AI VIETNAM
All-in-One Course

Random Forest and AdaBoost (Warm-up Class)

Quang-Vinh Dinh
Ph.D. in Computer Science

Year 2023

Random Forest

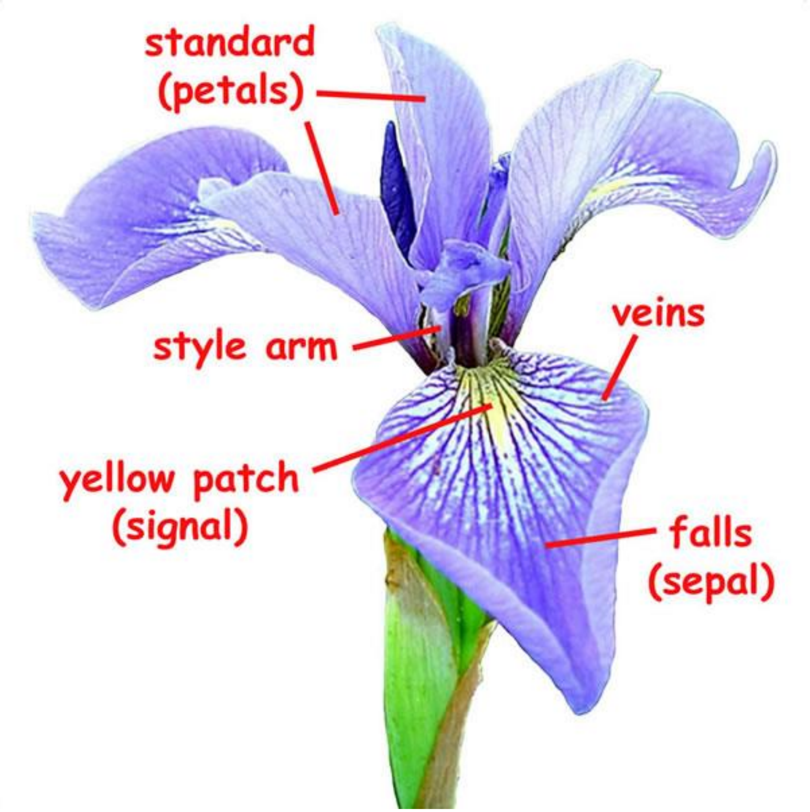
Quang-Vinh Dinh
Ph.D. in Computer Science

Year 2023

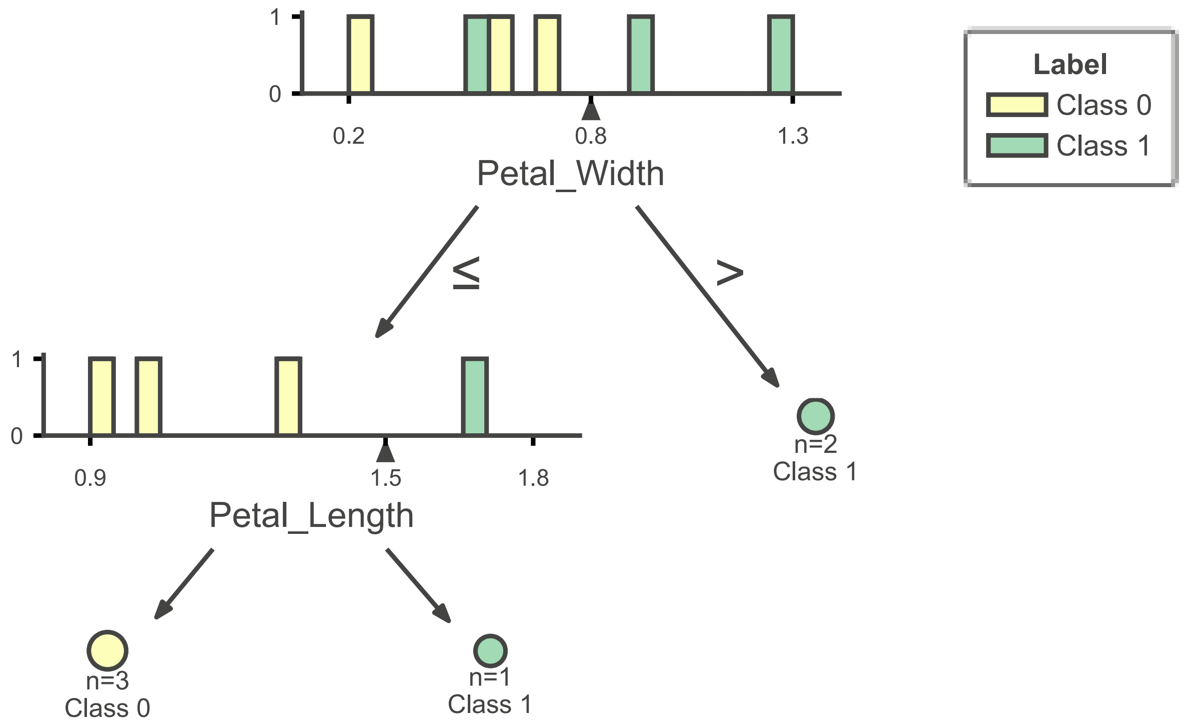
Decision Tree

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

❖ Observation



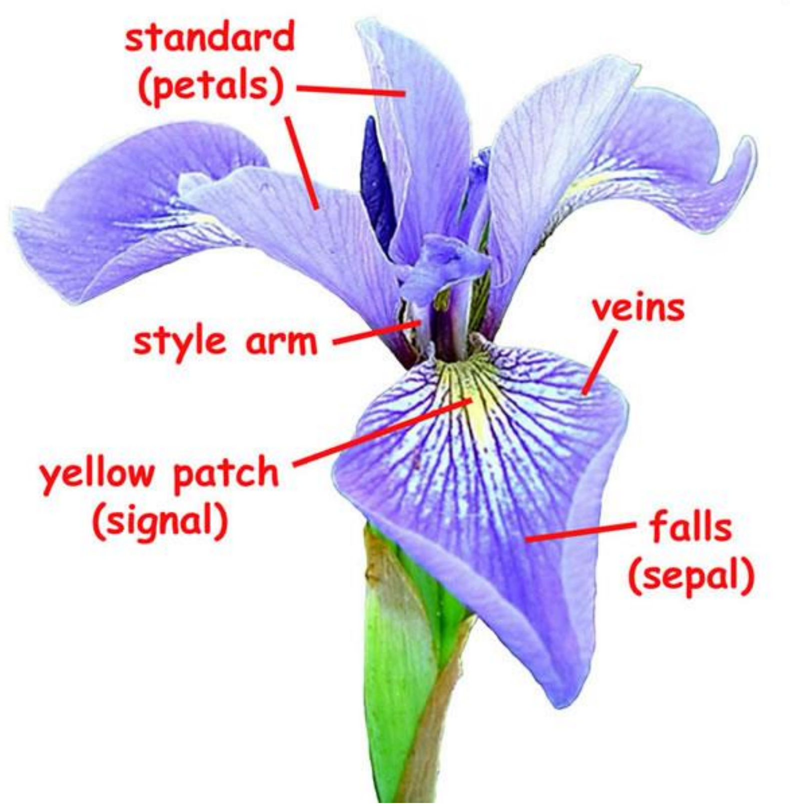
https://www.fs.usda.gov/wildflowers/beauty/iris/flower.shtml



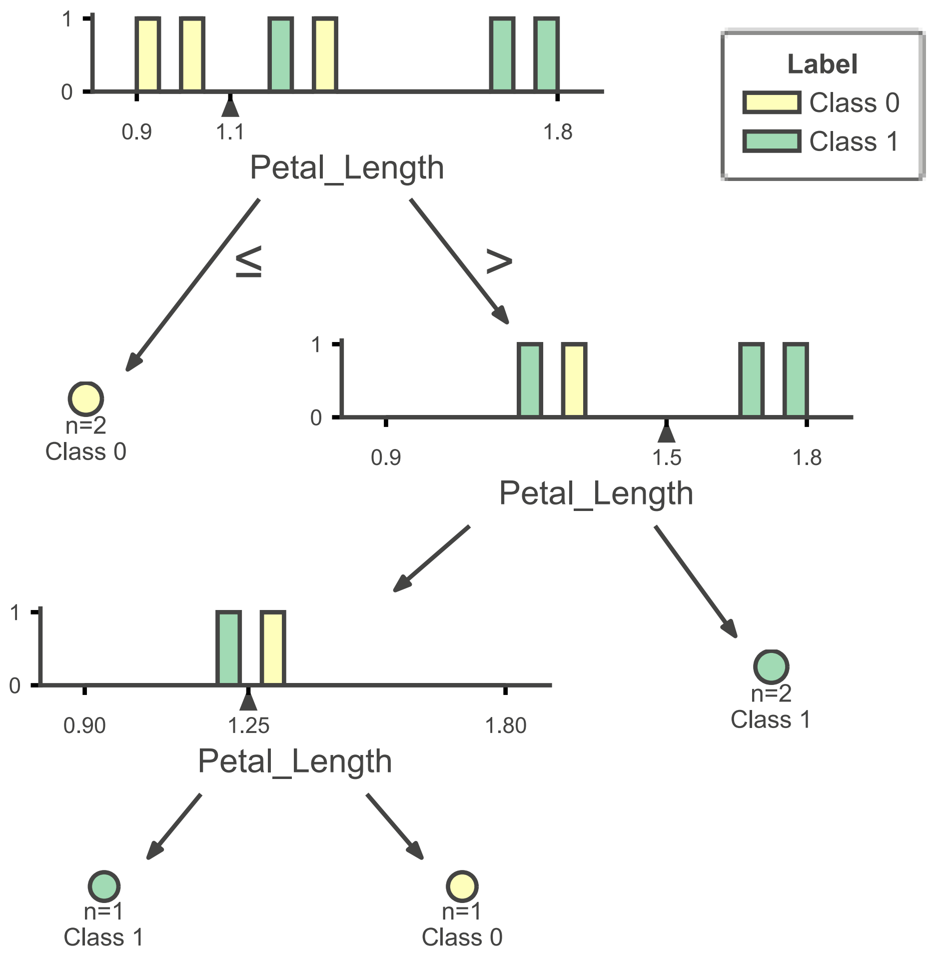
Decision Tree

❖ Observation

Petal_Length	Petal_Width	Sepal_length	Label
1	0.2	5.1	0
1.3	0.6	4.9	0
0.9	0.7	4.7	0
1.7	0.5	4.8	1
1.8	0.9	6.6	1
1.2	1.3	5.2	1



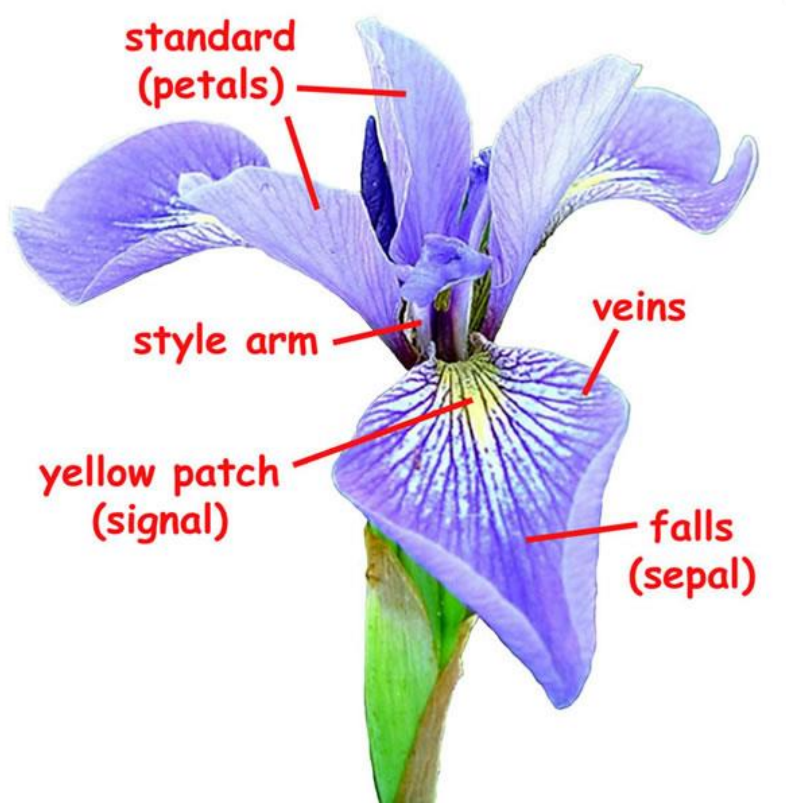
https://www.fs.usda.gov/wildflowers/beauty/iris/flower.shtml



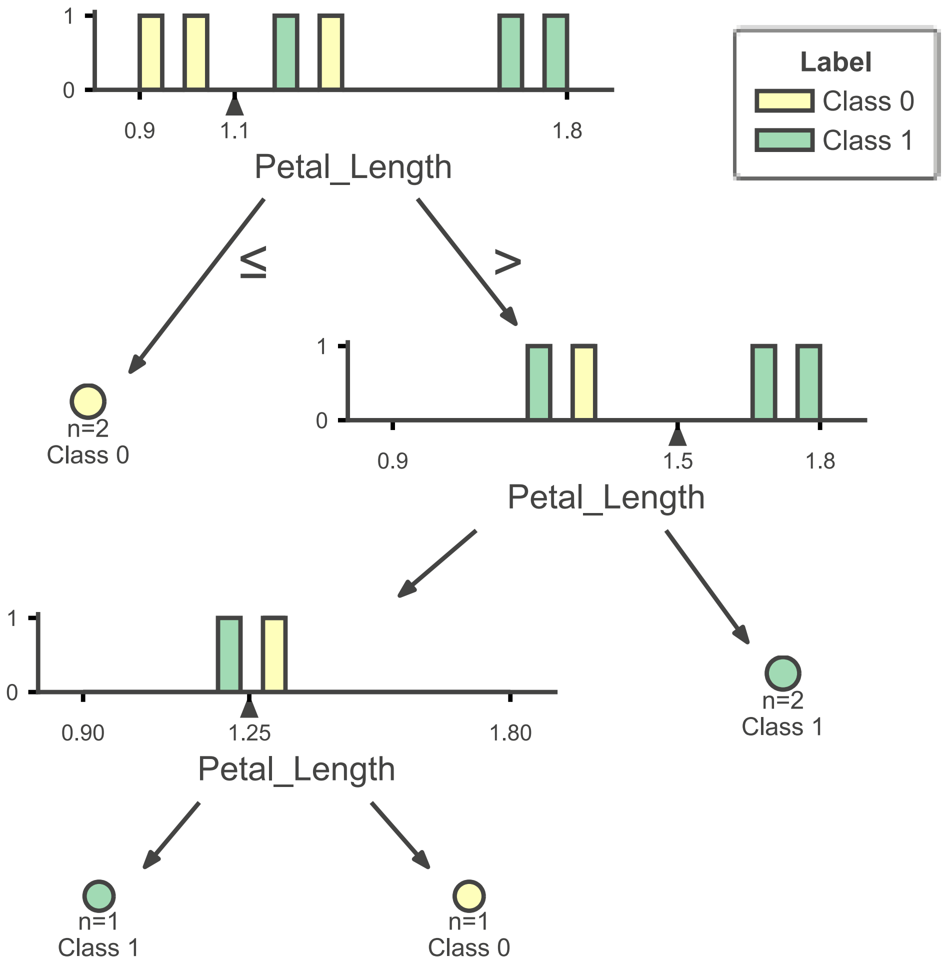
Decision Tree

Petal_Length	Petal_Width	Sepal_length	Sepal_Width	Label
1	0.2	5.1	3.5	0
1.3	0.6	4.9	3	0
0.9	0.7	4.7	3.2	0
1.7	0.5	4.8	2.8	1
1.8	0.9	6.6	3.3	1
1.2	1.3	5.2	2.4	1

❖ Observation



https://www.fs.usda.gov/wildflowers/
beauty/iris/flower.shtml



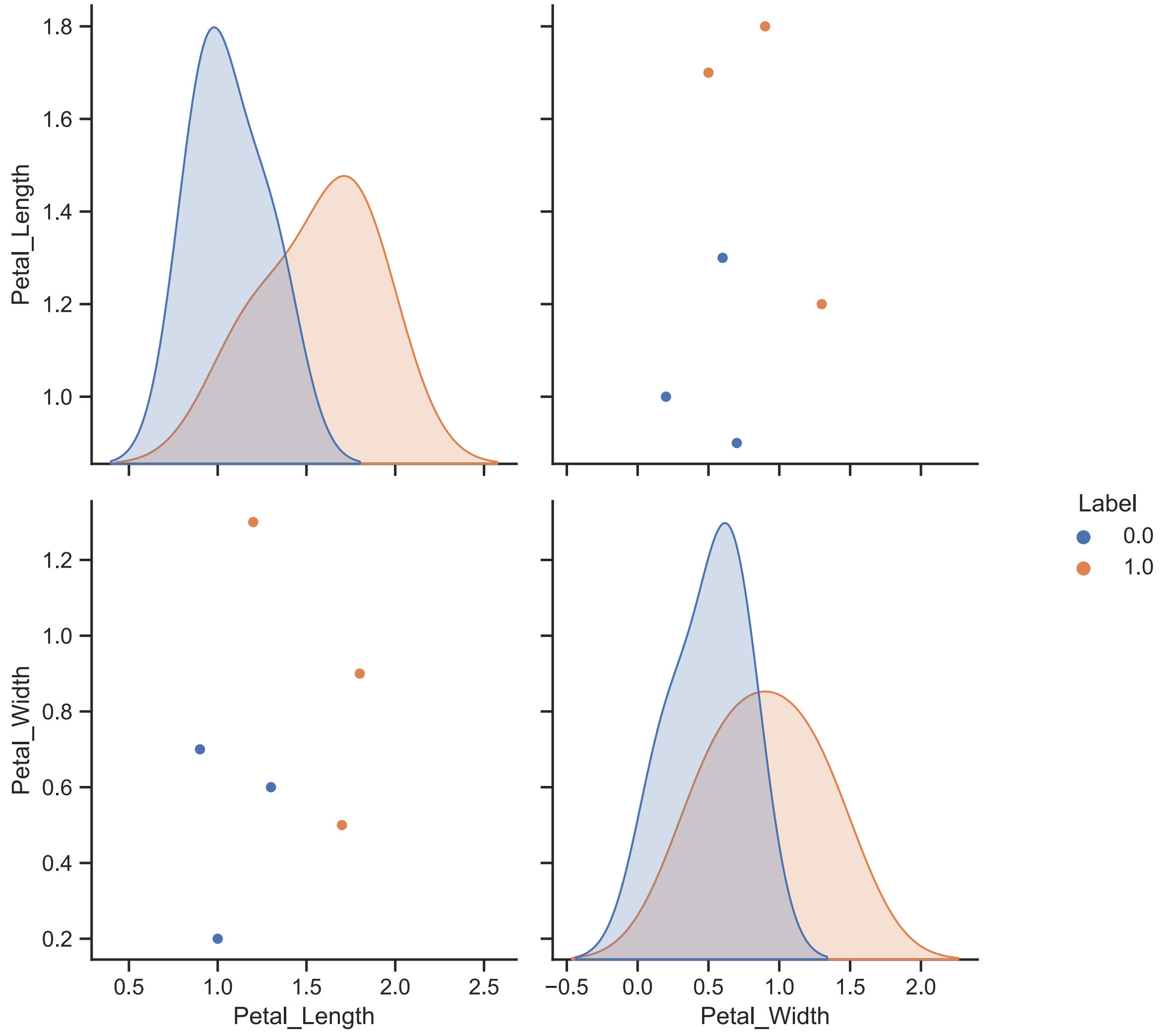
Discussion

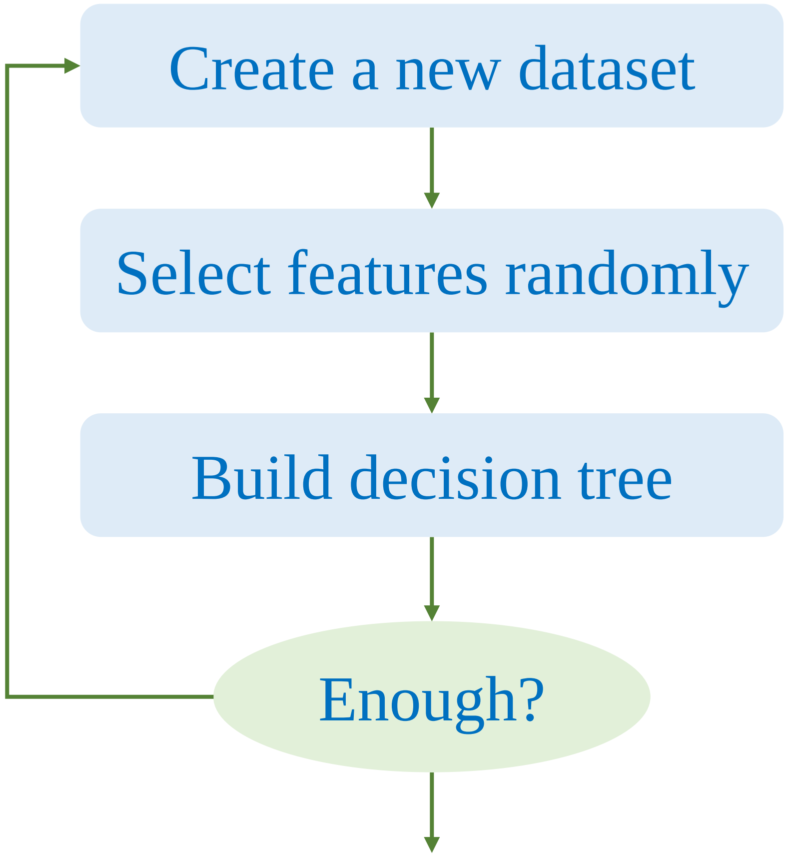
problem 1 $\Rightarrow$ problem 2

RF-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

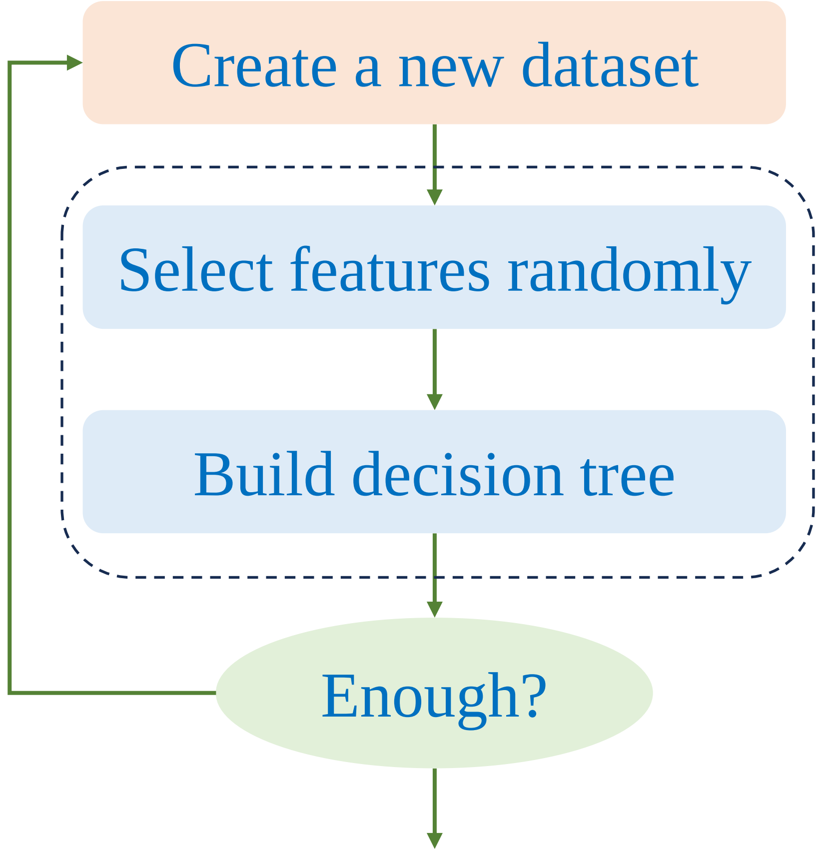




RF-Classification

❖ Simple IRIS

1



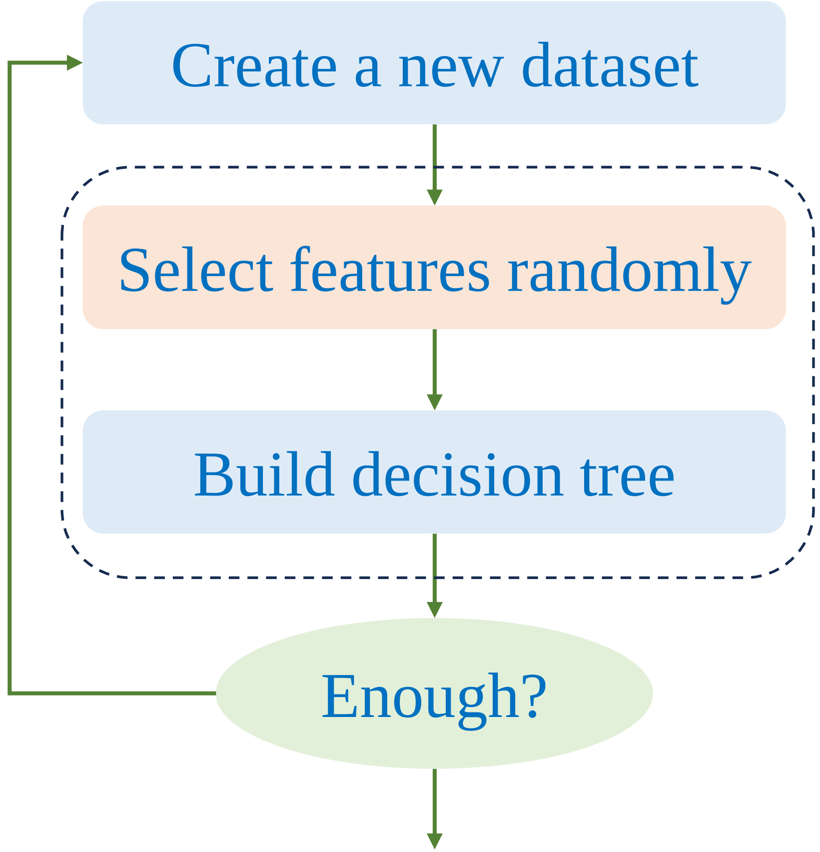
Petal_Length	Petal_Width	Label
1	0.2	0
1.3	0.6	0
0.9	0.7	0
1.7	0.5	1
1.8	0.9	1
1.2	1.3	1

