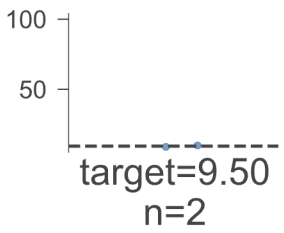
≤ >

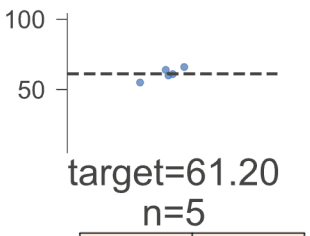
Experience	Salary
1	5
1.5	7
2	9
2.5	10

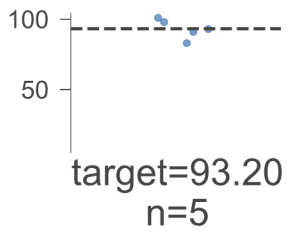


Experience	Salary
3	60
3.5	64
4	55
4.5	61
5	66
5.5	83
6	93
6.5	91
7	98
7.5	101









Experience	Salary
1	5
1.5	7

Experience	Salary
2	9
2.5	10

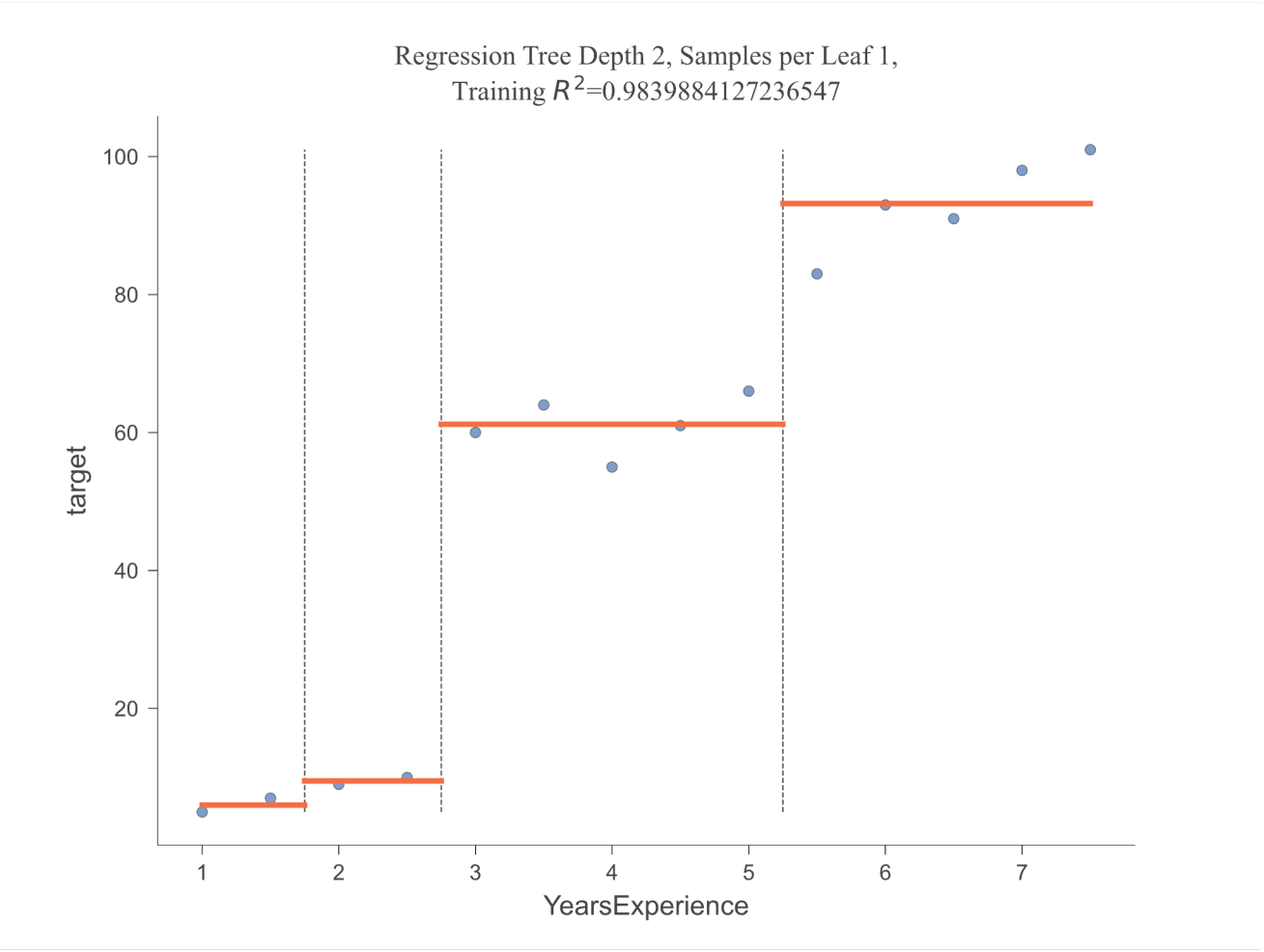
Experience	Salary
3	60
3.5	64
4	55
4.5	61
5	66

Experience	Salary
5.5	83
6	93
6.5	91
7	98
7.5	101

Level-wise Growth

❖ Salary

Experience	Salary
1	5
1.5	7
2	9
2.5	10
3	60
3.5	64
4	55
4.5	61
5	66
5.5	83
6	93
6.5	91
7	98
7.5	101



Leaf-wise tree growth strategy: Choose only branch with the lowest α_{mse} to split

Experience	Salary
1	5
1.5	7
2	9
2.5	10

$\alpha_{mse} = 202.02$

$\alpha_{mse} = 201.79$

$\alpha_{mse} = 202.25$

Experience	Salary
3	60
3.5	64
4	55
4.5	61
5	66
5.5	83
6	93
6.5	91
7	98
7.5	101

$\alpha_{mse} = 179.26$

$\alpha_{mse} = 161.48$

$\alpha_{mse} = 108.63$

$\alpha_{mse} = 61.86$

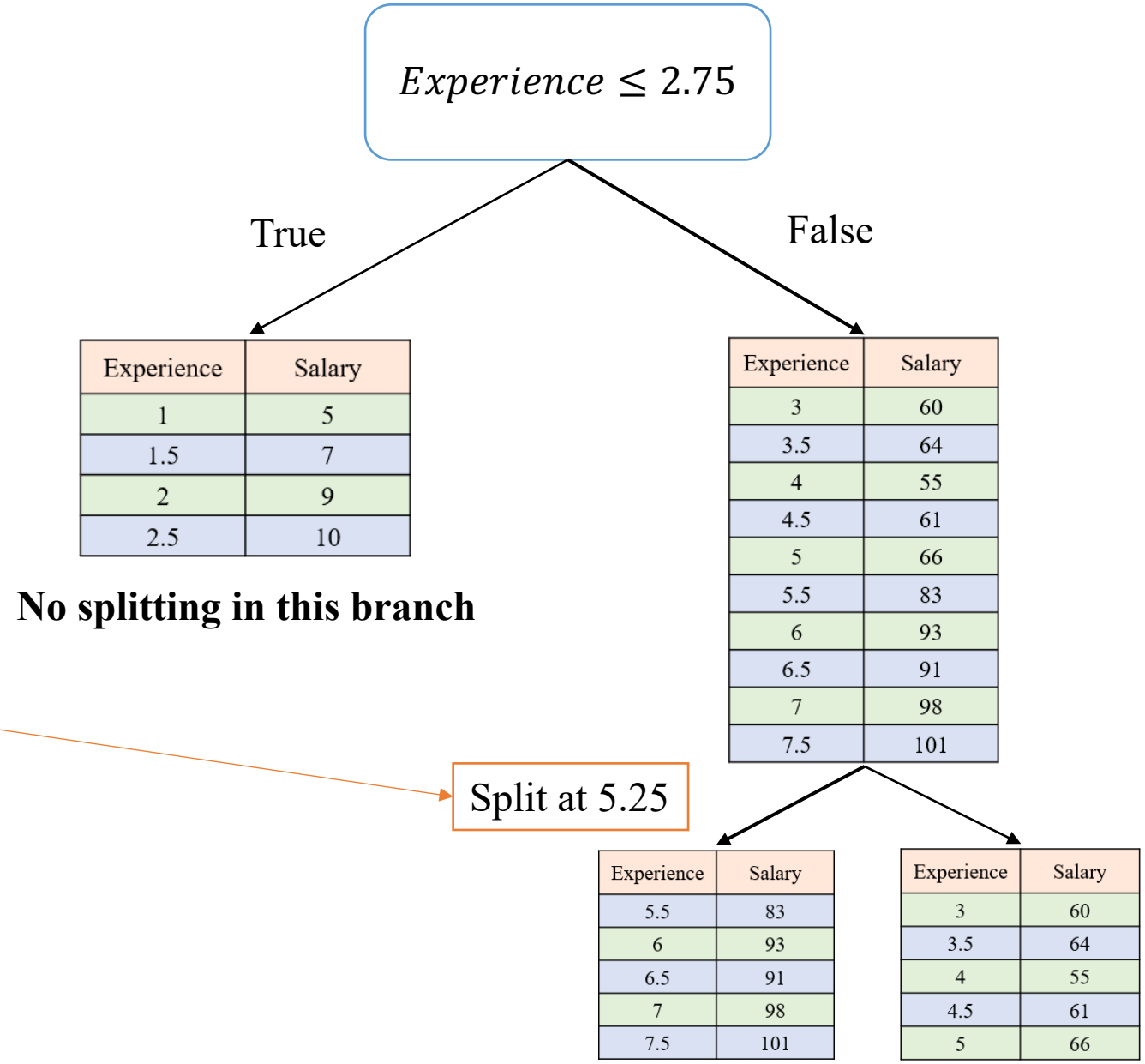
$\alpha_{mse} = 19.88$

$\alpha_{mse} = 38.88$

$\alpha_{mse} = 86.73$

$\alpha_{mse} = 113.94$

$\alpha_{mse} = 157.70$



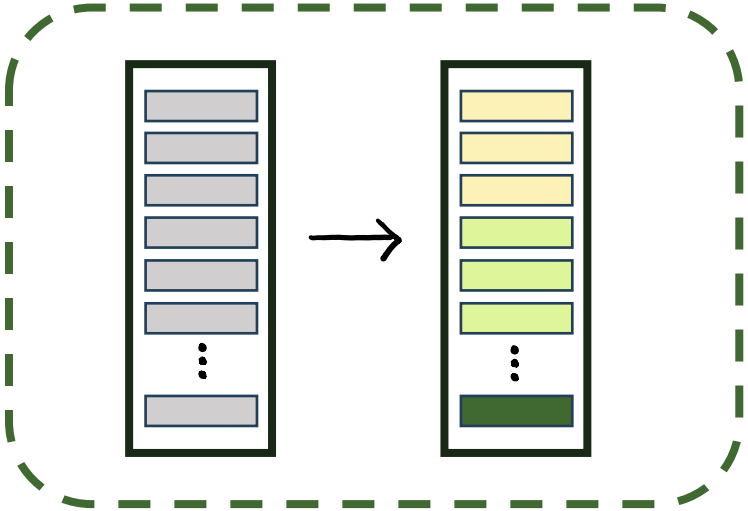
Exclusive Feature Bundling

Outlook	Temp	Humidity	Wind	Play Tennis
Sunny	Hot	High	Weak	No
Sunny	Hot	High	Strong	No
Overcast	Hot	High	Weak	Yes
Rain	Mild	High	Weak	Yes
Rain	Cool	Normal	Weak	Yes
Rain	Cool	Normal	Strong	No
Overcast	Cool	Normal	Strong	Yes
Sunny	Mild	High	Weak	No
Sunny	Cool	Normal	Weak	Yes
Rain	Mild	Normal	Weak	Yes
Sunny	Mild	Normal	Strong	Yes
Overcast	Mild	High	Strong	Yes
Overcast	Hot	Normal	Weak	Yes
Rain	Mild	High	Strong	No

