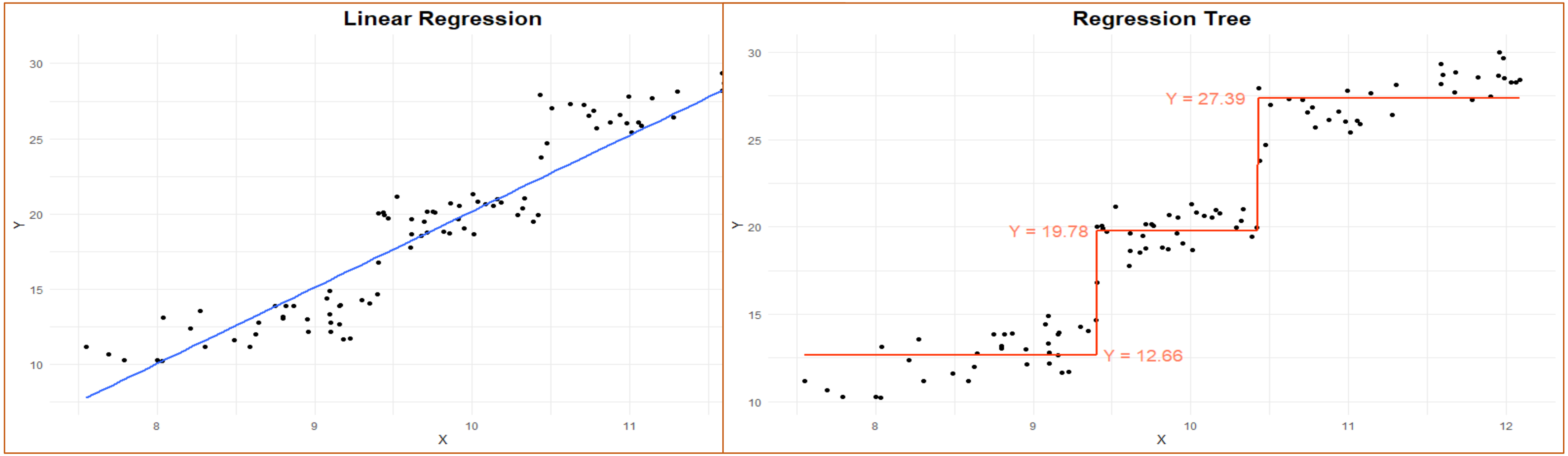
AI VIETNAM
All-in-One Course

Decision Tree for Regression



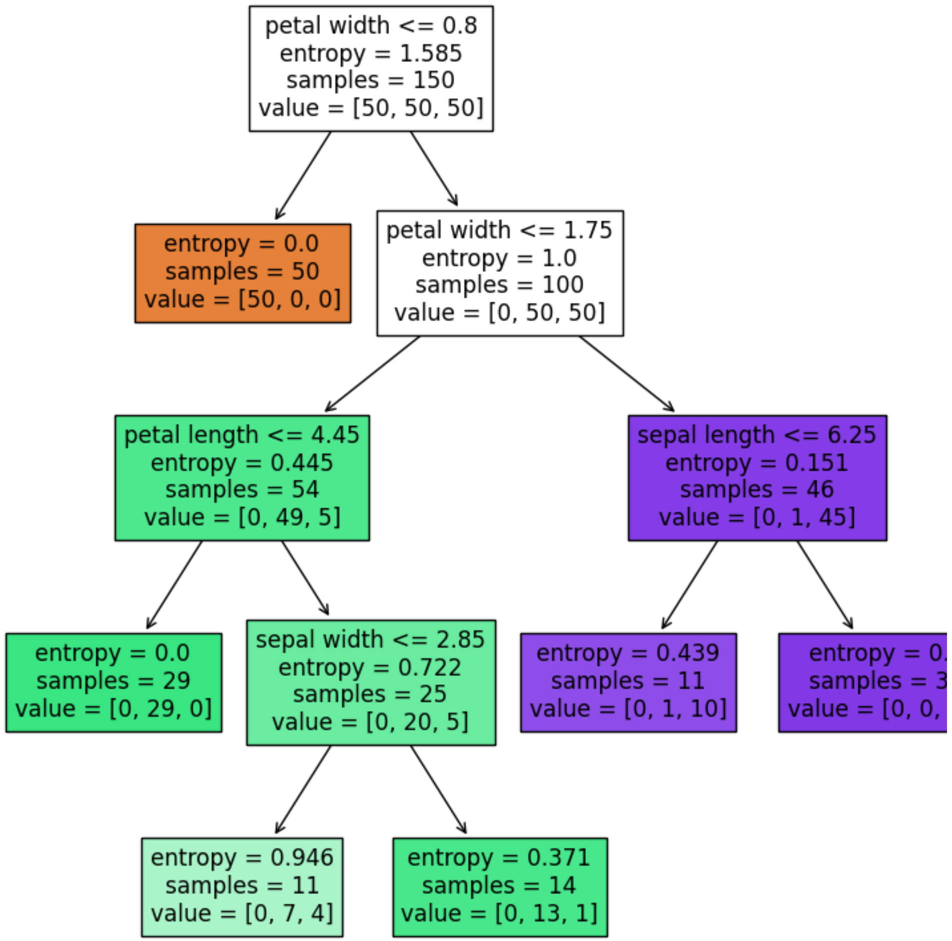
Vinh Dinh Nguyen
PhD in Computer Science

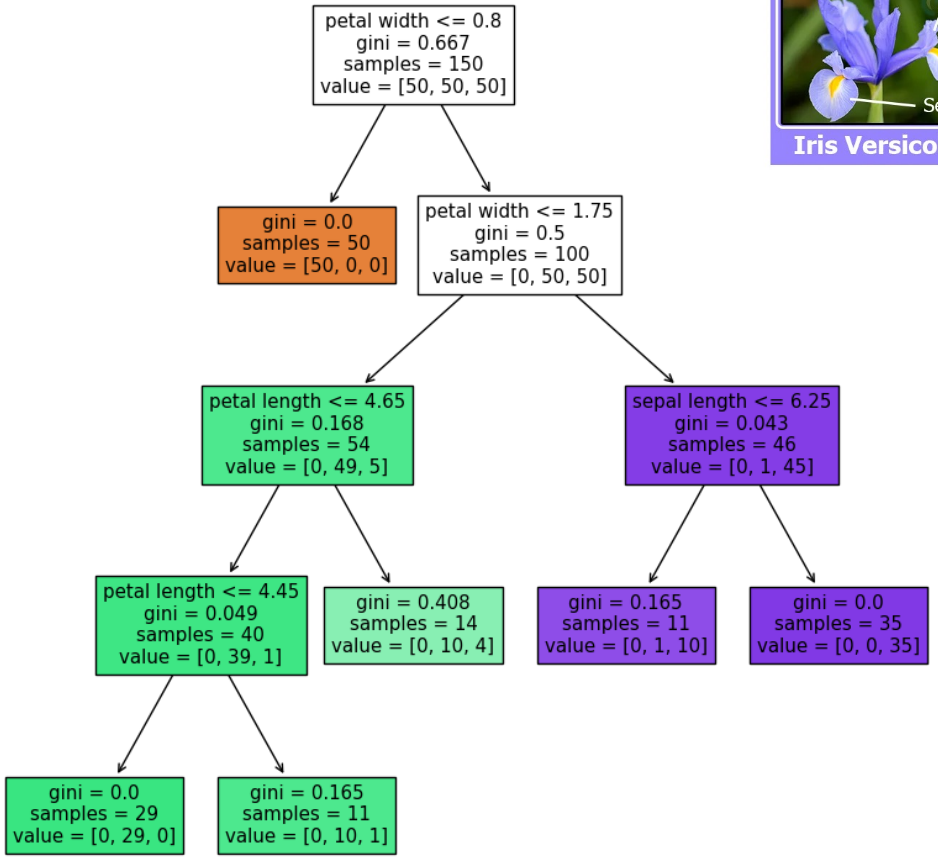
Year 2023

Outline

- Motivation for Regression Tree
- Regression Tree
- Overfitting in Regression Tree
- Case study

Classification Tree: Review

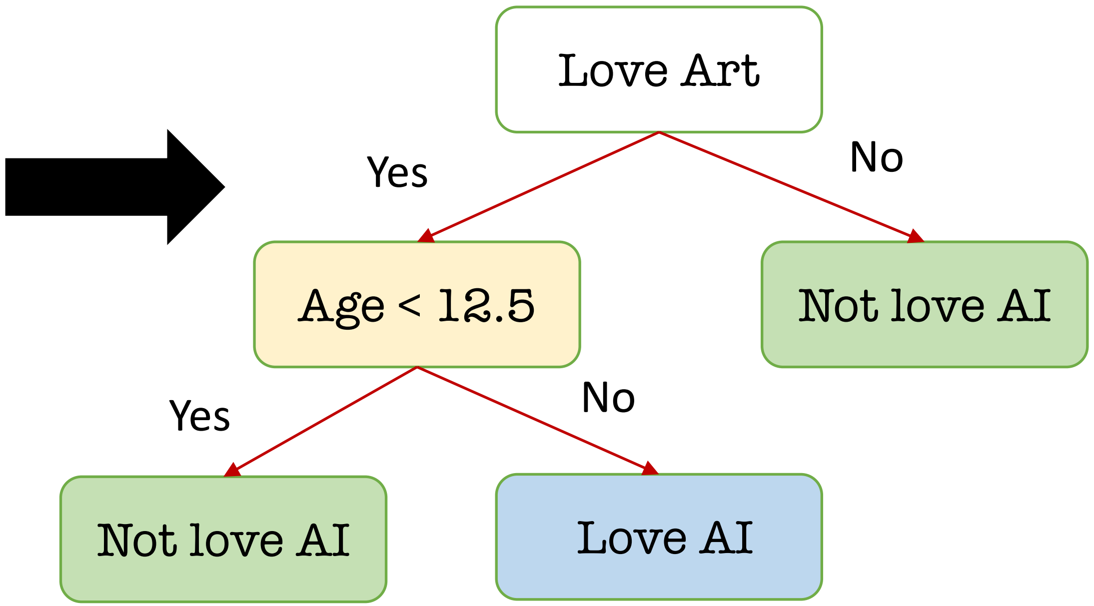




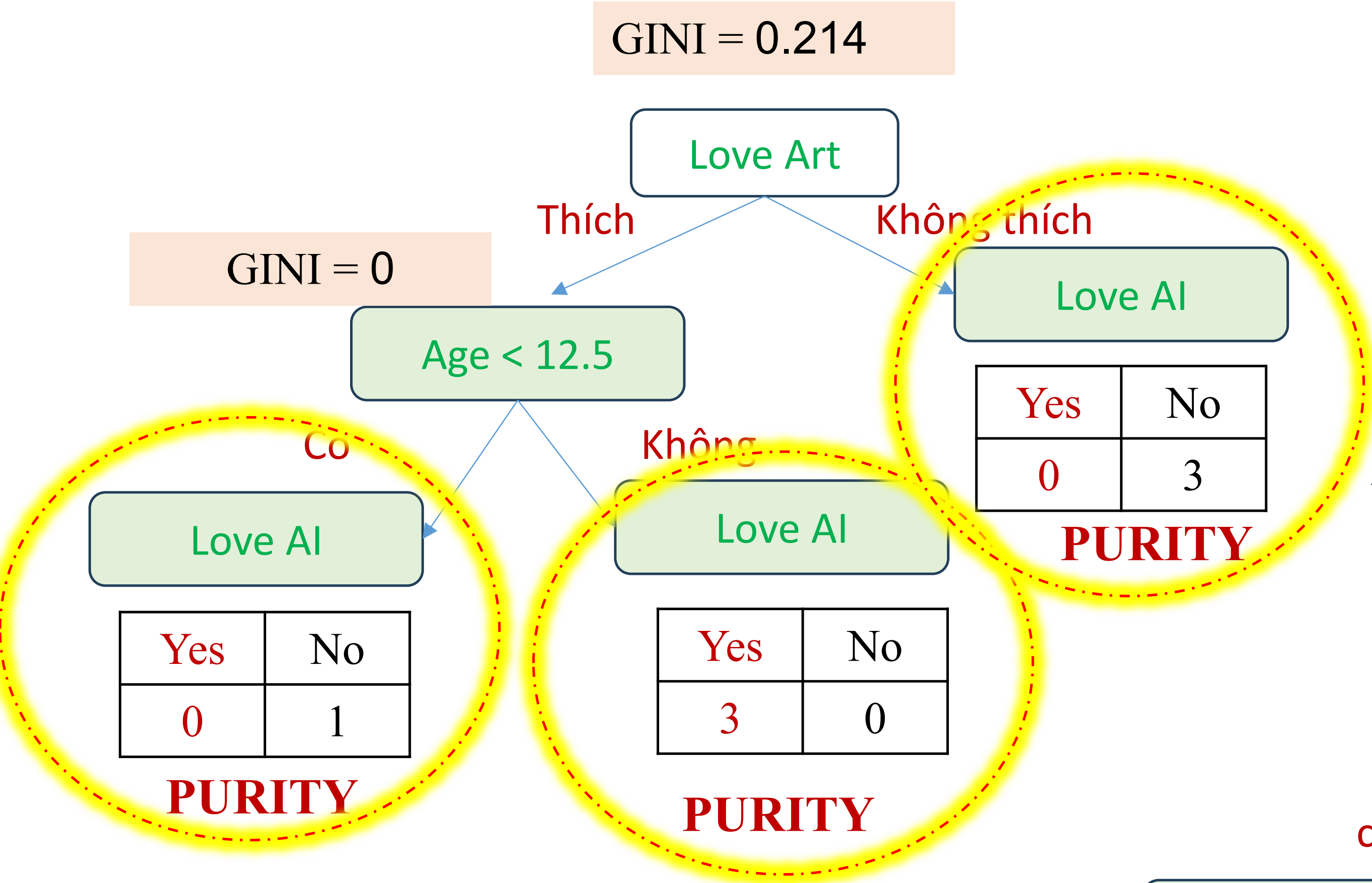
Classification Tree: Review

Love Math	Love Art	Age	Love AI
Yes	Yes	7	No
Yes	No	12	No
No	Yes	18	Yes
No	Yes	35	Yes
Yes	Yes	38	Yes
Yes	No	50	No
No	No	83	No

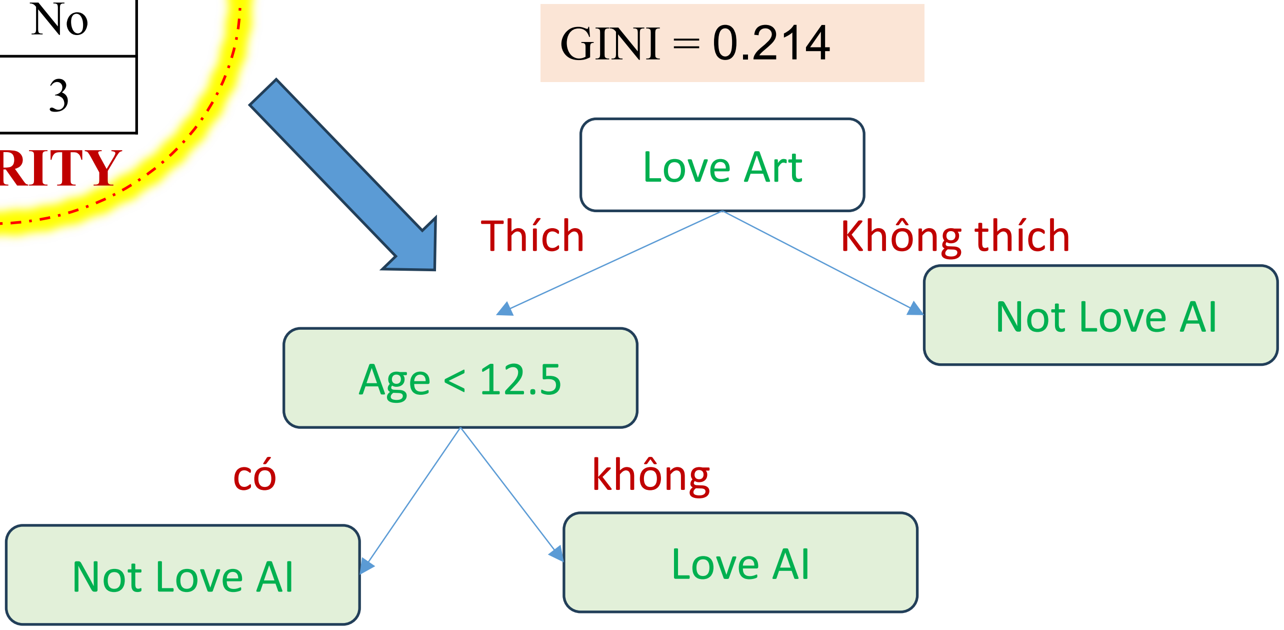
Features Labels



Classification Tree: Review



Previous Open Question:
How can we handle overfitting?



Metric Evaluation Review

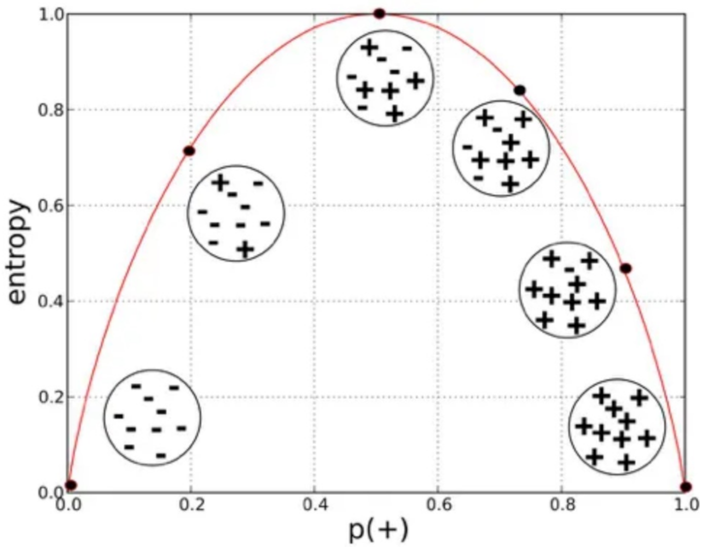
When should I use Gini Impurity as opposed to Information Gain (Entropy)

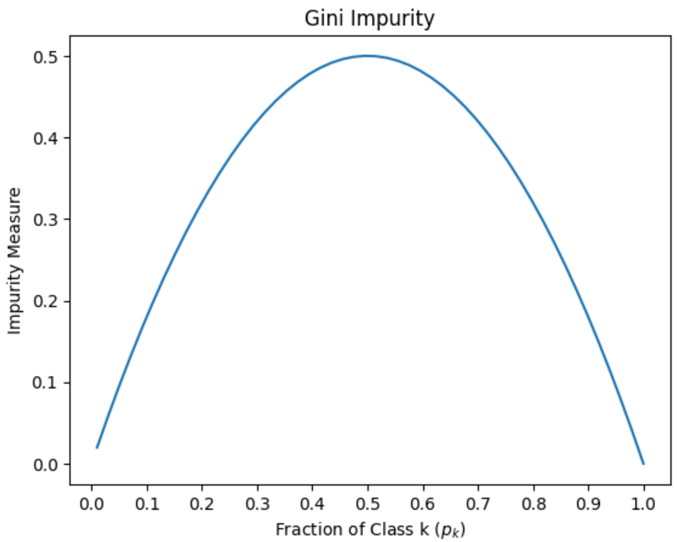
Entropy – Information Gain

$$GAIN_{split} = Entropy(p) - \left(\sum_{i=1}^{k} \frac{n_i}{n} Entropy(i) \right)$$

GNI IMPURITY

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





Gini Vs. Entropy

Laura Elena Raileanu and Kilian Stoffel compared both in "Theoretical comparison between the gini index and information gain criteria". The most important remarks were:

- It only matters in 2% of the cases whether you use gini impurity or entropy.
- Entropy might be a little slower to compute (because it makes use of the logarithm).