Petal_Length	Petal_Width	Label
1	0.2	0
1.3	0.6	0
1	0.2	0
1.8	0.9	1
1.8	0.9	1
1.2	1.3	1

RF-Classification

❖ Simple IRIS

1



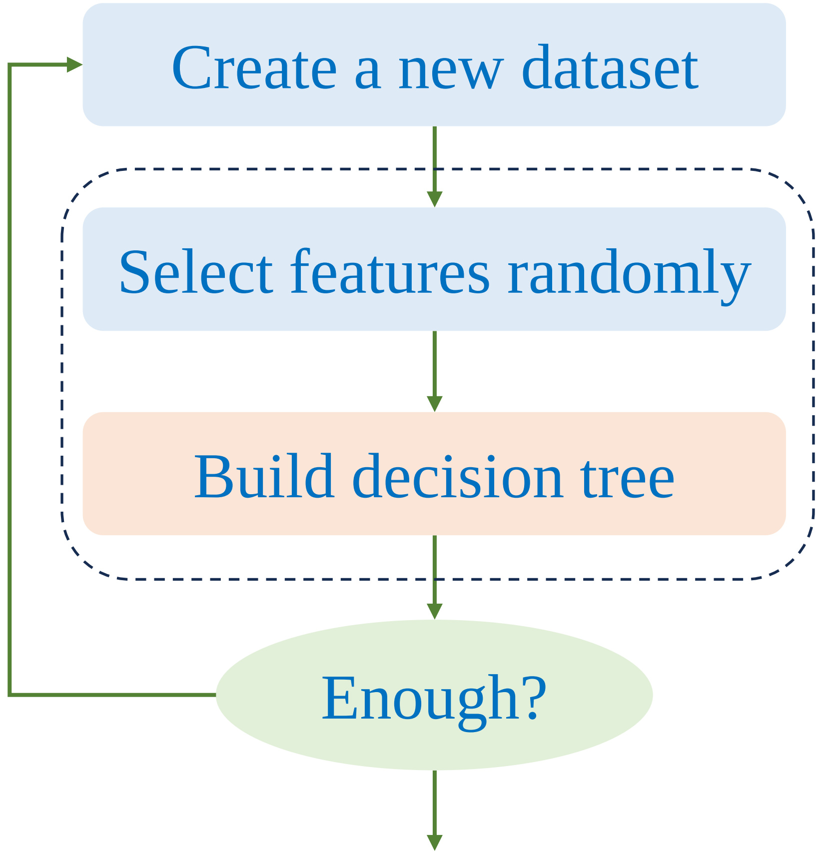
Petal_Length	Petal_Width	Label
1	0.2	0
1.3	0.6	0
0.9	0.7	0
1.7	0.5	1
1.8	0.9	1
1.2	1.3	1

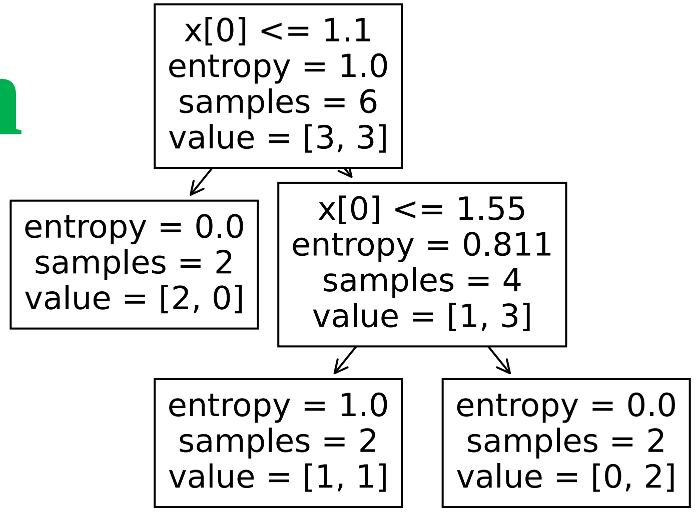
Petal_Length	Label
1	0
1.3	0
1	0
1.8	1
1.8	1
1.2	1

RF-Classification

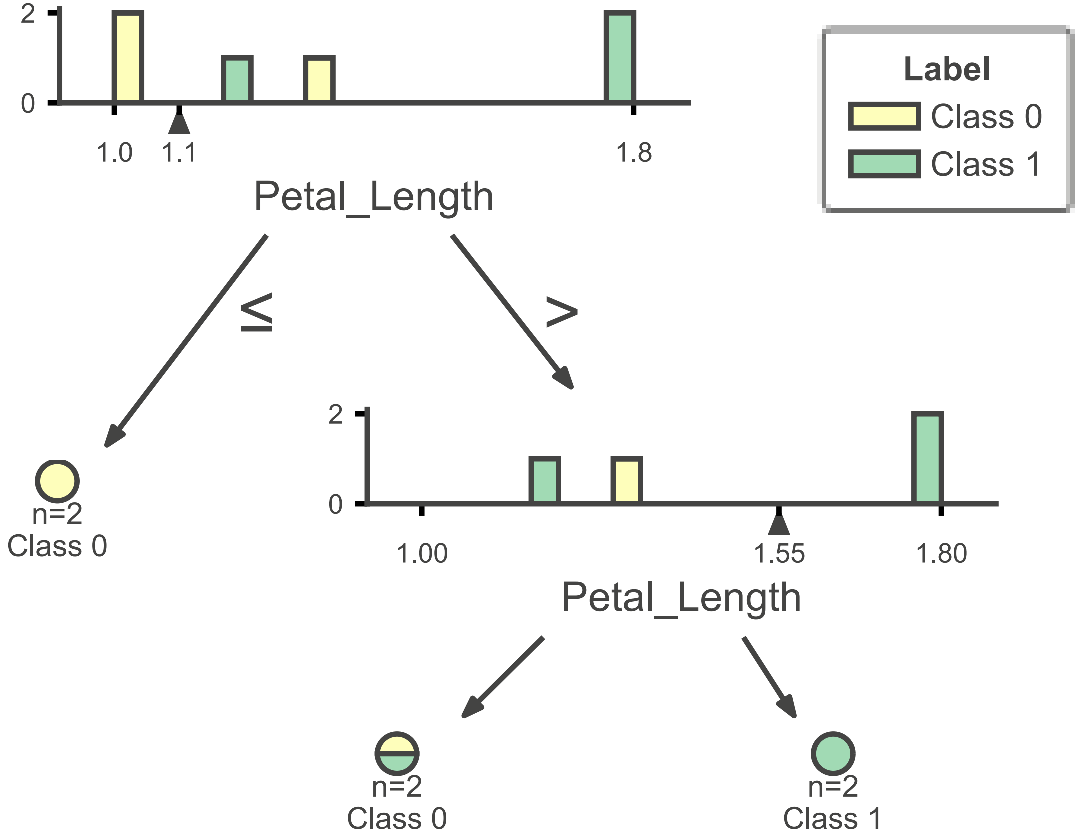
❖ Simple IRIS

1





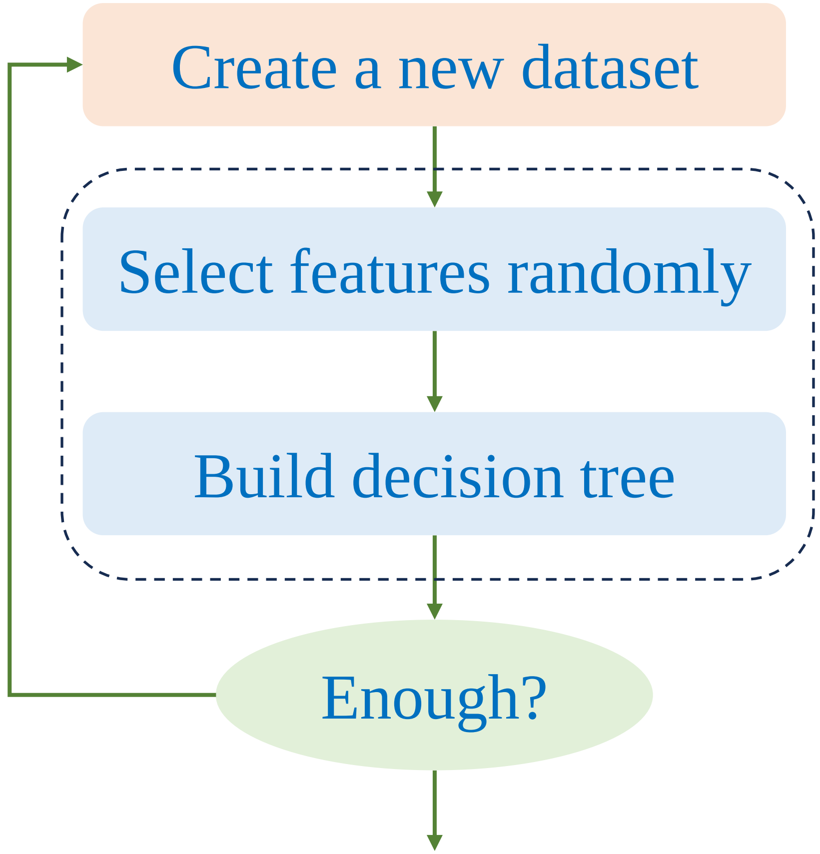
Petal_Length	Label
1	0
1.3	0
1	0
1.8	1
1.8	1
1.2	1



RF-Classification

❖ Simple IRIS

2



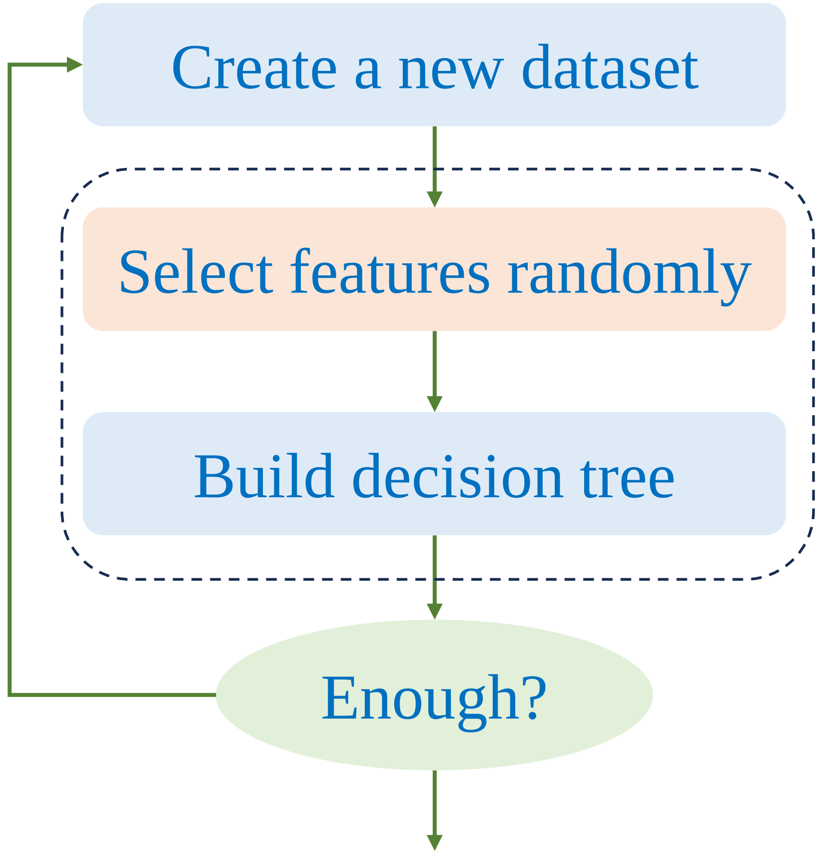
Petal_Length	Petal_Width	Label
1	0.2	0
1.3	0.6	0
0.9	0.7	0
1.7	0.5	1
1.8	0.9	1
1.2	1.3	1

Petal_Length	Petal_Width	Label
1.3	0.6	0
1.3	0.6	0
0.9	0.7	0
0.9	0.7	0
1.8	0.9	1
1.2	1.3	1

RF-Classification

❖ Simple IRIS

2



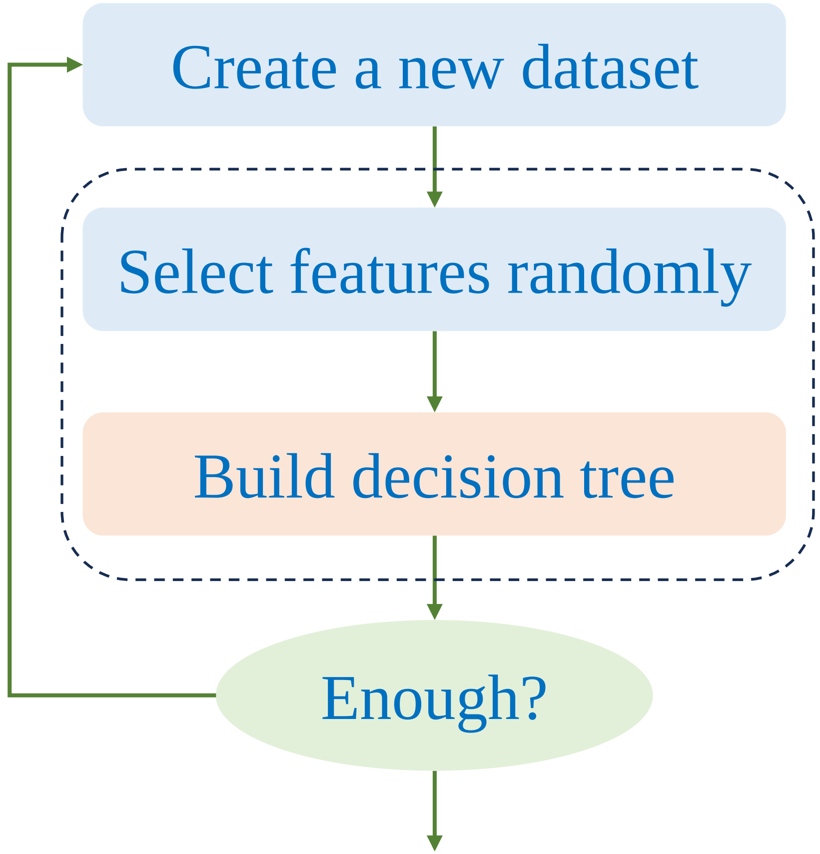
Petal_Length	Petal_Width	Label
1	0.2	0
1.3	0.6	0
0.9	0.7	0
1.7	0.5	1
1.8	0.9	1
1.2	1.3	1

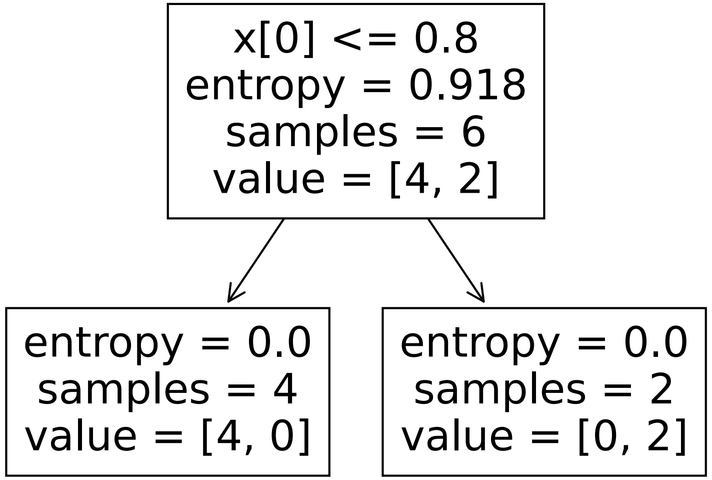
Petal_Width	Label
0.6	0
0.6	0
0.7	0
0.7	0
0.9	1
1.3	1

RF-Classification

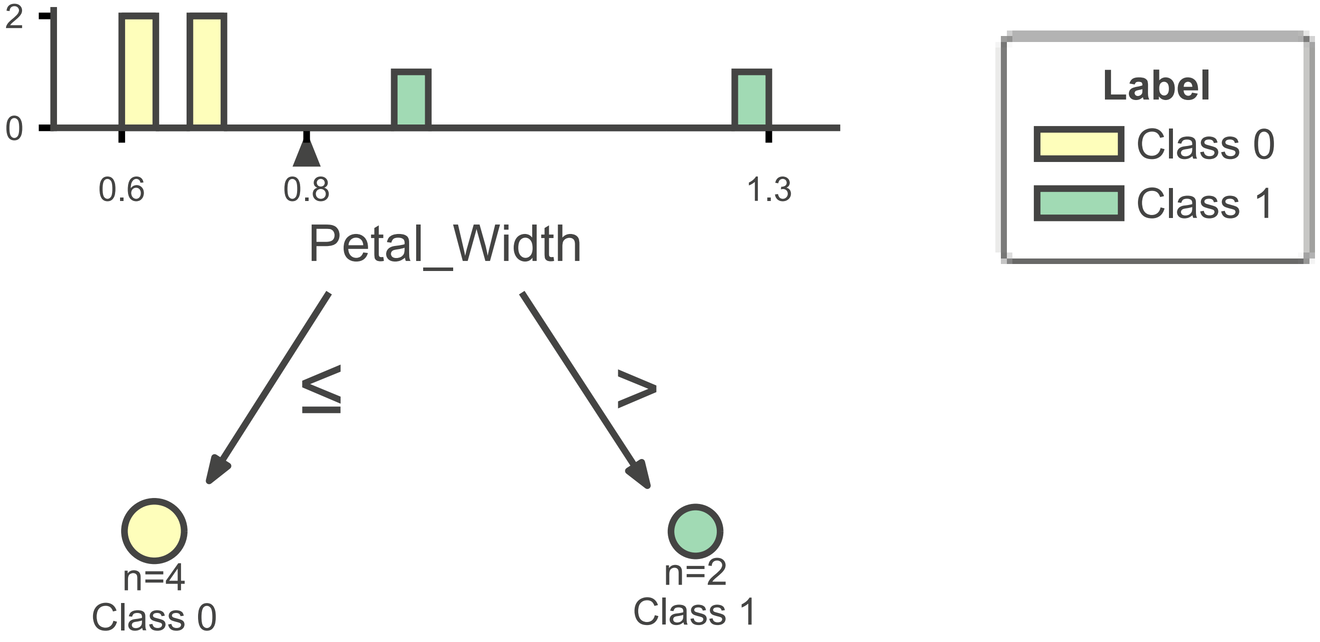
❖ Simple IRIS

2





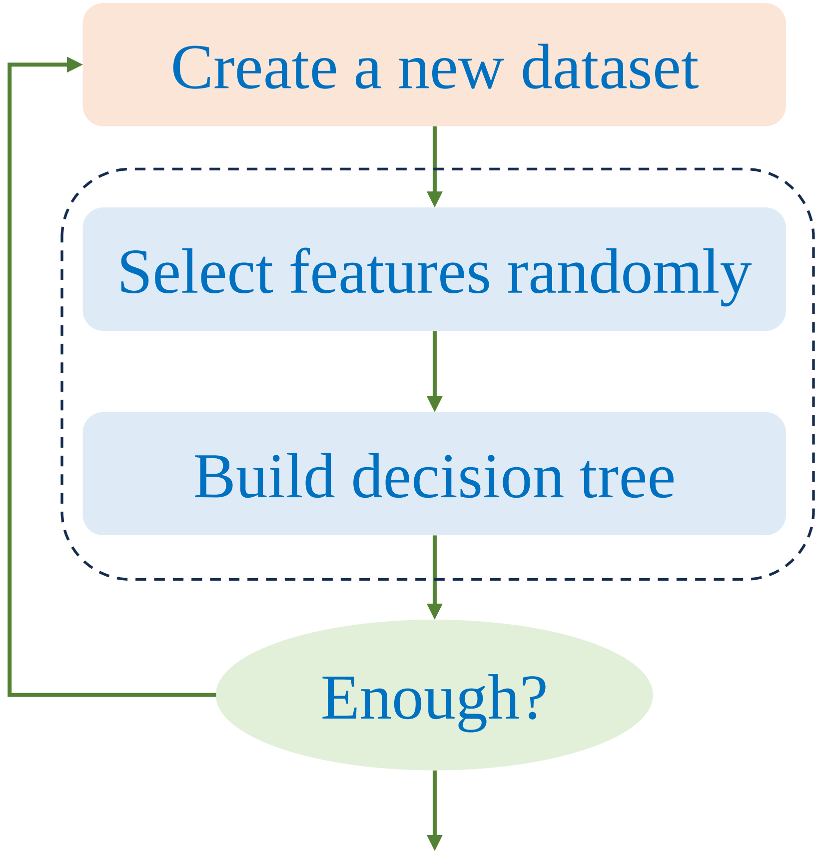
Petal_Width	Label
0.6	0
0.6	0
0.7	0
0.7	0
0.9	1
1.3	1



RF-Classification

❖ Simple IRIS

3



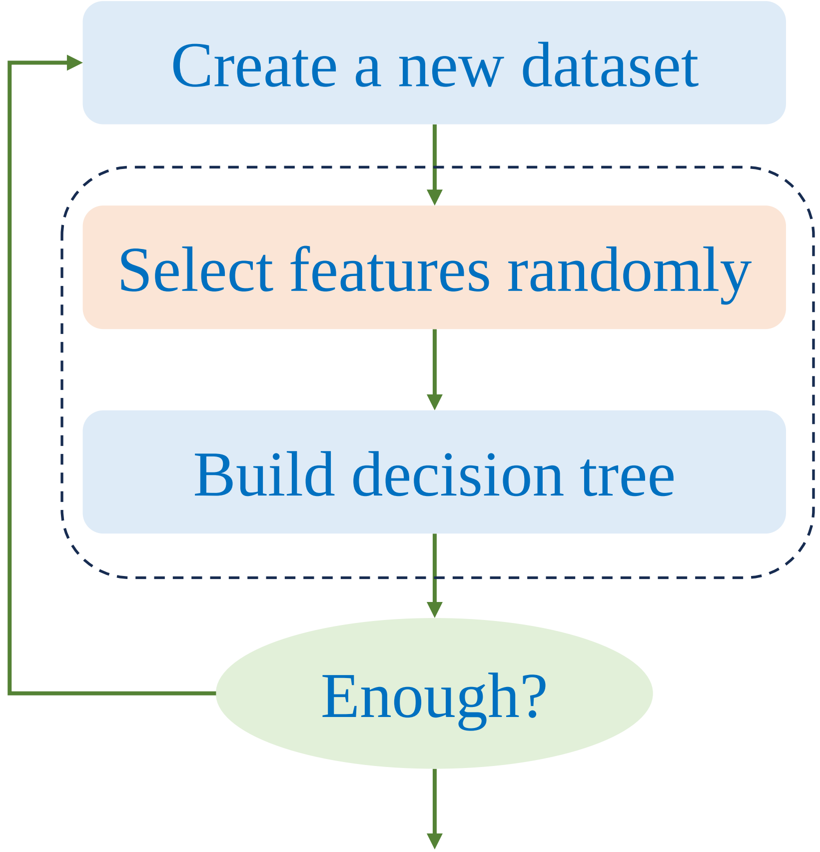
Petal_Length	Petal_Width	Label
1	0.2	0
1.3	0.6	0
0.9	0.7	0
1.7	0.5	1
1.8	0.9	1
1.2	1.3	1