0: Strong
1: Weak

ID	Age	Favorite Color	Class
S1	22	Red	0
S2	25	Blue	1
S3	28	Green	0
S4	35	Red	1
S5	40	Blue	1
S6	30	Green	0

0: Red
1: Blue
2: Green



Label conversion

Implicitly having ordinal relationship

How to solve the problem?

Exclusive Feature Bundling

❖ One-hot encoding

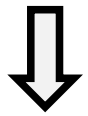
One-hot encoding converts categorical data into numeric

Each row has 1 for the true category, 0 for all others

Color
Red
Blue
Green
Red
Blue
Green

Color_Red	Color_Blue	Color_Green
1	0	0
0	1	0
0	0	1
1	0	0
0	1	0
0	0	1

Sparse features { Memory Inefficiency
Computational Overhead



Exclusive Feature Bundling (EFB)

Exclusive Feature Bundling

❖ Exclusive Feature Bundling

Groups mutually exclusive features into one feature

Reduces dimensionality and memory use

Color_Red	Color_Blue	Color_Green
1	0	0
0	1	0
0	0	1
1	0	0
0	1	0
0	0	1

EFB

Color_bundle
0
1
2
0
1
2

Mutually exclusive features: almost never have non-zero values at the same time, if so the others in the group are zero.

$$\text{Color\_bundle} = \begin{cases} 0 & \text{if Red} \\ 1 & \text{if Blue} \\ 2 & \text{if Green} \end{cases}$$

Exclusive Feature Bundling

❖ Exclusive Feature Bundling

Groups mutually exclusive features into one feature

Reduces dimensionality and memory use

Color	Color_Red	Color_Blue	Color_Green
Red	1	0	0
Blue	0	1	0
Green	0	0	1
Red	1	0	0
Blue	0	1	0
Green	0	0	1

EFB →

Color_bundle
0
1
2
0
1
2

Label Encoding	**EFB (LightGBM)**
Encodes categories as integers, may imply order	Bundles exclusive one-hot features, no order issue
Simple conversion, risk of misinterpretation	Efficient compression, preserves meaning

Exclusive Feature Bundling

❖ Example: Step-by-Step

ID	Age	Color_Red	Color_Blue	Color_Green	Class
S1	22	1	0	0	0
S2	25	0	1	0	1
S3	28	0	0	1	0
S4	35	1	0	0	1
S5	40	0	1	0	1
S6	30	0	0	1	0

Mutually exclusive features

EFB

ID	Age	Color_bundle	Class
S1	22	0	0
S2	25	1	1
S3	28	2	0
S4	35	0	1
S5	40	1	1
S6	30	2	0

$$\text{Color\_bundle} = \begin{cases} 0 & \text{if Red} \\ 1 & \text{if Blue} \\ 2 & \text{if Green} \end{cases}$$

❖ Example: Step-by-Step

ID	Age	Color_bundle	Class
S1	22	0	0
S2	25	1	1
S3	28	2	0
S4	35	0	1
S5	40	1	1
S6	30	2	0

Entropy:

$$E(S) = -\sum_{c \in C} p_c \log_2 p_c$$

Information Gain

$$IG(S,F) = E(S) - \sum_{f \in F} \frac{|S_f|}{|S|} E(S_f)$$

Root Entropy

- Class 0: $|\ S1, S3, S6\ | = 3$
- Class 1: $|\ S2, S4, S5\ | = 3$
- Total samples: $|S| = 6$

$$p_0 = \frac{3}{6} = 0.5, \qquad p_1 = \frac{3}{6} = 0.5$$

$$E(S) = -(0.5 \log_2 0.5 + 0.5 \log_2 0.5) = 1$$

Age	Class
22	0
25	1
28	0
30	0
35	1
40	1

23.5 | 26.5 | 29.0 | 32.5 | 37.5

Ascending sorting

Age≤23.5

True

Age	Class
22	0

$S_{Age \le 23.5} = \{Class\ 0: 1, Class\ 1: 0\}$

$$E(S_{Age \le 23.5}) = -(1 \cdot \log_2 1 + 0 \cdot \log_2 0) = 0$$

False

Age	Class
25	1
28	0
35	1
40	1
30	0

$S_{Age > 23.5} = \{Class\ 0: 2, Class\ 1: 3\}$

$$E(S_{Age > 23.5}) = -\left(\frac{2}{5}\log_2\frac{2}{5} + \frac{3}{5}\log_2\frac{3}{5}\right) = 0.97$$

$$E_{Split} = \frac{1}{6} \cdot 0 + \frac{5}{6} \cdot 0.97 = 0.81$$

$$IG(S, Age \le 23.5) = 1.0 - 0.81 = 0.19$$

❖ Example: Step-by-Step

ID	Age	Color_bundle	Class
S1	22	0	0
S2	25	1	1
S3	28	2	0
S4	35	0	1
S5	40	1	1
S6	30	2	0

Entropy:

$$E(S) = -\sum_{c \in C} p_c \log_2 p_c$$

Information Gain

$$IG(S,F) = E(S) - \sum_{f \in F} \frac{|S_f|}{|S|} E(S_f)$$

Root Entropy

- Class 0: $|\ S1, S3, S6\ | = 3$
- Class 1: $|\ S2, S4, S5\ | = 3$
- Total samples: $|S| = 6$

$$p_0 = \frac{3}{6} = 0.5, \qquad p_1 = \frac{3}{6} = 0.5$$

$$E(S) = -(0.5 \log_2 0.5 + 0.5 \log_2 0.5) = 1$$

Age	Class
22	0
25	1
28	0
30	0
35	1
40	1

23.5 | 26.5 | 29.0 | 32.5 | 37.5

Ascending sorting

Threshold	E_{Split}	IG
Age ≤ 23.5	0.81	0.19
Age ≤ 26.5	1.00	0.00
Age ≤ 29.0	0.92	0.08
Age ≤ 32.5	0.54	**0.46**
Age ≤ 37.5	0.81	0.19

$$\text{Color\_bundle} = \begin{cases} 0 & \text{if Red} \\ 1 & \text{if Blue} \\ 2 & \text{if Green} \end{cases} \rightarrow \begin{array}{l} \text{Red vs others: } S_{Color} = 0,\ S_{Color} \in \{1,2\} \\ \text{Blue vs others: } S_{Color} = 1,\ S_{Color} \in \{0,2\} \\ \text{Green vs others: } S_{Color} = 2,\ S_{Color} \in \{0,1\} \end{array}$$

Color = 0

True → Class 0 | Class 1: 1 | 1 → $p_0 = p_1 = \frac{1}{2} = 0.5,\ E(0) = 1.0$

False → Class 0 | Class 1: 2 | 2 → $p_0 = p_1 = \frac{2}{4} = 0.5,\ E(1,2) = 1.0$

$$E_{split} = \frac{2}{6} \cdot 1 + \frac{4}{6} \cdot 1 = 1,$$

$$IG(\text{S, Color } = 0\ ) = 1 - 1 = 0$$

❖ Example: Step-by-Step

ID	Age	Color_bundle	Class
S1	22	0	0
S2	25	1	1
S3	28	2	0
S4	35	0	1
S5	40	1	1
S6	30	2	0

Entropy:

$$E(S) = -\sum_{c \in C} p_c \log_2 p_c$$

Information Gain

$$IG(S,F) = E(S) - \sum_{f \in F} \frac{|S_f|}{|S|} E(S_f)$$

Root Entropy

- Class 0: $|\ S1, S3, S6\ | = 3$
- Class 1: $|\ S2, S4, S5\ | = 3$
- Total samples: $|S| = 6$

$$p_0 = \frac{3}{6} = 0.5, \qquad p_1 = \frac{3}{6} = 0.5$$

$$E(S) = -(0.5 \log_2 0.5 + 0.5 \log_2 0.5) = 1$$

Age	Class
22	0
25	1
28	0
30	0
35	1
40	1

23.5 | 26.5 | 29.0 | 32.5 | 37.5

Ascending sorting

Threshold	E_{Split}	IG
Age ≤ 23.5	0.81	0.19
Age ≤ 26.5	1.00	0.00
Age ≤ 29.0	0.92	0.08
Age ≤ 32.5	0.54	**0.46**
Age ≤ 37.5	0.81	0.19

Color = 1

True

Class 0	Class 1
0	2

$p_0 = 0, p_1 = 1$

$E(1) = 0$

False

Class 0	Class 1
3	1

$p_0 = \frac{3}{4}, p_1 = \frac{1}{4}$

$E(0,2) = -\left(\frac{3}{4}\log_2\frac{3}{4} + \frac{1}{4}\log_2\frac{1}{4}\right) = 0.81$

$$E_{split} = \frac{2}{6} \cdot 0 + \frac{4}{6} \cdot 0.81 = 0.54, \qquad IG(\text{S, Color } = 1\ ) = 1 - 0.54 = 0.46$$

❖ Example: Step-by-Step

ID	Age	Color_bundle	Class
S1	22	0	0
S2	25	1	1
S3	28	2	0
S4	35	0	1
S5	40	1	1
S6	30	2	0