Study the behavior of the Gini Index and Information Gain, to give an exact mathematical description of the situations when they are choosing the same test to split on and when they are choosing different tests.

Found that they disagree only in 2% of all cases, which explains why most previously published empirical results concluded that it is not possible to decide which one of the two tests performs better

Published: May 2004

Theoretical Comparison between the Gini Index and Information Gain Criteria

Laura Elena Raileanu & Kilian Stoffel

Annals of Mathematics and Artificial Intelligence **41**, 77–93 (2004) | Cite this article

2960 Accesses | **395** Citations | Metrics

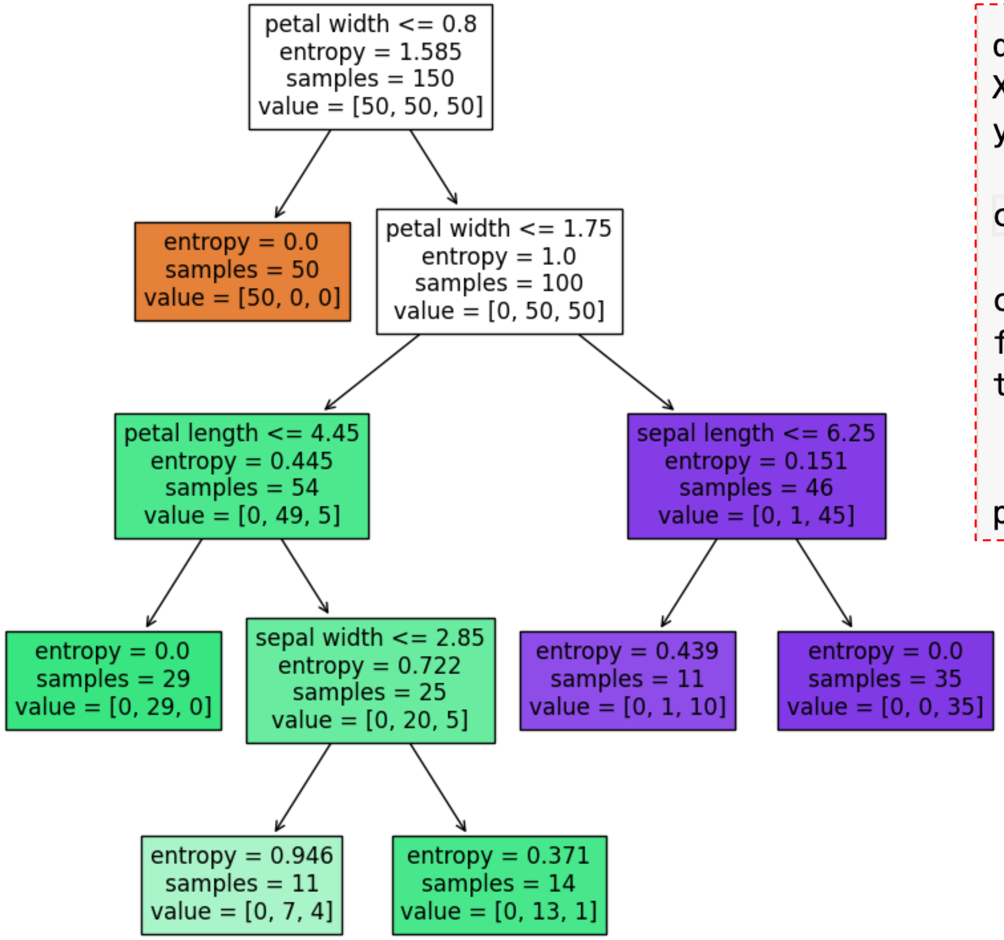
Abstract

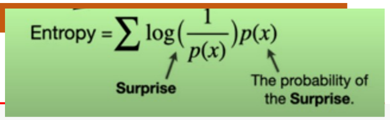
Knowledge Discovery in Databases (KDD) is an active and important research area with the promise for a high payoff in many business and scientific applications. One of the main tasks in KDD is classification. A particular efficient method for classification is decision tree induction. The selection of the attribute used at each node of the tree to split the data (split criterion) is crucial in order to correctly classify objects. Different split criteria were proposed in the literature (Information Gain, Gini Index, etc.). It is not obvious which of them will produce the best decision tree for a given data set. A large amount of empirical tests were conducted in order to answer this question. No conclusive results were found. In this paper we introduce a formal methodology, which allows us to compare multiple split criteria. This permits us to present fundamental insights into the decision process. Furthermore, we are

Classification Tree Review

	Sepal length	Sepal width	Petal length	Petal width	Class
1	5.1	3.5	1.4	0.2	setosa
2	4.9	3.0	1.4	0.2	setosa
3	4.7	3.2	1.3	0.2	setosa
⋮	⋮	⋮	⋮	⋮	⋮
150	5.9	3.0	5.1	1.8	virginica

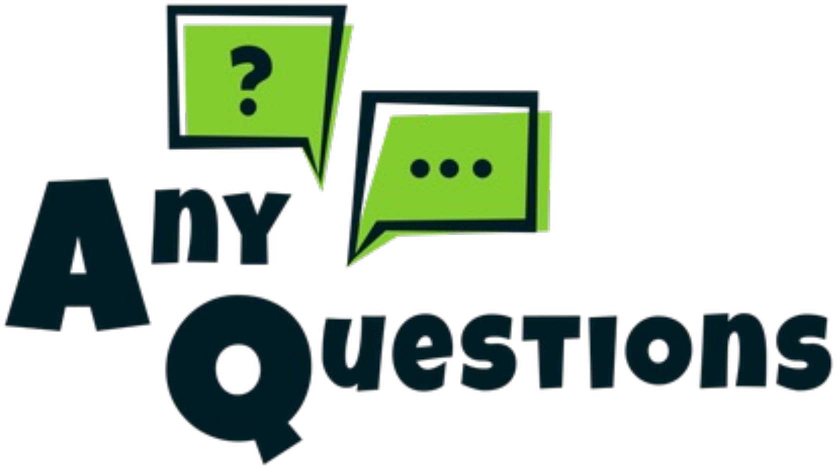
Iris Flower Classification (Entropy)





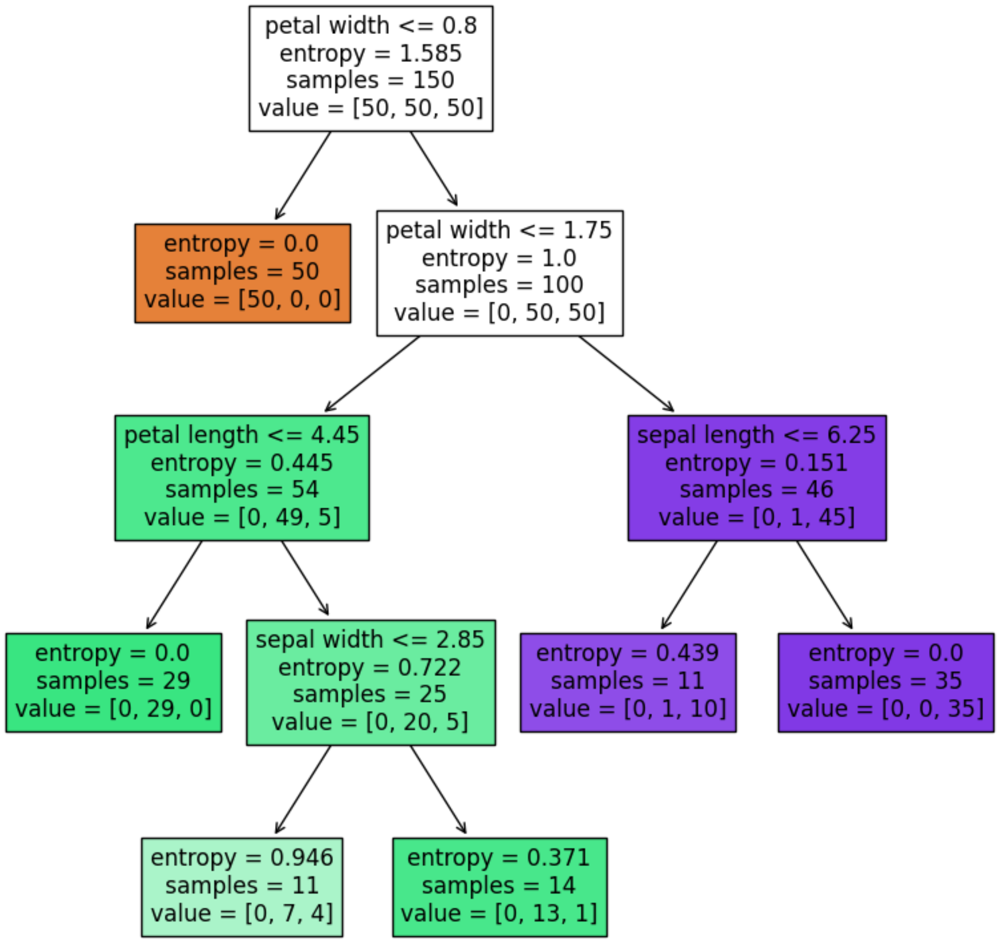
$$E(S) = \sum_{i=1}^{c} -p_i \log_2 p_i$$

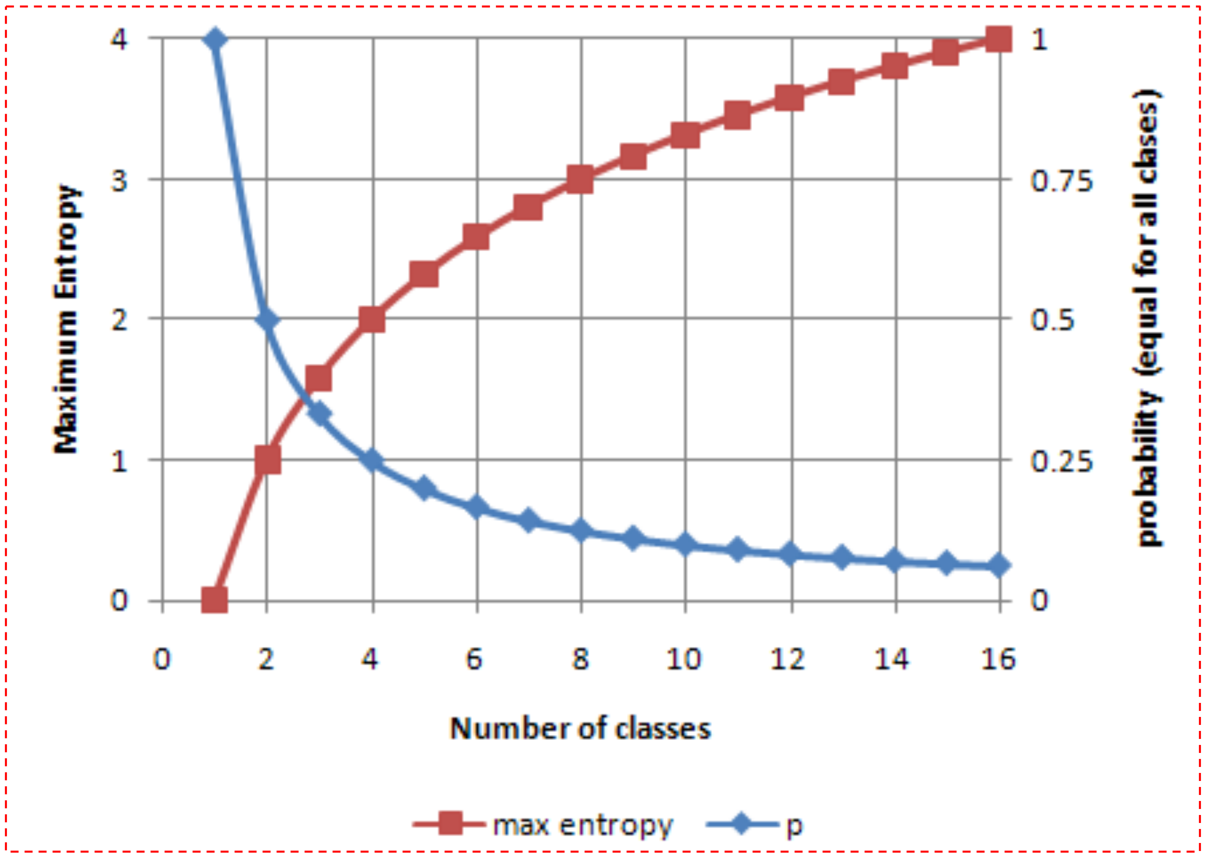
```
dataset = load_iris()
X = dataset.data
y = dataset.target

classifier = tree.DecisionTreeClassifier(criterion="entropy",
                                         max_depth=4, min_samples_leaf=10)
classifier.fit(X,y)
fig, ax = plt.subplots(figsize=(10,10))
tree.plot_tree(classifier,ax=ax, feature_names=["sepal length", "sepal width",
                                                "petal length", "petal width"],
                                                 filled=True)
plt.show()
```



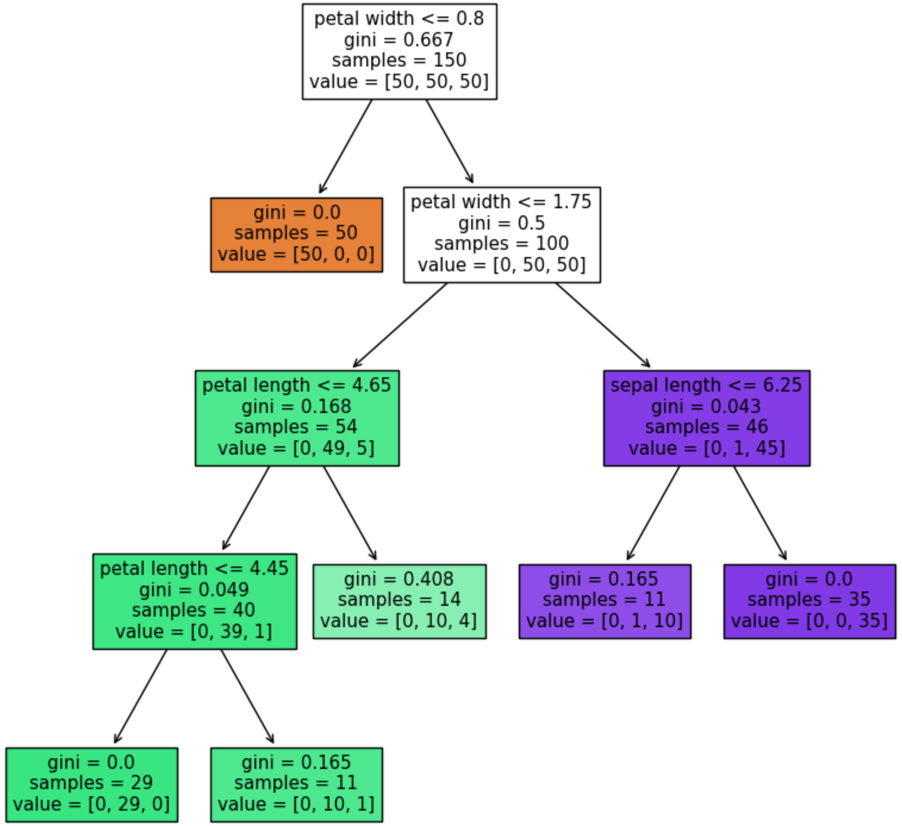
Các bạn có thấy điều gì bất thường ở đây không?

Iris Flower Classification (Entropy)



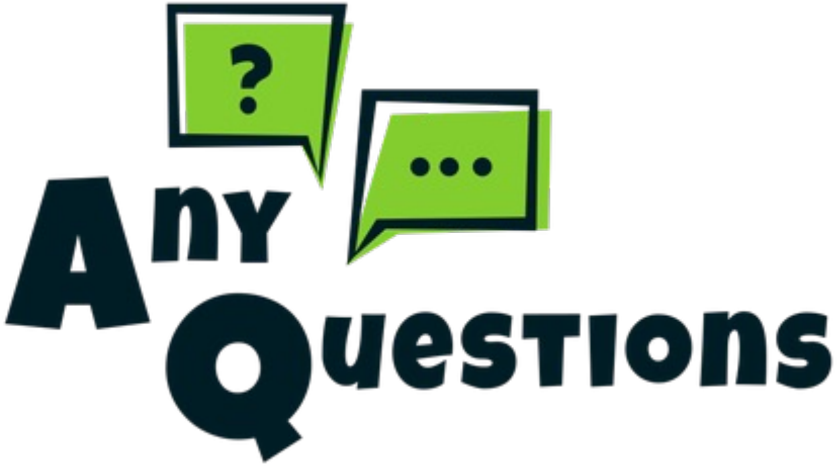


Iris Flower Classification (Gini)



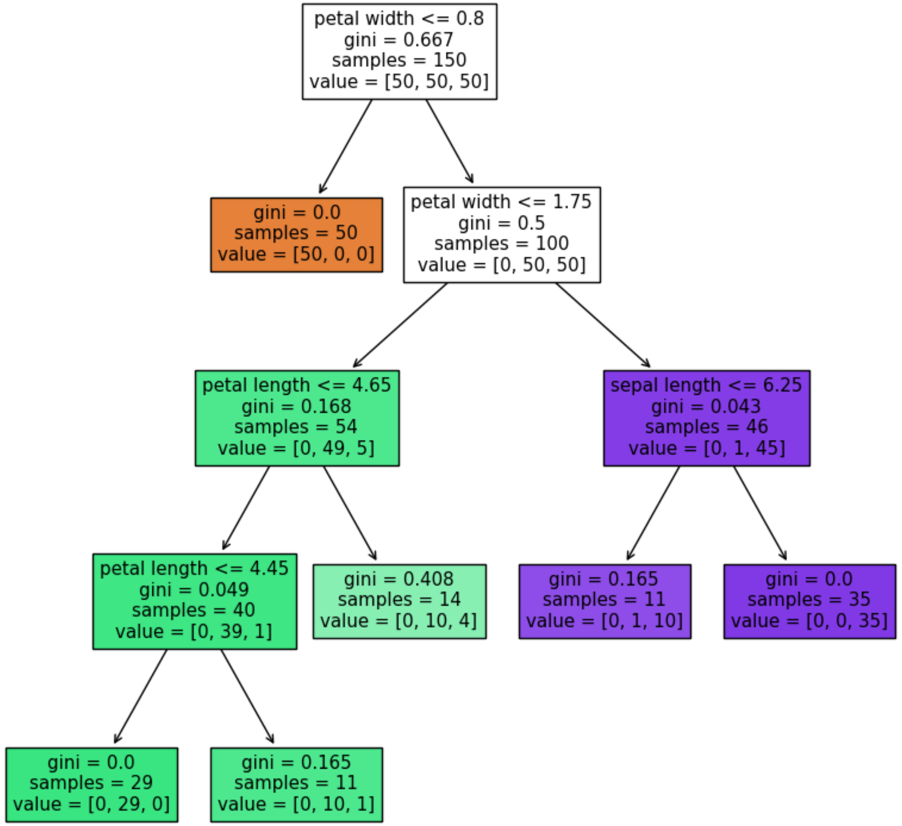

```python
dataset = load_iris()
X = dataset.data
y = dataset.target

classifier = tree.DecisionTreeClassifier(criterion="gini",
                                         max_depth=4, min_samples_leaf=10)
classifier.fit(X,y)
fig, ax = plt.subplots(figsize=(10,10))
tree.plot_tree(classifier,ax=ax, feature_names=["sepal length", "sepal width",
                                                "petal length", "petal width"],
                                                 filled=True)
plt.show()
```