Petal_Length	Petal_Width	Label
1	0.2	0
1.3	0.6	0
1.2	1.3	1
1.8	0.9	1
1.8	0.9	1
1.2	1.3	1

RF-Classification

❖ Simple IRIS

3



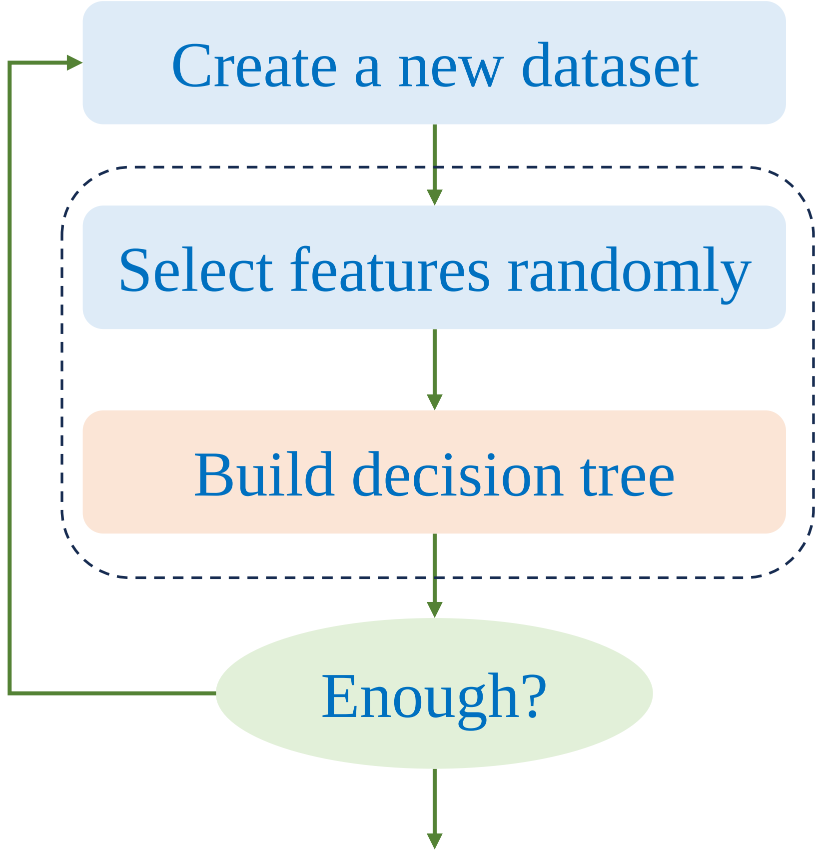
Petal_Length	Petal_Width	Label
1	0.2	0
1.3	0.6	0
0.9	0.7	0
1.7	0.5	1
1.8	0.9	1
1.2	1.3	1

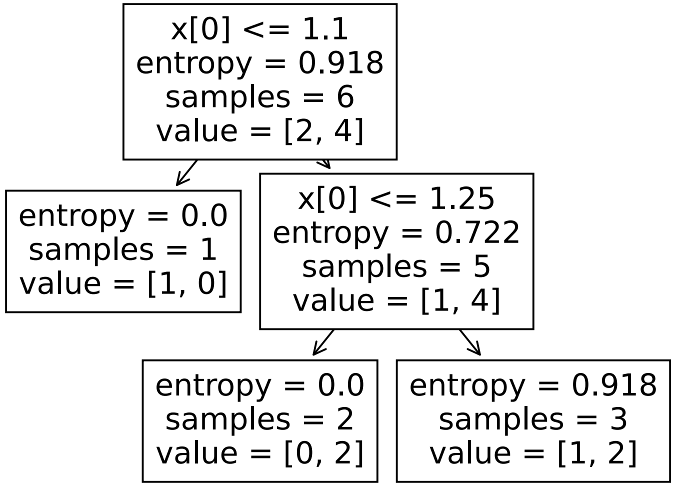
Petal_Length	Label
1	0
1.3	0
1.2	1
1.8	1
1.8	1
1.2	1

RF-Classification

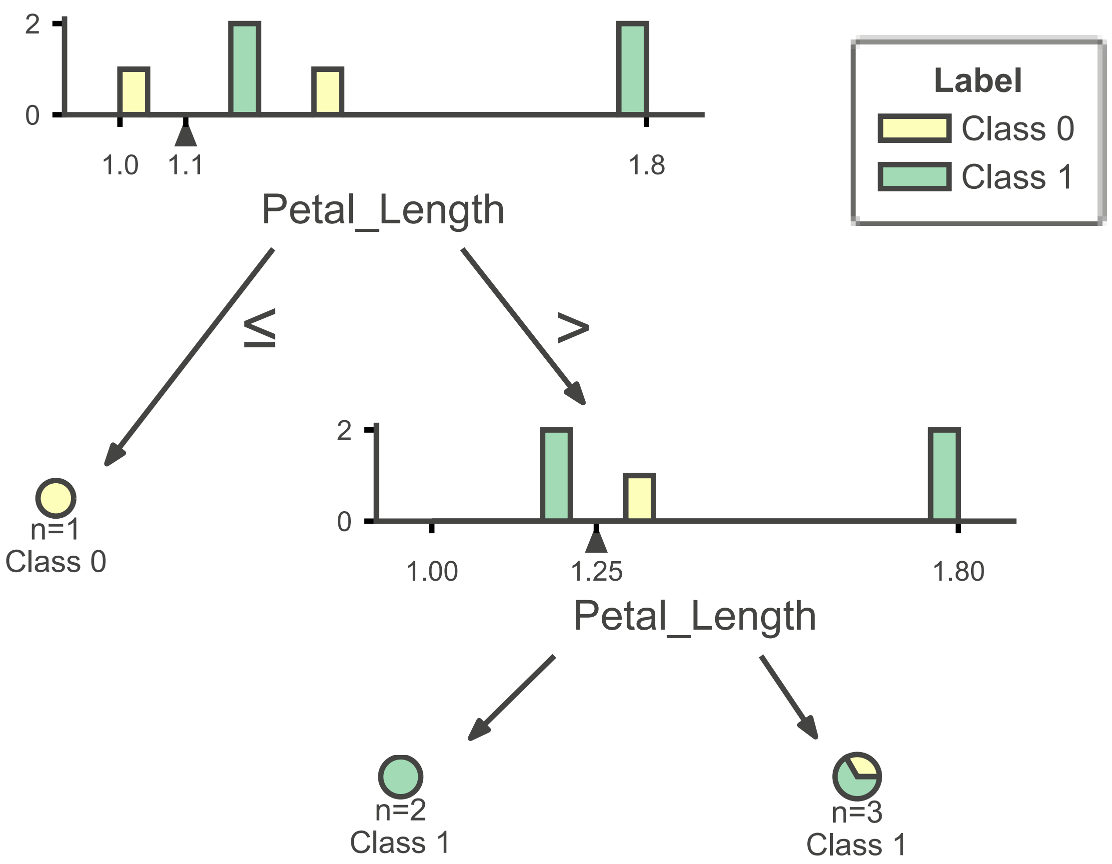
❖ Simple IRIS

3





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



RF-Classification

❖ Simple IRIS

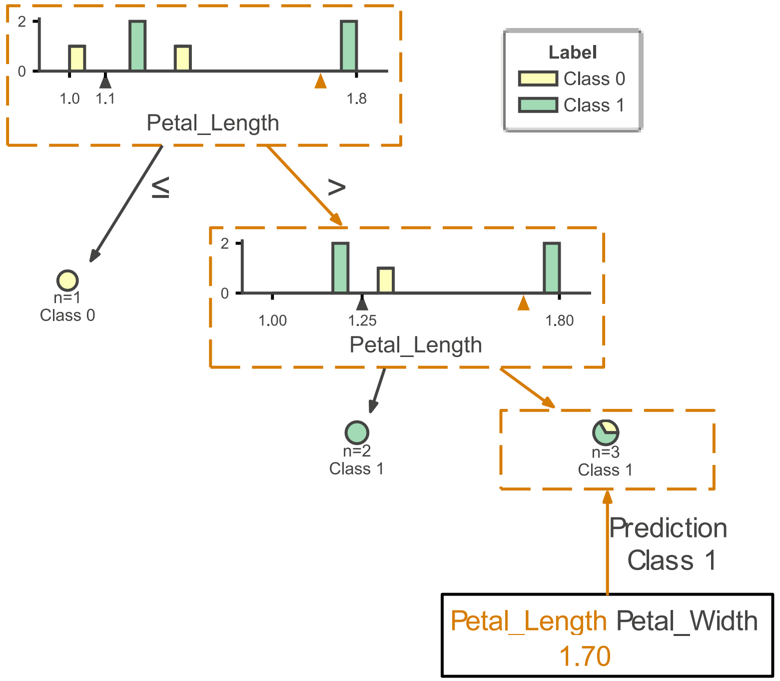
Petal_Length	Petal_Width	Label
1	0.2	0
1.3	0.6	0
0.9	0.7	0
1.7	0.5	1
1.8	0.9	1
1.2	1.3	1

inference

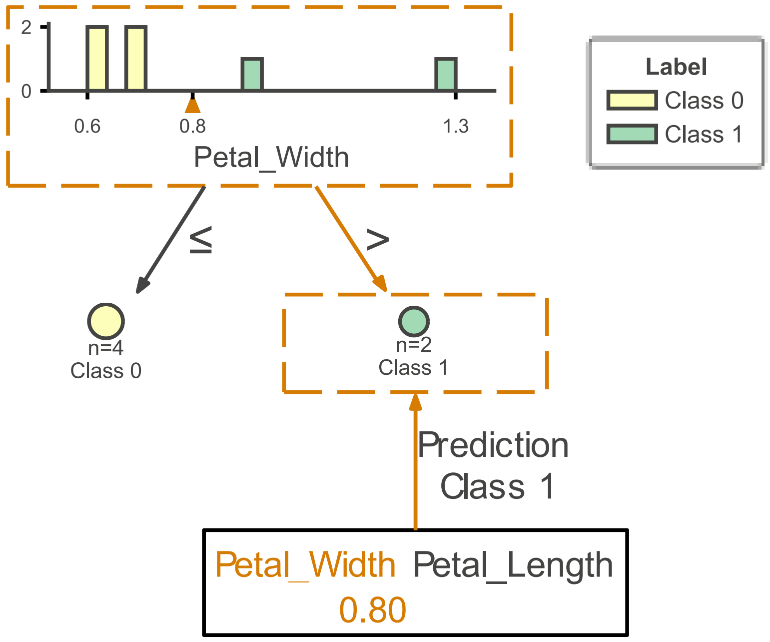
Petal_Length = 1.7

Petal_Width = 0.8

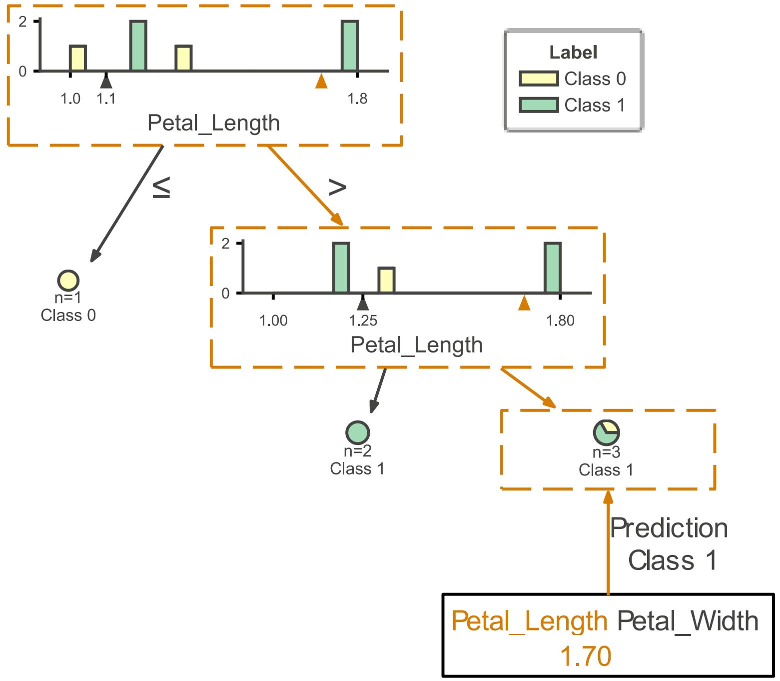
1



2



3



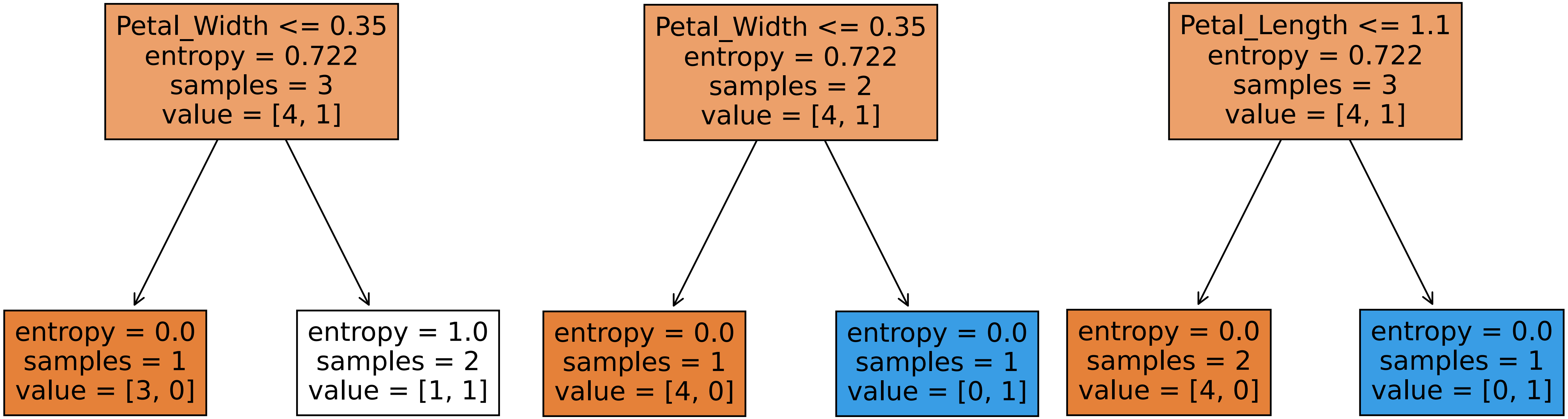
RF-Classification

❖ **Using sklearn**

Another experiment

```
rf_classifier = RandomForestClassifier(n_estimators=3,
                                       max_features=1,
                                       max_depth=1,
                                       criterion='entropy',
                                       max_samples=5)
rf_classifier.fit(x_data, y_train)
rf_classifier.predict(np.array([[2.7, 0.8]]))
```

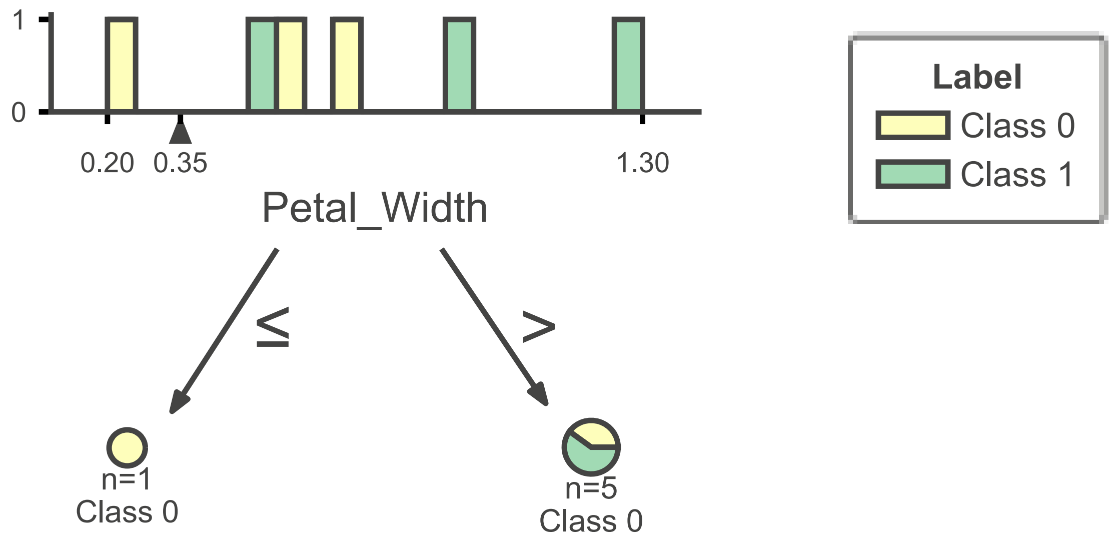
```
array([0])
```

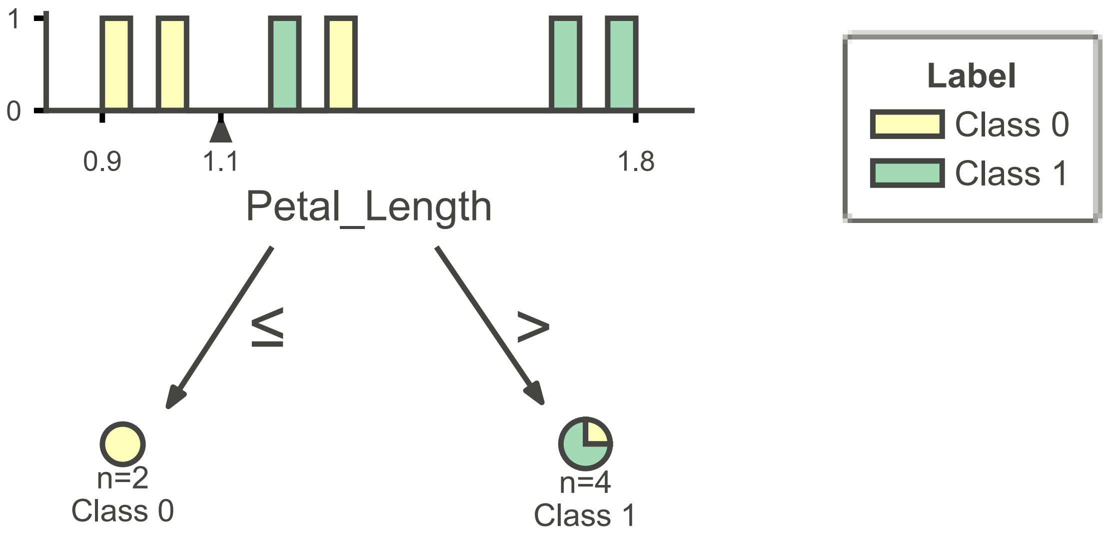


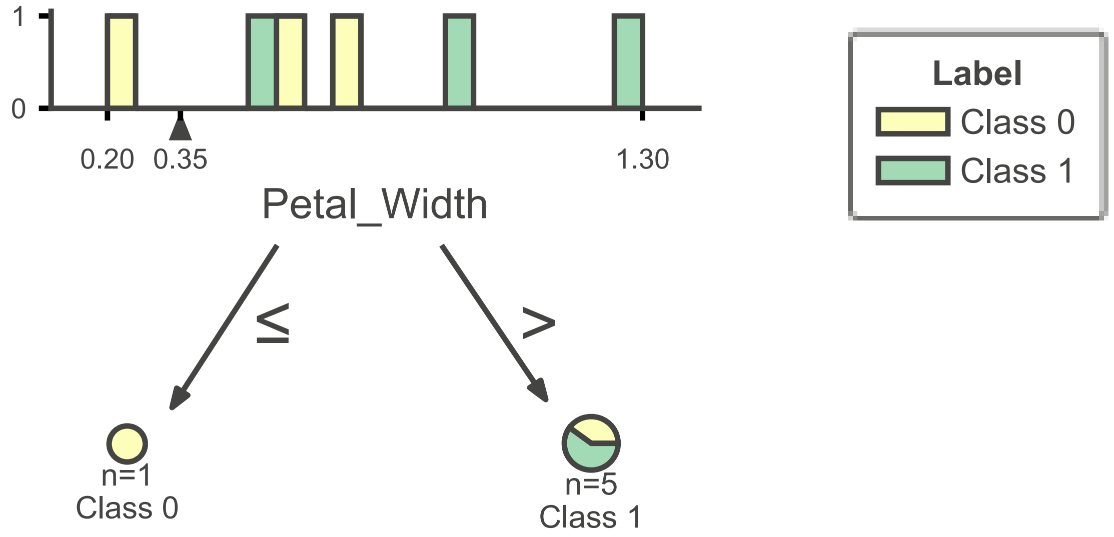
RF-Classification

❖ **Using sklearn**

Using all the training samples







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

Entropy:

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

Information Gain

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

Decision Tree

$$S = \{9{:}\,Yes, 5{:}\,No\} \longrightarrow E(S) = -\frac{9}{14} log_2\left(\frac{9}{14}\right) - \frac{5}{14} log_2\left(\frac{5}{14}\right) = 0.94$$

$$S_{weak} = \{6{:}\,Yes, 2{:}\,No\} \longrightarrow E(S_{weak}) = -\frac{6}{8} log_2\left(\frac{6}{8}\right) - \frac{2}{8} log_2\left(\frac{6}{8}\right) = 0.811$$

$$S_{Strong} = \{3{:}\,Yes, 3{:}\,No\} \longrightarrow E(S_{Strong}) = -\frac{3}{6} log_2\left(\frac{3}{6}\right) - \frac{3}{6} log_2\left(\frac{3}{6}\right) = 1$$

$$\Rightarrow Gain(S, Wind) = E(S) - \frac{8}{14} E(S_{weak}) - \frac{6}{14} E(S_{Strong})$$