Entropy:

$$E(S) = -\sum_{c \in C} p_c log_2 p_c$$

Information Gain

$$IG(S,F) = E(S) - \sum_{f \in F} \frac{|S_f|}{|S|} E(S_f)$$

Root Entropy

- Class 0: $|\ S1, S3, S6\ | = 3$
- Class 1: $|\ S2, S4, S5\ | = 3$
- Total samples: $|S| = 6$

$$p_0 = \frac{3}{6} = 0.5, \qquad p_1 = \frac{3}{6} = 0.5$$

$$E(S) = -(0.5 \log_2 0.5 + 0.5 \log_2 0.5) = 1$$

Age	Class
22	0
25	1
28	0
30	0
35	1
40	1

23.5 | 26.5 | 29.0 | 32.5 | 37.5

Ascending sorting

Threshold	E_{Split}	IG
Age ≤ 23.5	0.81	0.19
Age ≤ 26.5	1.00	0.00
Age ≤ 29.0	0.92	0.08
<u>Age ≤ 32.5</u>	0.54	**<u>0.46</u>**
Age ≤ 37.5	0.81	0.19

Color = 2

True

Class 0	Class 1
2	0

$p_0 = 1, p_1 = 0$

$E(2) = 0$

False

Class 0	Class 1
1	3

$$p_0 = \frac{1}{4}, p_1 = \frac{3}{4}$$

$$E(0,1) = -\left(\frac{1}{4}\log_2\frac{1}{4} + \frac{3}{4}\log_2\frac{3}{4}\right) = 0.81$$

$$E_{split} = \frac{2}{6} \cdot 0 + \frac{4}{6} \cdot 0.81 = 0.54, \qquad IG(\text{S, Color } = 2\ ) = 1 - 0.54 = 0.46$$

❖ Example: Step-by-Step

ID	Age	Color_bundle	Class
S1	22	0	0
S2	25	1	1
S3	28	2	0
S4	35	0	1
S5	40	1	1
S6	30	2	0

Entropy:

$$E(S) = -\sum_{c \in C} p_c log_2 p_c$$

Information Gain

$$IG(S,F) = E(S) - \sum_{f \in F} \frac{|S_f|}{|S|} E(S_f)$$

Root Entropy

- Class 0: $|\ S1, S3, S6\ | = 3$
- Class 1: $|\ S2, S4, S5\ | = 3$
- Total samples: $|S| = 6$

$$p_0 = \frac{3}{6} = 0.5, \qquad p_1 = \frac{3}{6} = 0.5$$

$$E(S) = -(0.5 \log_2 0.5 + 0.5 \log_2 0.5) = 1$$

Age	Class
22	0
25	1
28	0
30	0
35	1
40	1

23.5 | 26.5 | 29.0 | 32.5 | 37.5

Ascending sorting

Threshold	E_{Split}	IG
Age ≤ 23.5	0.81	0.19
Age ≤ 26.5	1.00	0.00
Age ≤ 29.0	0.92	0.08
Age ≤ 32.5	0.54	**0.46**
Age ≤ 37.5	0.81	0.19

$$\text{Color\_bundle} = \begin{cases} 0 \text{ if Red} \\ 1 \text{ if Blue} \\ 2 \text{ if Green} \end{cases}$$

→ Red vs others: $S_{Color} = 0,\ S_{Color} \in \{1,2\}$
Blue vs others: $S_{Color} = 1,\ S_{Color} \in \{0,2\}$
Green vs others: $S_{Color} = 2,\ S_{Color} \in \{0,1\}$

Color	E_{Split}	IG
0 (Red)	1	0
1 (Blue)	0.54	**0.46**
2 (Green)	0.54	0.46

There is a tie-break between "Age ≤ 32.5", "Color = 1 (Blue)" with highest gain score (0.46), choose Color = Blue for splitting

❖ Example: Step-by-Step

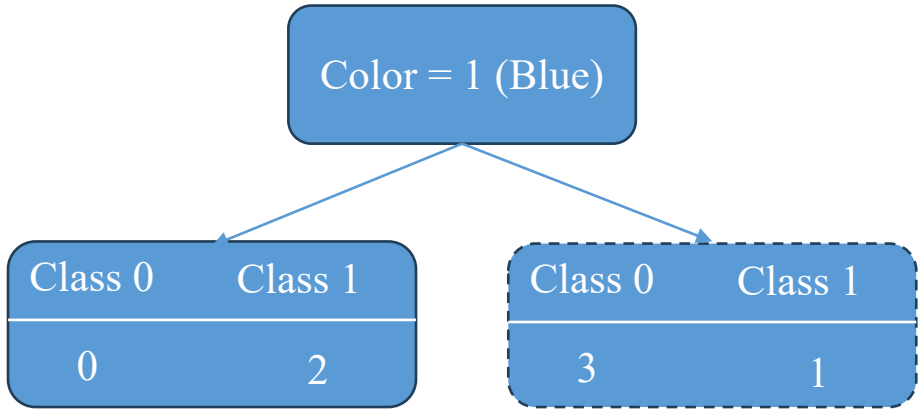
ID	Age	Color_bundle	Class
S1	22	0	0
S2	25	1	1
S3	28	2	0
S4	35	0	1
S5	40	1	1
S6	30	2	0

Entropy:

$$E(S) = -\sum_{c \in C} p_c \log_2 p_c$$

Information Gain

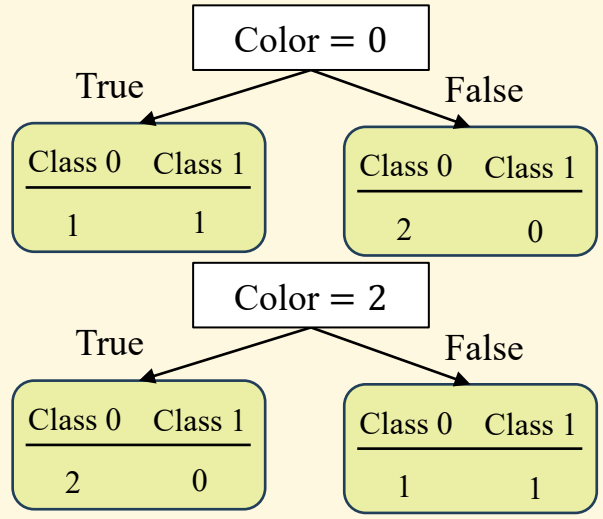
$$IG(S,F) = E(S) - \sum_{f \in F} \frac{|S_f|}{|S|} E(S_f)$$



Continue split

ID	Age	Color_bundle	Class
S1	22	0	0
S3	28	2	0
S4	35	0	1
S6	30	2	0

Color_bundle = { 0 if Red; 1 if Blue; 2 if Green }



Color	E_{Split}	IG
0 (Red)	0.50	0.31
2 (Green)	0.50	0.31

Ascending sorting

Age	Class
22	0
28	0
30	0
35	1

25
29
32.5

Threshold	E_{Split}	IG
Age ≤ 25	0.69	0.12
Age ≤ 29	0.50	0.31
Age ≤ 32.5	0.00	**0.81**

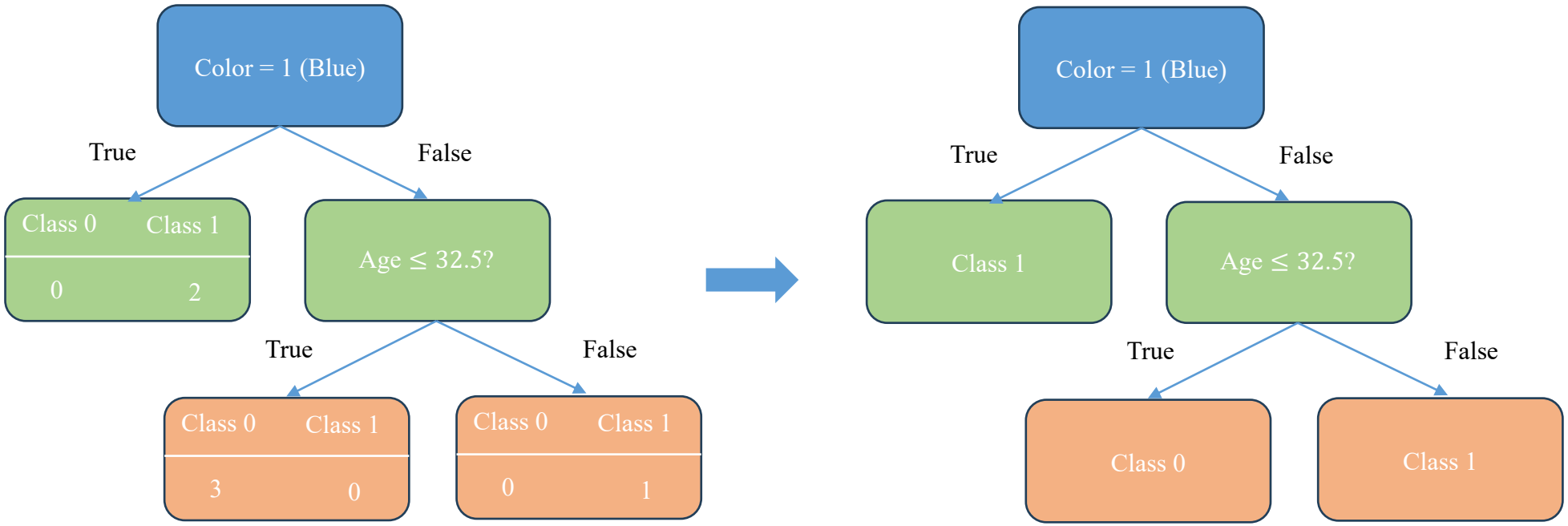
choose Age ≤ 32.5 with **highest gain score** for split

$$p_0 = \frac{3}{4}, p_1 = \frac{1}{4} \longrightarrow E(S) = -\left(\frac{3}{4}\log_2\frac{3}{4} + \frac{1}{4}\log_2\frac{1}{4}\right) = 0.81$$