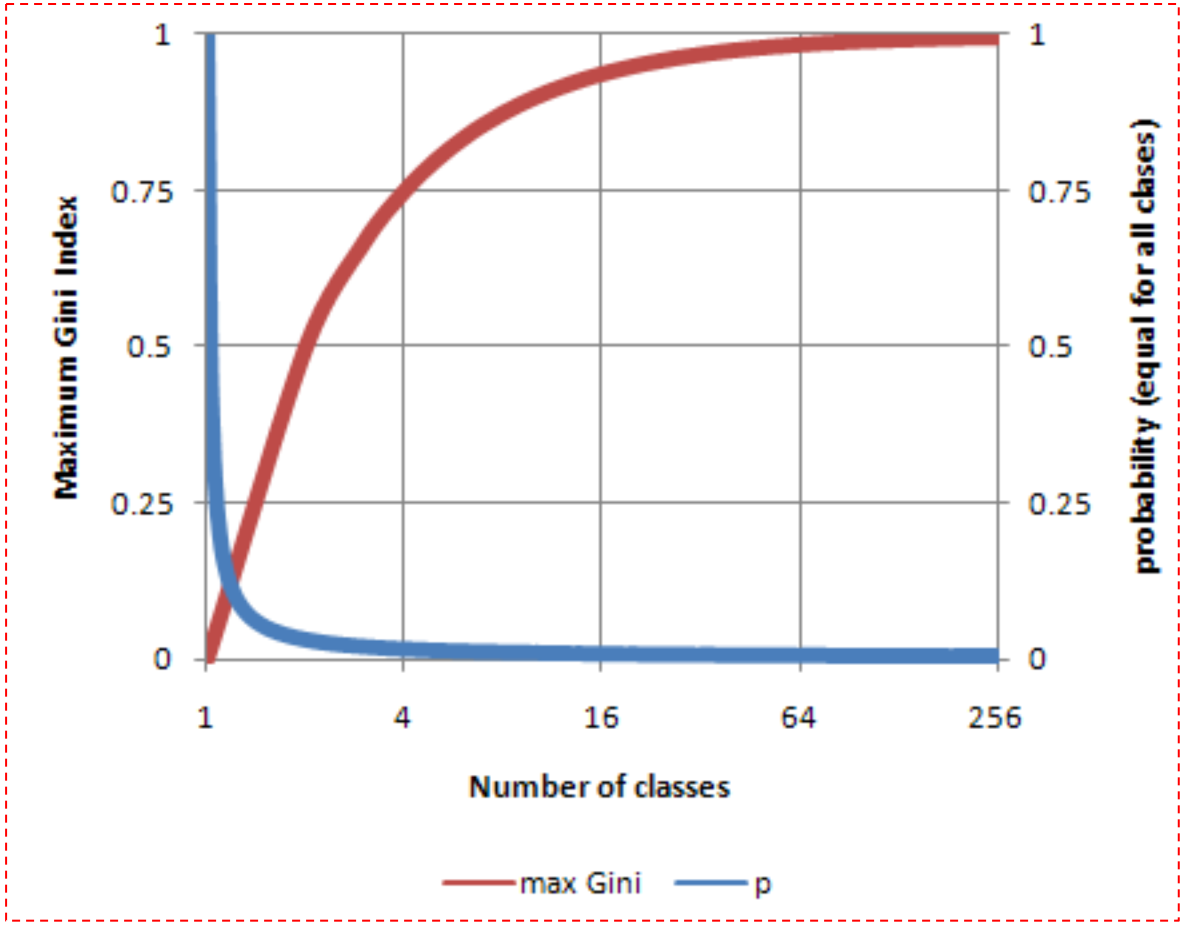
$$Gini = 1 - \sum_{i=1}^{C} (p_i)^2$$



Các bạn có thấy điều gì bất thường ở đây không?

Iris Flower Classification (Gini)

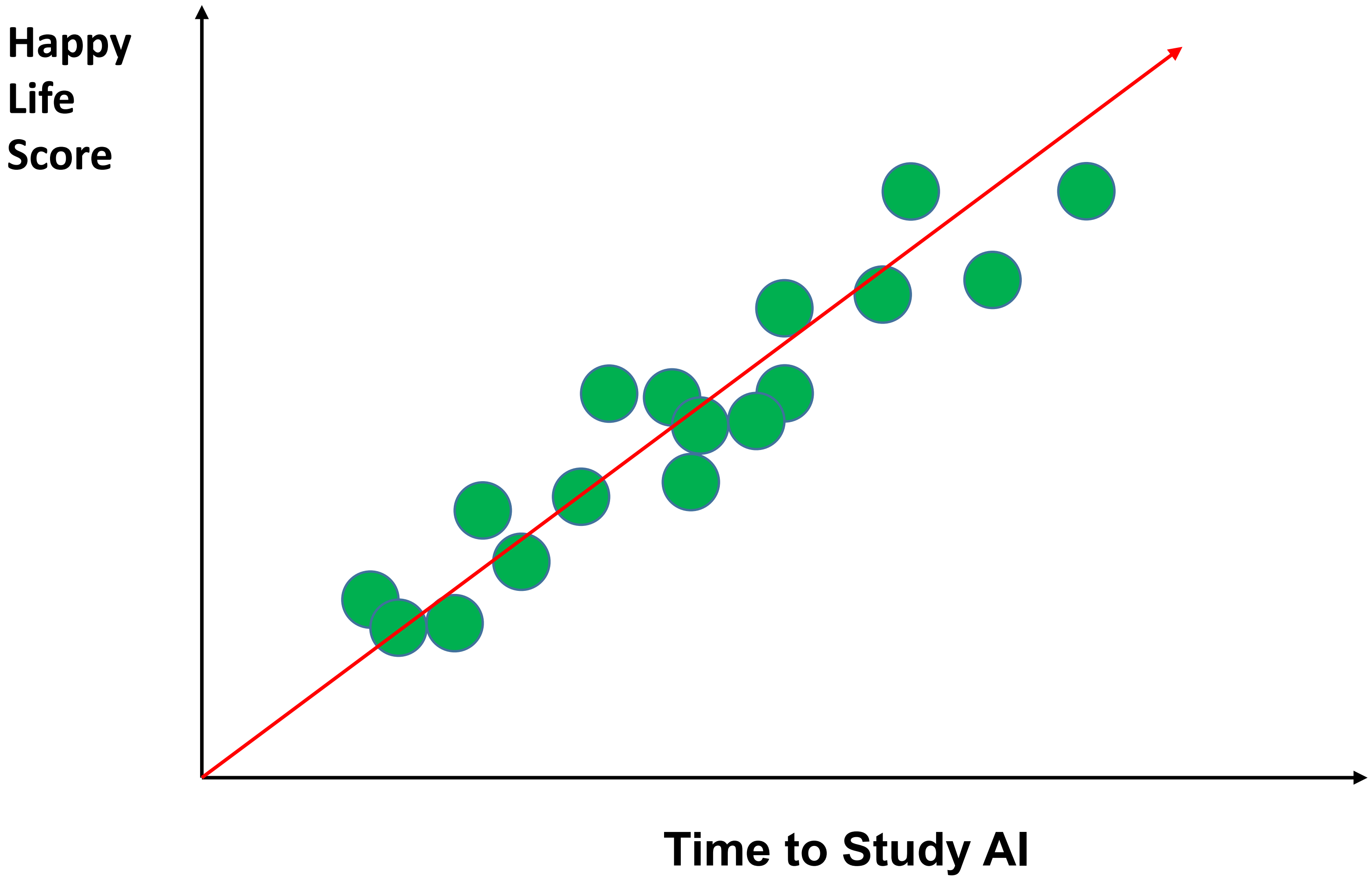




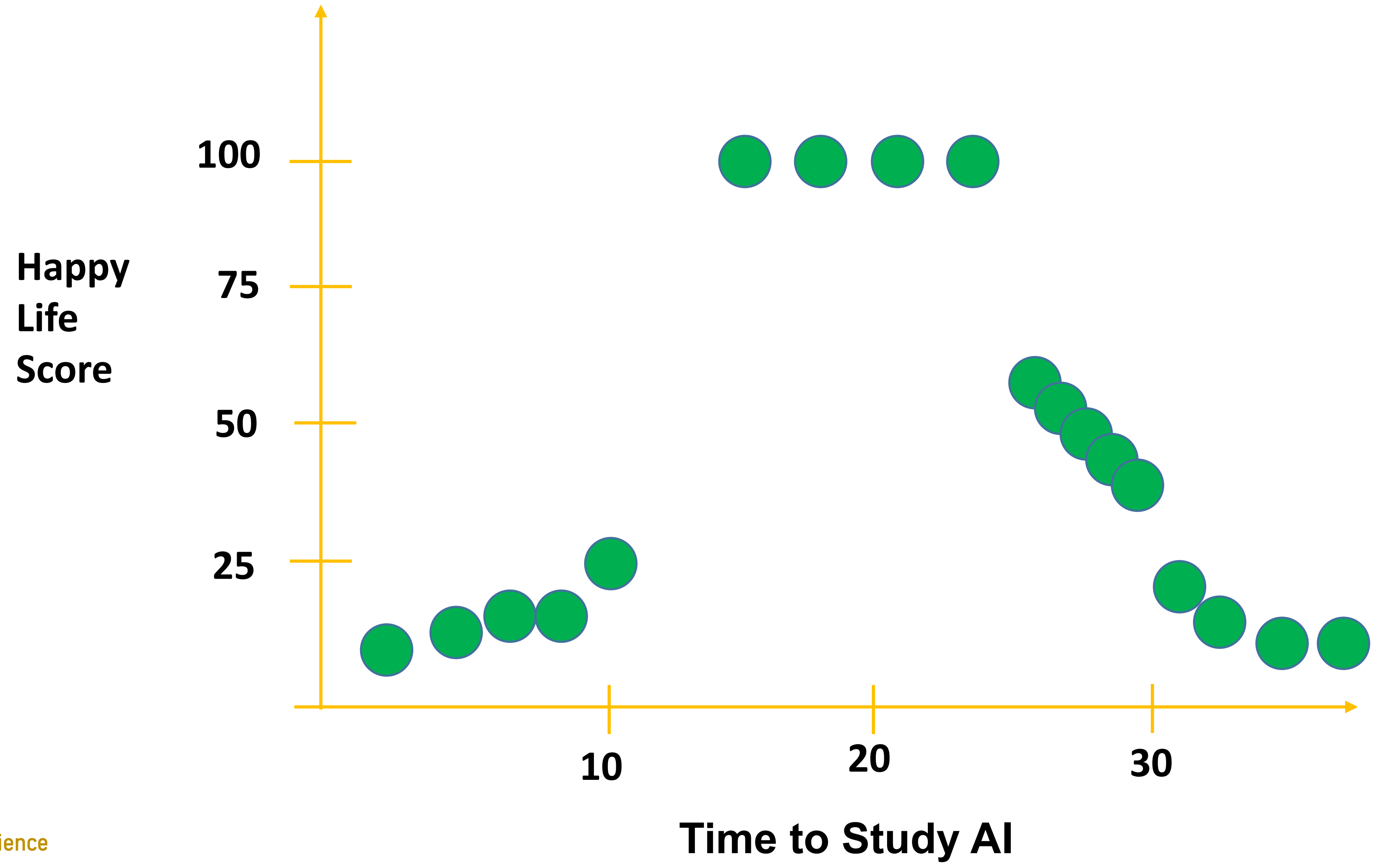
Outline

- ➢ Motivation for Regression Tree
- ➢ Regression Tree
- ➢ Overfitting in Regression Tree
- ➢ Case study

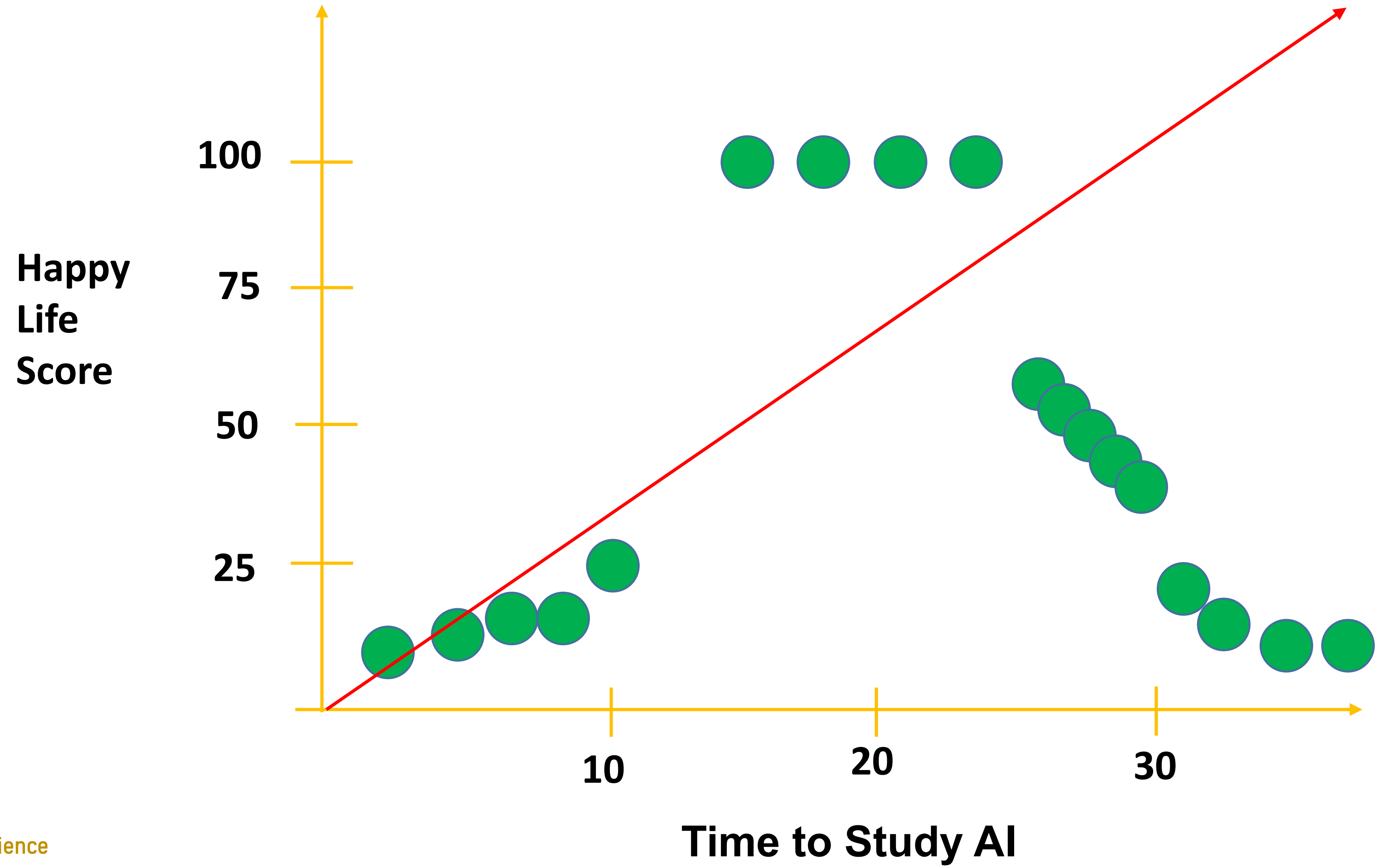
Motivation



Motivation



Motivation



Case study

Supposing that, you want to research and develop a new vaccine to cure the Covid-19

The StatQuest Illustrated Guide To Machine Learning, Josh Starmer PhD

Case study

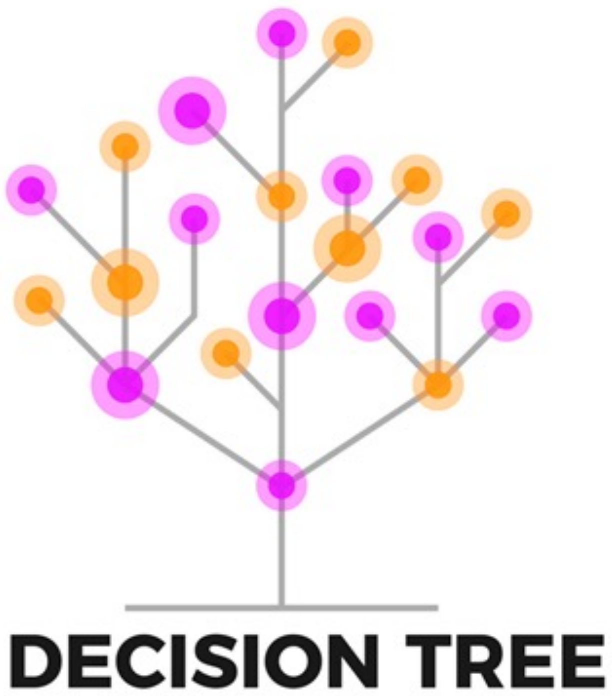
Unit	Age	Sex	Effect (%)
10	25	Female	98
20	73	Male	0
35	54	Female	100
5	12	Male	44
…	…	…	…

Tiêm 5 đơn vị vaccine, 12 tuổi, giới tính nam

Hiệu quả vaccine: 44%

Khi có 1 vaccine ra đời, chúng ta muốn dự đoán xem nó hiệu quả bao nhiêu % ứng với liều lượng dùng cố định (unit), tuổi (age) và giới tính (sex) của bệnh nhân.

Can we use Decision Tree for solving this research?



Outline

- Motivation for Regression Tree
- Regression Tree
- Overfitting in Regression Tree
- Case study

Which node is root?

Unit(đơn vị)	Effect (hiệu quả) (%)
10	98
20	0
35	100
5	44
…	…

Tiêm 5 đơn vị vaccine

Hiệu quả vaccine: 44%

Khi có 1 vaccine ra đời, chúng ta muốn dự đoán xem nó hiệu quả bao nhiêu % ứng với từng liều lượng (unit) dùng trên bệnh nhân.

Which node is root?