Category = 2

$$= 0.94 - \frac{8}{14} * 0.811 - \frac{6}{14} * 1 = 0.048$$

Category = 3 > 2 → **Combine** →
Option_1: Sunny - (Overcast, Rain)
Option_2: Overcast - (Sunny, Rain)
Option_3: Rain – (Sunny, Overcast)

$$\text{Gain(S, Outlook)} = \max \begin{cases} IG(S, Option\_1) = 0.102 \\ IG(S, Option\_2) = 0.226 \\ IG(S, Option\_3) = 0.003 \end{cases}$$

$$S_{Sunny} = \{2{:}\,Yes, 3{:}\,No\} \longrightarrow E(S_{Sunny}) = 0.97$$

$$S_{Overcast,Rain} = \{7{:}\,Yes, 2{:}\,No\} \longrightarrow E(S_{Overcast,Rain}) = 0.764$$

$$IG(S, Option\_1)$$

$$= E(S) - \frac{5}{14} E(S_{Sunny}) - \frac{9}{14} E(S_{Overcast,Rain})$$

$$= 0.94 - \frac{5}{14} * 0.97 - \frac{9}{14} * 0.764 = 0.102$$

<u>Gain(S, Outlook) = 0.226</u>

Gain(S, Temp) = 0.015

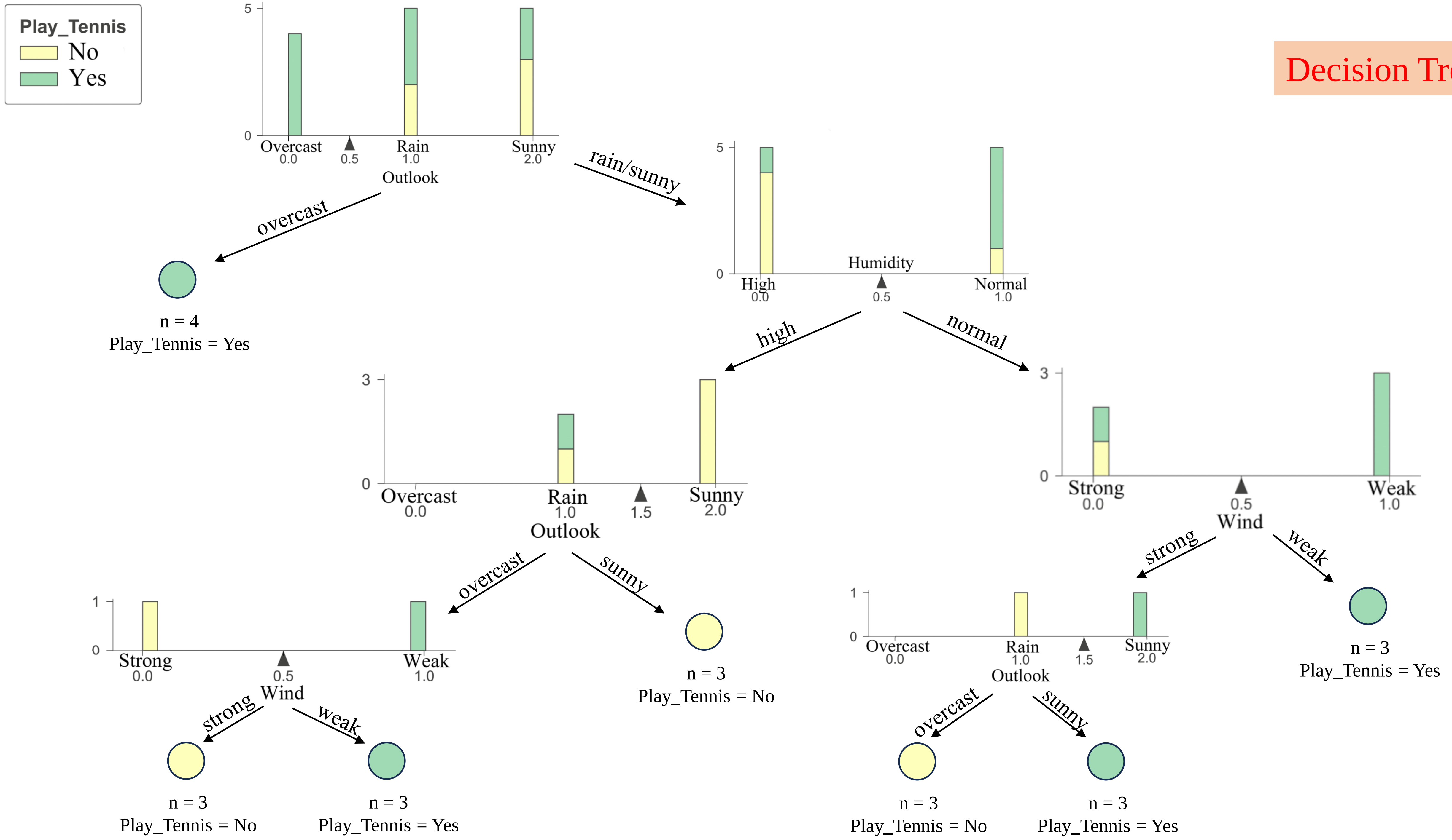
Gain(S, Humidity) = 0.151

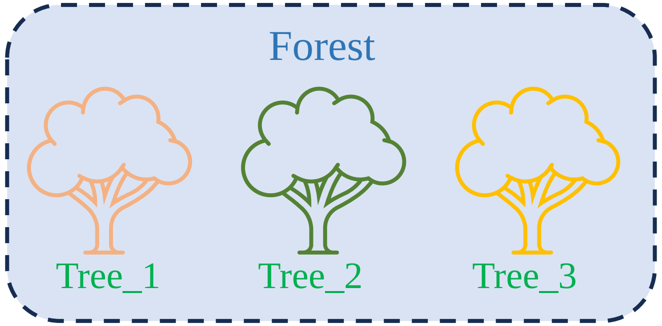
Gain(S, Wind) = 0.048

Choose Outlook with highest Gain score for root node

Option_2 is used to split

Decision Tree





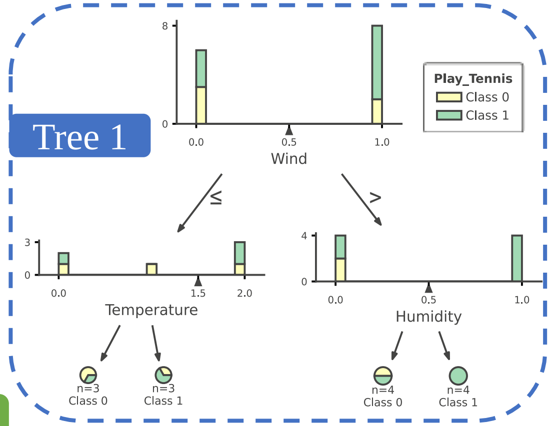
Outlook	Temp	Humidity	Wind	Label
Sunny	Hot	High	Weak	No
Sunny	Hot	High	Strong	No
Overcast	Hot	High	Weak	Yes
Rain	Mild	High	Weak	Yes
Rain	Cool	Normal	Weak	Yes
Rain	Cool	Normal	Strong	No
Overcast	Cool	Normal	Strong	Yes
Sunny	Mild	High	Weak	No
Sunny	Cool	Normal	Weak	Yes
Rain	Mild	Normal	Weak	Yes
Sunny	Mild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

n_sample = 10

Bootstrapping (random sampling with replacement)

Outlook	Temp	Humidity	Wind	Label
Sunny	Hot	High	Strong	No
Rain	Cool	Normal	Weak	Yes
Rain	Cool	Normal	Strong	No
Sunny	Mild	High	Weak	No
Sunny	Cool	Normal	Weak	Yes
Rain	Mild	Normal	Weak	Yes
Sunny	Mild	Normal	Strong	Yes
Sunny	Mild	Normal	Strong	Yes
Sunny	Mild	Normal	Strong	Yes
Sunny	Mild	Normal	Strong	Yes

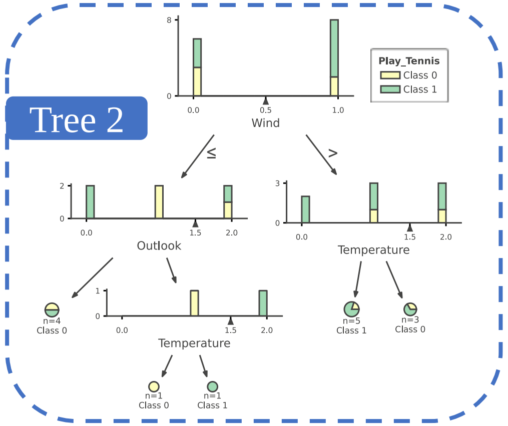
Build tree



replacement

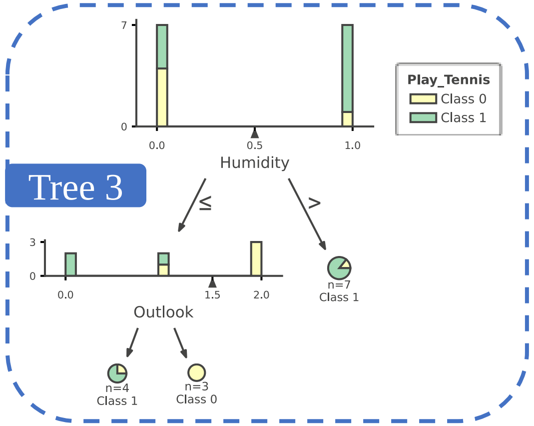
Outlook	Temp	Humidity	Wind	Label
Sunny	Hot	High	Strong	No
Overcast	Hot	High	Weak	Yes
Overcast	Hot	High	Weak	Yes
Rain	Cool	Normal	Strong	No
Sunny	Mild	High	Weak	No
Sunny	Mild	High	Weak	No
Sunny	Cool	Normal	Weak	Yes
Sunny	Mild	Normal	Strong	Yes
Overcast	Hot	Normal	Weak	Yes
Rain	Mild	High	Strong	No

Build tree



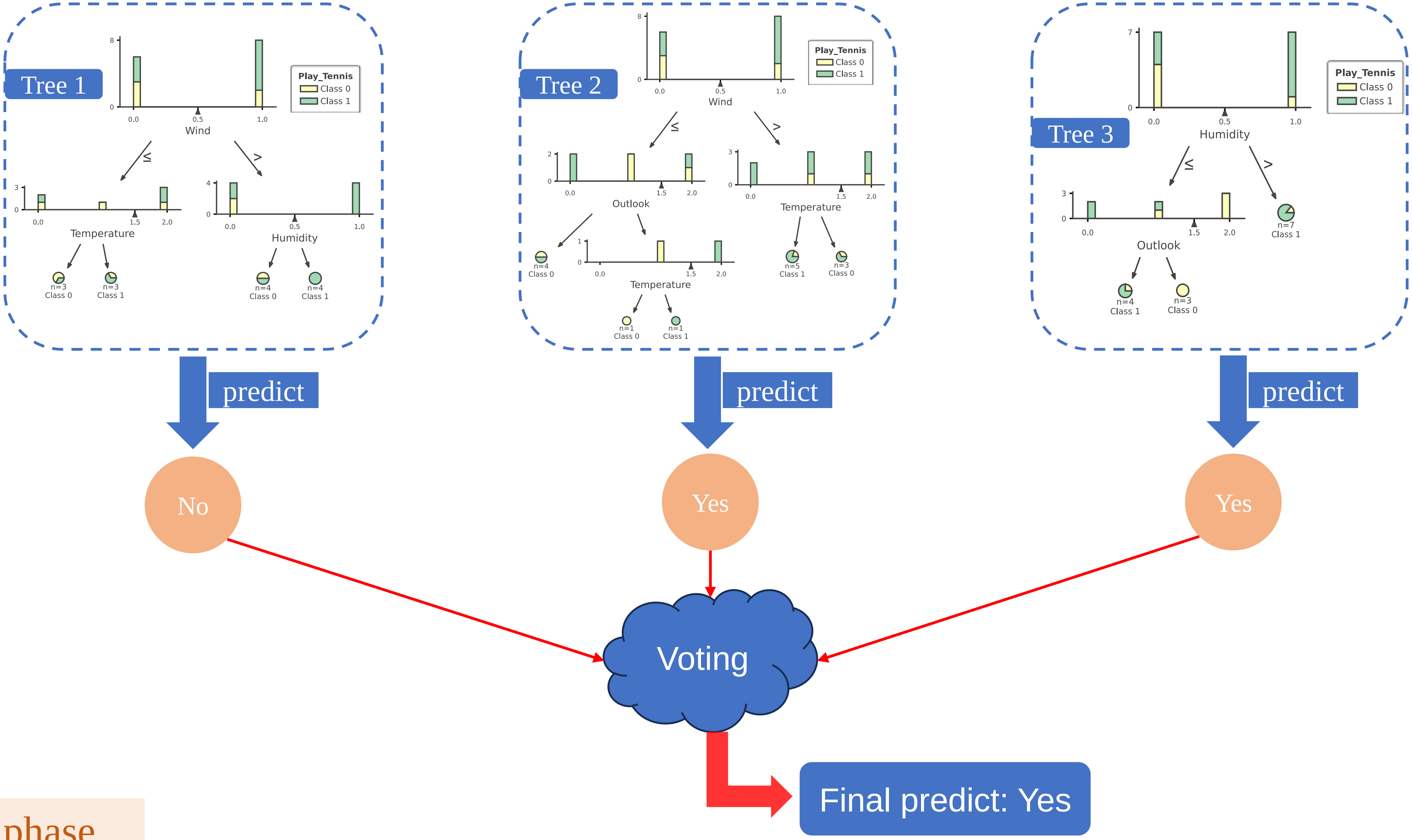
Outlook	Temp	Humidity	Wind	Label
Sunny	Hot	High	Weak	No
Rain	Mild	High	Weak	Yes
Rain	Mild	High	Weak	Yes
Overcast	Cool	Normal	Strong	Yes
Overcast	Cool	Normal	Strong	Yes
Rain	Mild	Normal	Weak	Yes
Rain	Mild	Normal	Weak	Yes
Overcast	Mild	High	Strong	Yes
Overcast	Hot	Normal	Weak	Yes
Overcast	Hot	Normal	Weak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

Build tree



Training phase

Test = <outlook=Sunny, temperature=Hot, humidity=High, Wind=Weak>



Test phase

Using sklearn

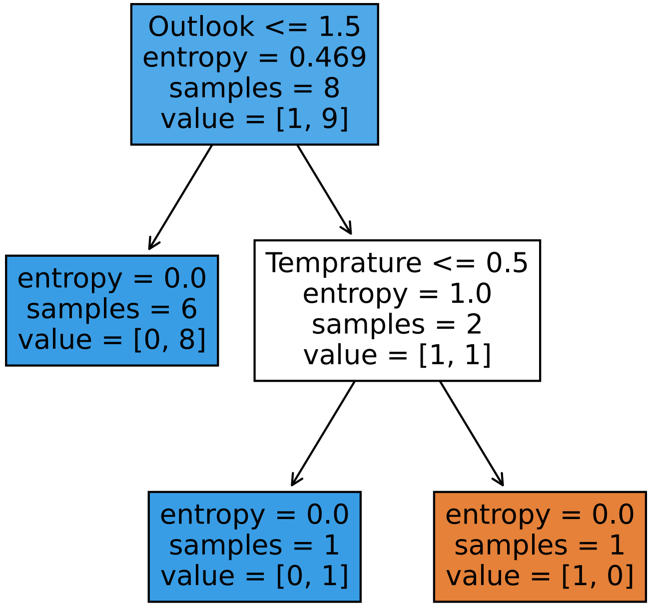
Another run

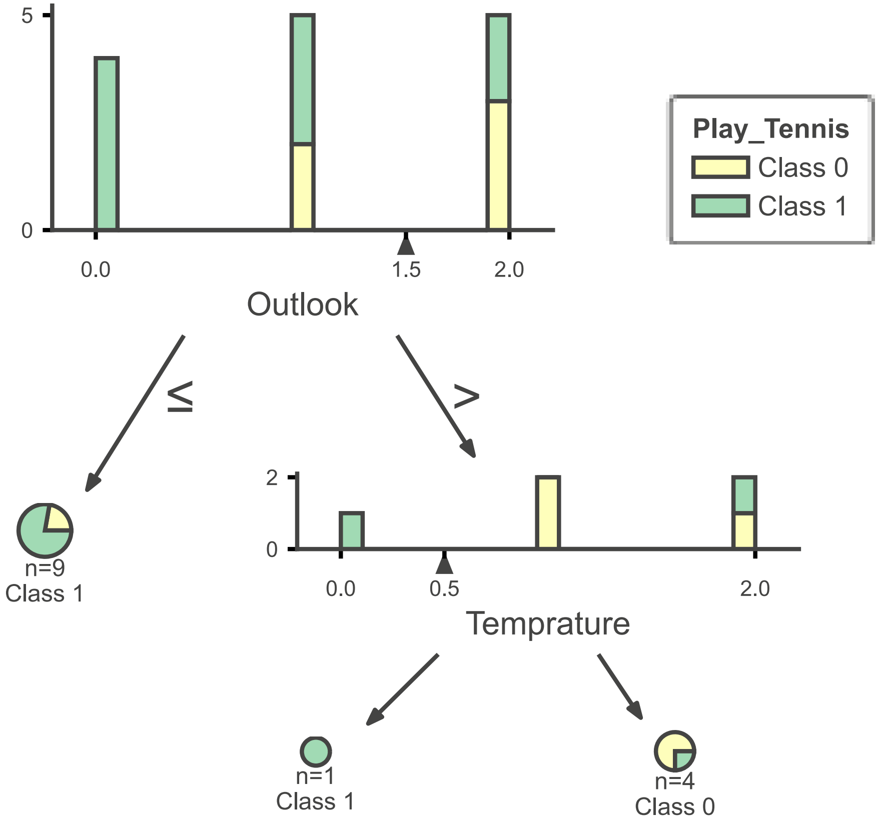
```
from sklearn.ensemble import RandomForestClassifier

classifier = RandomForestClassifier(n_estimators=3,
                                    max_features=2,
                                    criterion='entropy',
                                    max_samples=10)
classifier.fit(X, y)
```

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

Tree 1





Using sklearn

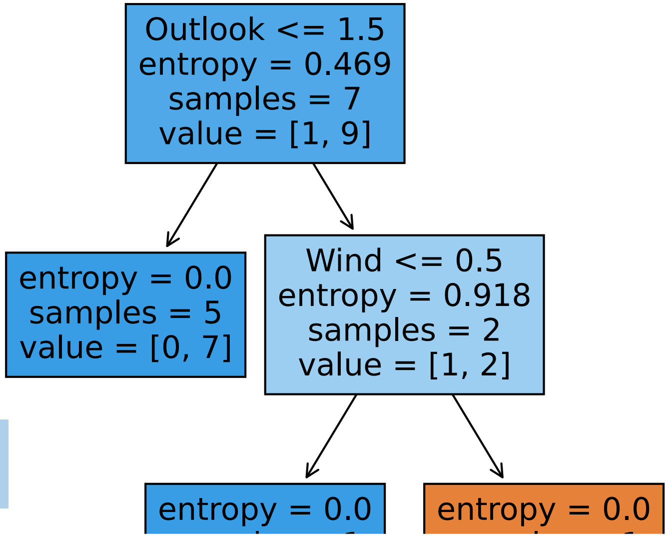
Another run

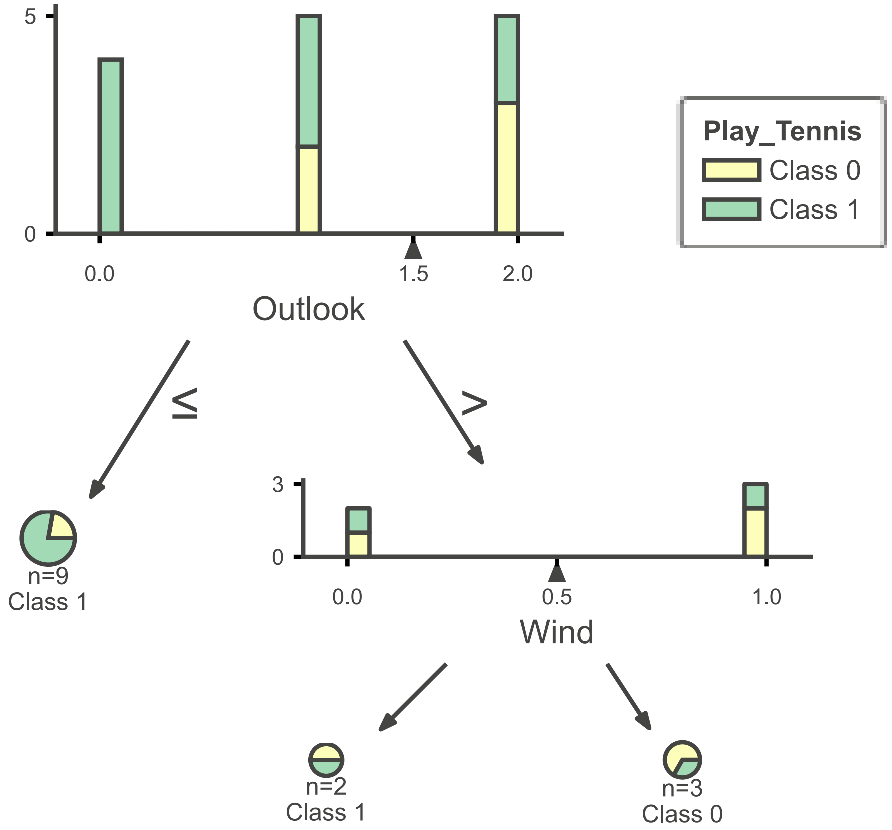
```python
from sklearn.ensemble import RandomForestClassifier

classifier = RandomForestClassifier(n_estimators=3,
                                    max_features=2,
                                    criterion='entropy',
                                    max_samples=10)
classifier.fit(X, y)
```

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

Tree 2



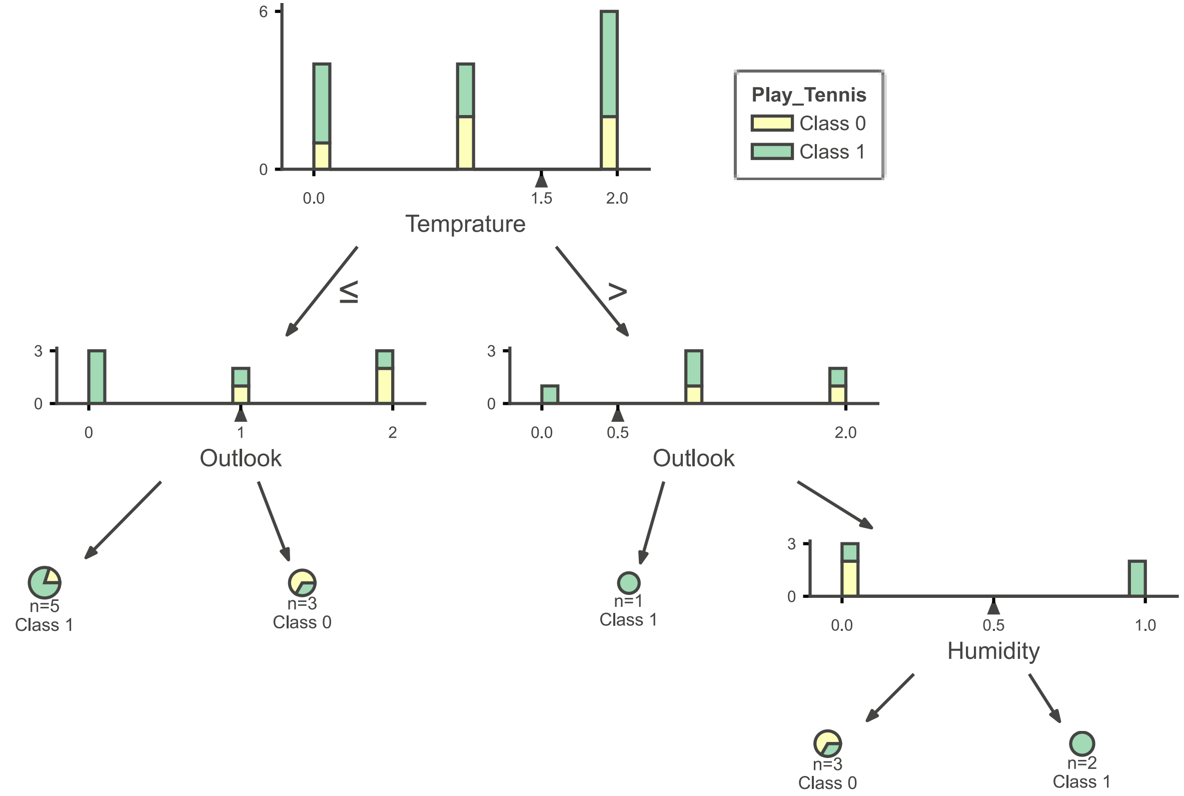


Using sklearn

Another run

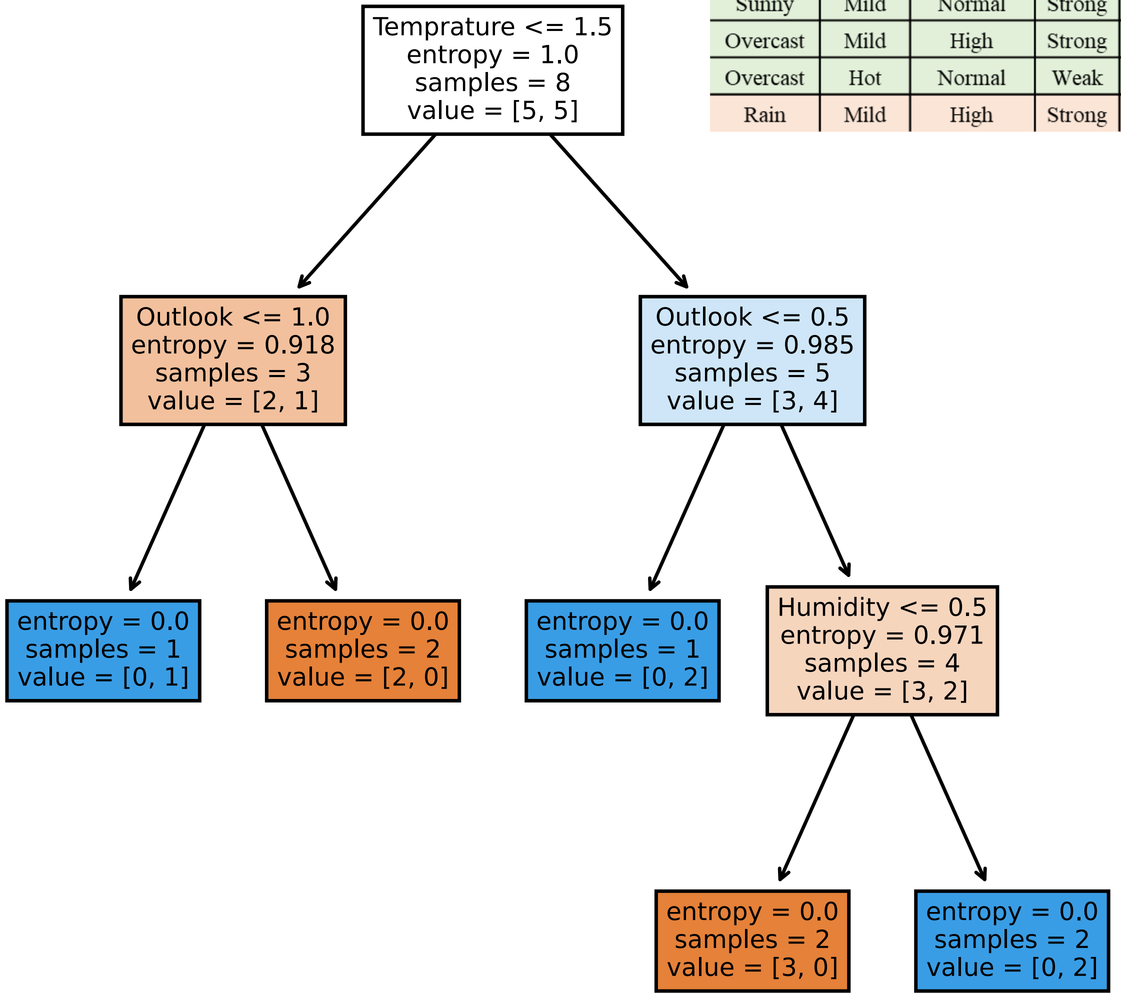
```python
from sklearn.ensemble import RandomForestClassifier

classifier = RandomForestClassifier(n_estimators=3,
                                    max_features=2,
                                    criterion='entropy',
                                    max_samples=10)
classifier.fit(X, y)
```