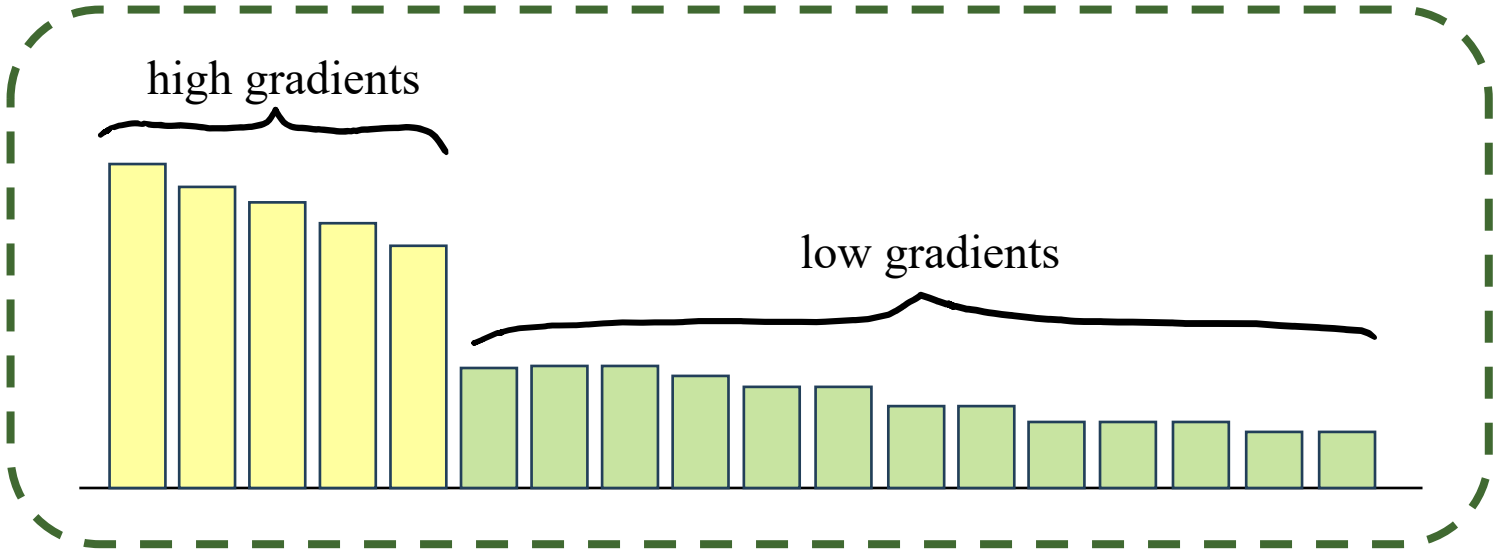
❖ Example

ID	Age	Favorite Color	Class
S1	22	Red	0
S2	25	Blue	1
S3	28	Green	0
S4	35	Red	1
S5	40	Blue	1
S6	30	Green	0

ID	Age	Color_bundle	Class
S1	22	0	0
S2	25	1	1
S3	28	2	0
S4	35	0	1
S5	40	1	1
S6	30	2	0



Gradient-based One-Side Sampling



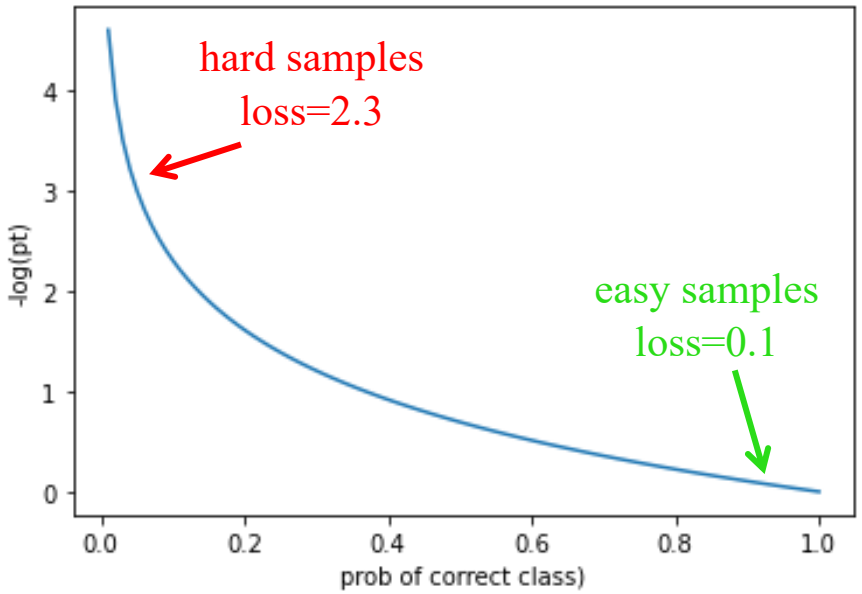
Easy samples loss = 100000*0.1
= 10000

Hard samples loss = 100*2.3
= 230

When having a large number of samples

Instead of using all the samples

Only using the samples with top gradient magnitudes + some randomly selected samples of the remaining samples



❖ Example 1

	x	y_{reg}	r_{i1}
1.5	1	2	-2
2.5	2	3	-1
3.5	3	5	1
	4	6	2

$F_0(x) = 4.0$

(i) Compute pseudo-residuals:
$r_{i1} = [-2, -1, 1, 2]$

(ii) Choose 2 samples with largest gradients

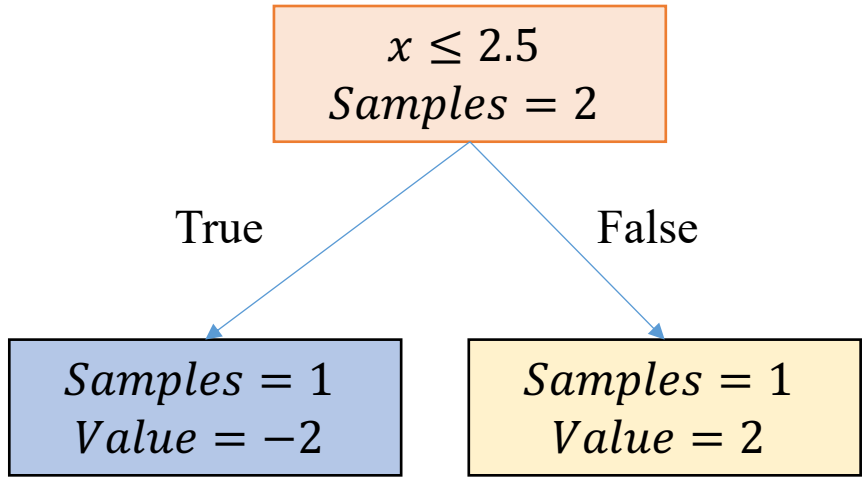
(iii) Choose the best splitting threshold: $x \leq 2.5$

(iv) Compute leaf values: $\gamma_{1,1} = -2$, $\gamma_{2,1} = 2$

(v) Update the model ($v = 1$):

$$F_1(x_i) = \left(F_1(x_1), F_1(x_2), F_1(x_3), F_1(x_4)\right)$$
$$= (2.0, 2.0, 6.0, 6.0)$$

	x	y_{reg}	r_{i1}
2.5	1	2	-2
	4	6	2



x	y_{reg}	r_{i1}	r_{i2}
1	2	-2	0
2	3	-1	1
3	5	1	-1
4	6	2	0
		6	2

❖ Example 2

	x	y_{reg}	r_{i1}
1.5	1	2	-2
2.5	2	3	-1
3.5	3	5	1
	4	6	2

$F_0(x) = 4.0$

(i) Compute pseudo-residuals:

$$r_{i1} = [-2, -1, 1, 2]$$

(ii) Choose 2 samples with largest gradients

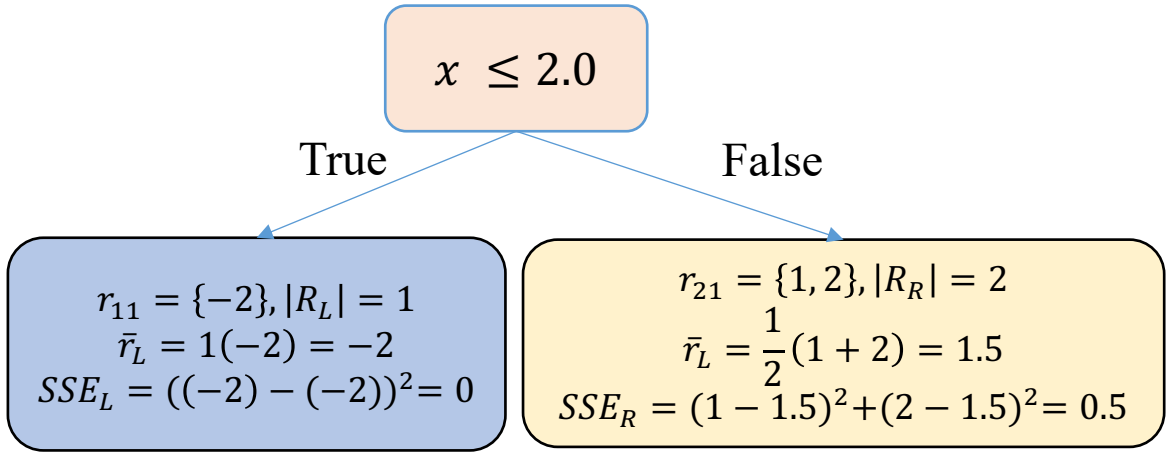
(iii) Choose the best splitting threshold: $x \leq 3.5$

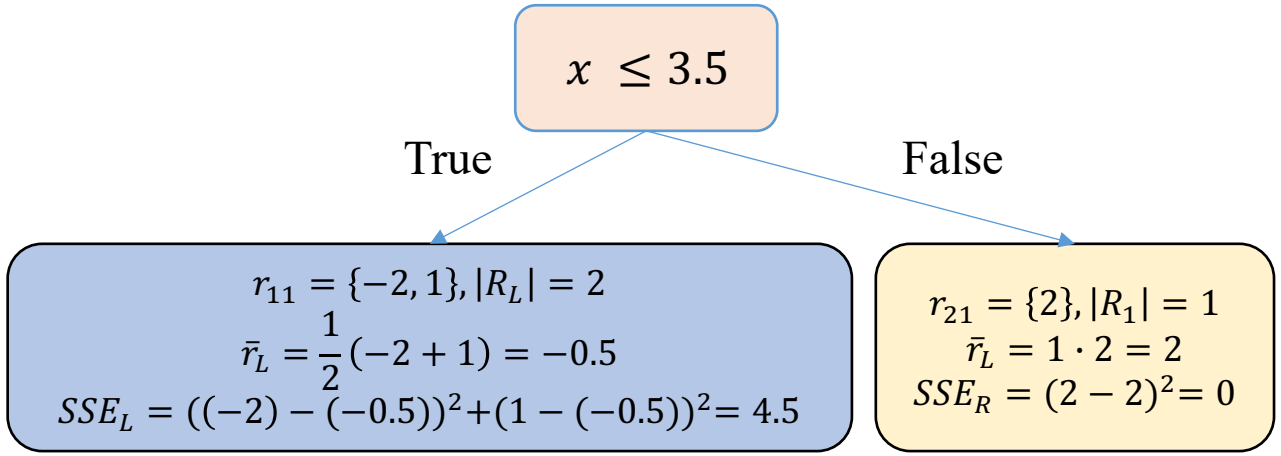
(iv) Compute leaf values: $\gamma_{1,1} = -0.5$, $\gamma_{2,1} = 2$