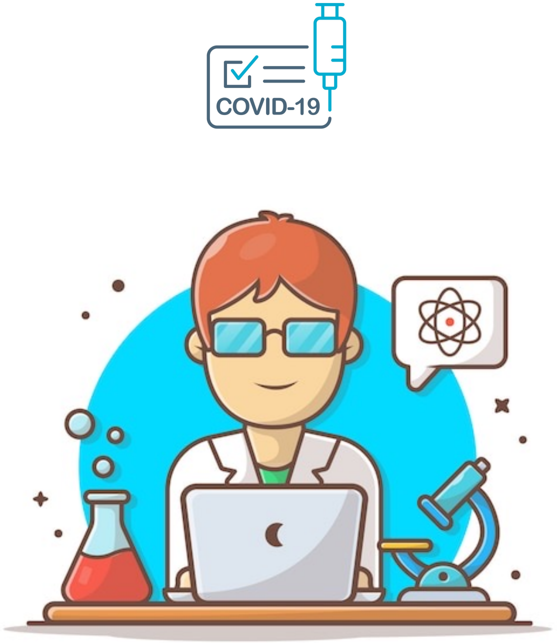
Age	Effect (hiệu quả) (%)
25	98
73	0
54	100
12	44
…	…

12 tuổi

Hiệu quả vaccine: 44%

Khi có 1 vaccine ra đời, chúng ta muốn dự đoán xem nó hiệu quả bao nhiêu % ứng với tuổi (age) của bệnh nhân.

Which node is root?



Sex	Effect (hiệu quả) (%)
Female	98
Male	0
Female	100
Male	44
…	…

Giới tính Male

Hiệu quả vaccine: 44%

Khi có 1 vaccine ra đời, chúng ta muốn dự đoán xem nó hiệu quả bao nhiêu % ứng với giới tính (sex) của bệnh nhân.

Unit is a root node

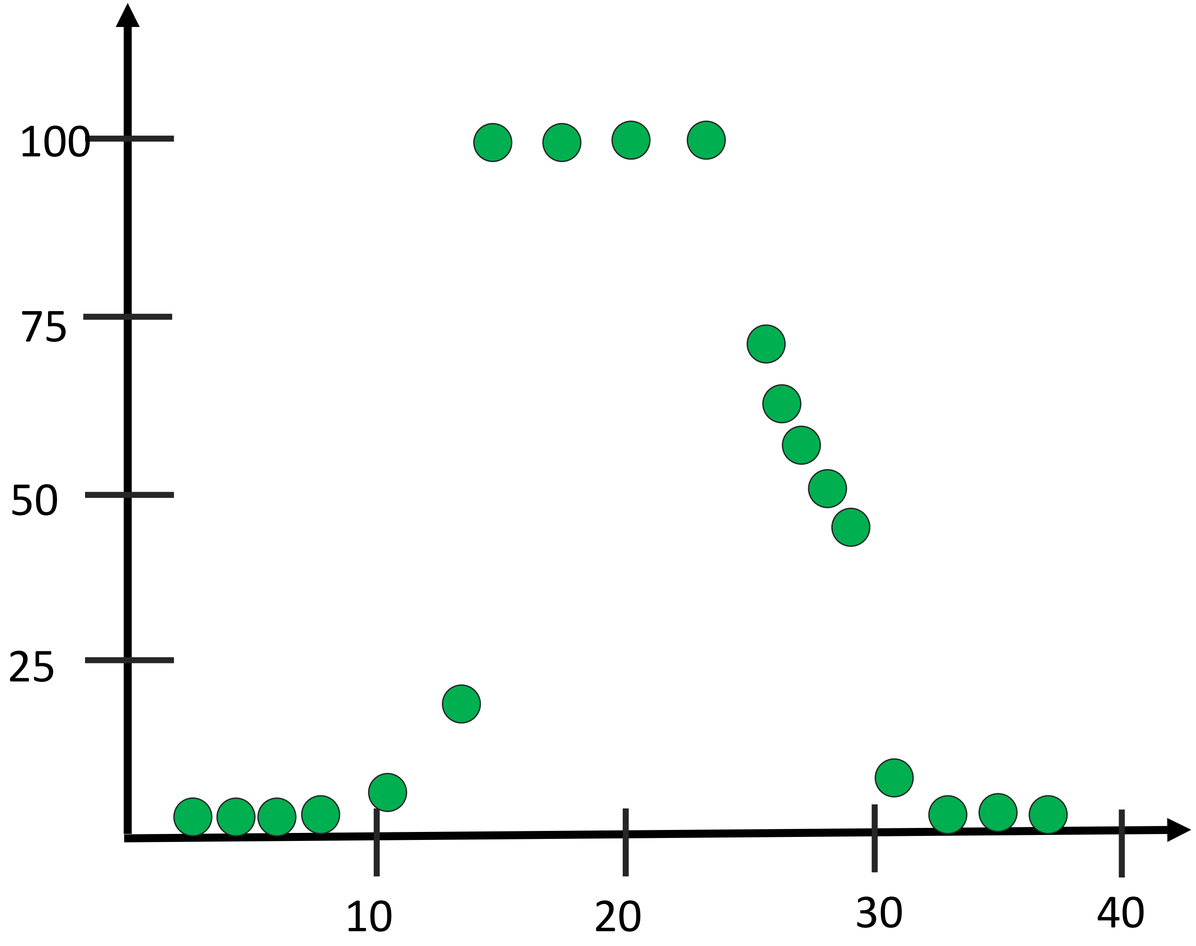
Unit(đơn vị)	Effect (hiệu quả) (%)
10	98
20	0
35	100
5	44
…	…

Tiêm 5 đơn vị vaccine

Hiệu quả vaccine: 44%

Khi có 1 vaccine ra đời, chúng ta muốn dự đoán xem nó hiệu quả bao nhiêu % ứng với từng liều lượng (unit) dùng trên bệnh nhân.

Unit is a root node



Hiệu quả (%)

Đơn vị vaccine

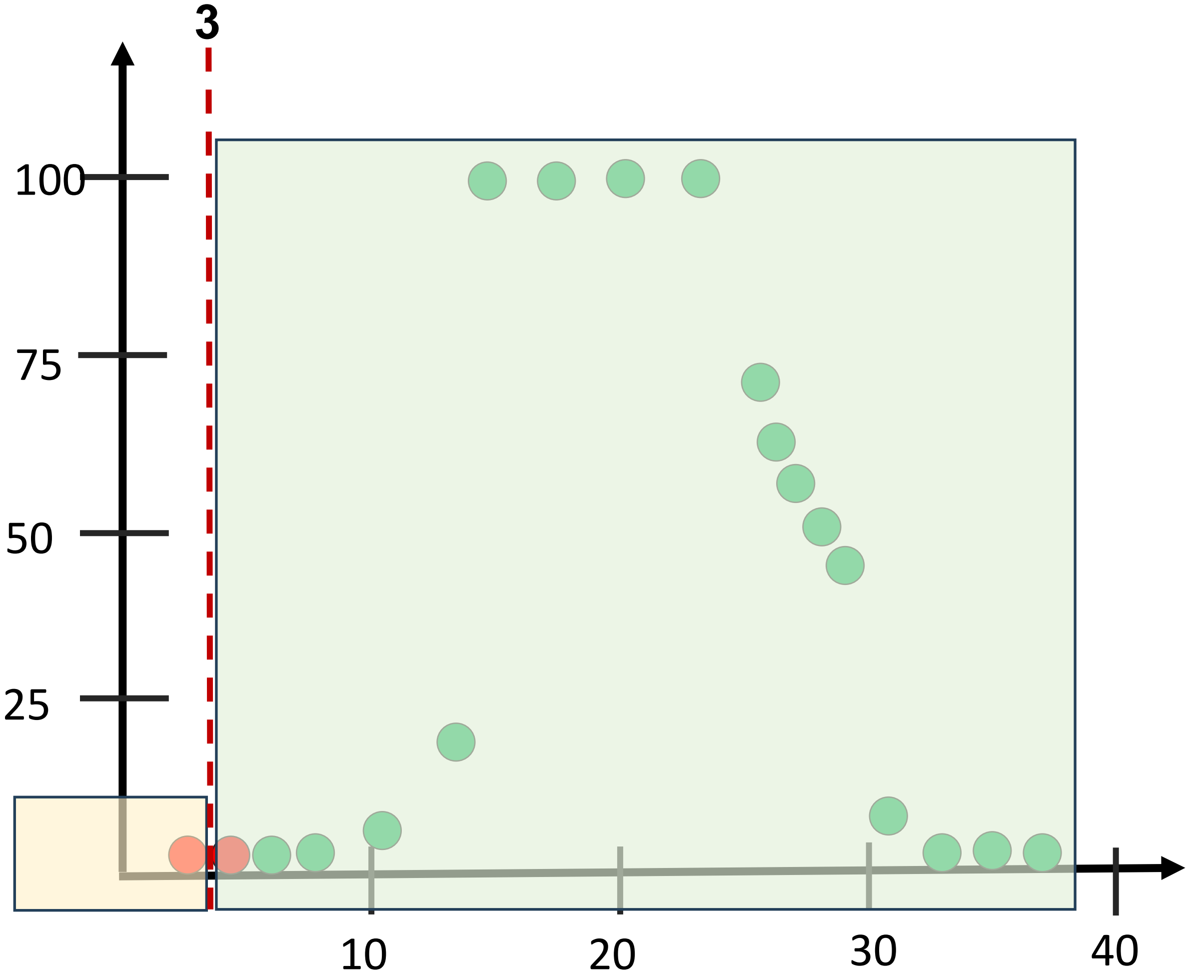
Tiêm 5 đơn vị vaccine

Hiệu quả vaccine: 44%

Khi có 1 vaccine ra đời, chúng ta muốn dự đoán xem nó hiệu quả bao nhiêu % ứng với từng liều lượng dùng (unit) trên bệnh nhân.

Unit is a root node

Effectiveness (Hiệu quả) (%)

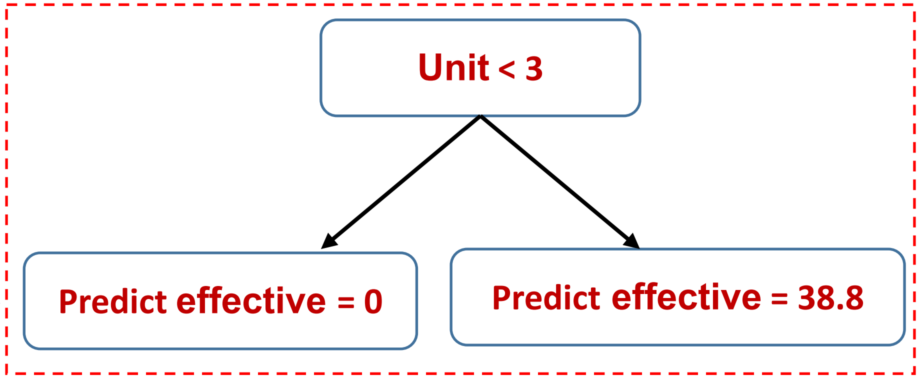


Unit (Đơn vị) vaccine

Average in unit( ) = 3

Average in effectiveness( ) = 0

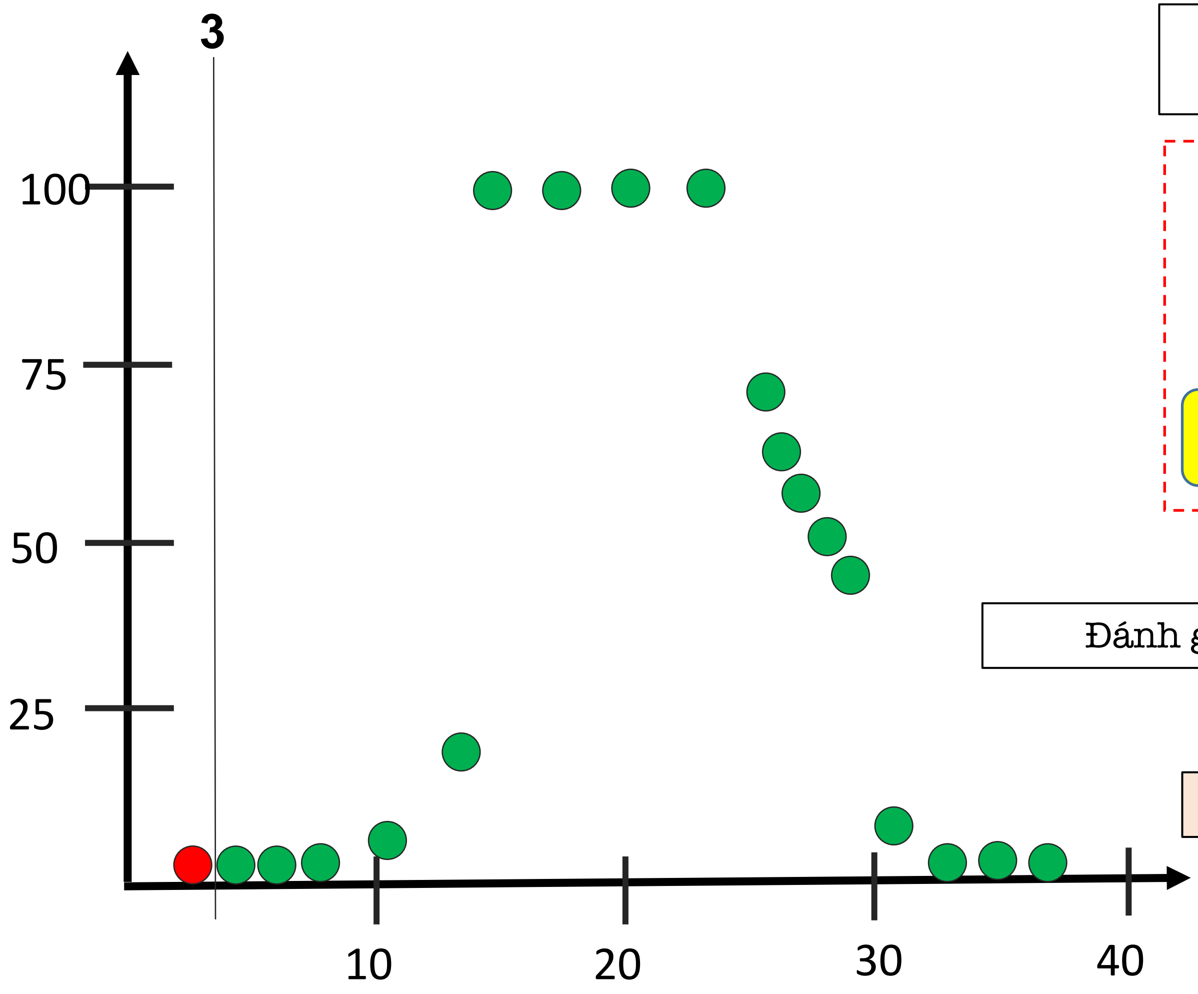
Average in effectiveness( ) = 38.8



Khi có 1 vaccine ra đời, chúng ta muốn dự đoán xem nó hiệu quả bao nhiêu % ứng với từng liều lượng dùng (unit) trên bệnh nhân.

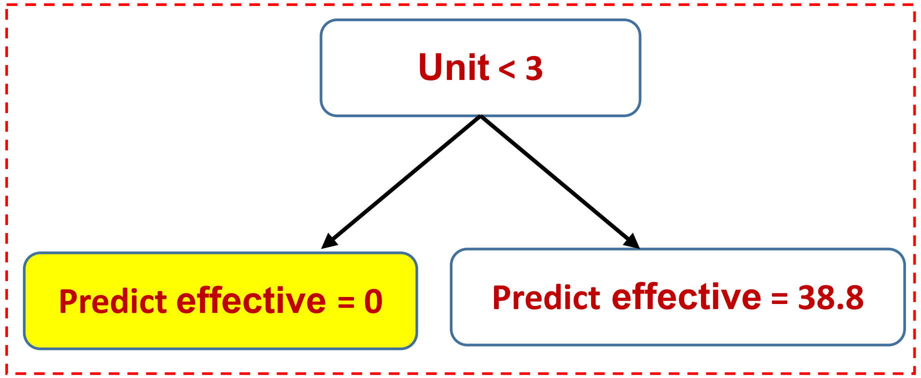
Unit is a root node

Effectiveness
(Hiệu quả)
(%)



Unit (Đơn vị) vaccine

Để đánh giá chất lượng cây mới tạo, chúng ta cần đánh giá residual error (lỗi) của nó



Đánh giá residual error trong trường hợp unit < 3

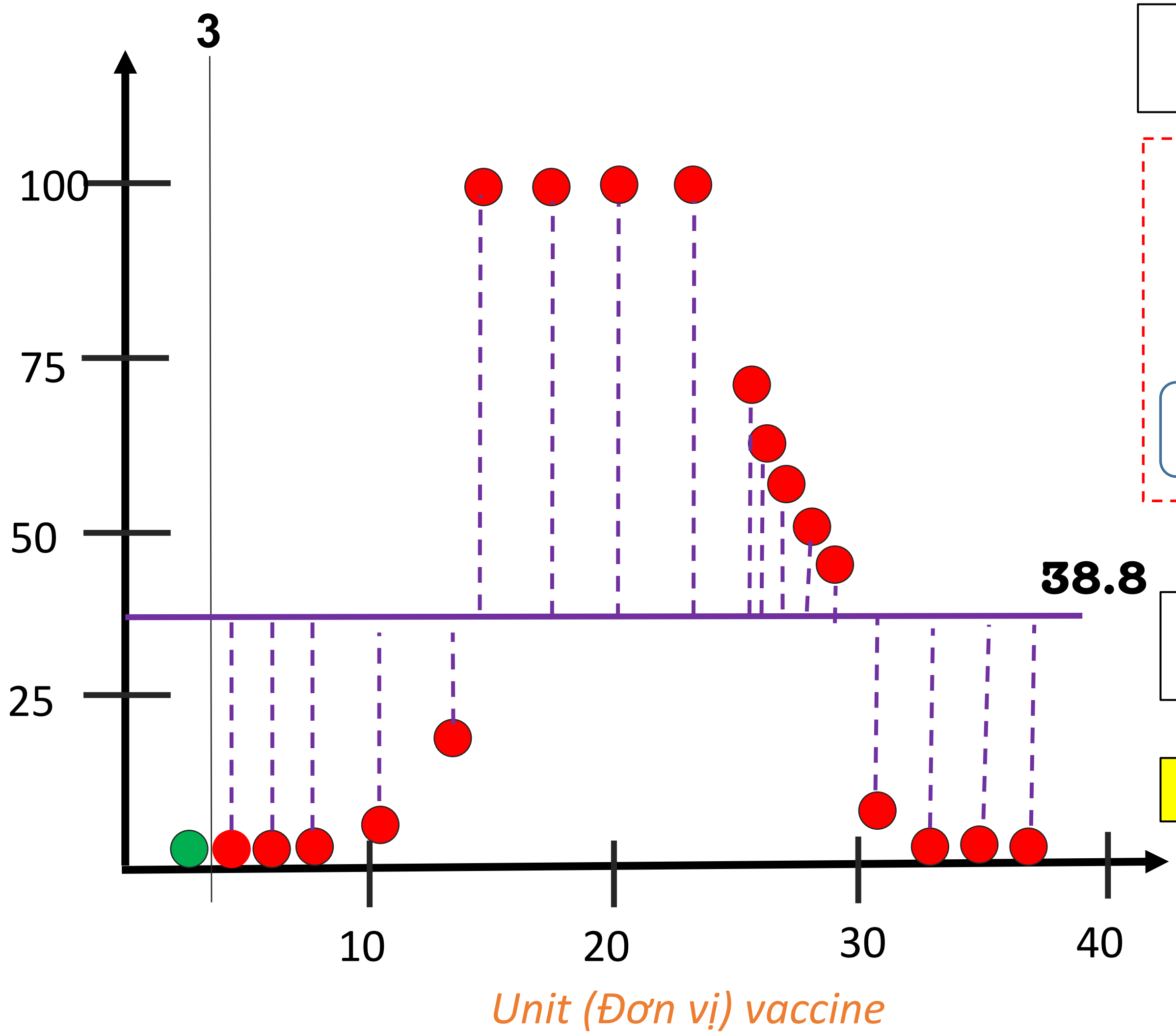
Sum of Square Error (SSR) = 0

number of samples — real value — predicted value

$$\underbrace{\sum_{i=1}^{\boxed{n}} \left(\boxed{Y_i} - \boxed{\hat{Y}_i}\right)^2}_{\text{sum of the errors of all samples}}$$