Tree 3

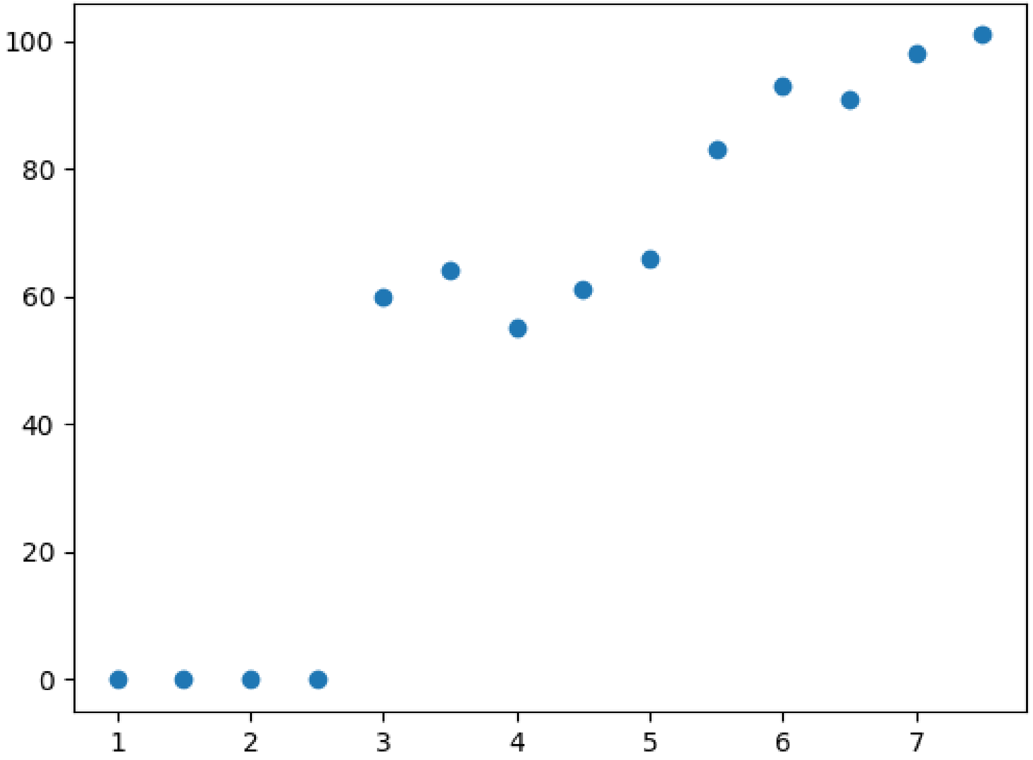
Outlook	Temp	Humidity	Wind	Play Tennis
Sunny	Hot	High	Weak	No
Sunny	Hot	High	Strong	No
Overcast	Hot	High	Weak	Yes
Rain	Mild	High	Weak	Yes
Rain	Cool	Normal	Weak	Yes
Rain	Cool	Normal	Strong	No
Overcast	Cool	Normal	Strong	Yes
Sunny	Mild	High	Weak	No
Sunny	Cool	Normal	Weak	Yes
Rain	Mild	Normal	Weak	Yes
Sunny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



Decision Tree - Regression

❖ Salary prediction

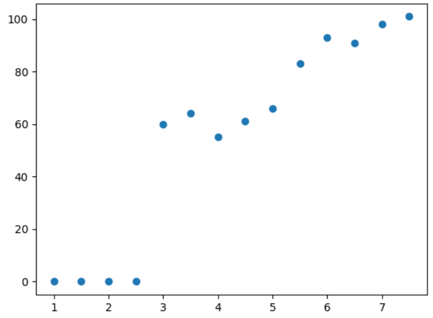
Experience	Salary
1	0
1.5	0
2	0
2.5	0
3	60
3.5	64
4	55
4.5	61
5	66
5.5	83
6	93
6.5	91
7	98
7.5	101



When Experience = 5.3,

Salary = ?

Experience	Salary
1	0
1.5	0
2	0
2.5	0
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	0

$$\mu_L = \frac{1}{|L|}\sum_i L_i = 0$$

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

Experience	Salary
1.5	0
2	0
2.5	0
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

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

$$= \frac{1}{14} * 0 + \frac{13}{14} * 1275.15$$

$$= 1184.07$$

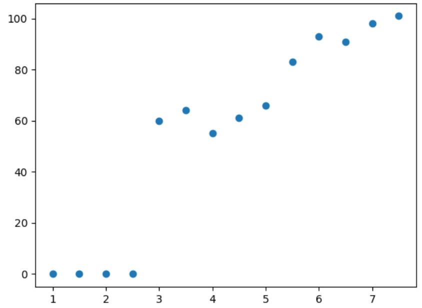
$$\mu = \frac{1}{|S|}\sum_i S_i = 55.14$$

$$mse = \frac{1}{|S|}\sum_i (S_i - \mu)^2 = 1417.97$$

$$\mu_R = \frac{1}{|R|}\sum_i R_i = 59.38$$

$$mse_R = \frac{1}{|R|}\sum_i (R_i - \mu)^2 = 1275.15$$

Experience	Salary
1	0
1.5	0
2	0
2.5	0
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	0
1.5	0
2	0
2.5	0

$$\mu_L = \frac{1}{|L|}\sum_i L_i = 0$$

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

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

$$a_{mse} = \frac{|L|}{|S|} mse_L + \frac{|R|}{|S|} mse_R$$
$$= \frac{4}{14} * 0 + \frac{10}{14} * 282.35$$
$$= 201.68$$

$$\mu = \frac{1}{|S|}\sum_i S_i = 55.14$$

$$mse = \frac{1}{|S|}\sum_i (S_i - \mu)^2 = 1417.97$$

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

$$mse_R = \frac{1}{|R|}\sum_i (R_i - \mu)^2 = 282.35$$

Experience	Salary
1	0
1.5	0
2	0
2.5	0
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

$a_{mse} = 1184.07$

$a_{mse} = 911.19$

$a_{mse} = 588.68$

$\boxed{a_{mse} = 201.68}$

$a_{mse} = 383.92$

$a_{mse} = 526.52$

$a_{mse} = 543.51$

$a_{mse} = 575.09$

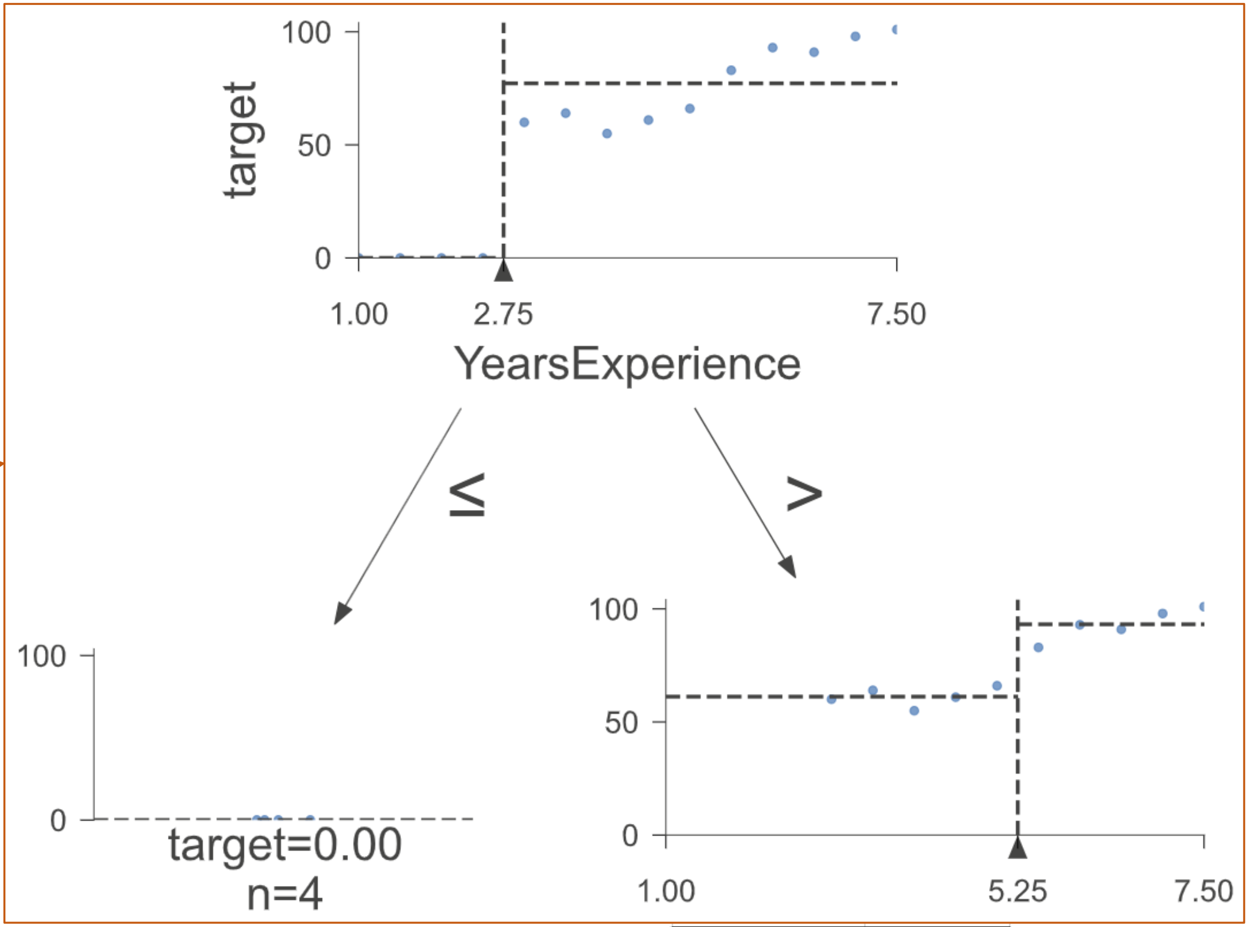
$a_{mse} = 613.34$

$a_{mse} = 758.4$

$a_{mse} = 947.73$

$a_{mse} = 1090.05$

$a_{mse} = 1256.21$



Experience	Salary
1	0
1.5	0
2	0
2.5	0

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

Decision Tree - Regression

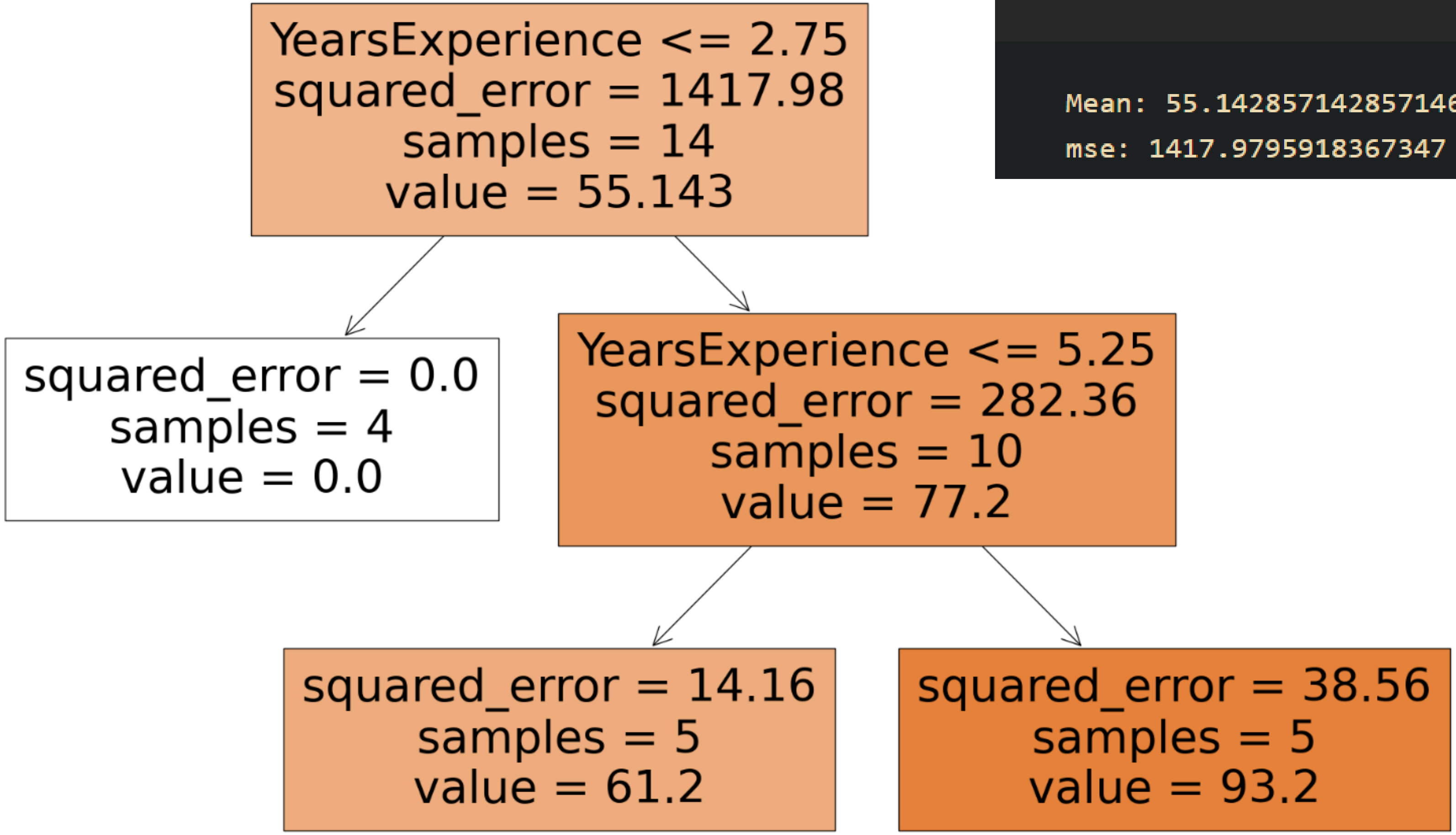
```
y_mean = y.mean()
print('Mean:', y_mean)

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

```
Mean: 55.142857142857146
mse: 1417.9795918367347
```

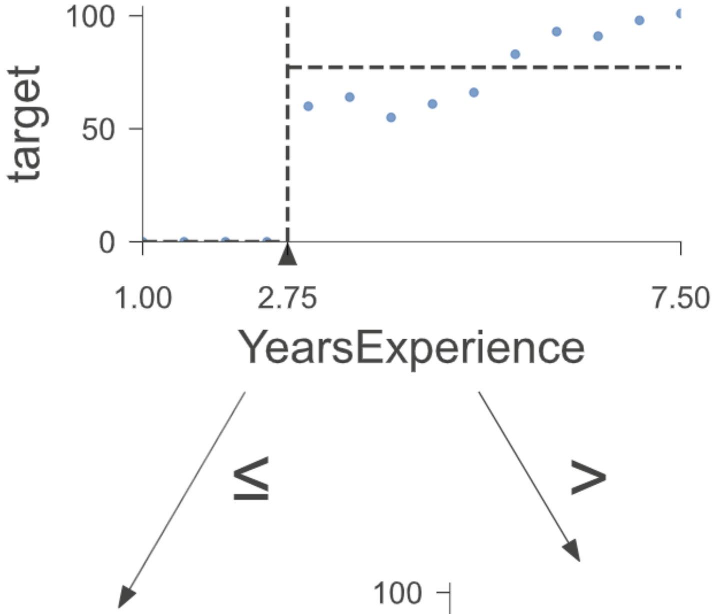
❖ Salary prediction

Experience	Salary
1	0
1.5	0
2	0
2.5	0
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



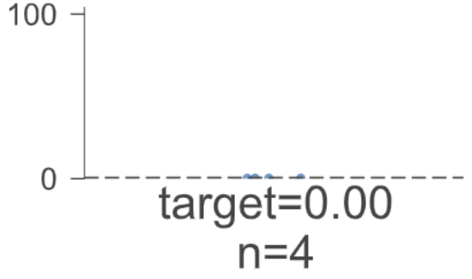
Decision Tree Regression

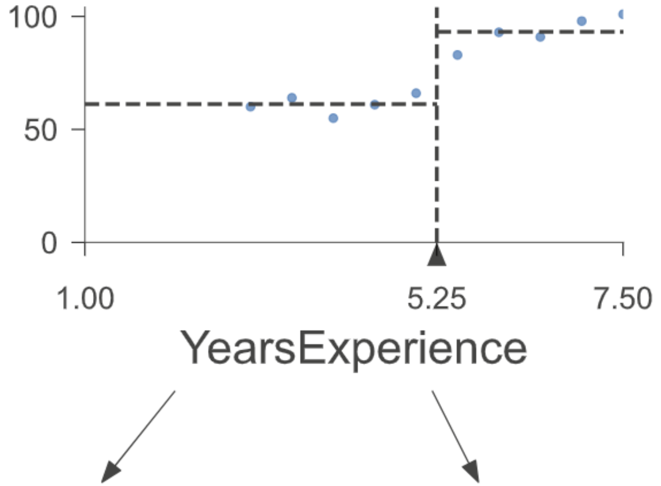
Experience	Salary
1	0
1.5	0
2	0
2.5	0
3	60
3.5	64
4	55
4.5	61
5	66
5.5	83
6	93
6.5	91
7	98
7.5	101



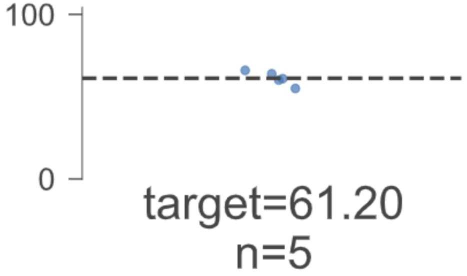
Experience	Salary
3	60
3.5	64
4	55
4.5	61
5	66
5.5	83
6	93
6.5	91
7	98
7.5	101

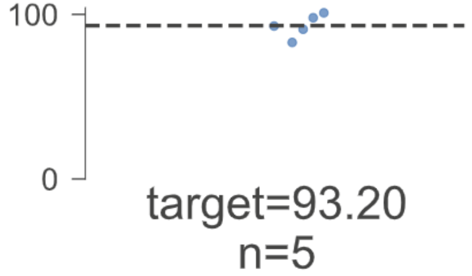
Experience	Salary
1	0
1.5	0
2	0
2.5	0





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



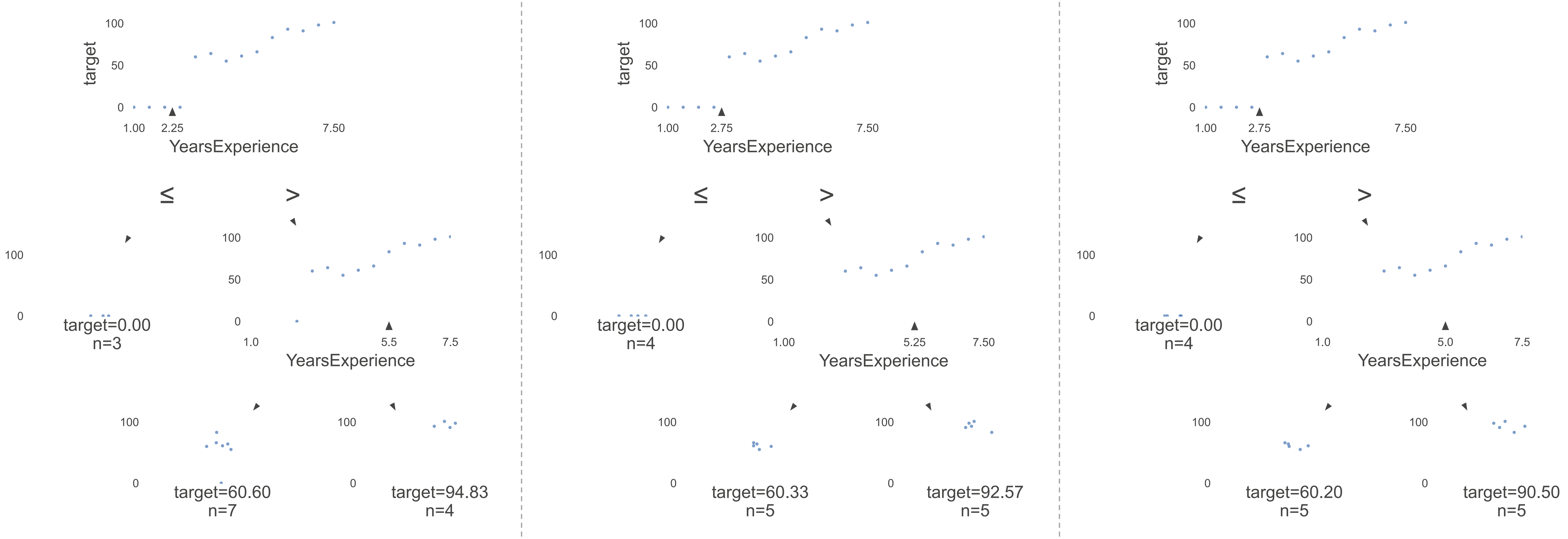


Experience	Salary
5.5	83
6	93
6.5	91
7	98
7.5	101

Random Forest Regression