(v) Update the model (v = 1):

$$F_1(x_i) = \left(F_1(x_1), F_1(x_2), F_1(x_3), F_1(x_4)\right)$$
$$= (2, 2, 5.5, 5.5)$$

	x	y_{reg}	r_{i1}
2.0	1	2	-2
3.5	3	5	1
	4	6	2





❖ Example

	x	y_{reg}	r_{i1}
1.5	1	2	-2
2.5	2	3	-1
3.5	3	5	1
	4	6	2

$F_0(x) = 4.0$

(i) Compute pseudo-residuals:

$r_{i1} = [-2, -1, 1, 2]$

(ii) Choose 2 samples with largest gradients

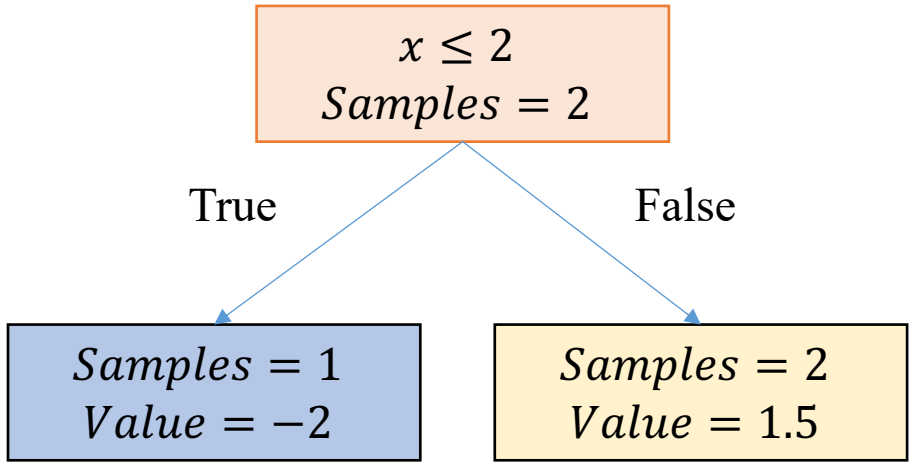
(iii) Choose the best splitting threshold: $x \leq 3.5$

(iv) Compute leaf values: $\gamma_{1,1} = -0.5$, $\gamma_{2,1} = 2$

(v) Update the model ($v = 1$):

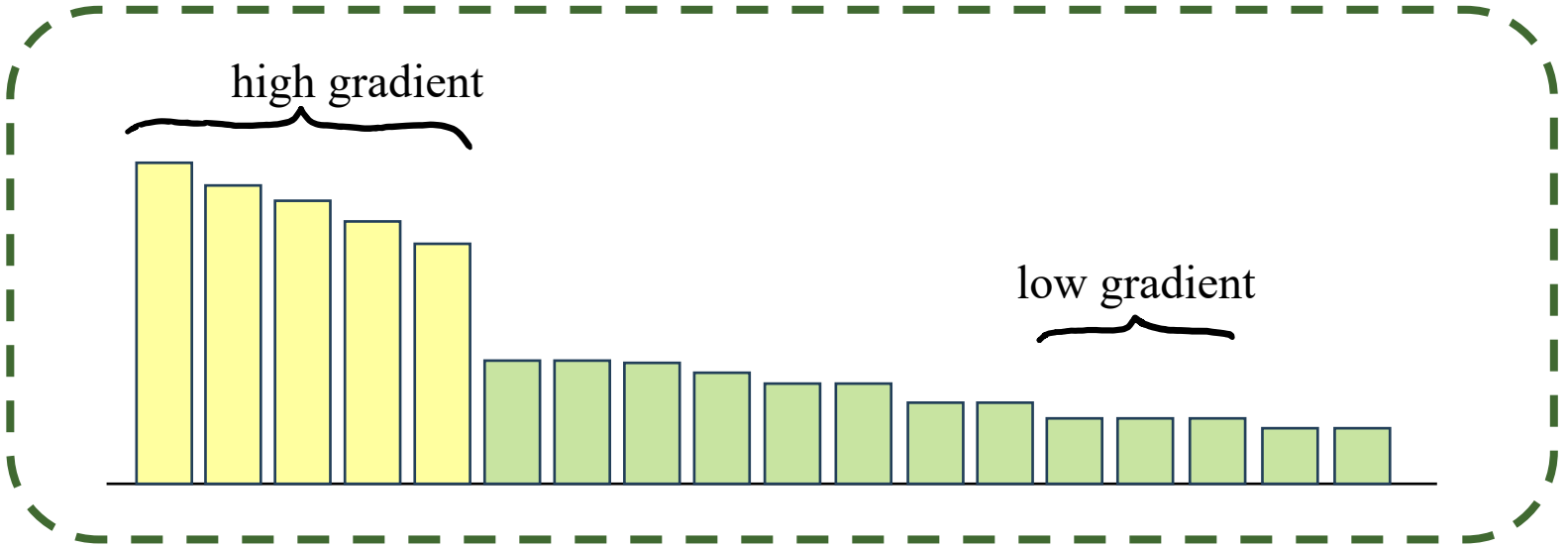
$$F_1(x_i) = \left(F_1(x_1), F_1(x_2), F_1(x_3), F_1(x_4)\right)$$
$$= (2, 2, 5.5, 5.5)$$

	x	y_{reg}	r_{i1}
2.0	1	2	-2
	3	5	1
3.5	4	6	2

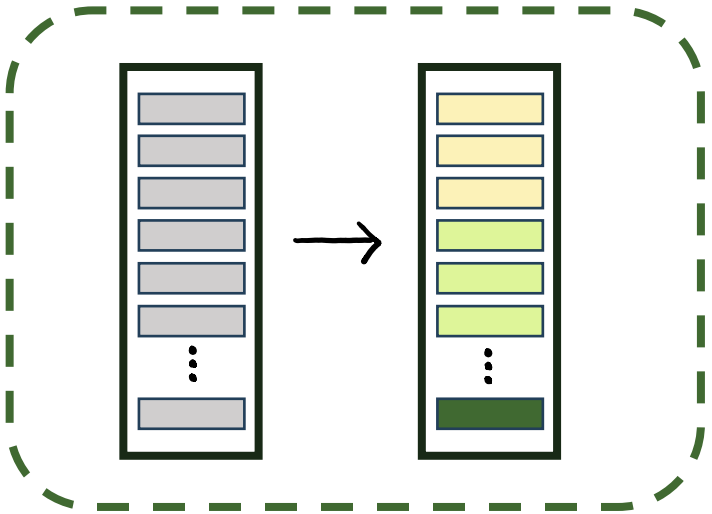


x	y_{reg}	r_{i1}	r_{i2}
1	2	-2	0
2	3	-1	1
3	5	1	0.5
4	6	2	-0.5
		6	2

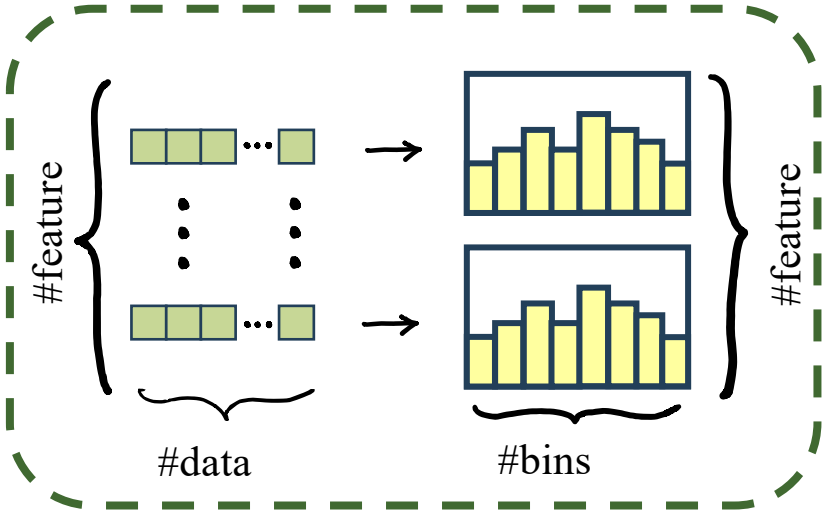
Gradient-based One-Side Sampling (GOSS)



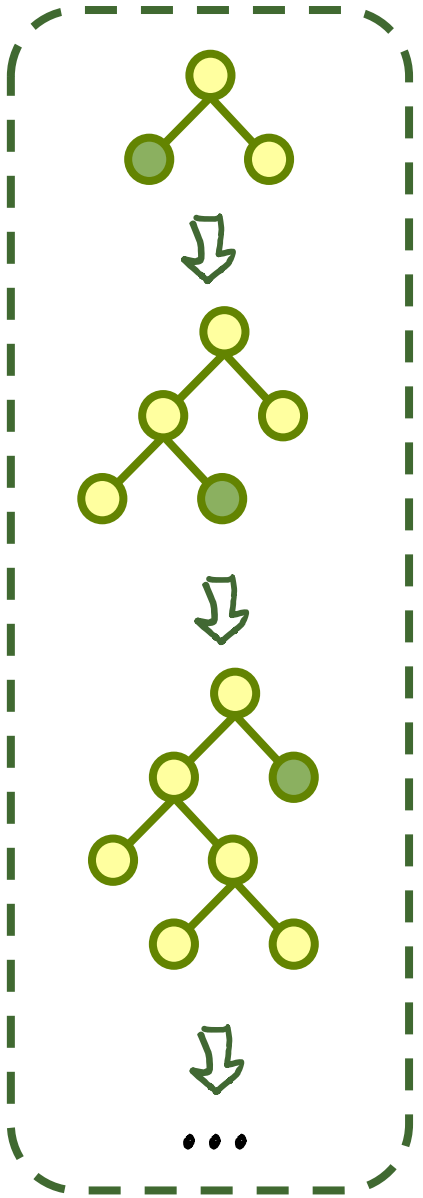
Exclusive feature bundling (EFB)



Histogram-based Algorithm



Leaf-wise tree growth



Khởi tạo với dự đoán ban đầu

$$\hat{y}^{(0)} = \arg\min_c \sum_i L(y_i, c) = \frac{1}{n}\sum y_i$$

1) Tính Gradient & Hessian

$$L = \frac{1}{2n}\sum (y_i - \hat{y}_i)^2$$

Gradient: $g_i = \frac{\partial L}{\partial \hat{y}_i} = (\hat{y}_i - y_i)$

Hessian: $h_i = \frac{\partial^2 L}{\partial \hat{y}_i^2} = 1$

2) Áp dụng GOSS

Chọn **top a%** mẫu có |gradient| lớn nhất (dữ liệu khó)

Chọn ngẫu nhiên **b%** trong phần còn lại (dữ liệu dễ)

Tính lại trọng số cho mẫu để tránh unbiased

$$g_i \leftarrow g_i \times \frac{1-a}{b} \qquad h_i \leftarrow h_i \times \frac{1-a}{b}$$

3) Tạo Histogram cho mỗi feature

Chia mỗi feature thành **B bins**

(Tính tổng gradient & hessian cho từng bin)

❖ Altogether (LightGBM for Regression)

4) Tìm Split tốt nhất & Xây dựng Cây

Duyệt qua các bin → tính Gain:

$$Gain = \frac{1}{2}\left(\frac{G_L^2}{H_L + \lambda} + \frac{G_R^2}{H_R + \lambda} - \frac{(G_L + G_R)^2}{H_L + H_R + \lambda}\right) - \gamma$$

Chọn feature và điểm chia với Gain cao nhất

Chia node thành 2 nhánh (Leaf-wise growth)

$$G_{L|R} = \sum_{i \in S_{L|R}} g_i\,, \quad H_{L|R} = \sum_{i \in S_{L|R}} h_i$$

5) Cập nhật giá trị lá

Giá trị dự đoán tại mỗi lá:

$$w_j = -\frac{G_j}{H_j + \lambda}$$

Cập nhật $\hat{y}_i \leftarrow \hat{y}_i + \eta w_j$ với η là learning_rate.