Unit is a root node

Effectiveness
(Hiệu quả)
(%)



Để đánh giá chất lượng cây mới tạo, chúng ta cần đánh giá residual error (lỗi) của nó

Unit < 3

Predict effective = 0

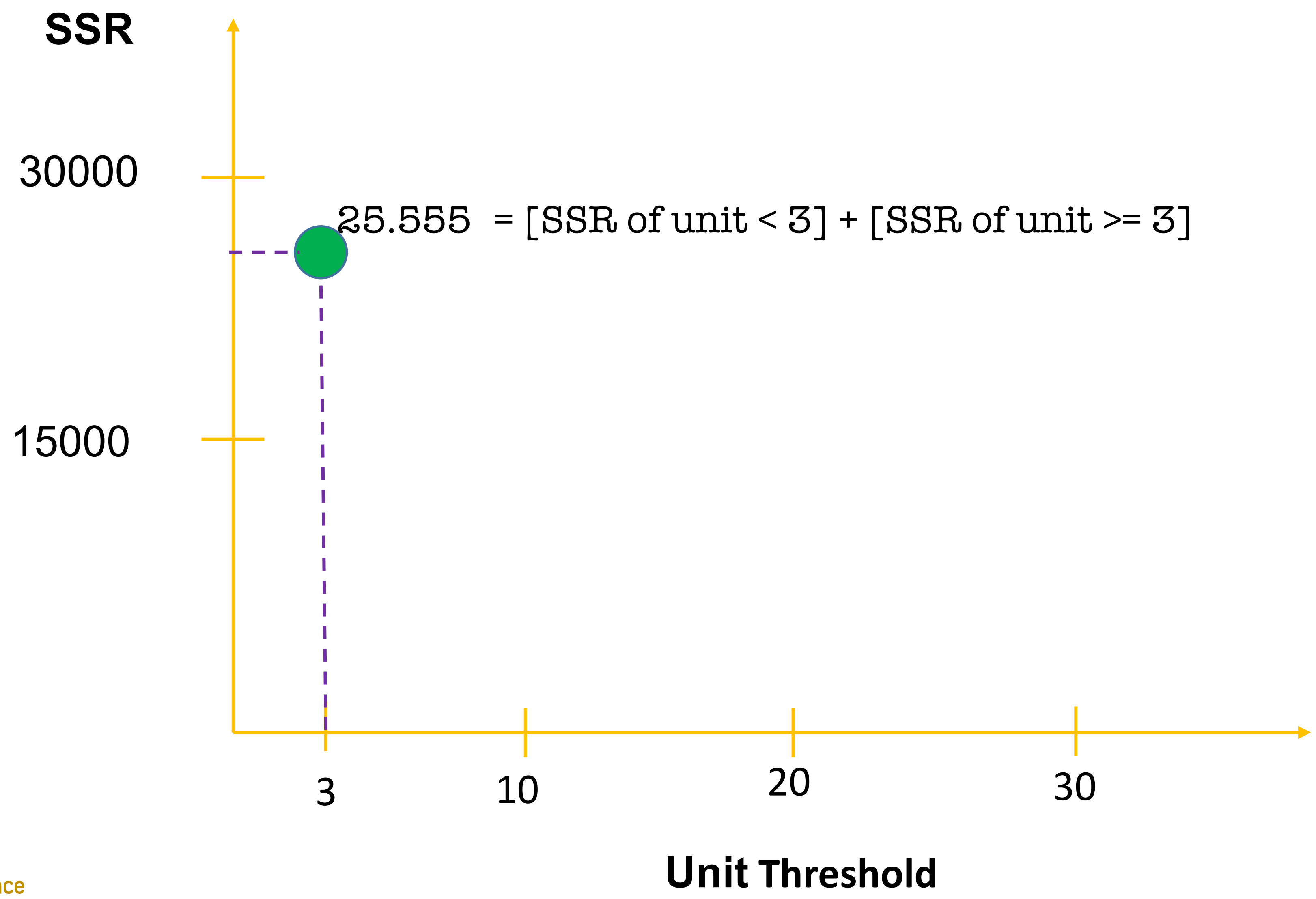
Predict effective = 38.8

Đánh giá residual error trong trường hợp unit >= 3

Sum of Square Error (SSR) = 25.555

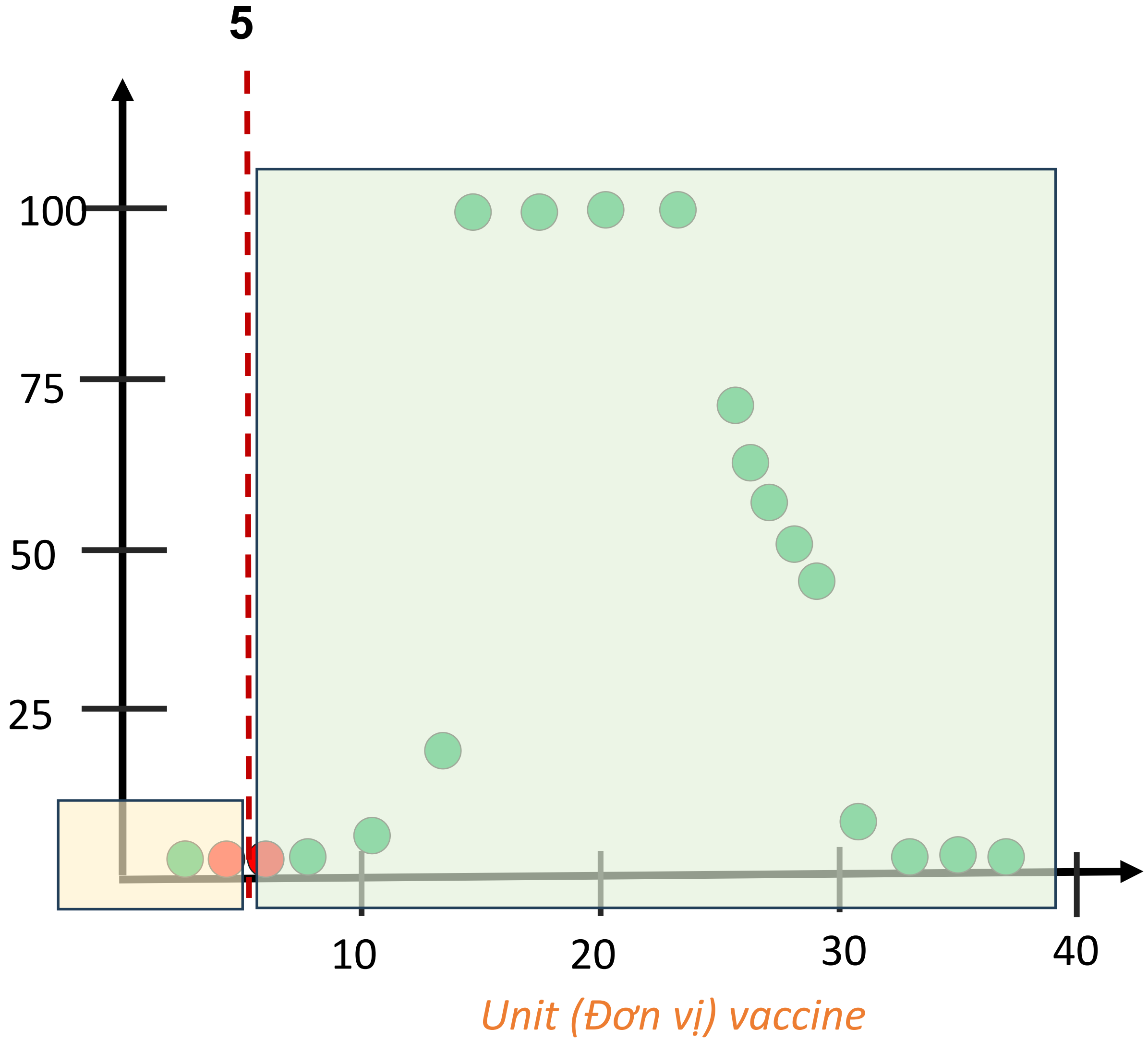
$$\underbrace{\sum_{i=1}^{\boxed{n}} (\boxed{Y_i} - \boxed{\hat{Y}_i})^2}_{\text{sum of the errors of all samples}}$$

number of samples; real value; predicted value


SSR
30000
15000
25.555 = [SSR of unit < 3] + [SSR of unit >= 3]
3
10
20
30
Unit Threshold

Unit is a root node

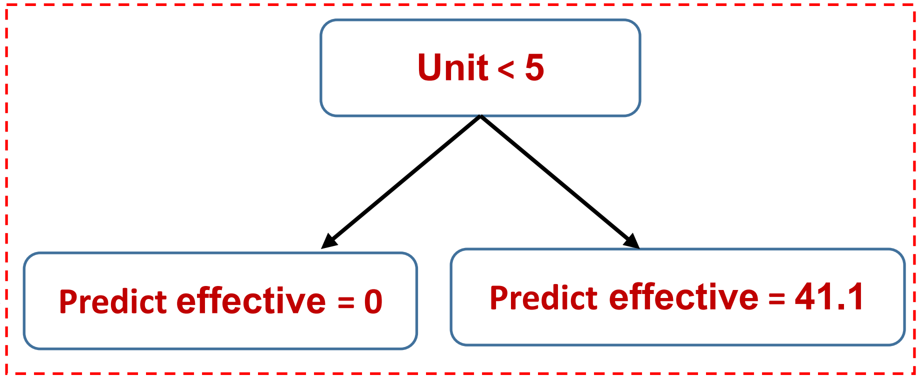
Effectiveness
(Hiệu quả)
(%)



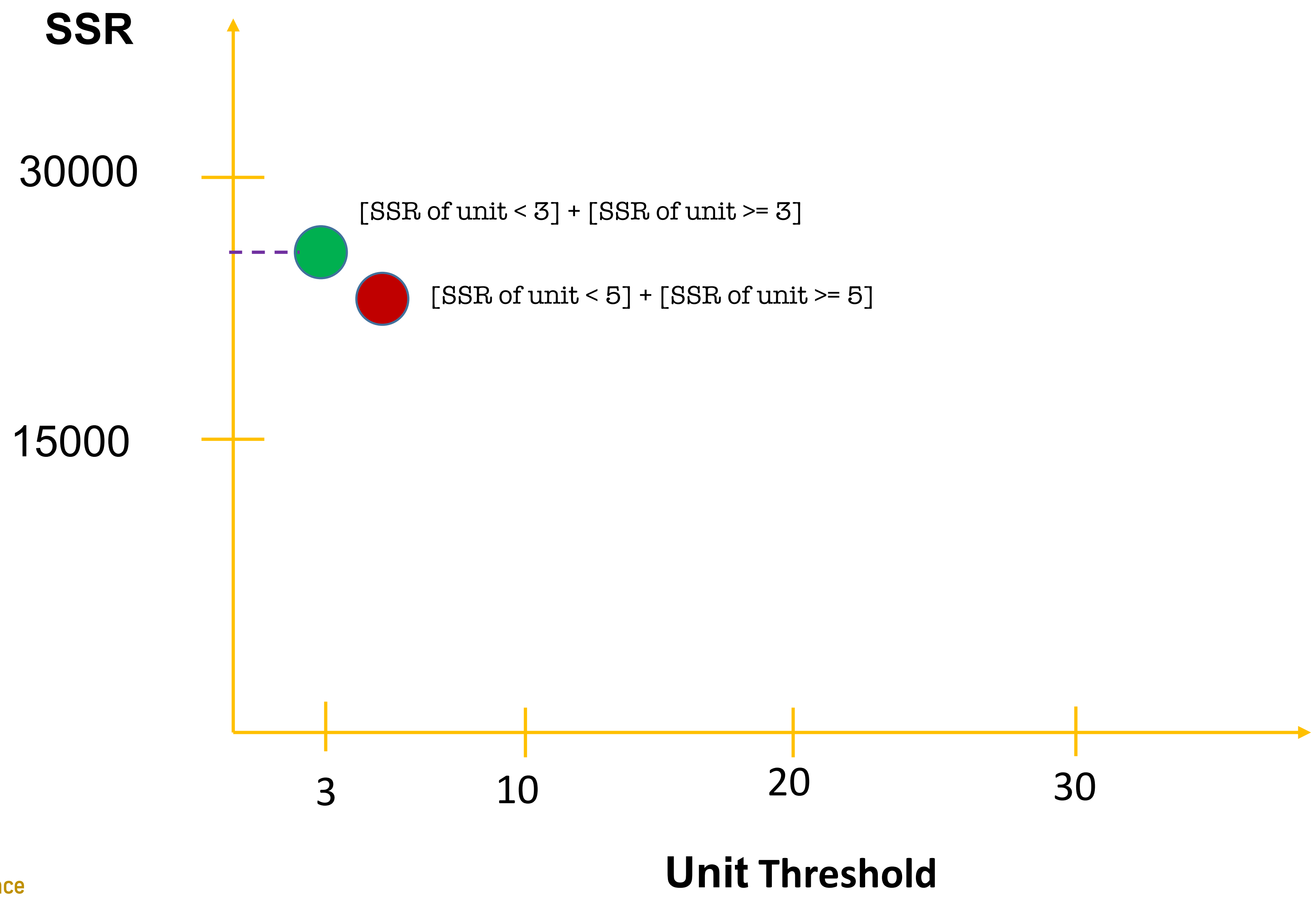
Average in unit( ) = 5

Average in effectiveness( ) = 0

Average in effectiveness( ) = 41.1

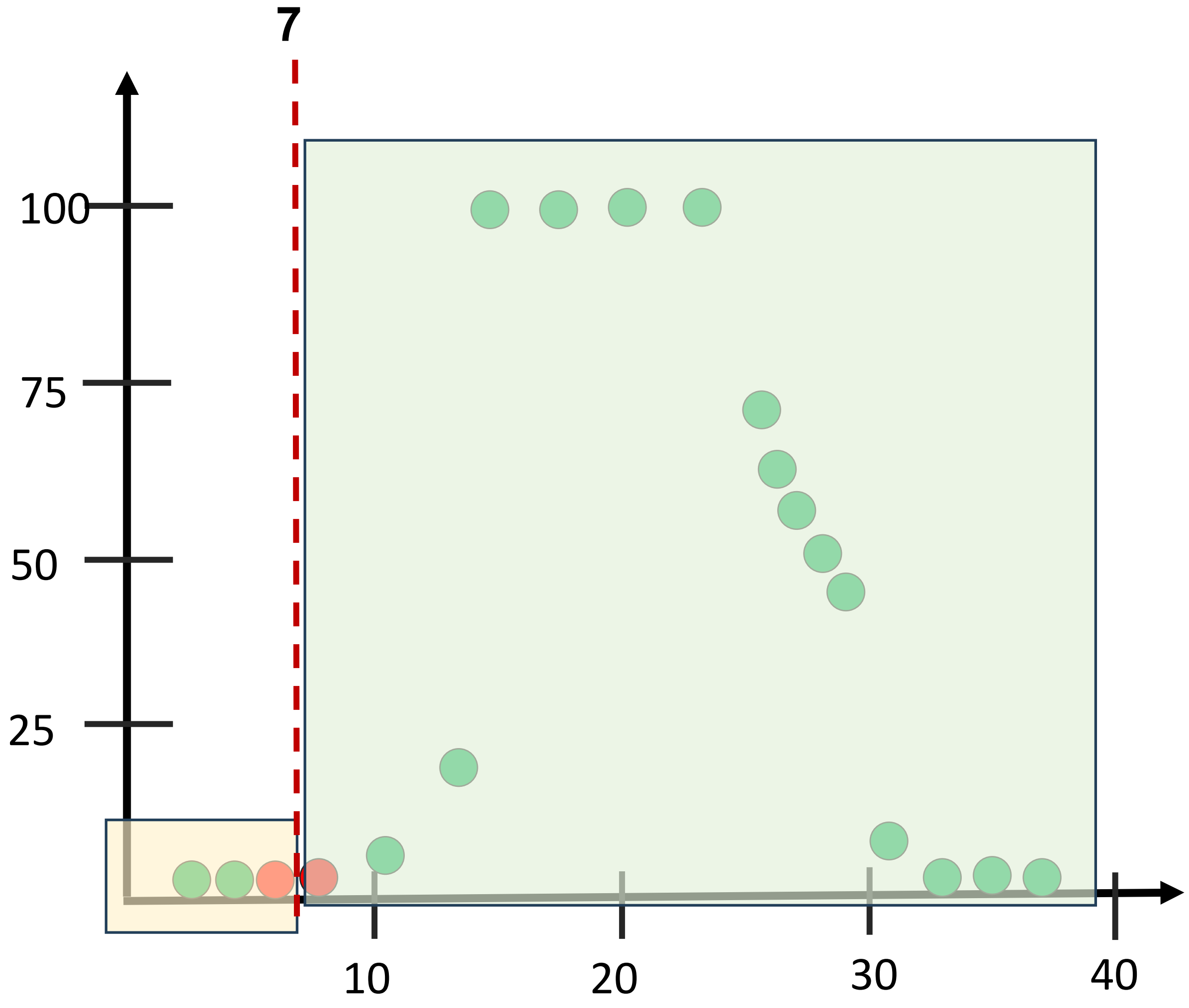


Compute SSR


SSR
30000
15000
[SSR of unit < 3] + [SSR of unit >= 3]
[SSR of unit < 5] + [SSR of unit >= 5]
3
10
20
30
Unit Threshold

Unit is a root node

Effectiveness (Hiệu quả) (%)