```
dt_regressor = RandomForestRegressor(n_estimators=3,
                                     max_depth=2)
dt_regressor.fit(X, y)
```

❖ Salary prediction



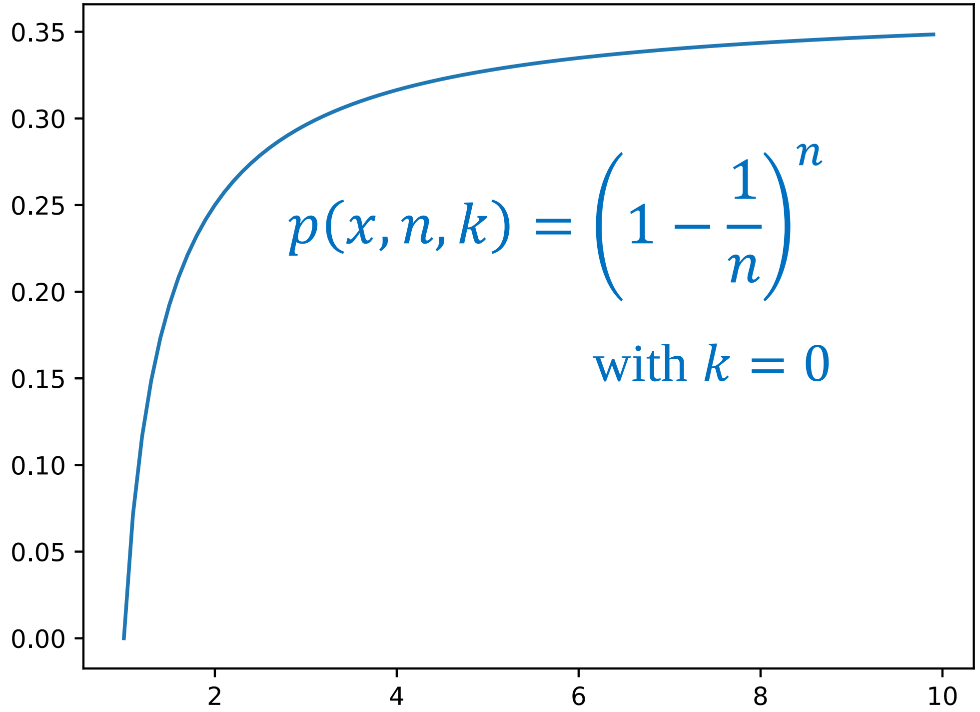
Random Forest

❖ Bernoulli Random variables

A numerical description of the outcome of a statistical experiment

$$p(x) = p\{X = x\} = \begin{cases} p & when\ x = 1 \\ 1 - p & when\ x = 0 \end{cases}$$

$$p(x, n, k) = C_n^k \left(\frac{1}{n}\right)^k \left(1 - \frac{1}{n}\right)^{n-k}$$



Toss a coin

Sample space: S = {tail, head}

X= {0, 1}

AI VIETNAM
All-in-One Course

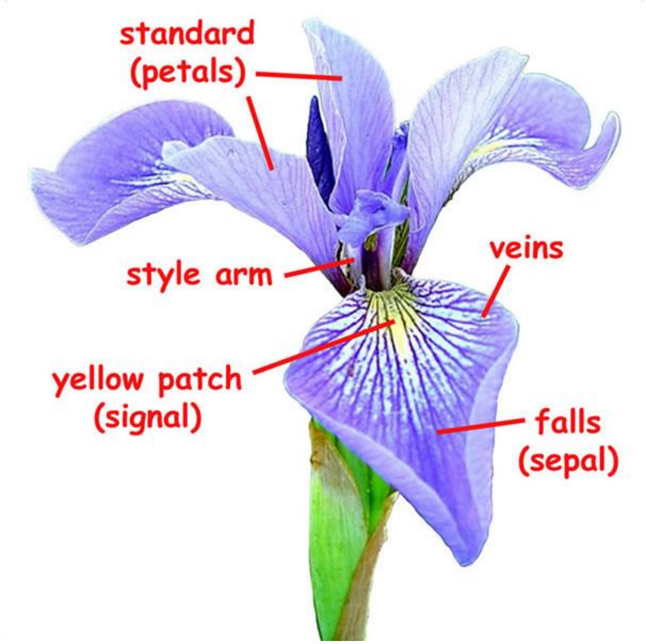
Adaptive Boosting (Warm-up Class)

Quang-Vinh Dinh
Ph.D. in Computer Science

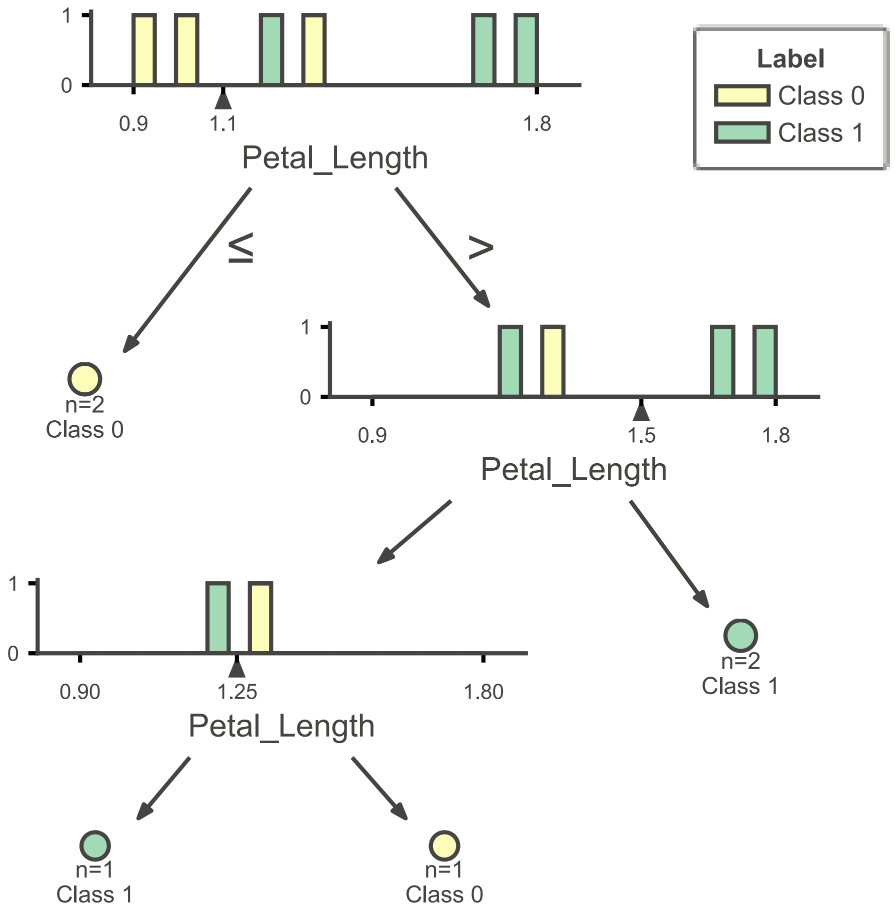
Year 2023

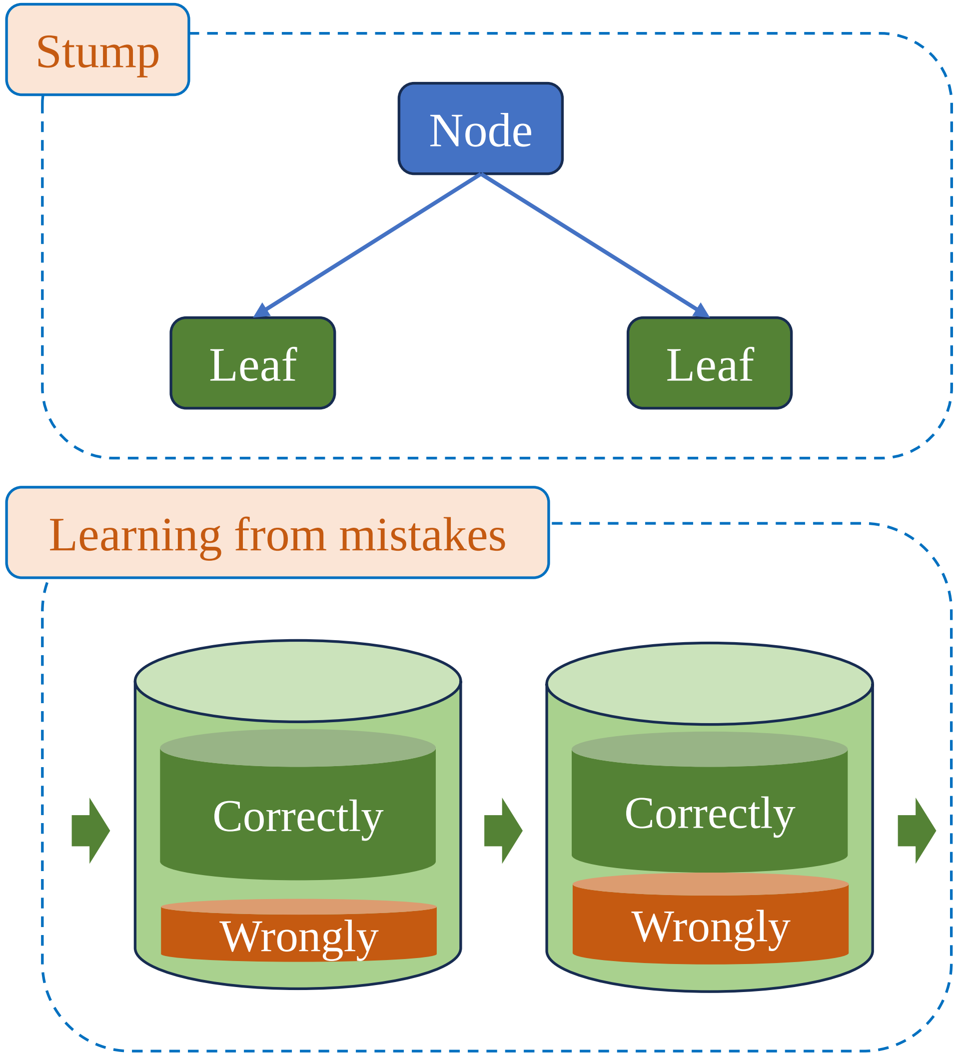
AdaBoost

❖ Idea



https://www.fs.usda.gov/wildflowers/beauty/iris/flower.shtml



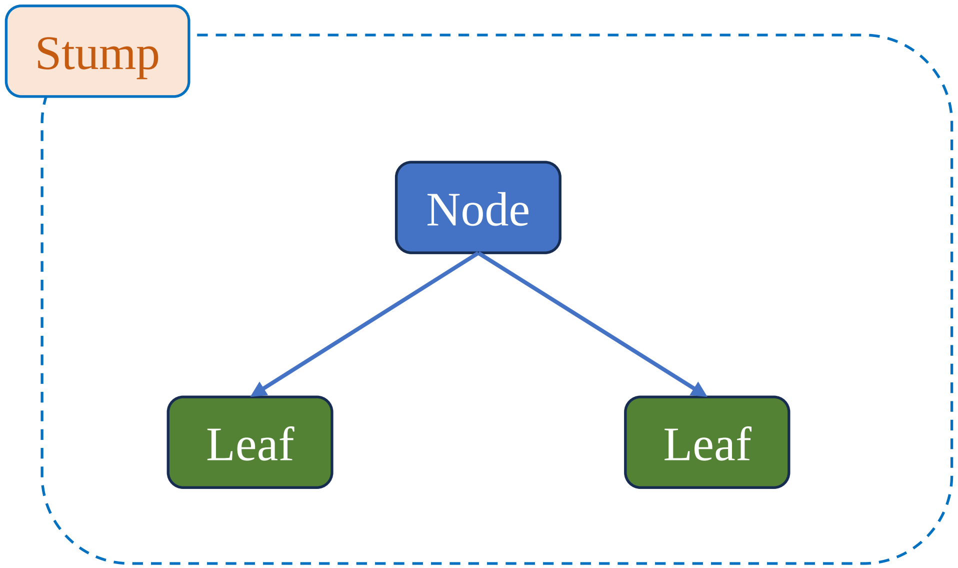


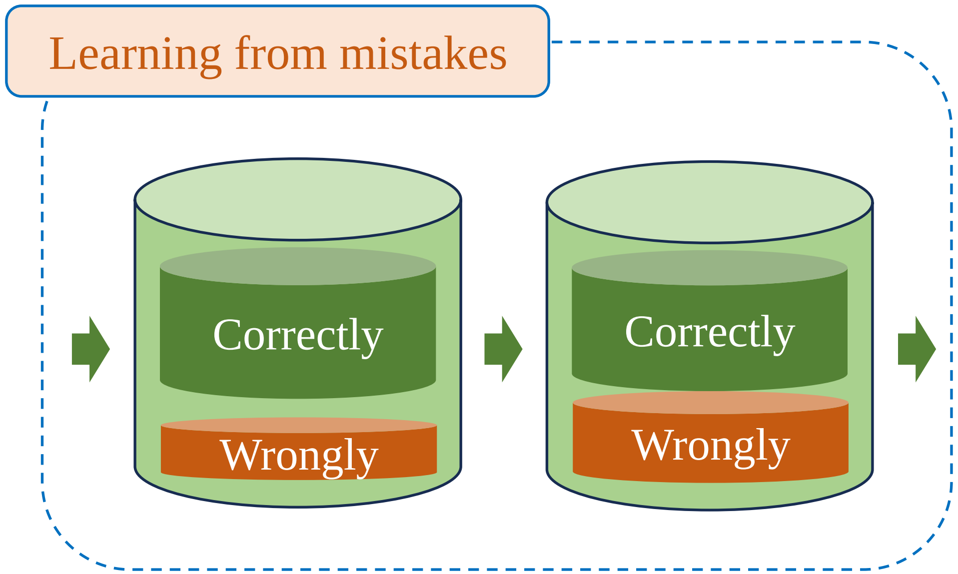
AdaBoost

❖ Discussion

1) Are a wide range of features used to build a forest?

2) How to create a new dataset?

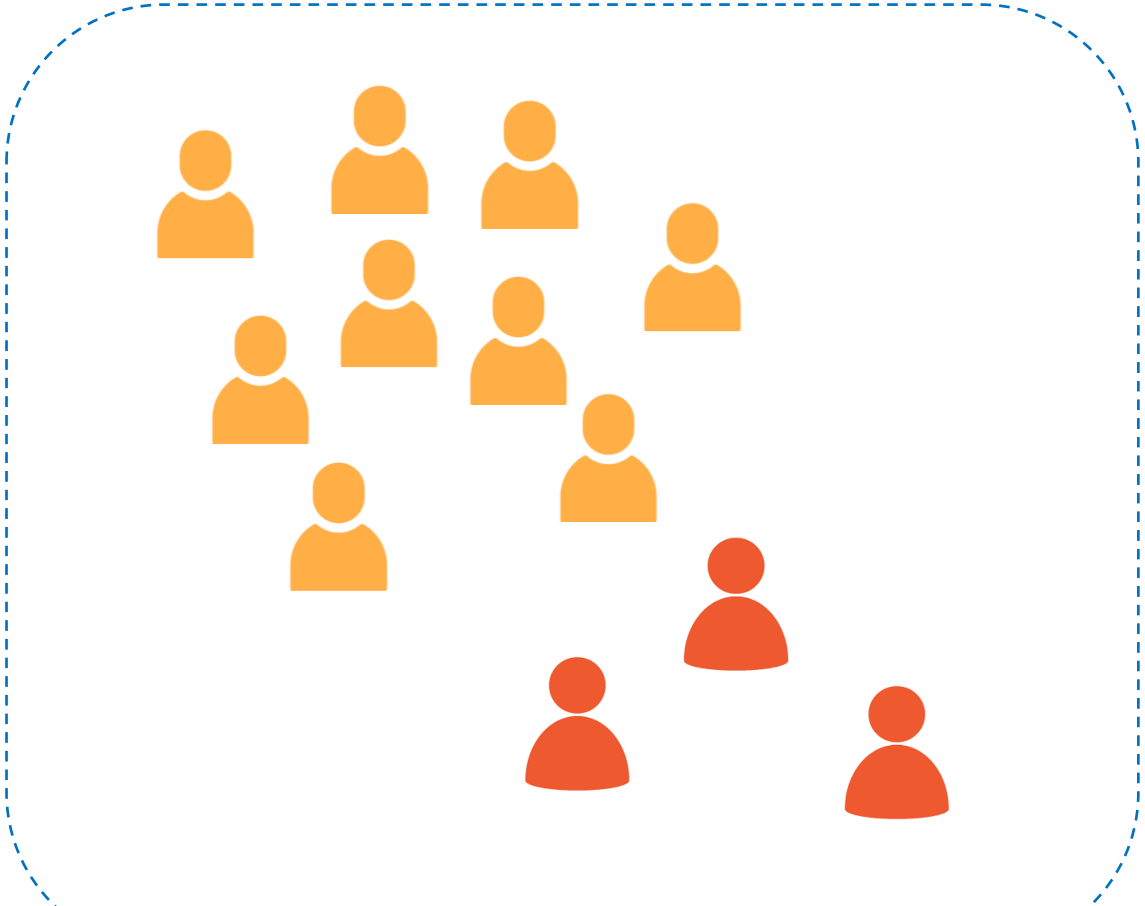




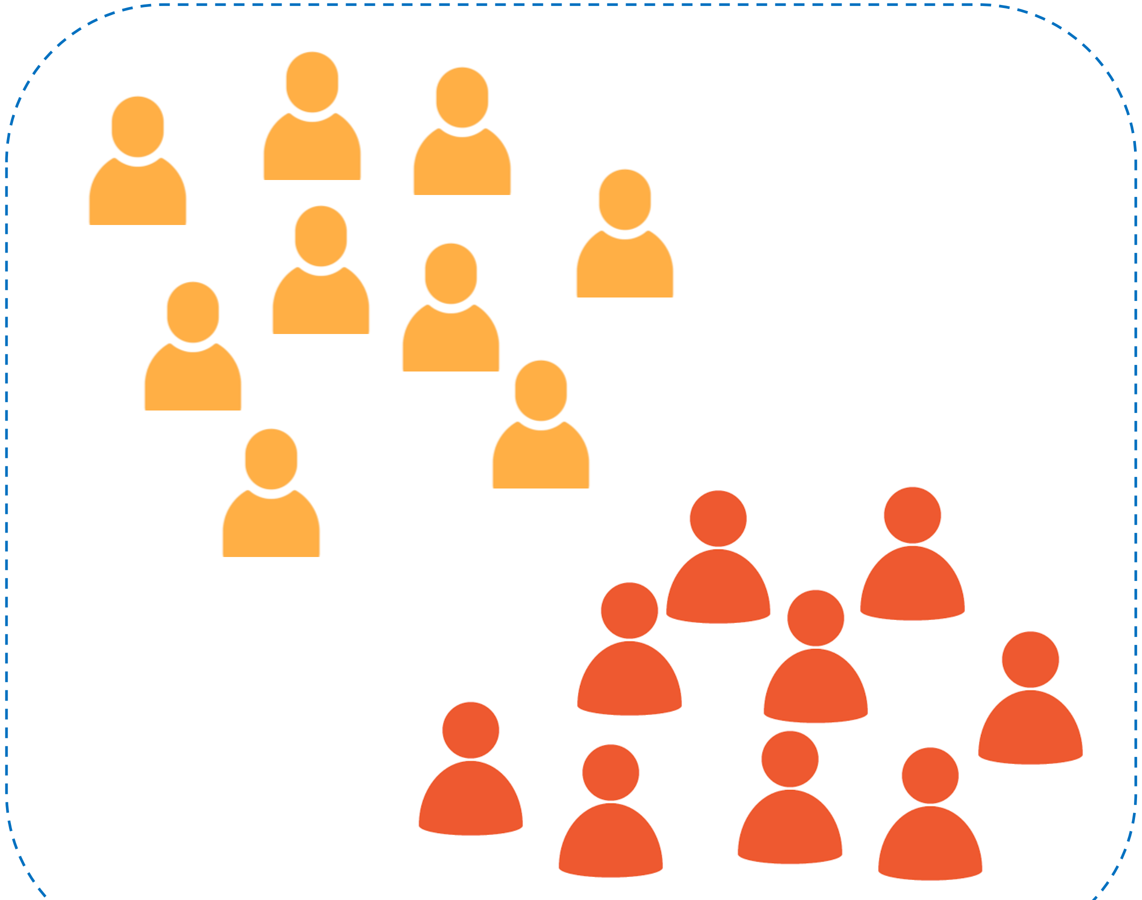
AdaBoost

❖ Idea

How to balance the two groups' values?



Each person has a value

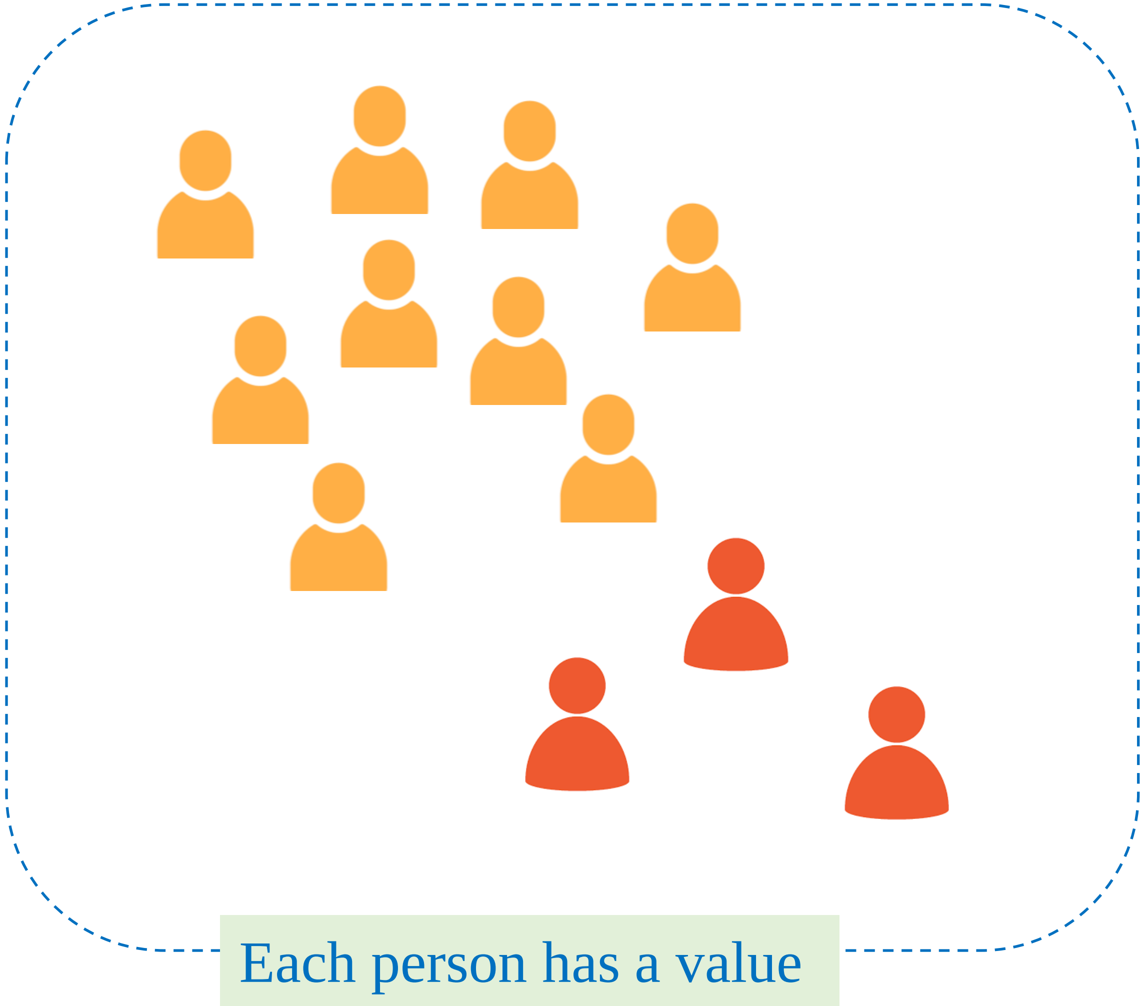


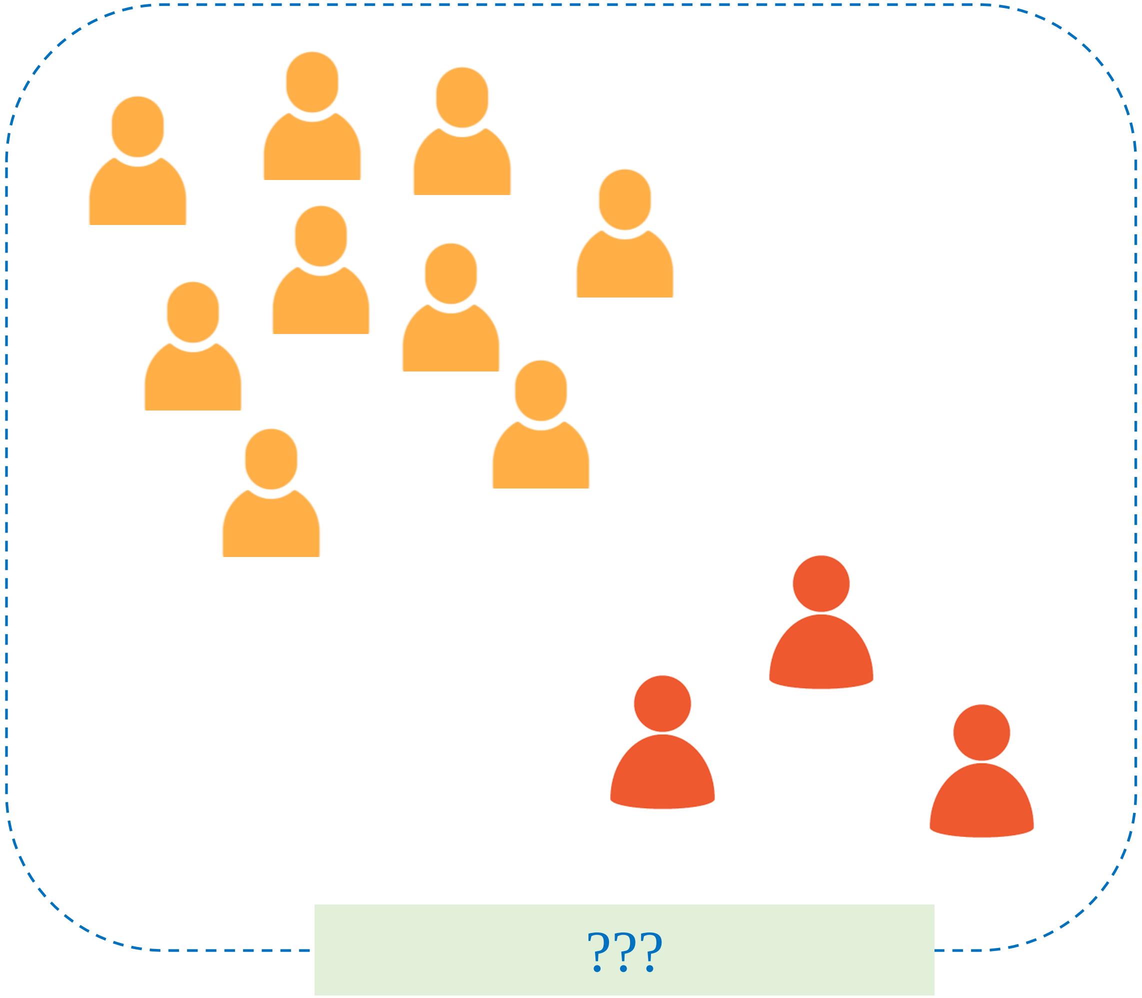
Balance #sources

AdaBoost

❖ Idea

Any other ideas?





AdaBoost

Naïve

❖ Create a new dataset

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

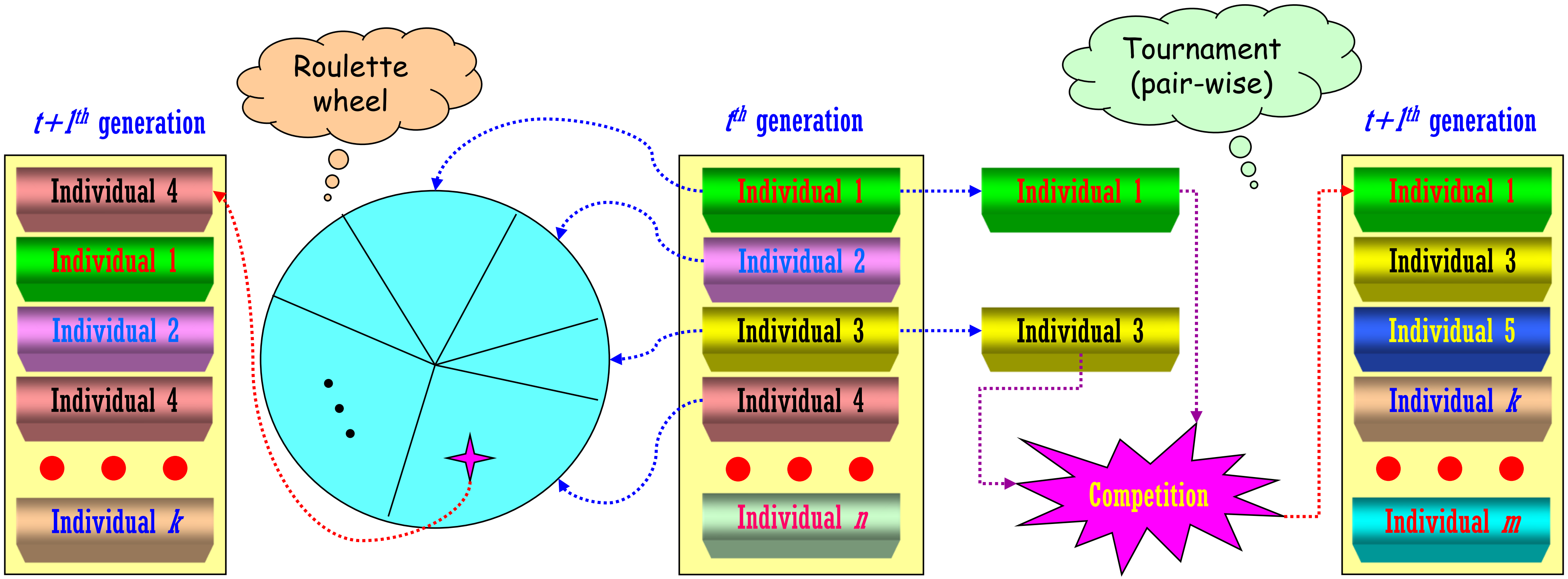
Petal_Length	Petal_Width	Label	Evaluation
1	0.2	0	T
1.3	0.6	0	F
0.9	0.7	0	T
1.7	0.5	1	T
1.8	0.9	1	F
1.2	1.3	1	T

Petal_Length	Petal_Width	Label	Evaluation
1.3	0.6	0	F
1.3	0.6	0	F
1.8	0.9	1	F
1.8	0.9	1	F

Petal_Length	Petal_Width	Label	Evaluation
1	0.2	0	T
0.9	0.7	0	T
1.7	0.5	1	T
1.2	1.3	1	T

Ideas from Genetic Algorithms

- **Roulette Wheel Selection**
  - The probability of selecting a given chromosome is proportional to its fitness
- **Tournament Selection**
  - Combine the fitness proportional concept with the random selection



AdaBoost

❖ Create a new dataset

Add more randomness

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

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

normalize

Petal_Length	Petal_Width	Label	Evaluation	Score	Probability
1	0.2	0	T	1	0.125
1.3	0.6	0	F	2	0.25
0.9	0.7	0	T	1	0.125
1.7	0.5	1	T	1	0.125
1.8	0.9	1	F	2	0.25
1.2	1.3	1	T	1	0.125

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

AdaBoost

❖ Create a new dataset

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	Evaluation	Score	Probability
1	0.2	0	T	1	0.125
1.3	0.6	0	F	2	0.25
0.9	0.7	0	T	1	0.125
1.7	0.5	1	T	1	0.125
1.8	0.9	1	F	2	0.25
1.2	1.3	1	T	1	0.125

Case 1

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

Petal_Length	Petal_Width	Label	Evaluation	Score	Probability
1	0.2	0	T	1	0.142
1.3	0.6	0	T	1	0.142
1.2	1.3	1	F	2	0.29
1.7	0.5	1	T	1	0.142
1.8	0.9	1	T	1	0.142
1.8	0.9	1	T	1	0.142

AdaBoost

❖ Create a new dataset

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	Evaluation	Score	Probability
1	0.2	0	T	1	0.125
1.3	0.6	0	F	2	0.25
0.9	0.7	0	T	1	0.125
1.7	0.5	1	T	1	0.125
1.8	0.9	1	F	2	0.25
1.2	1.3	1	T	1	0.125

Case 2

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

Petal_Length	Petal_Width	Label	Evaluation	Score	Probability
1	0.2	0	T	1	0.142
1.3	0.6	0	F	2	0.29
1.2	1.3	1	T	1	0.142
1.7	0.5	1	T	1	0.142
1.8	0.9	1	T	1	0.142
1.8	0.9	1	T	1	0.142

Problem and solution?

AdaBoost

❖ Create a new dataset

Petal_Length	Petal_Width	Label	Probability
1	0.2	0	0.166
1.3	0.6	0	0.166
0.9	0.7	0	0.166