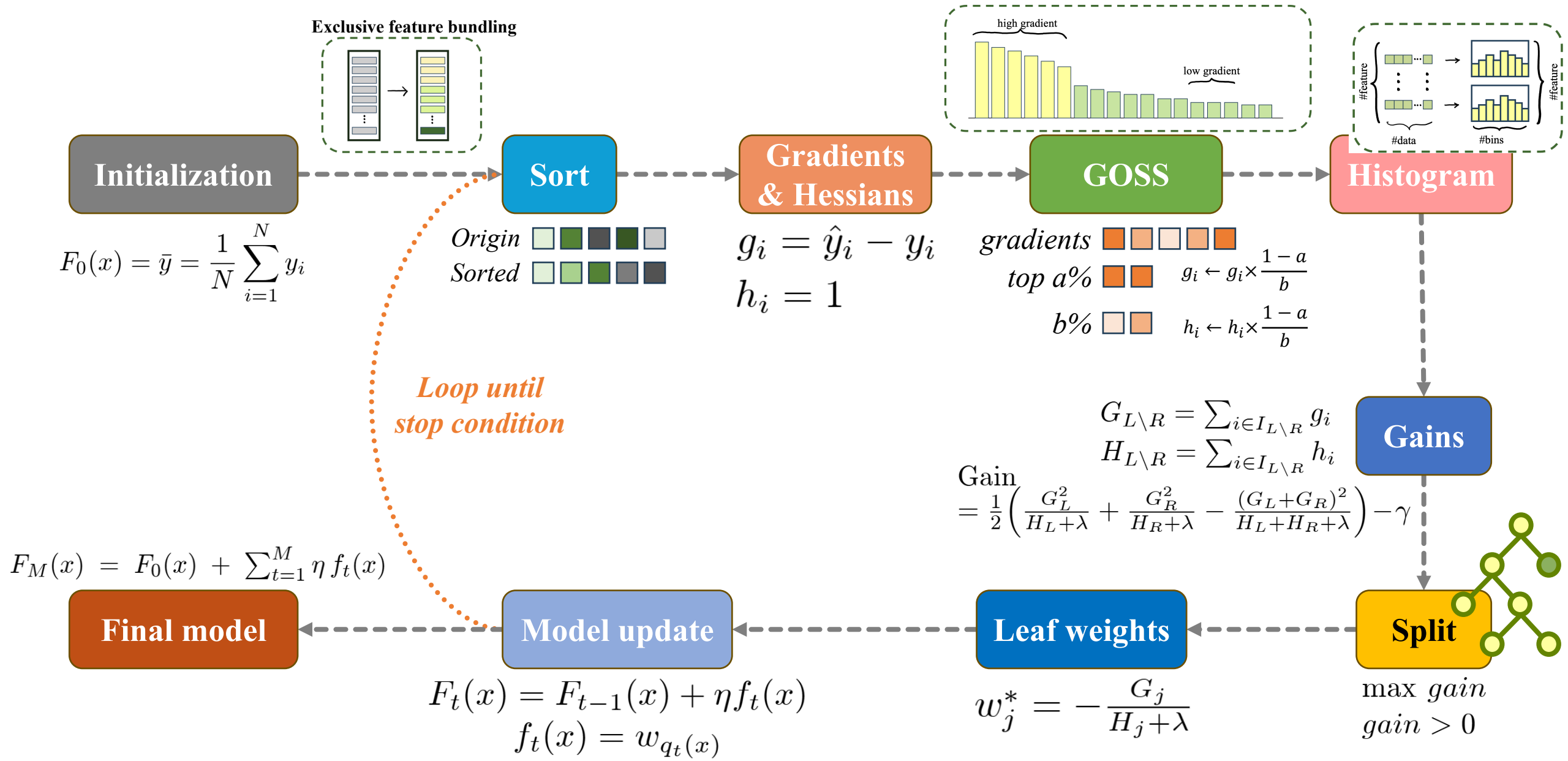
6) Lặp lại cho đến khi đủ số cây

Mỗi cây giảm loss một ít (mô hình dần tốt lên).

Cuối cùng $\hat{y}^{(t)}$ là:

$$\hat{y}^{(t)} = \sum_{t=1}^{T} \eta f_t(X)$$

❖ LightGBM for Regression




QUIZ
QUIZ
TIME

Outline

SECTION 1

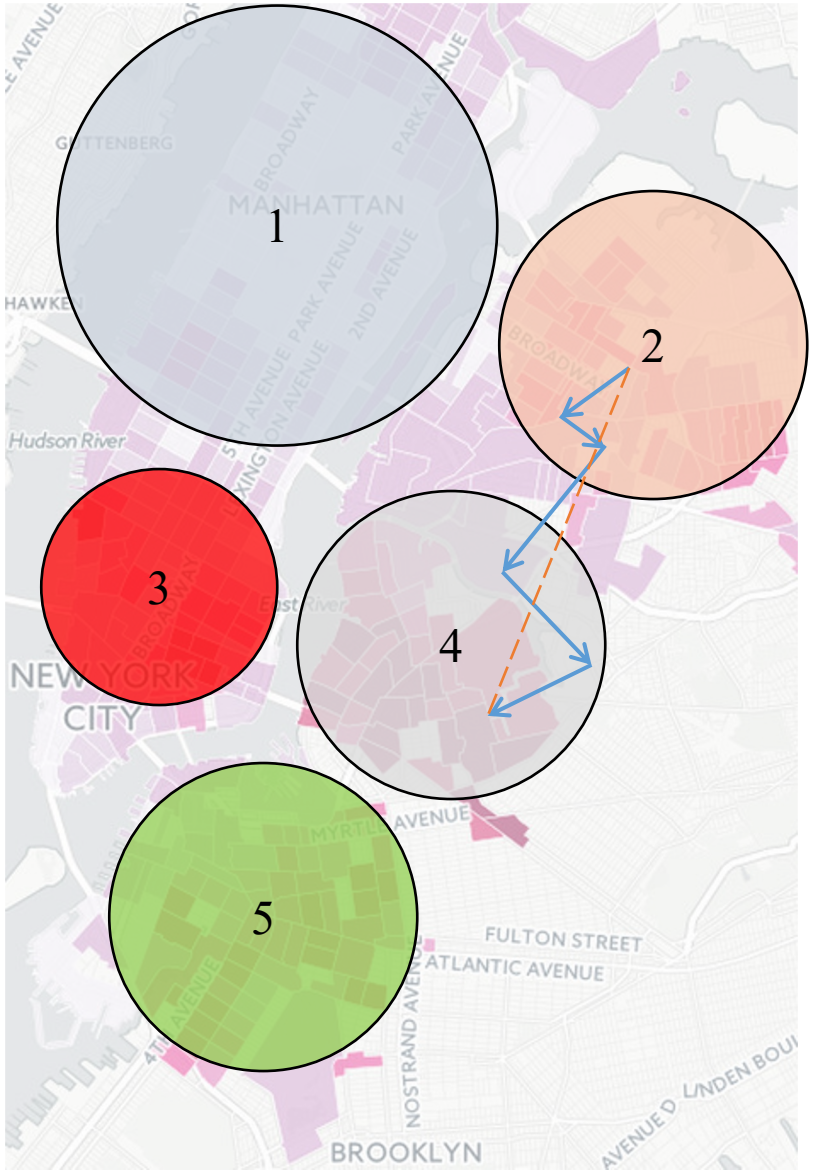
Discussion&Motivation

SECTION 2

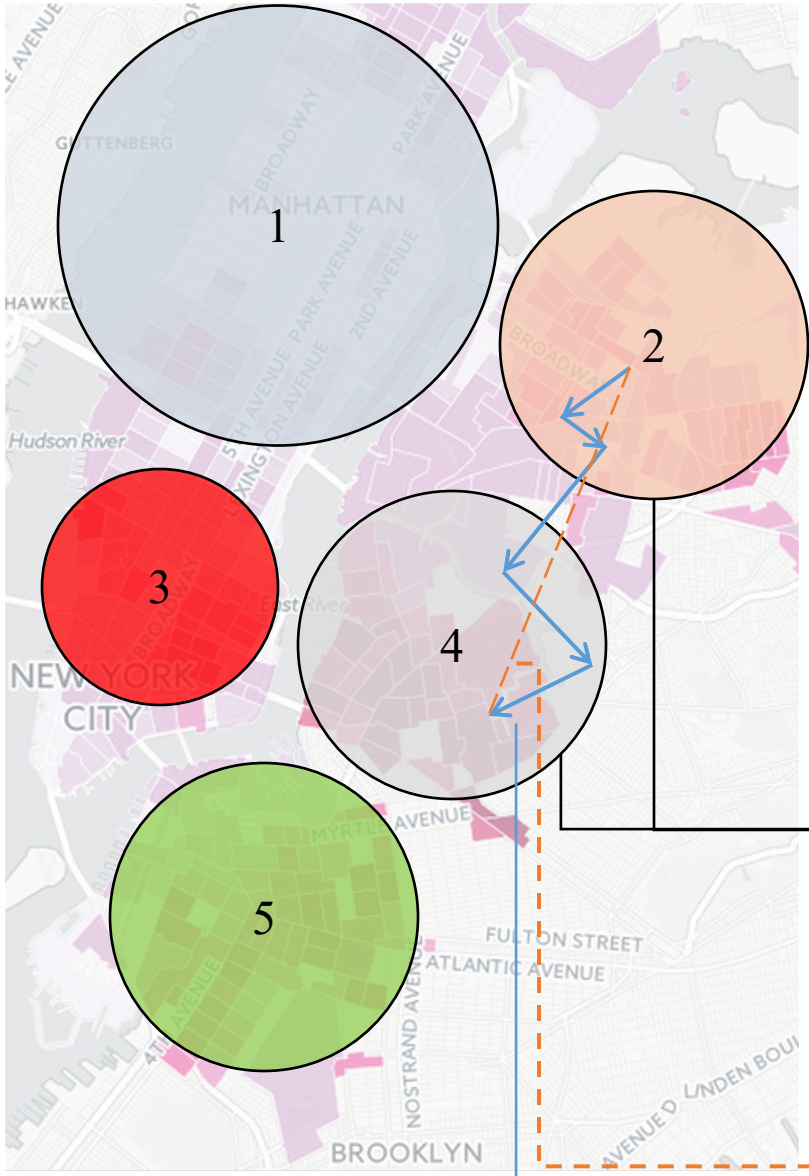
Improvements

SECTION 3

Case Studies



Taxi Trip Dataset



Number of samples: 1,400,000 trips +
Task: Predict taxi trip duration in NYC.