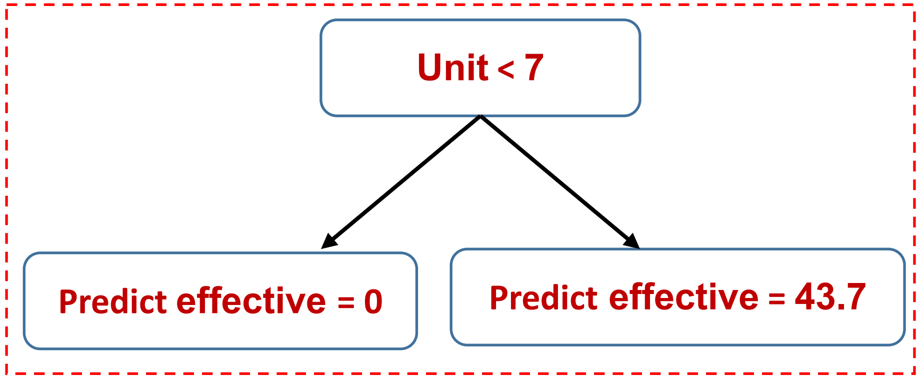


Unit (Đơn vị) vaccine

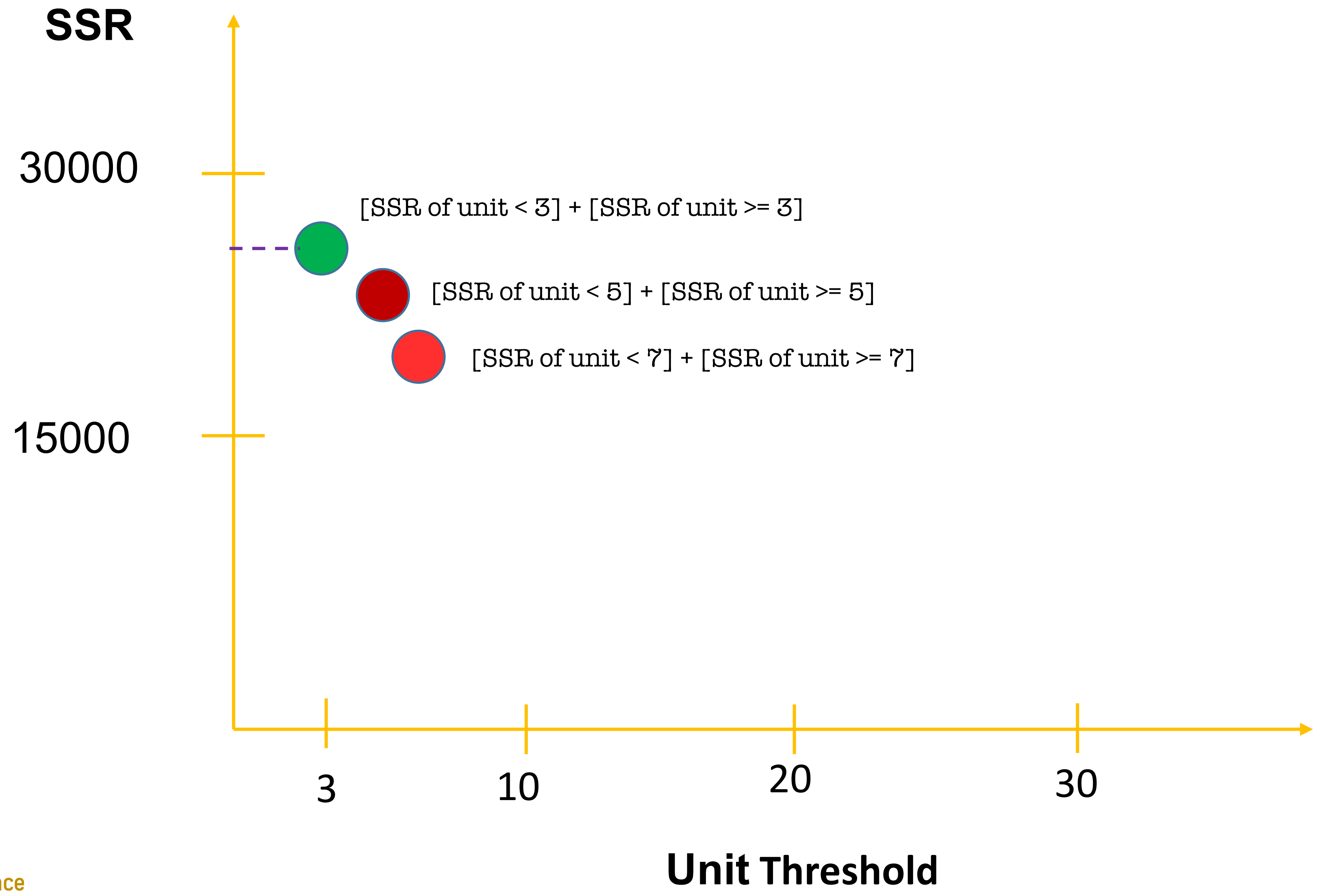
Average in unit( ) = 7

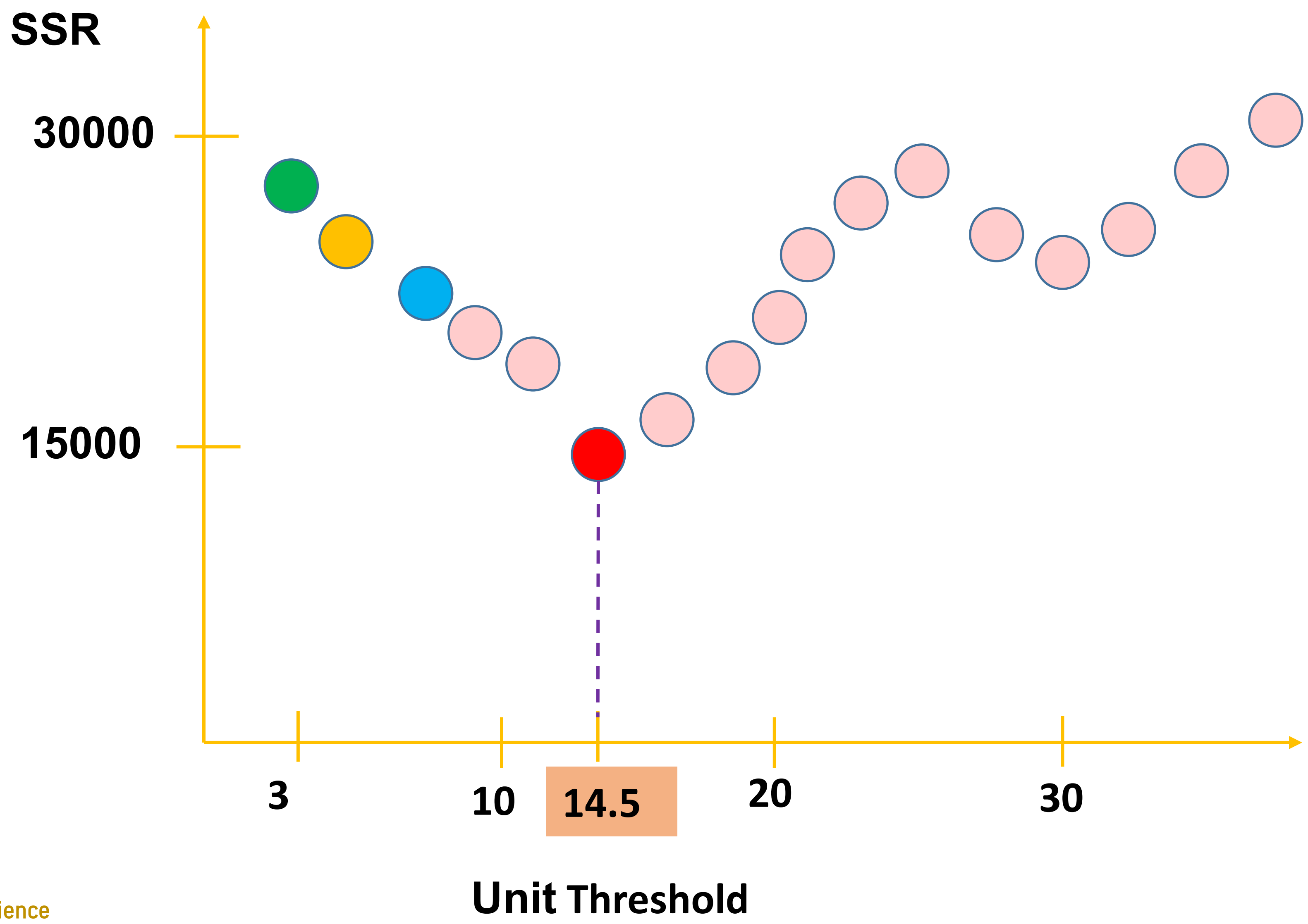
Average in effectiveness( ) = 0

Average in effectiveness( ) = 43.7



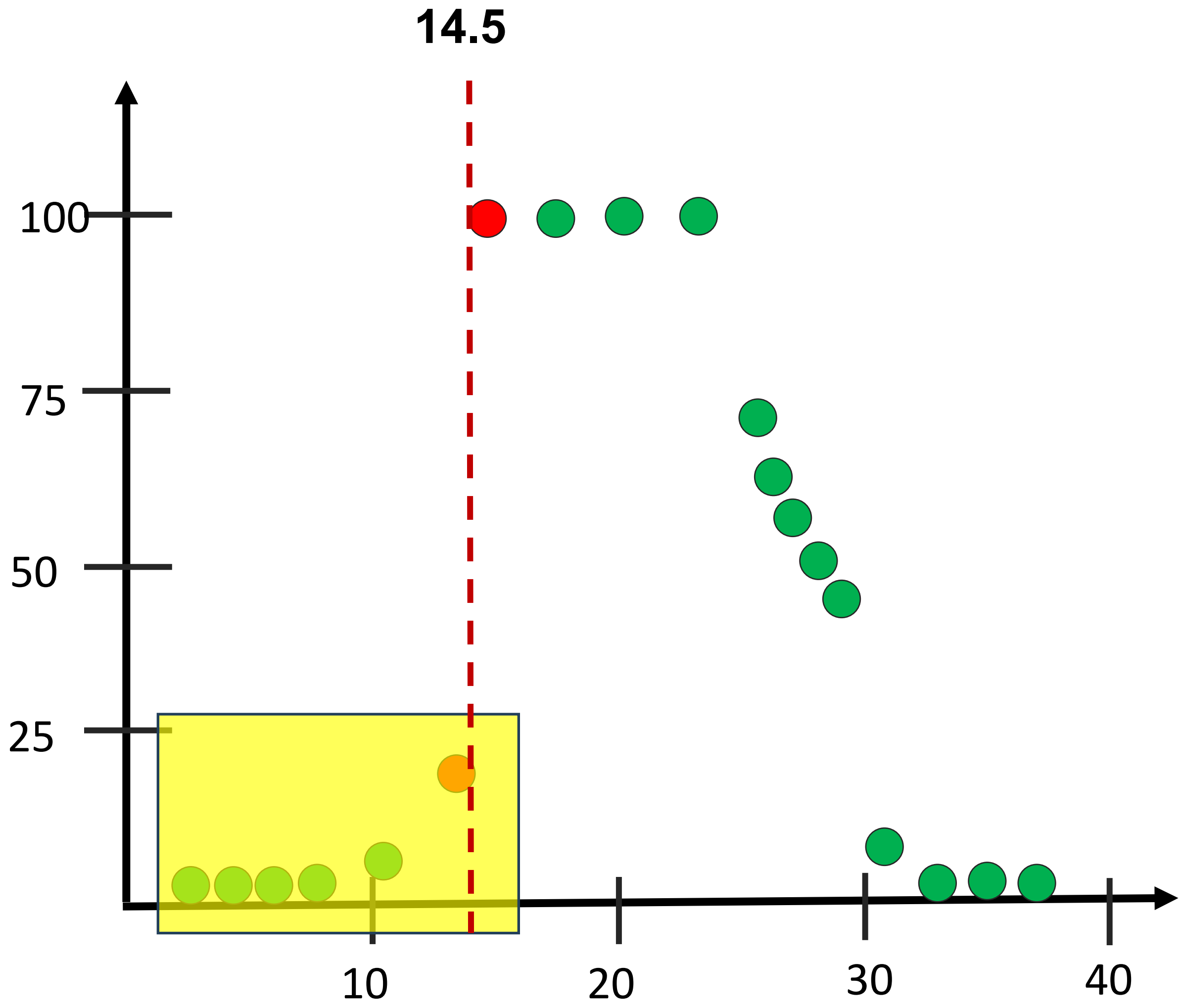
Compute SSR


SSR
30000
15000
[SSR of unit < 3] + [SSR of unit >= 3]
[SSR of unit < 5] + [SSR of unit >= 5]
[SSR of unit < 7] + [SSR of unit >= 7]
3
10
20
30
Unit Threshold


SSR
30000
15000
3
10
14.5
20
30
Unit Threshold

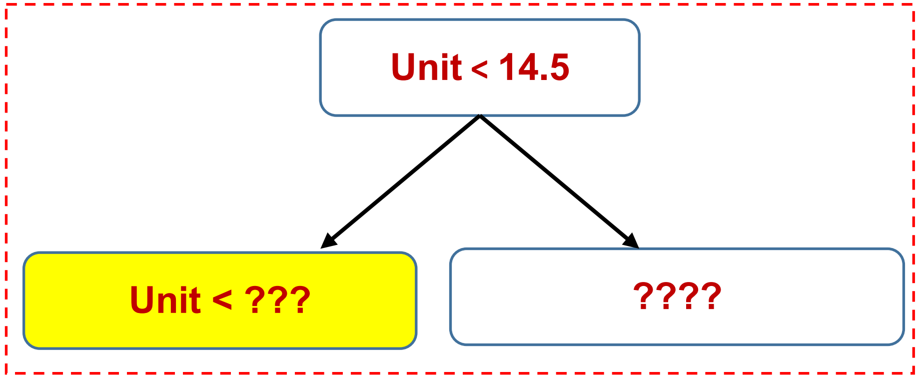
Unit is a root node

Effectiveness
(Hiệu quả)
(%)