1.7	0.5	1	0.166
1.8	0.9	1	0.166
1.2	1.3	1	0.166

Petal_Length	Petal_Width	Label	Probability
1	0.2	0	0.166
1.3	0.6	0	0.166
0.9	0.7	0	0.166
1.7	0.5	1	0.166
1.8	0.9	1	0.166
1.2	1.3	1	0.166

True

False

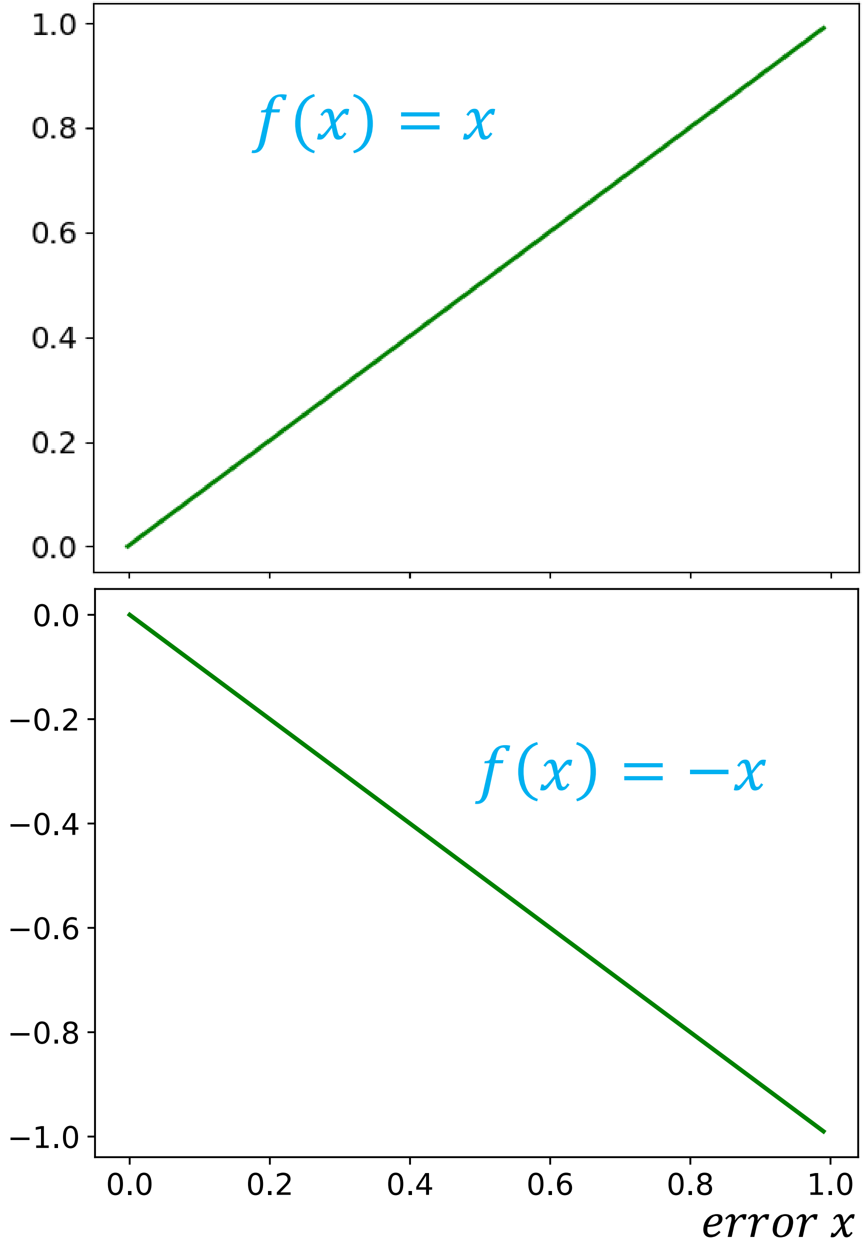
Update

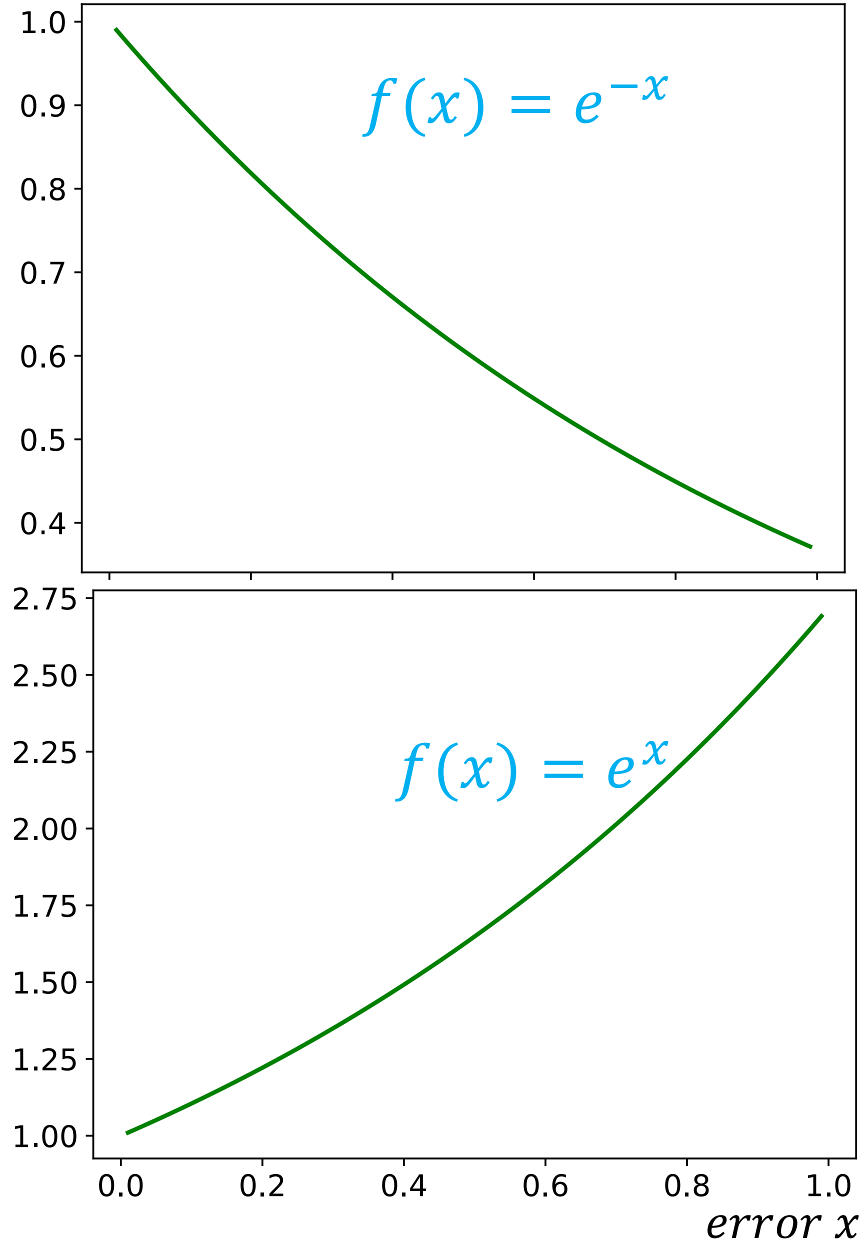
How much?

Error < 0.5

For incorrect samples

When a model is good (small error), scaled weights should increase/decrease slightly/significantly?



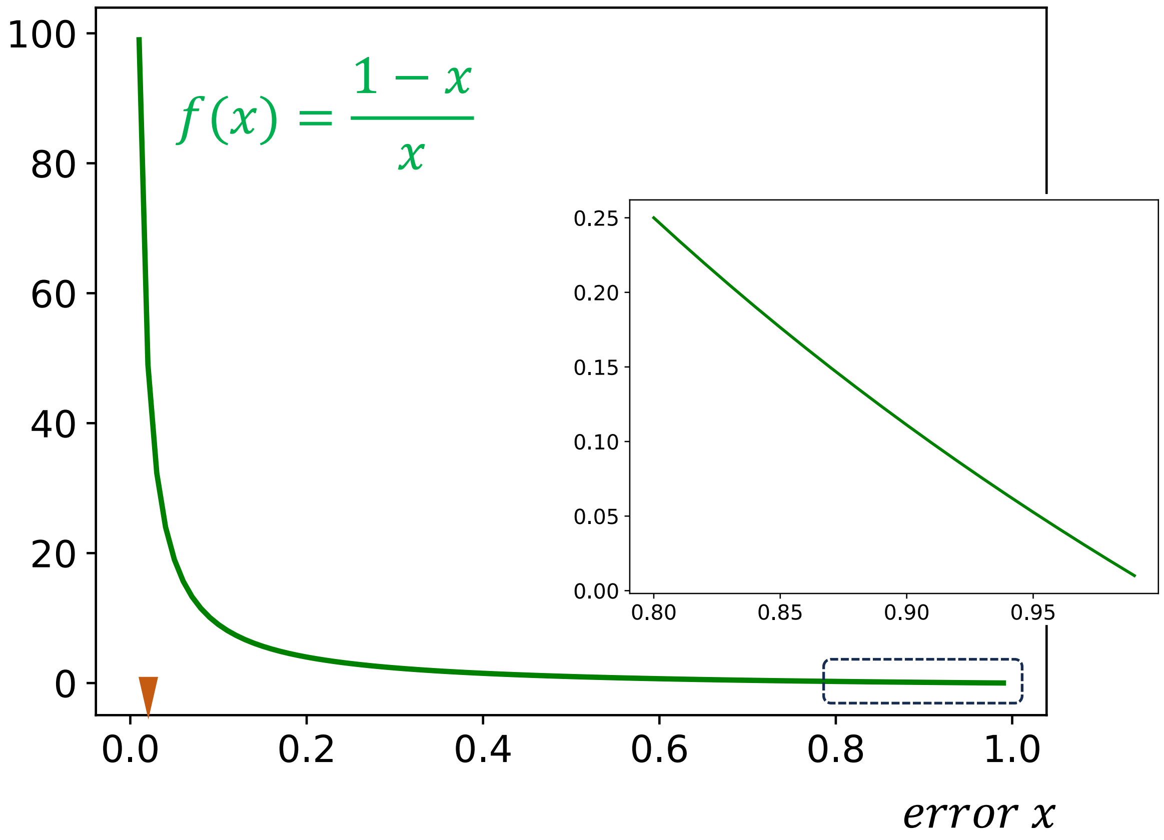


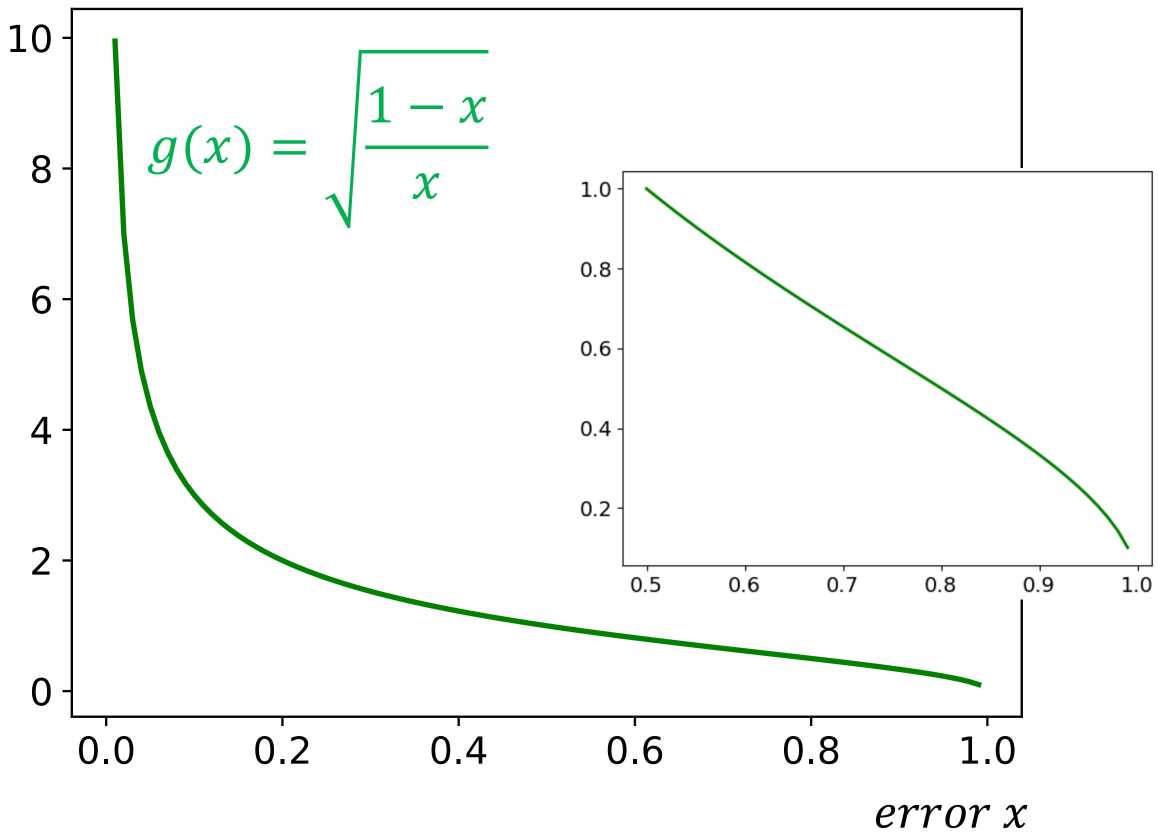
AdaBoost

Error < 0.5

For incorrect samples

When a model has a small error, increase significantly





AdaBoost

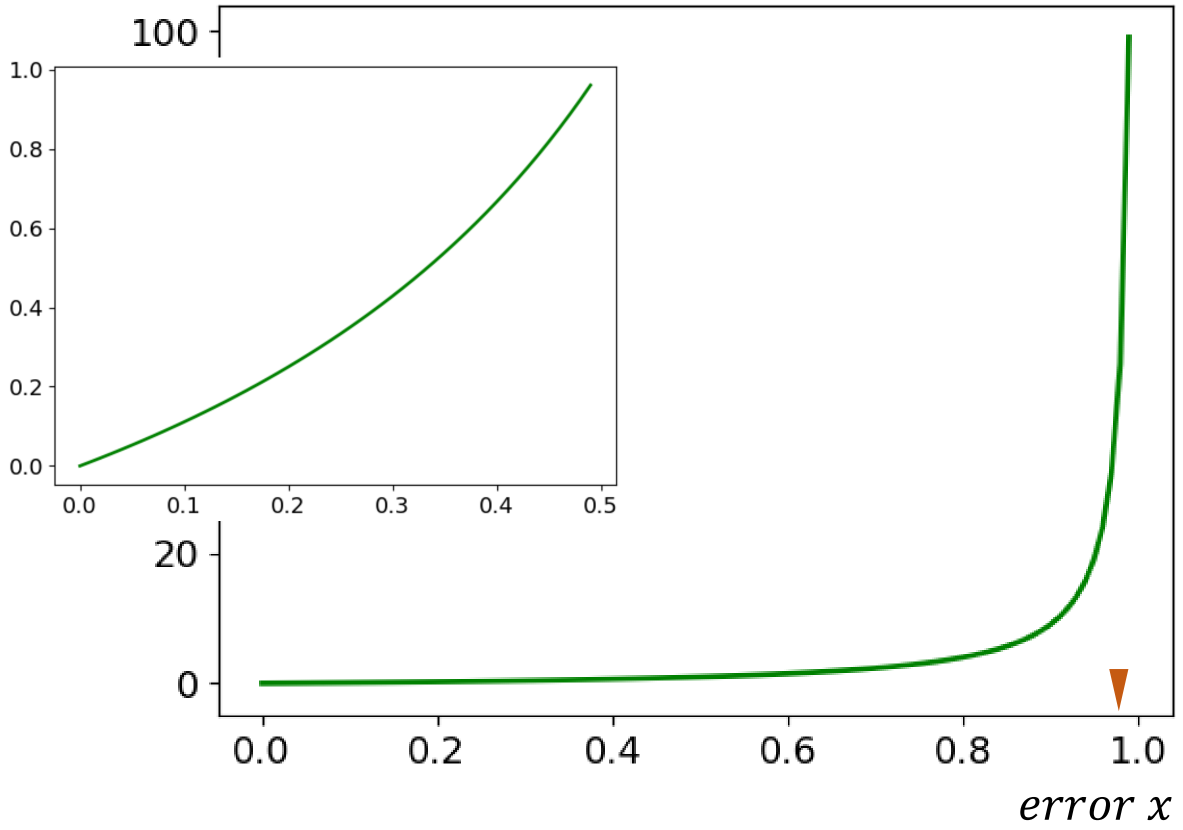
Error < 0.5

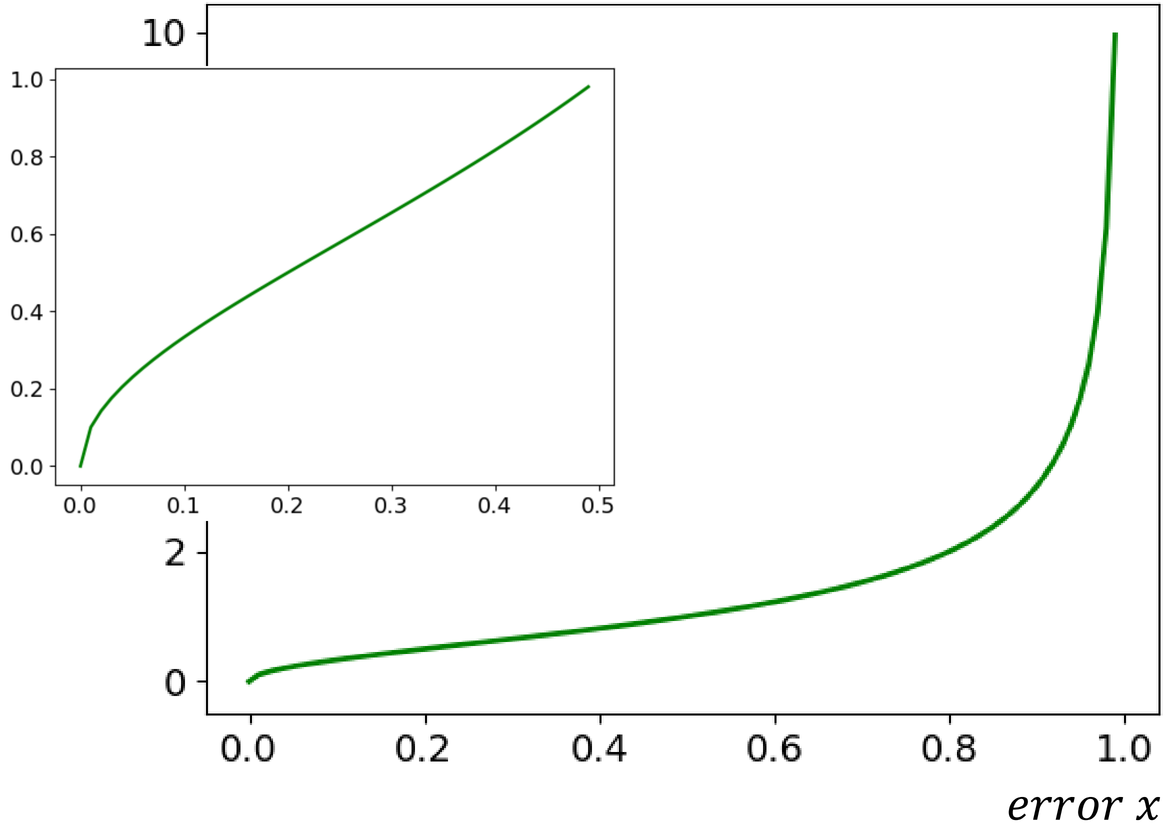
For correct samples

Decrease significantly

$$k(x) = \frac{1}{f(x)} = \frac{x}{1-x}$$

$$h(x) = \frac{1}{g(x)} = \sqrt{\frac{x}{1-x}}$$





❖ Create a new dataset

Petal_Length	Petal_Width	Label	Probability
1	0.2	0	0.166
1.3	0.6	0	0.166
0.9	0.7	0	0.166
1.7	0.5	1	0.166
1.8	0.9	1	0.166
1.2	1.3	1	0.166

Petal_Length	Petal_Width	Label	Probability
1	0.2	0	0.166
1.3	0.6	0	0.166
0.9	0.7	0	0.166
1.7	0.5	1	0.166
1.8	0.9	1	0.166
1.2	1.3	1	0.166

True

False

Update

$$g(E) = \sqrt{\frac{1-E}{E}}$$

$$p_i = p_i g(E)$$

$$= 0.166 * 1.41 = 2.347$$

$$h(E) = \sqrt{\frac{E}{1-E}}$$

$$p_i = p_i h(E)$$

$$= 0.166 * 0.707 = 1.17$$

Petal_Length	Petal_Width	Label	Probability
1	0.2	0	1.17
1.3	0.6	0	2.347
0.9	0.7	0	1.17
1.7	0.5	1	1.17
1.8	0.9	1	2.347
1.2	1.3	1	1.17

❖ Create a new dataset

Petal_Length	Petal_Width	Label	Probability
1	0.2	0	0.166
1.3	0.6	0	0.166
0.9	0.7	0	0.166
1.7	0.5	1	0.166
1.8	0.9	1	0.166
1.2	1.3	1	0.166

Petal_Length	Petal_Width	Label	Probability
1	0.2	0	0.166
1.3	0.6	0	0.166
0.9	0.7	0	0.166
1.7	0.5	1	0.166
1.8	0.9	1	0.166
1.2	1.3	1	0.166

True

False

Update

$$g(E) = \sqrt{\frac{1-E}{E}}$$

$$p_i = p_i g(E)$$

$$= 0.166 * 1.41 = 2.347$$

$$h(E) = \sqrt{\frac{E}{1-E}}$$

$$p_i = p_i h(E)$$

$$= 0.166 * 0.707 = 1.17$$

Petal_Length	Petal_Width	Label	Probability
1	0.2	0	0.124
1.3	0.6	0	0.25
0.9	0.7	0	0.124
1.7	0.5	1	0.124
1.8	0.9	1	0.25
1.2	1.3	1	0.124

Normalized

❖ Create a new dataset

Petal_Length	Petal_Width	Label	Probability
1	0.2	0	0.166
1.3	0.6	0	0.166
0.9	0.7	0	0.166
1.7	0.5	1	0.166
1.8	0.9	1	0.166
1.2	1.3	1	0.166

Petal_Length	Petal_Width	Label	Probability
1	0.2	0	0.166
1.3	0.6	0	0.166
0.9	0.7	0	0.166
1.7	0.5	1	0.166
1.8	0.9	1	0.166
1.2	1.3	1	0.166

True

False

Update

$$g(E) = \sqrt{\frac{1-E}{E}}$$

$$p_i = p_i g(E) = p_i e^{\ln(g(E))}$$

$$= p_i e^{\ln\left(\sqrt{\frac{1-E}{E}}\right)} = p_i e^{\frac{1}{2}\ln\left(\frac{1-E}{E}\right)}$$

$$h(E) = \sqrt{\frac{E}{1-E}}$$

$$p_i = p_i h(E) = p_i\, e^{\ln(h(E))}$$

$$= p_i e^{\ln\left(\sqrt{\frac{E}{1-E}}\right)} = p_i e^{-\frac{1}{2}\ln\left(\frac{1-E}{E}\right)}$$

Petal_Length	Petal_Width	Label	Probability
1	0.2	0	0.124
1.3	0.6	0	0.25
0.9	0.7	0	0.124
1.7	0.5	1	0.124
1.8	0.9	1	0.25
1.2	1.3	1	0.124

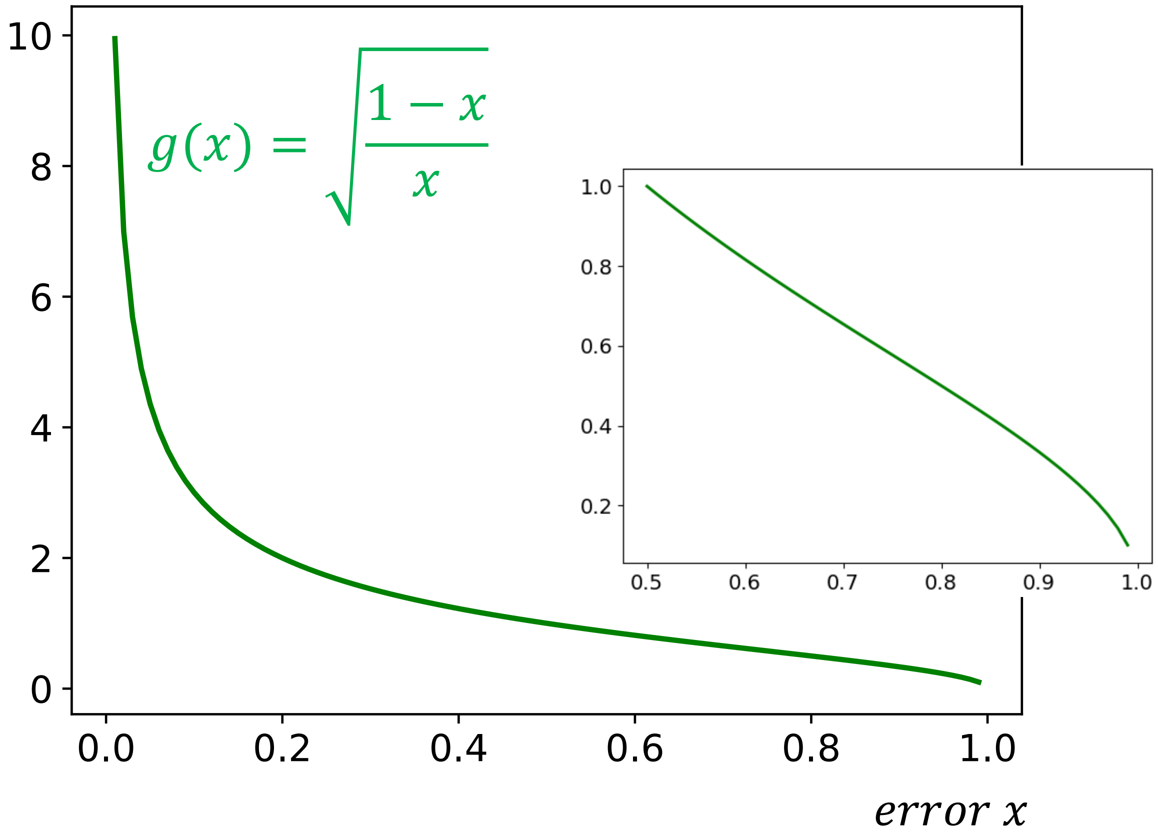
Normalized

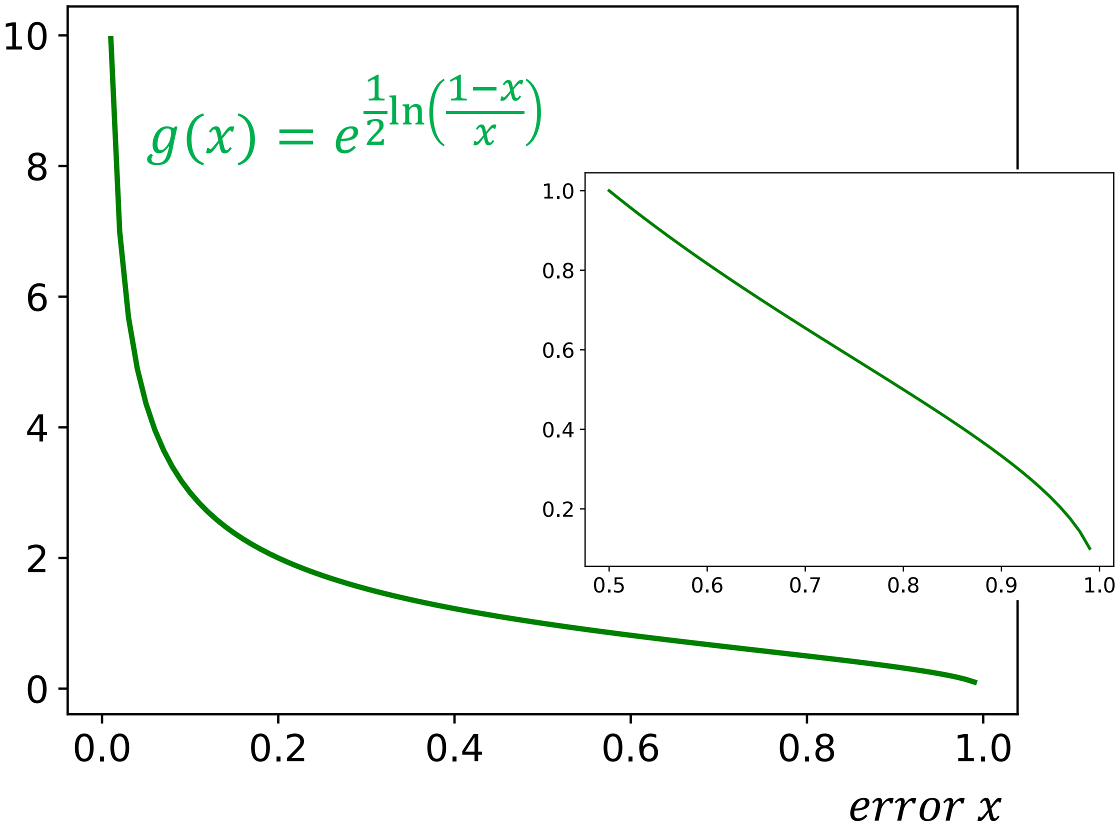
AdaBoost

Error < 0.5

For incorrect samples

Increase significantly





https://mbernste.github.io/files/notes/AdaBoost.pdf

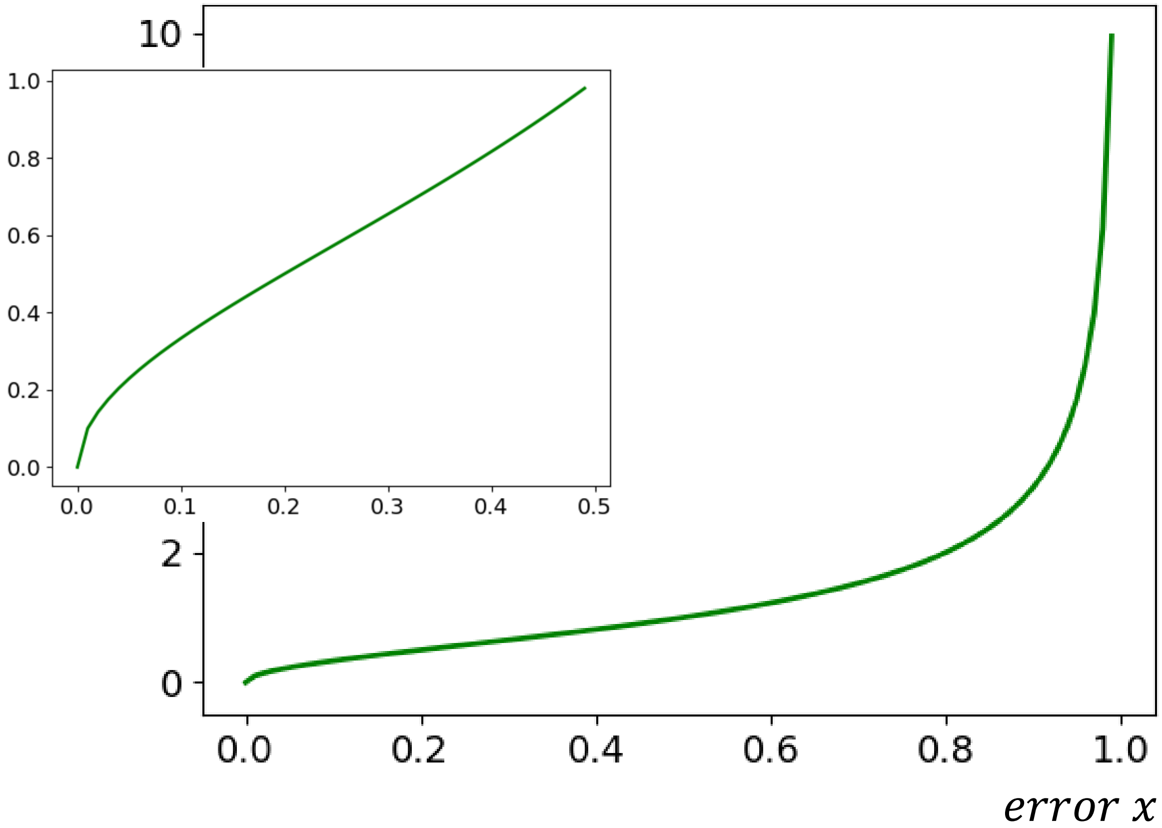
Error < 0.5

AdaBoost

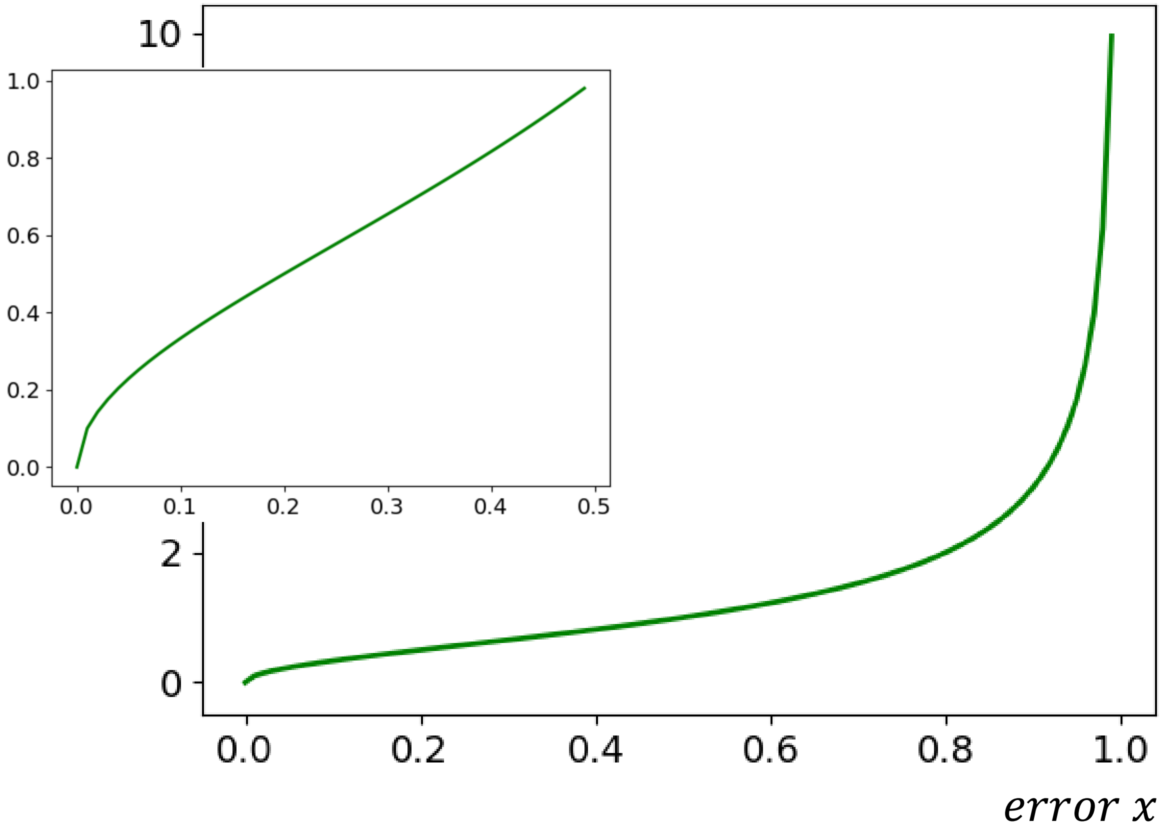
For correct samples

Decrease significantly

$$h(x) = \frac{1}{g(x)} = \sqrt{\frac{x}{1-x}}$$



$$h(x) = e^{-\frac{1}{2}\ln\left(\frac{1-x}{x}\right)}$$

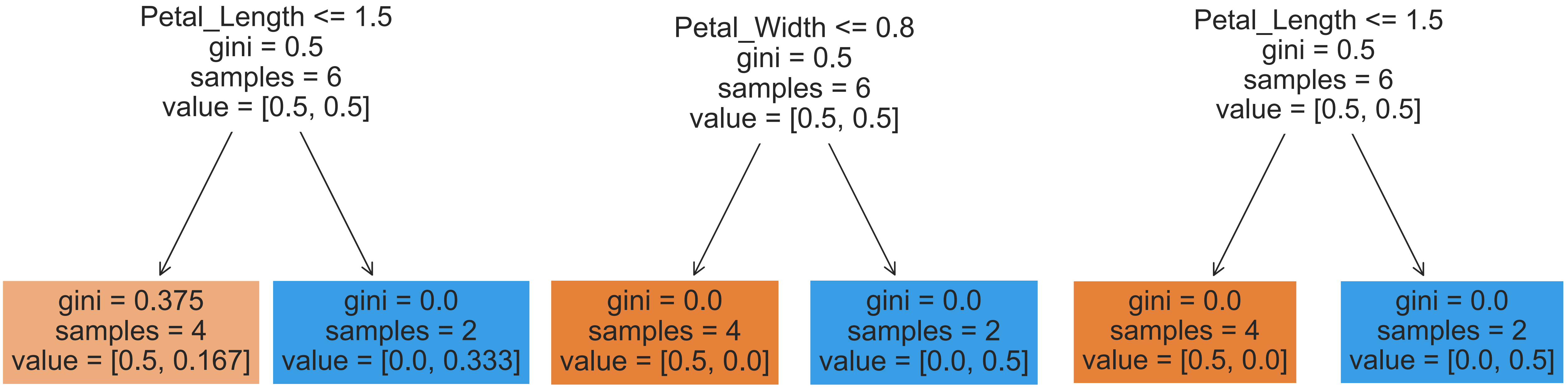


AdaBoost

Implementation using sklearn

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