Train: 1,100,000 trips +
Test: 300,000 trips +

Basic features

Pickup time	Drop-off time	Pickup loc	Drop-off loc	Trip duration
17:24:55	17:32:30	(-73.9, 40.7)	(-73.9, 40.6)	455

This dataset has mixed data types (both numerical and categorical).

Using Kmean for clustering

Engineered features

Pickup cluster	Drop-off cluster	Haversine distance	Manhattan distance
2	4	60	74

Boosting in Regression

Metrics	GradientBoosting	XGBoost	LightGBM
RMSE ↓	**0.3466**	0.3497	0.3482
Time Training ↓	**452s**	3.39s	**2.65s**

LightGBM gives near-best RMSE in ~2.7 s ⇒ best accuracy–speed trade-off.

XGBoost and LightGBM are suitable for large tabular datasets.

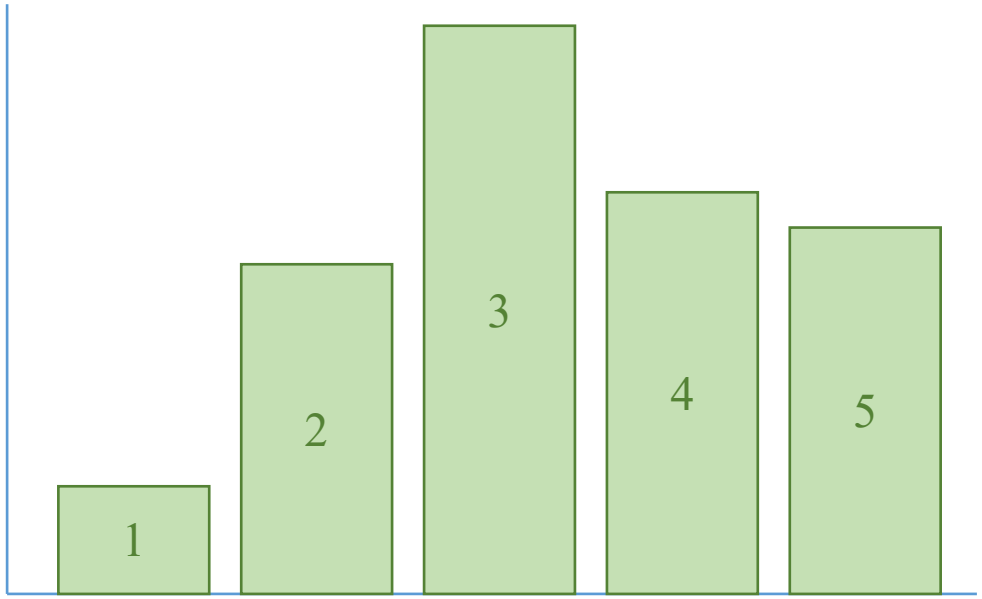
XGBoost

Cluster
1
1
1
2
2
2
2

Class 1	Class 2
1	0
1	0
1	0
0	1
0	1
0	1
0	1

XGBoost have to use one hot encoding.

Why does LightGBM work best here?



Categorical Features (Cluster)

LightGBM

Cluster
1
1
1
2
2
2
…

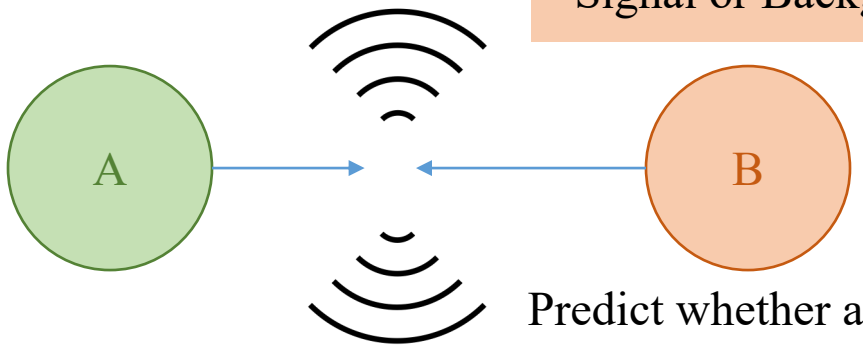
Bin 1

The Rest

LightGBM can work directly on categorical data.

HEPMASS Dataset

Signal or Background



Predict whether a collision event is signal or background

Number of samples: 3,500,000 collisions +
Task: Binary classification (signal vs. background events).

Train: 2,800,000 collision +
Test: 700,000 collision +

Perfect balance dataset.

All features are normalized at start.

22 low-level features
5 high-level features
→ Very high correlation.

Signal	F0	F1	…	F26
1				
1				
1				
…				
0				
0				
0				

This dataset has only one data type (dense numerical).

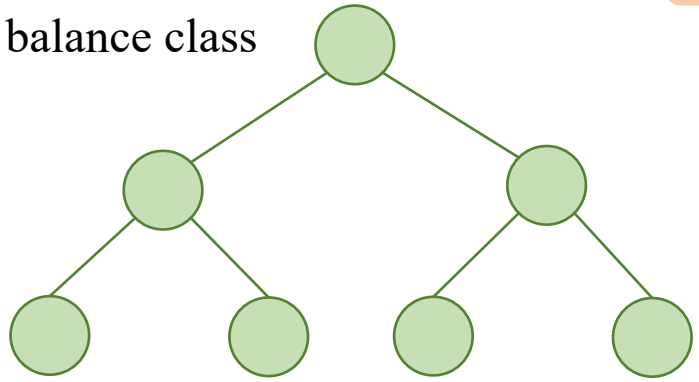
All particles have same mass
→ Same characteristics and many samples are correlated.

Boosting in Classification

Metrics	GradientBoosting	XGBoost	LightGBM
Accuracy ↑	**0.9181**	0.9178	0.9179
AUC-ROC ↑	0.9711	**0.9712**	0.9711
Time Training ↓	**4100s**	**20s**	21s

XGBoost slightly leads on AUC-ROC; training time is on par with LightGBM (~1s difference).

Level-wise Tree Method benefit with the balance class



On dense numeric tables, XGBoost's hist is highly optimized.

Why does XGBoost work best here?

XGBoost is in its **best case** because the dataset is totally balance and numerical
→ Node choosing method is at its best case

LightGBM remains competitive across metrics.

XGBoost for dense numerical task. LightGBM for large, mixed-type dataset with minimal setup.

LightGBM Library

❖ Parameter

```
lgb_params = {
    'objective': 'regression',
    'metric': 'rmse',
    'boosting_type': 'gbdt',
    'num_leaves': 63,
    'max_depth': 6,
    'learning_rate': 0.1,
    'feature_fraction': 0.8,
    'bagging_fraction': 0.8,
    'bagging_freq': 5,
    'min_child_samples': 20,
    'verbosity': -1,
    'random_state':
}
```

Mục tiêu bài toán là **hồi quy** (dự đoán giá trị liên tục)

Dùng Gradient Boosting Decision Tree

Ở mỗi cây, chỉ lấy 80% số feature ngẫu nhiên

Ở mỗi cây, chỉ dùng 80% số sample ngẫu nhiên

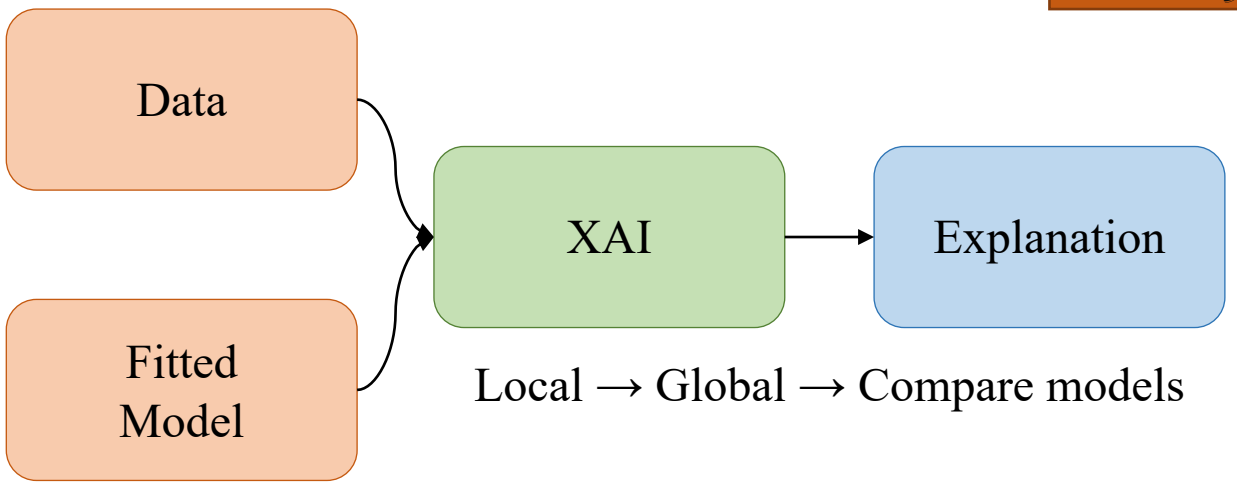
Cứ 5 iteration thì thực hiện bagging một lần

```
lgb_model = lgb.train(
    lgb_params,
    lgb_train,
    num_boost_round=num_rounds,
    valid_sets=[lgb_train, lgb_test],
    valid_names=['train', 'eval'],
)
```