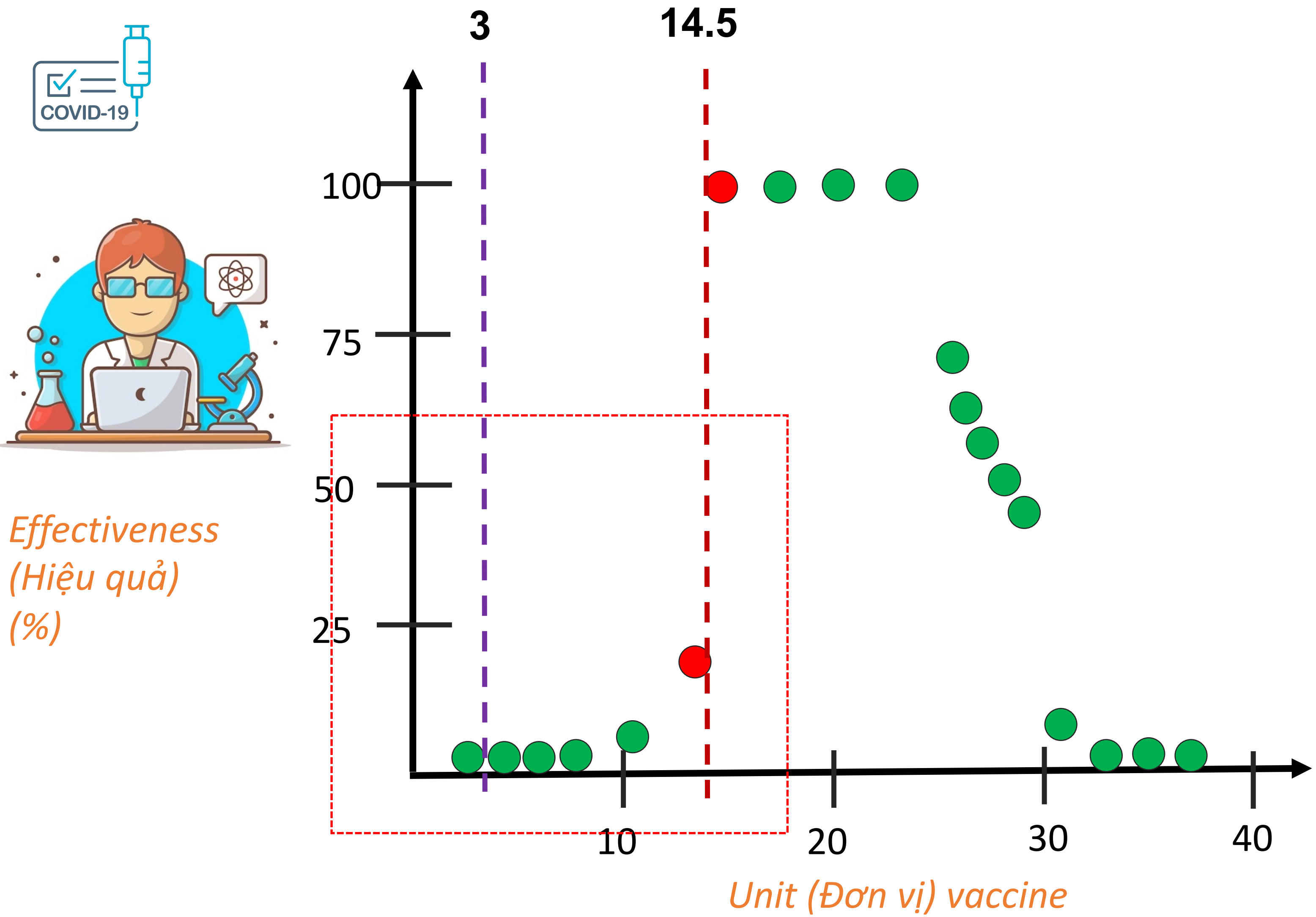
Unit (Đơn vị) vaccine

Average in unit( ) = 14.5

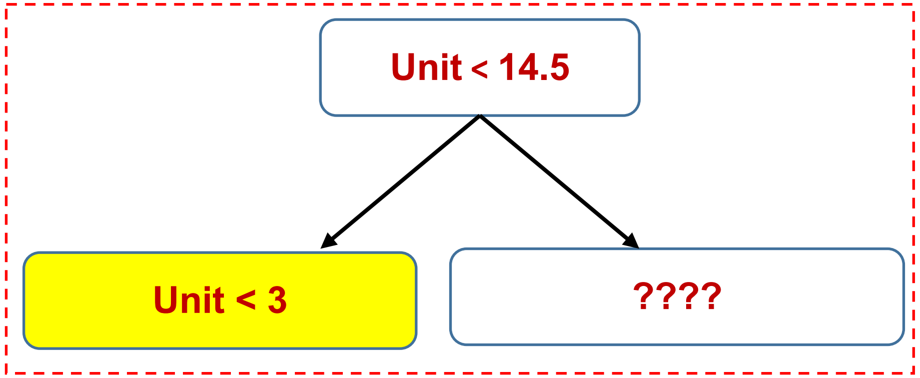


AI VIETNAM
All-in-One Course

Unit is a root node

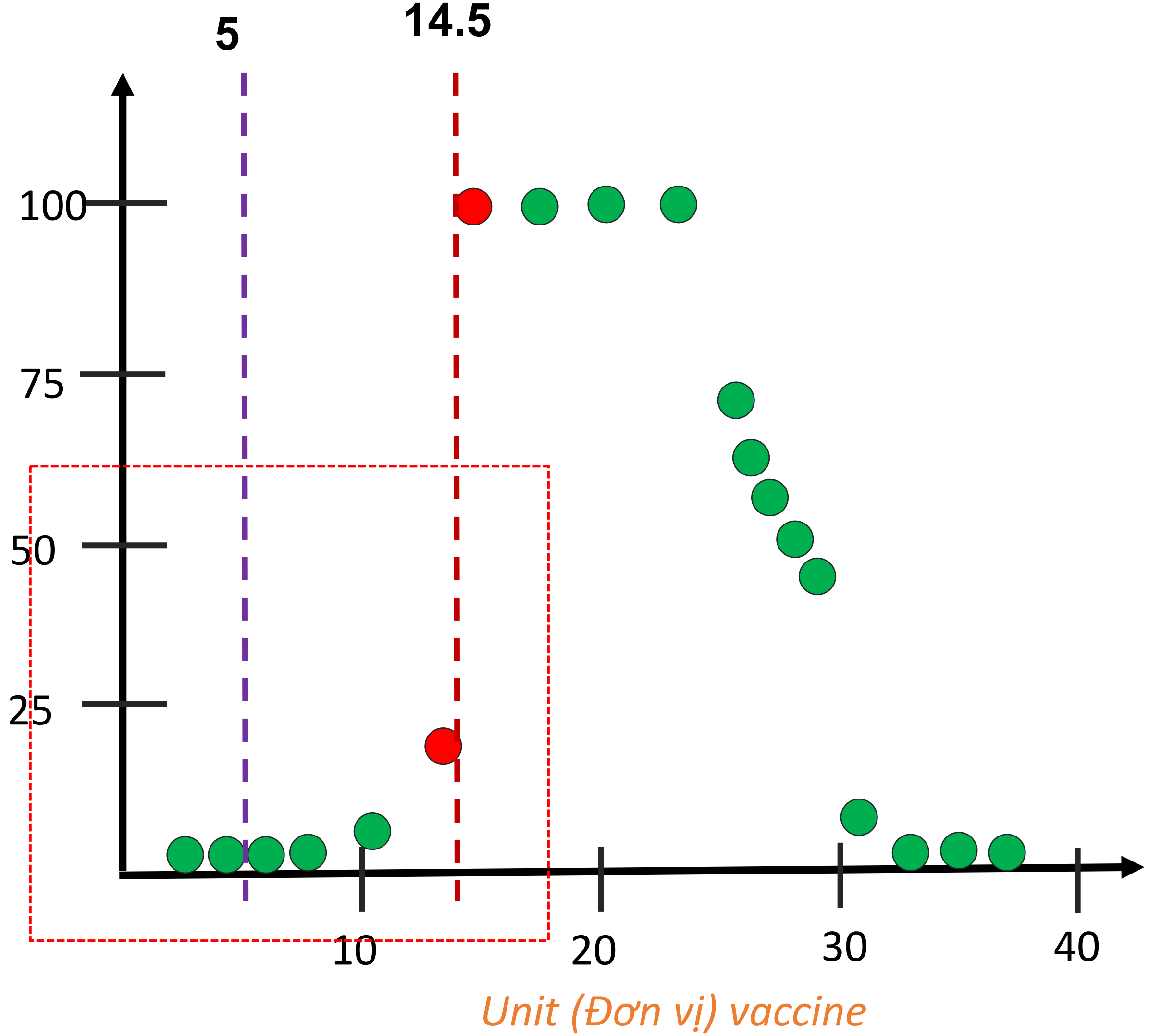


Average in unit( ) = 14.5

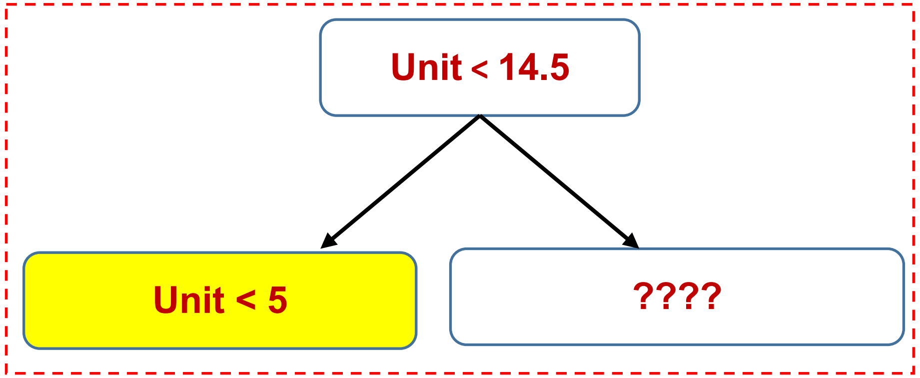


Compute SSR for unit 3

Unit is a root node



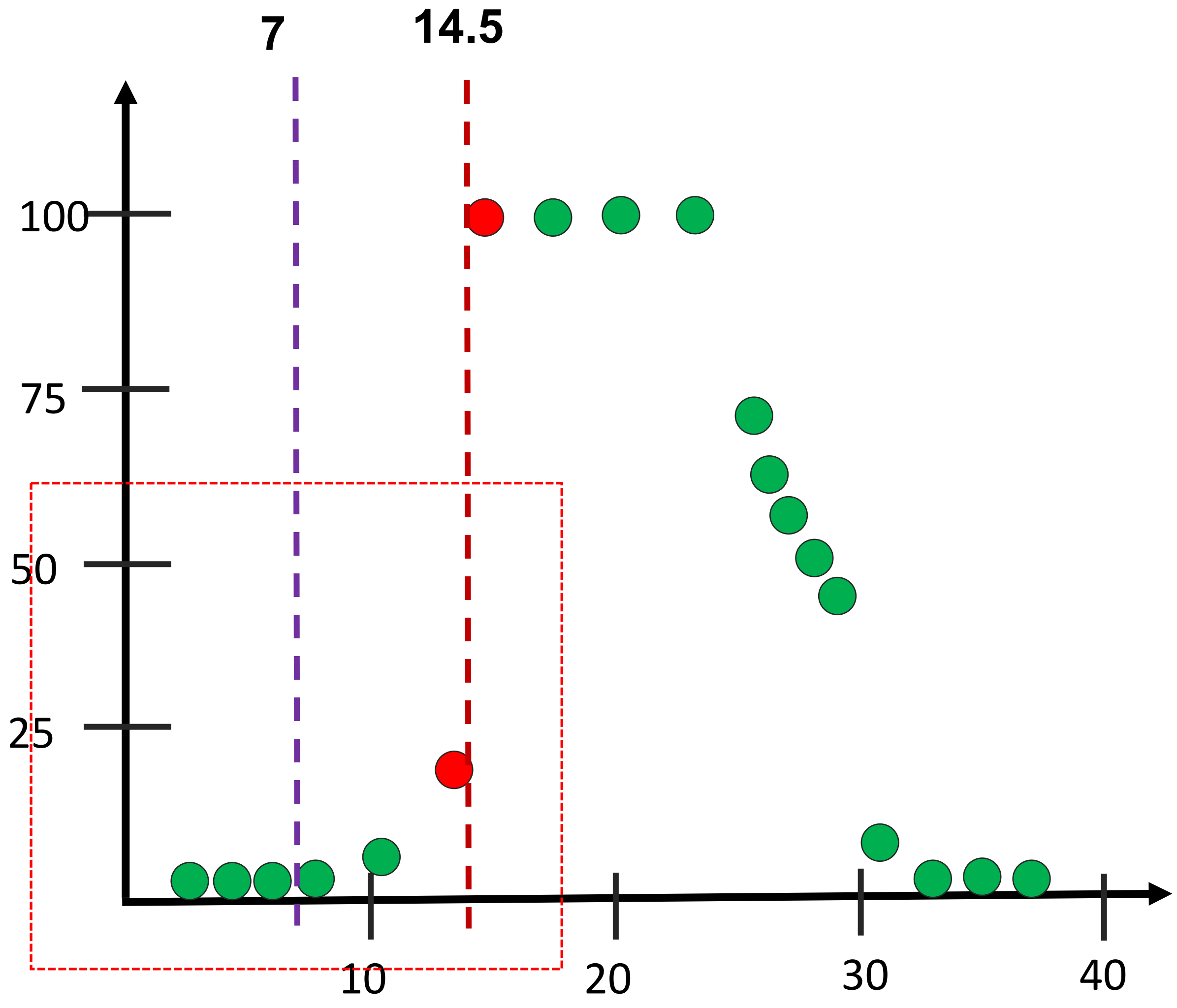
Average in unit( ) = 14.5



Compute SSR for unit 5

AI VIETNAM
All-in-One Course

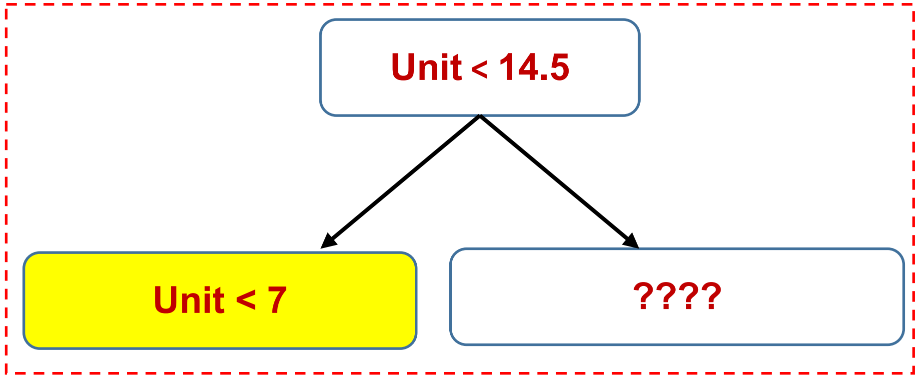
Unit is a root node



Effectiveness (Hiệu quả) (%)

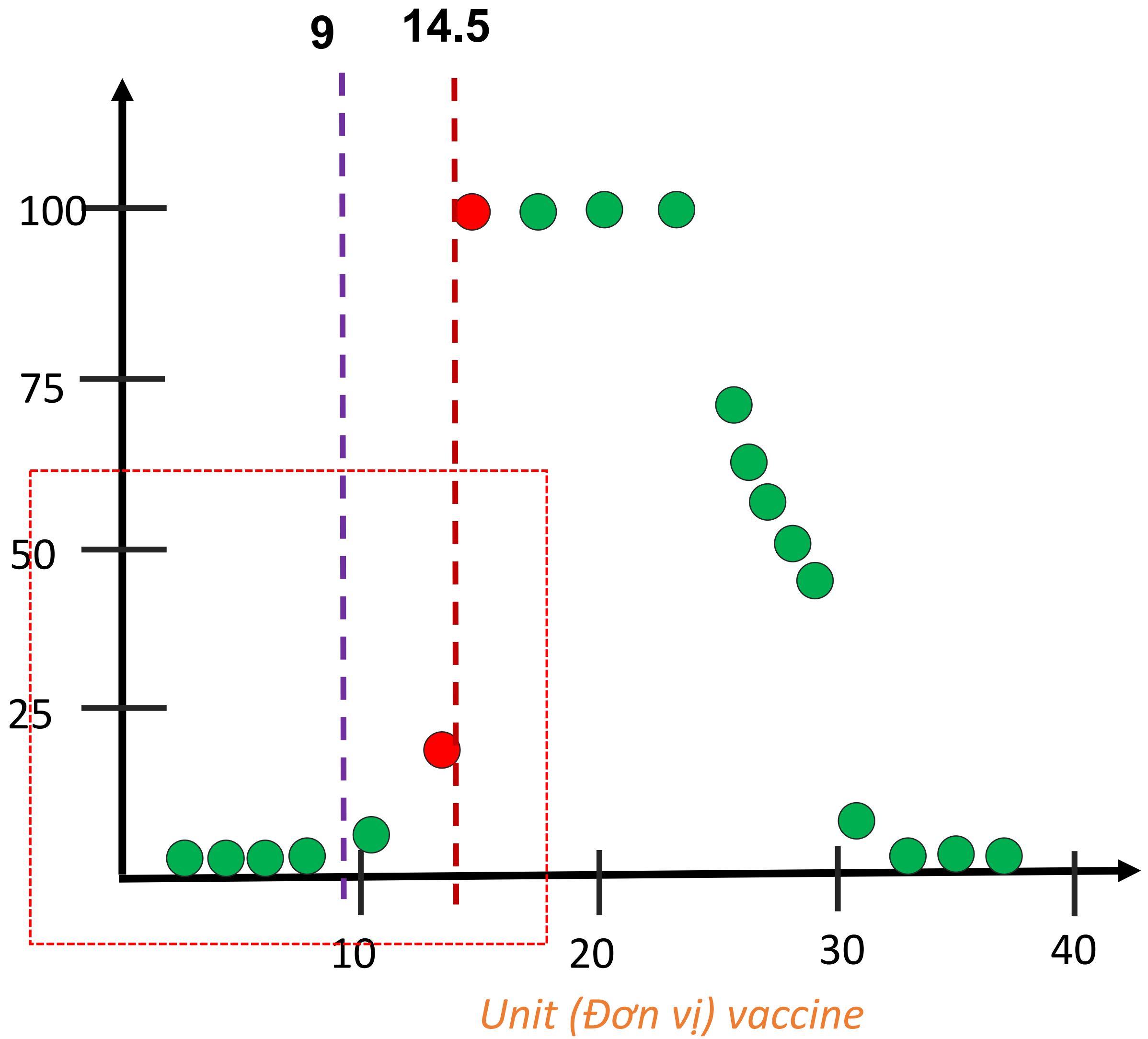
Unit (Đơn vị) vaccine

Average in unit( ) = 14.5

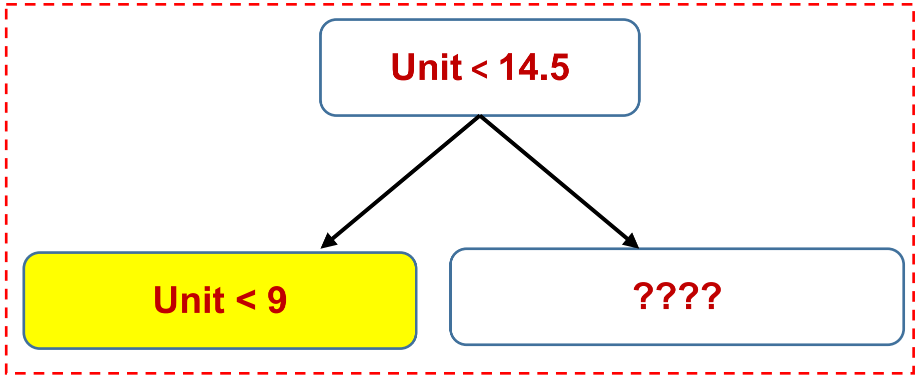


Compute SSR for unit 7

Unit is a root node



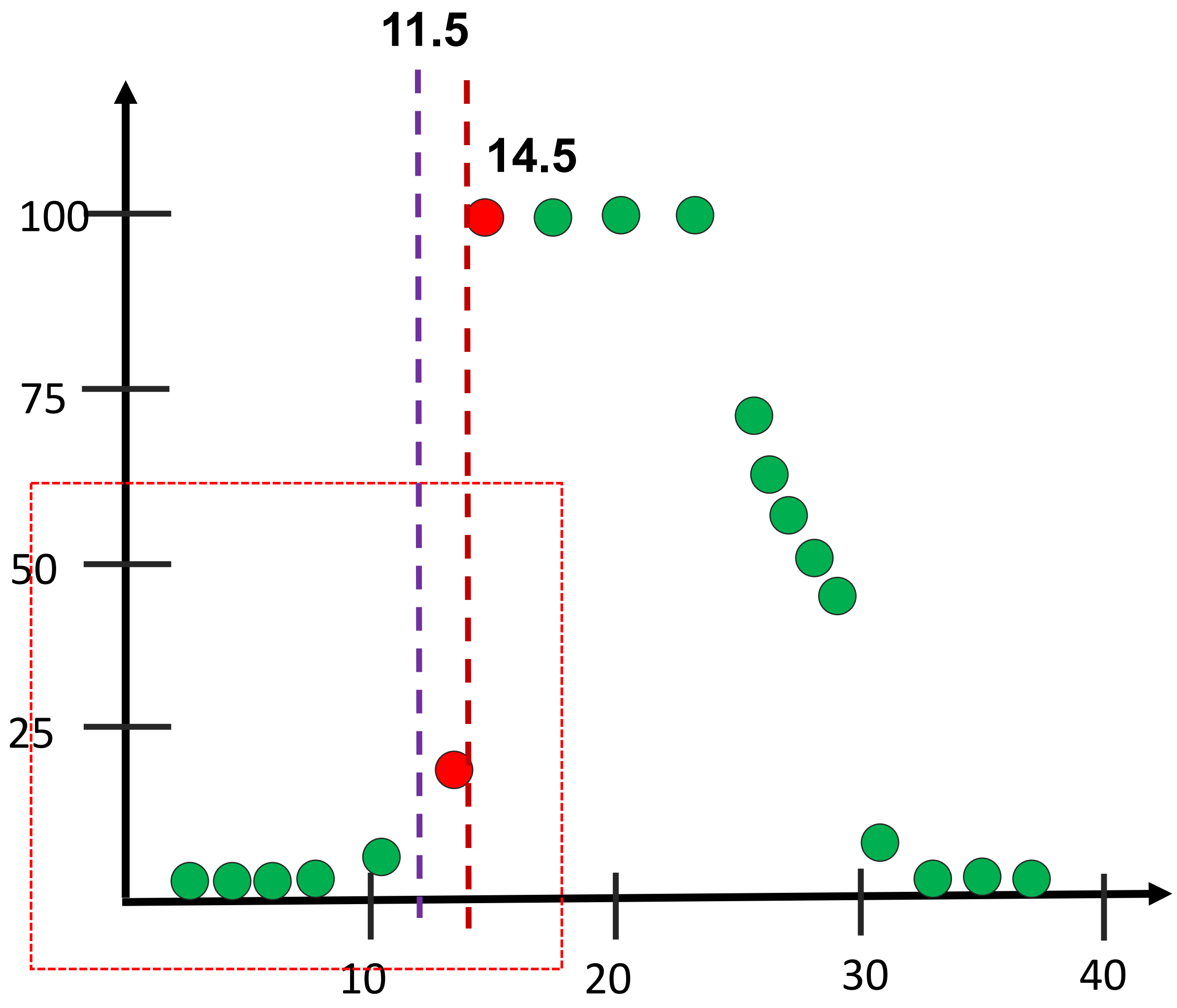
Average in unit( ) = 14.5



Compute SSR for unit 9

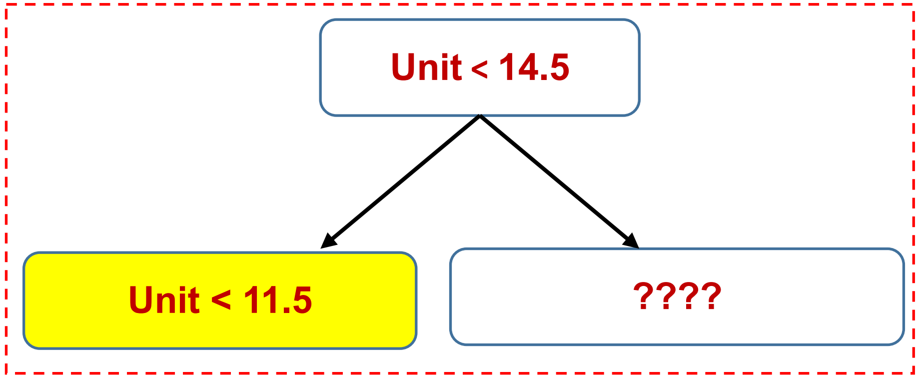
Unit is a root node

Effectiveness (Hiệu quả) (%)