```
dt_classifier = AdaBoostClassifier(n_estimators=3)
dt_classifier.fit(x_data, y_train)
```

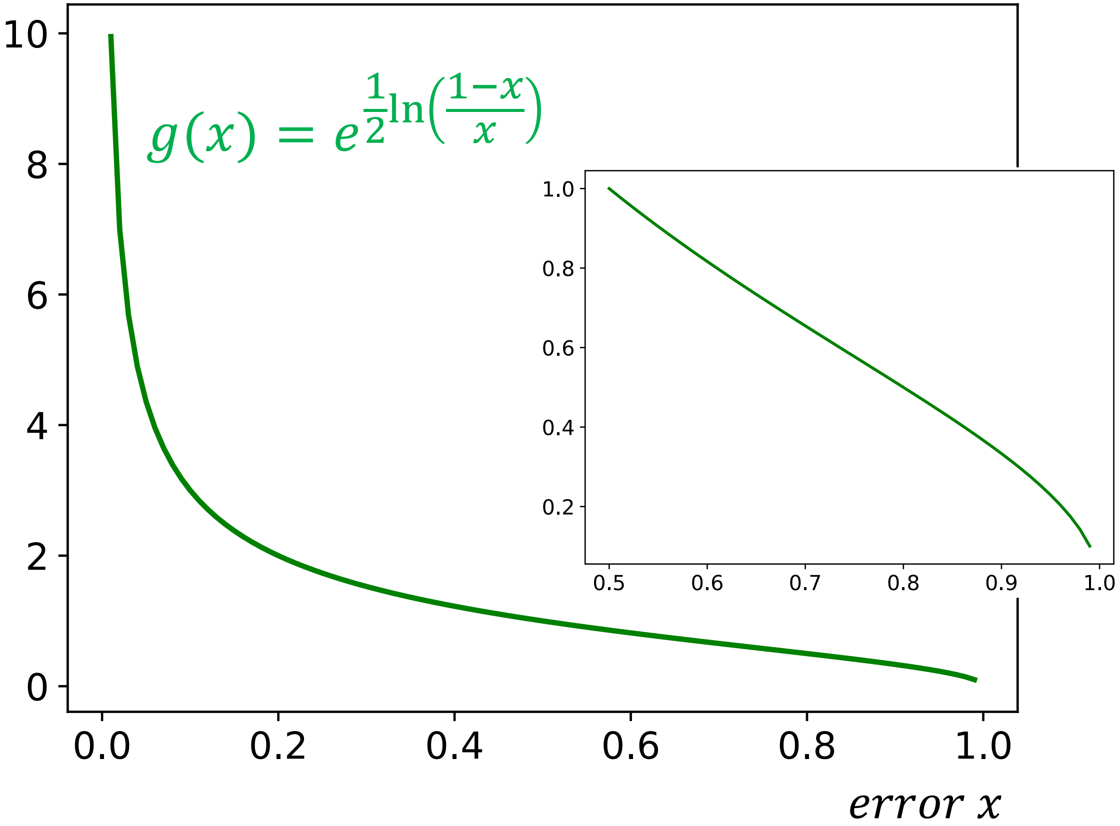


AdaBoost

Error > 0.5

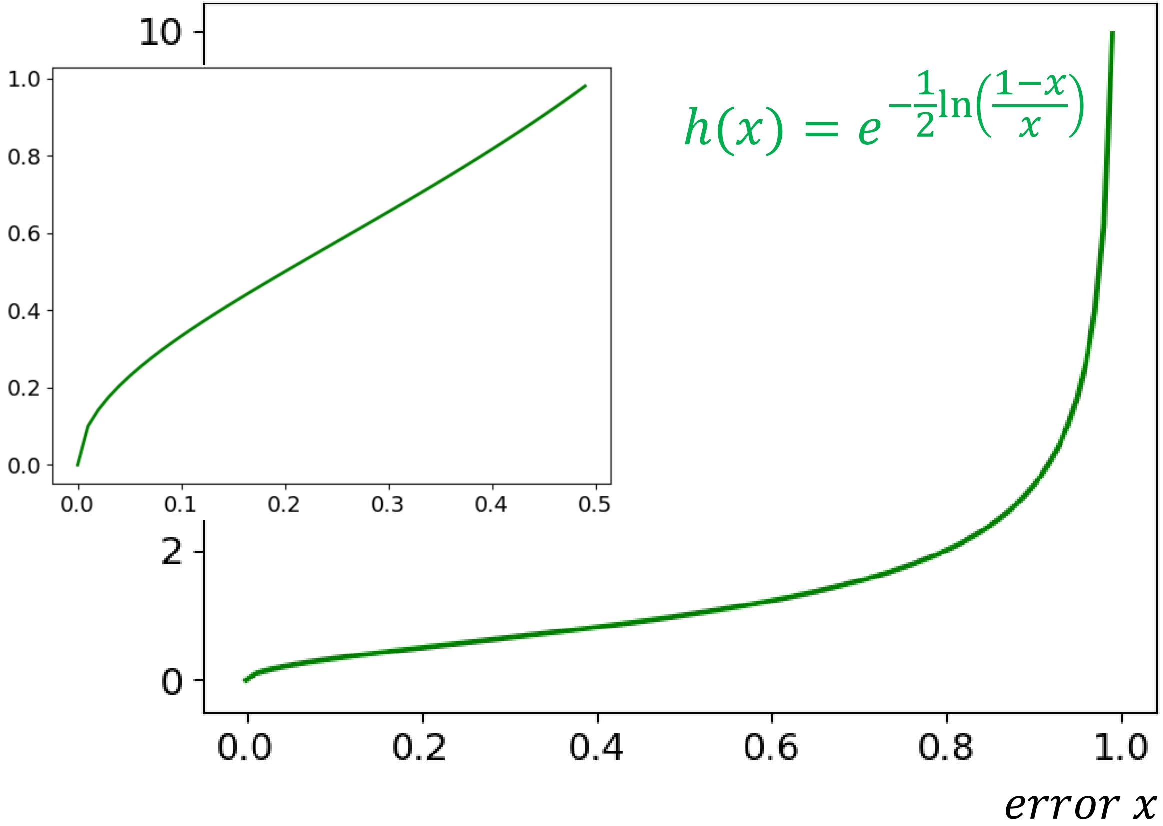
For incorrect samples

Increase



For correct samples

Decrease



AdaBoost

Implementation using sklearn

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

```
dt_classifier = AdaBoostClassifier(n_estimators=3)
dt_classifier.fit(x_data, y_train)
```

How to do inference?