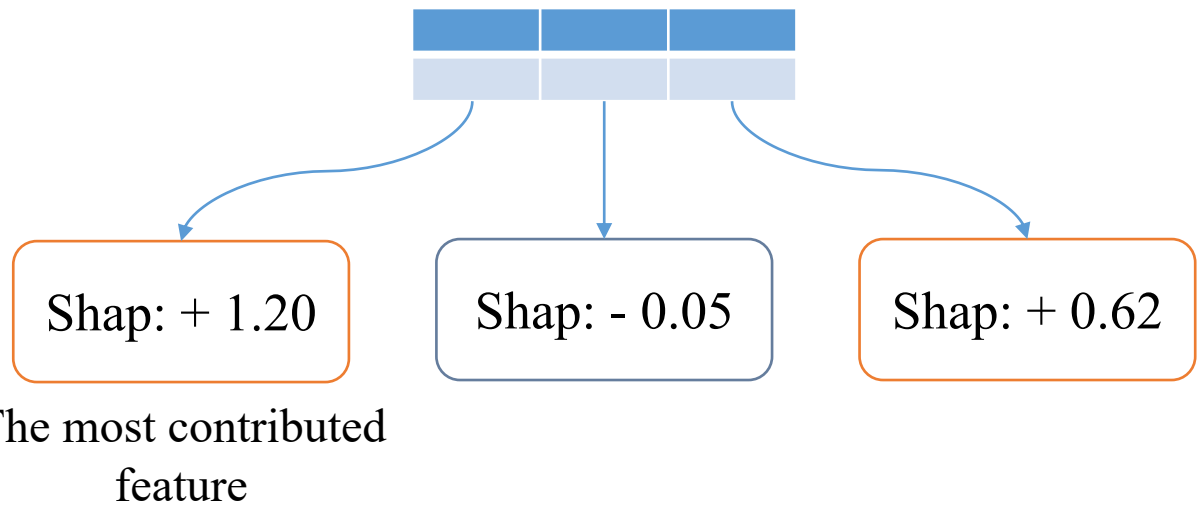
Model Interpretation

Which factors drive the predictions? Are XGBoost and LightGBM using the same signals?

How can we make the model more trustworthy?



How important is each feature to the model's prediction?



SHAP explains a prediction by attributing contributions to features.

- **Consistent explanations**: prediction = baseline + feature contributions.
- **Tree-native and fast:** exact and fast for XGBoost/LightGBM.
- **Model comparison:** Same units across models.

SHAP


```python
import shap
import numpy as np

# Add background sample for SHAP for speed/stability
bg = shap.sample(X_train, 200)

# Explain black-box models
xgb_exp = shap.TreeExplainer(xgb_model, data=bg)
xgb_shap = xgb_exp.shap_values(dtest)  # (n_samples, n_features) for reg
xgb_base = xgb_exp.expected_value

lgb_exp = shap.TreeExplainer(lgb_model, data=bg)
lgb_shap = lgb_exp.shap_values(X_test)
lgb_base = lgb_exp.expected_value

# Global summaries
shap.summary_plot(xgb_shap, dtest, show=False)   # beeswarm
shap.summary_plot(lgb_shap, X_test, show=False)

# Local explanations
shap.plots.waterfall(shap.Explanation(values=lgb_shap[i],
                                      base_values=xgb_base,
                                      data=X_test.iloc[i],
                                      feature_names=X_test.columns))
```

A background set sampled from the training data helps SHAP estimate the data distribution.

SHAP is Model-Agnostic method: can works with many models/libraries (even Transformer!)

We explain predictions on the validation/test set to understand generalization and detect overfitting.

We can use SHAP to explain each individual sample comparing to global expectation.

SHAP is simple framework that work with any model and library

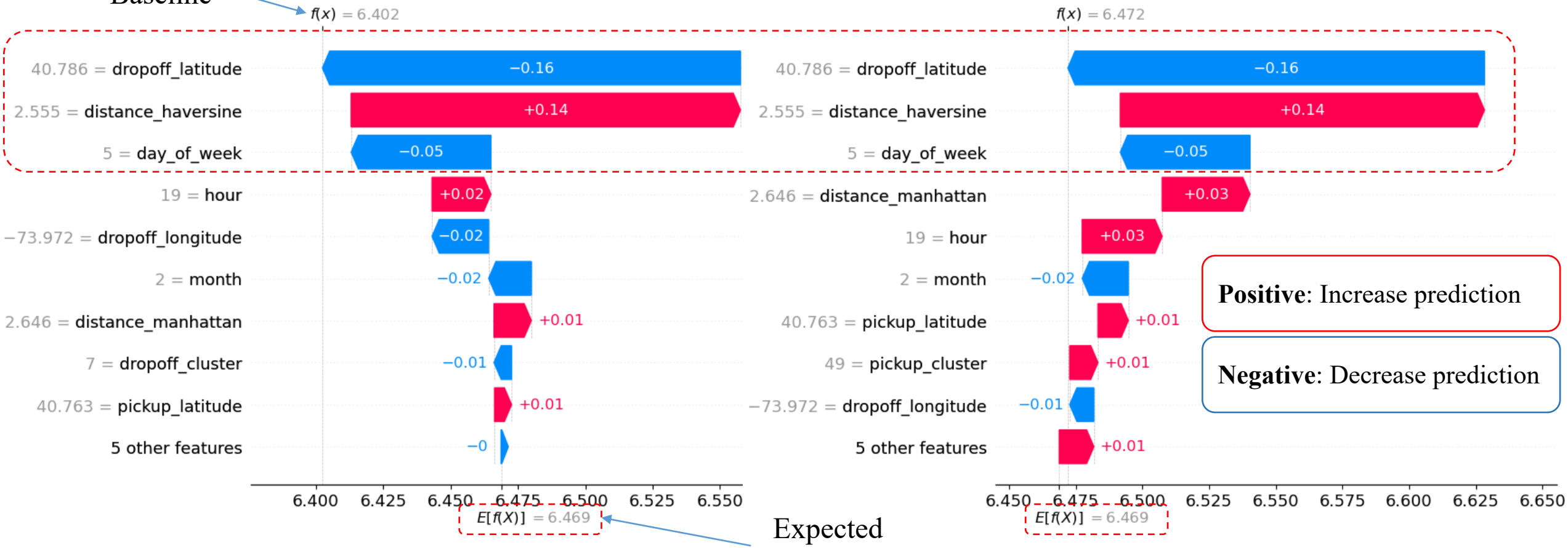
For large datasets, sample a background (100 - 1000) and a subset of points to plot.

SHAP for Taxi Trip Dataset

XGBoost

LightGBM

Baseline



Expected Baseline

Positive: Increase prediction

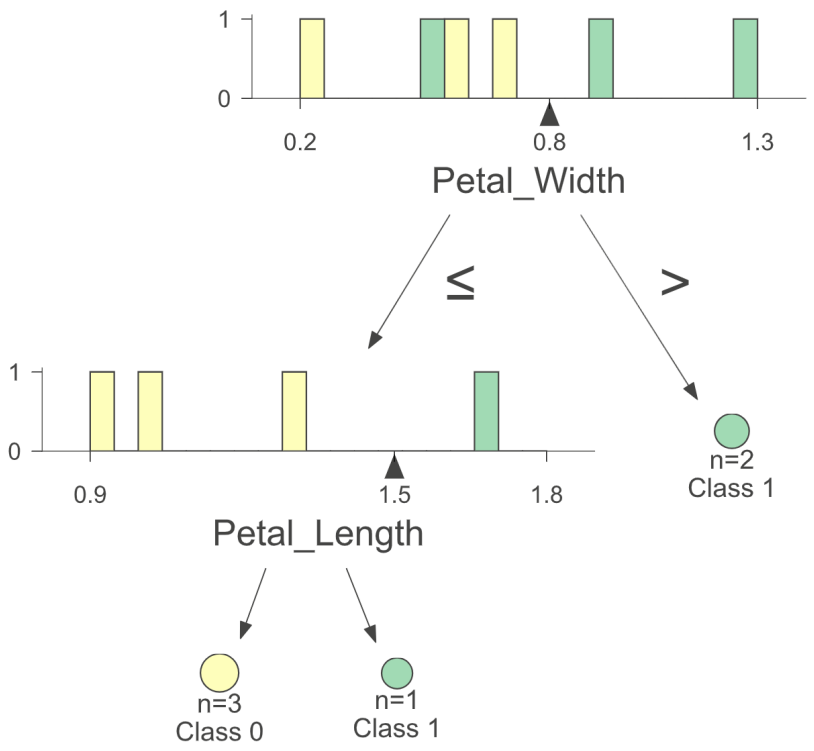
Negative: Decrease prediction

Both models have the same expected value E[f(X)] → same scale.

Top-3 features are the same across both models → we can trust this prediction.

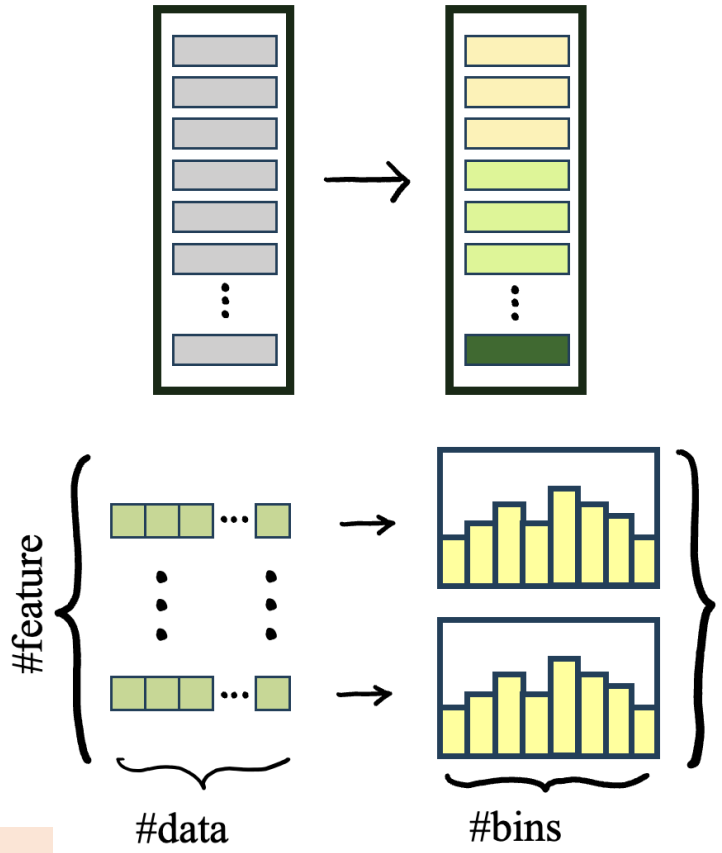
Summary

Dis. & Mov.



Quiz: https://forms.gle/JxuJPXyiC229DQxm9

Improvements



Case Studies