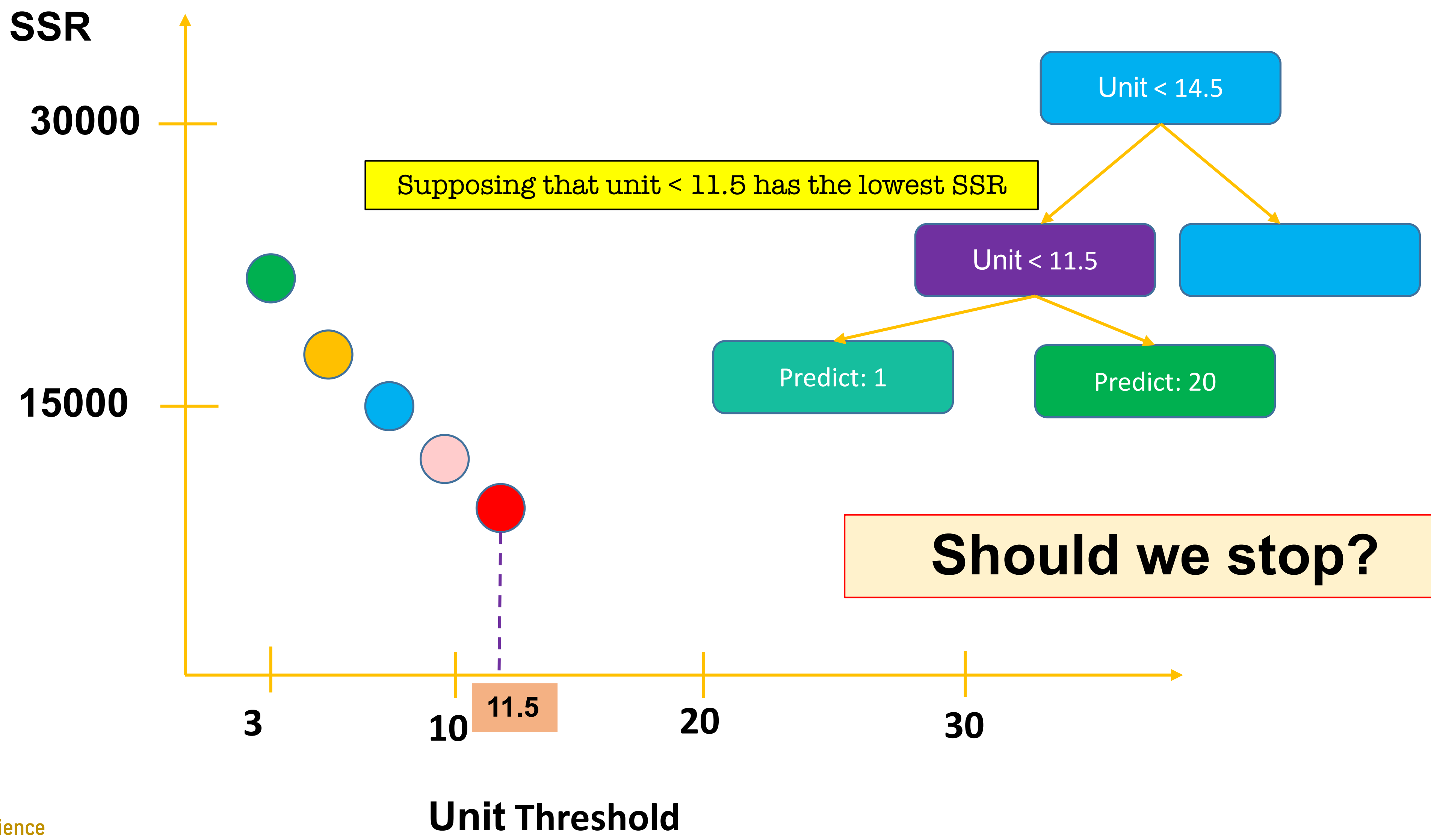


Unit (Đơn vị) vaccine

Average in unit(●●) = 14.5

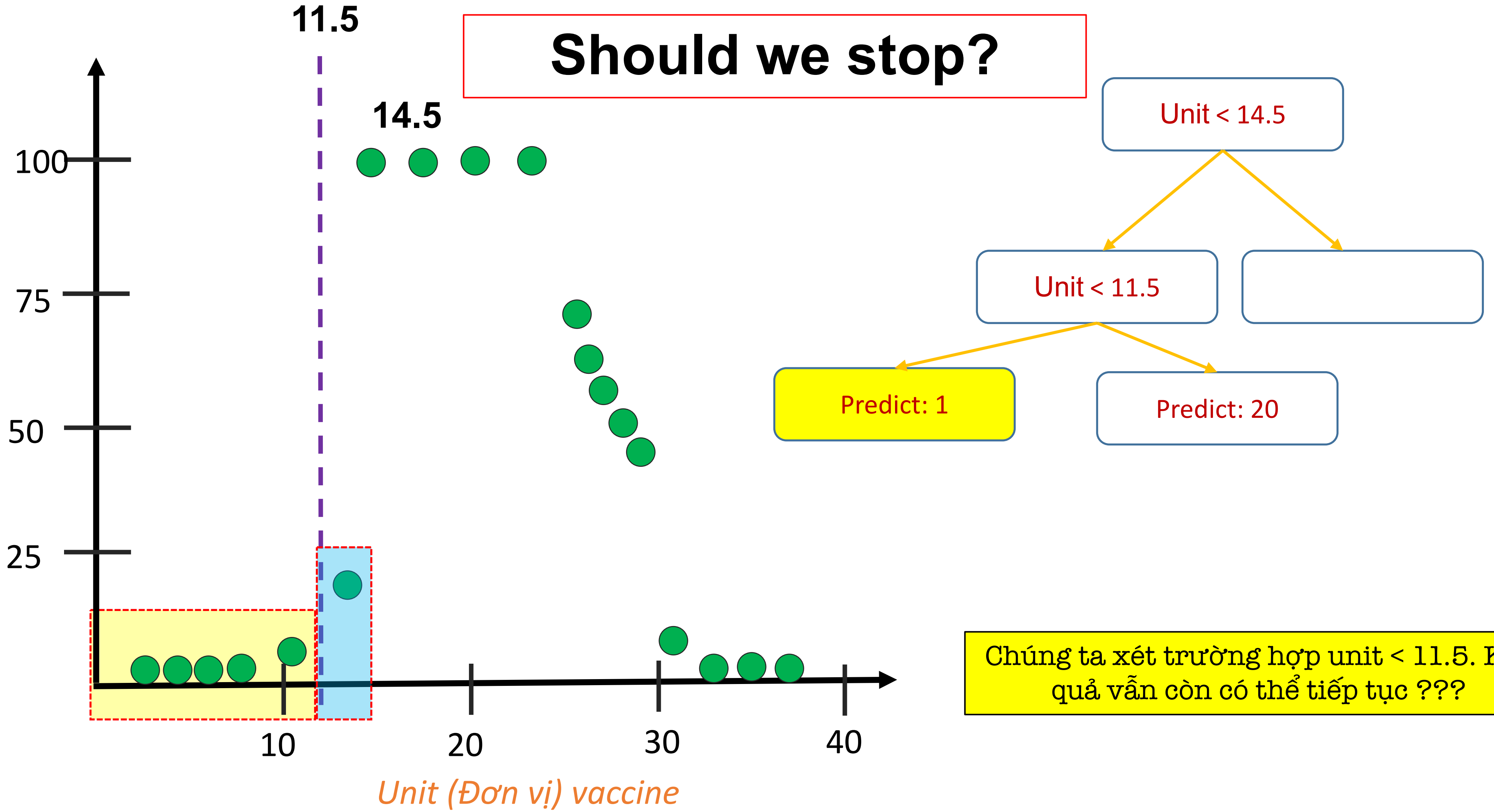


Compute SSR for unit 11.5

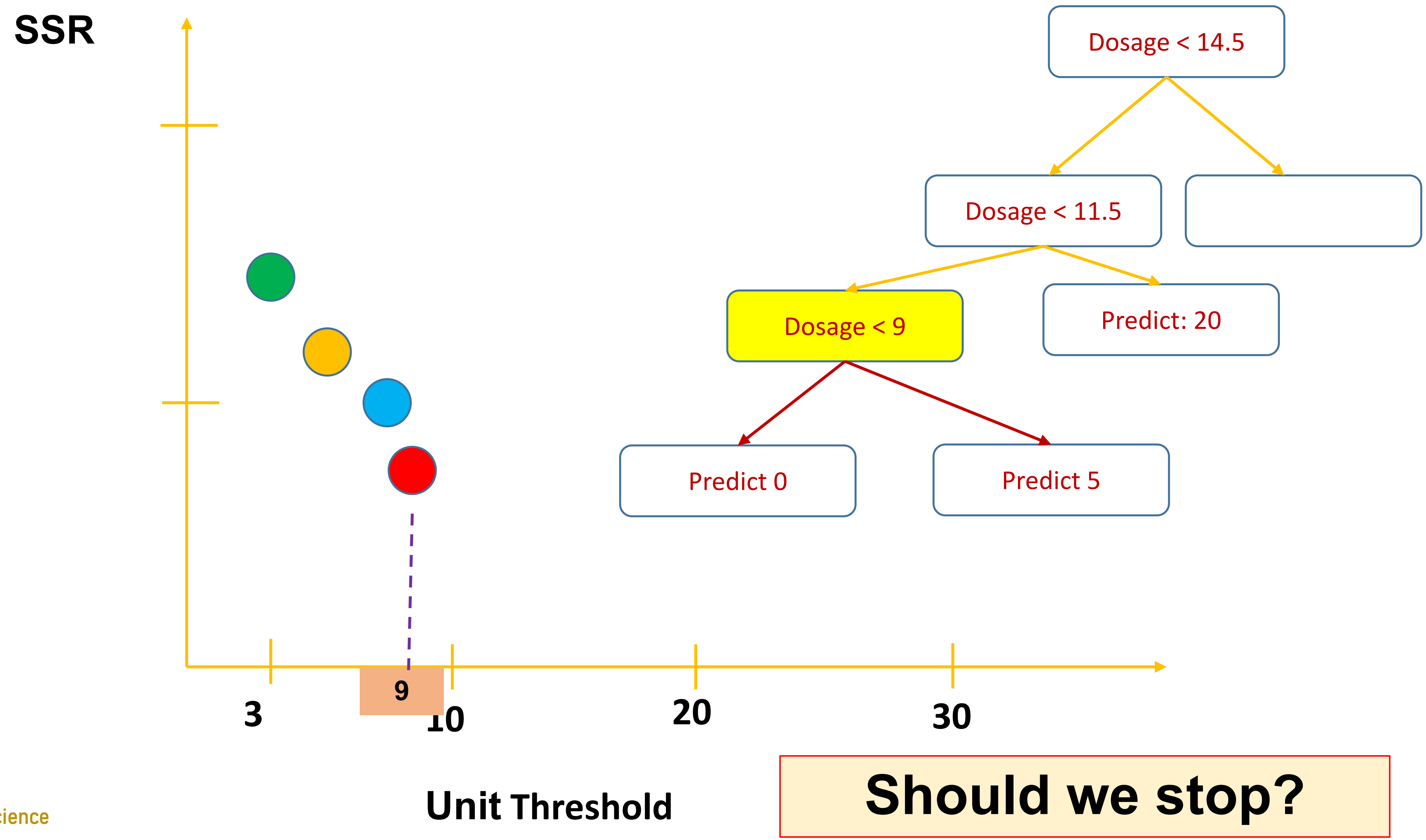


Should we stop?

Unit is a root node



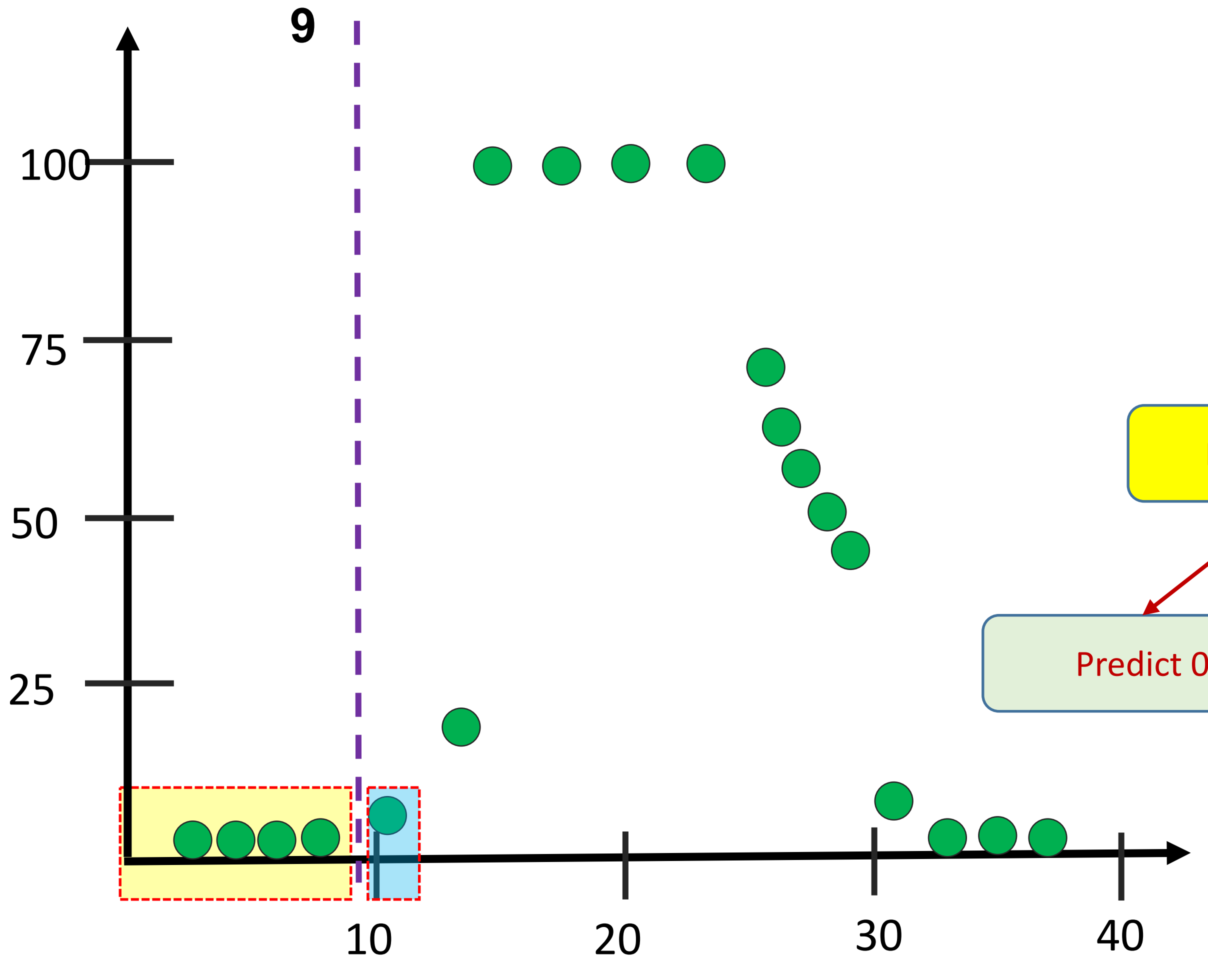
Chúng ta xét trường hợp unit < 11.5. Kết quả vẫn còn có thể tiếp tục ???



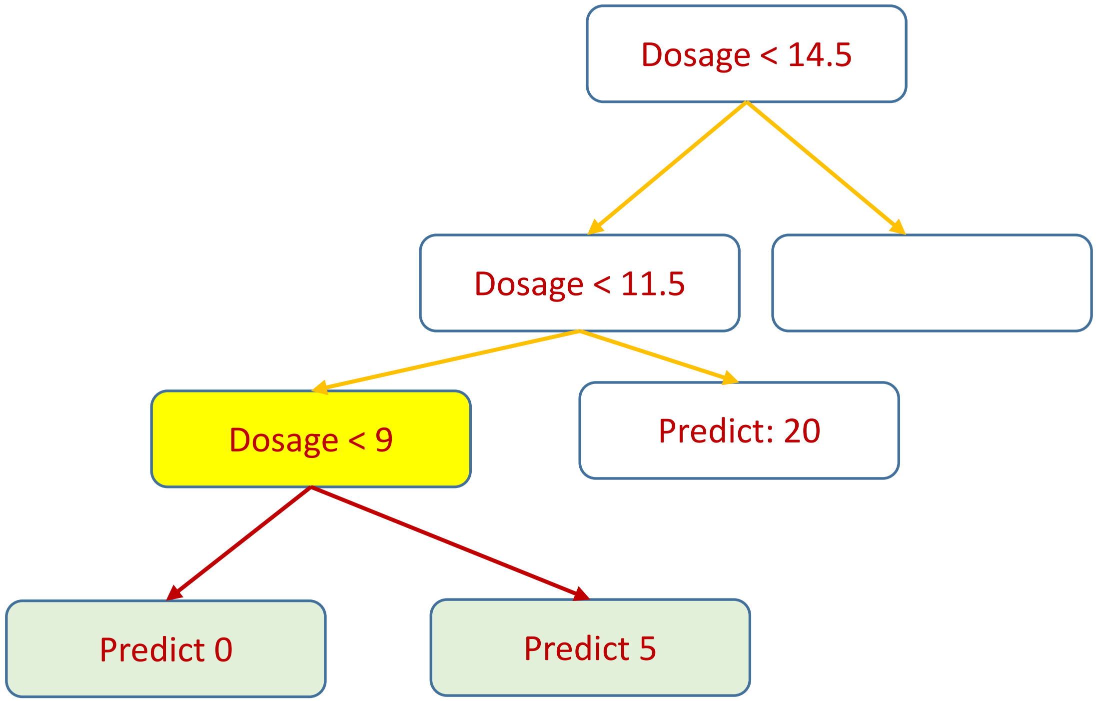
Should we stop?

Unit is a root node

Effectiveness
(Hiệu quả)
(%)

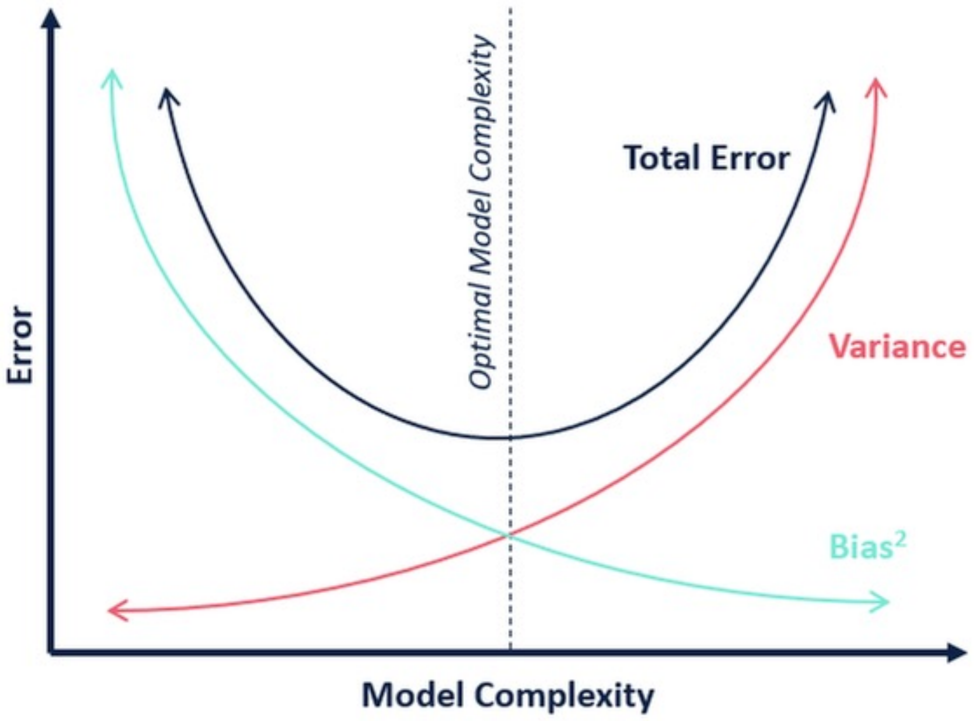


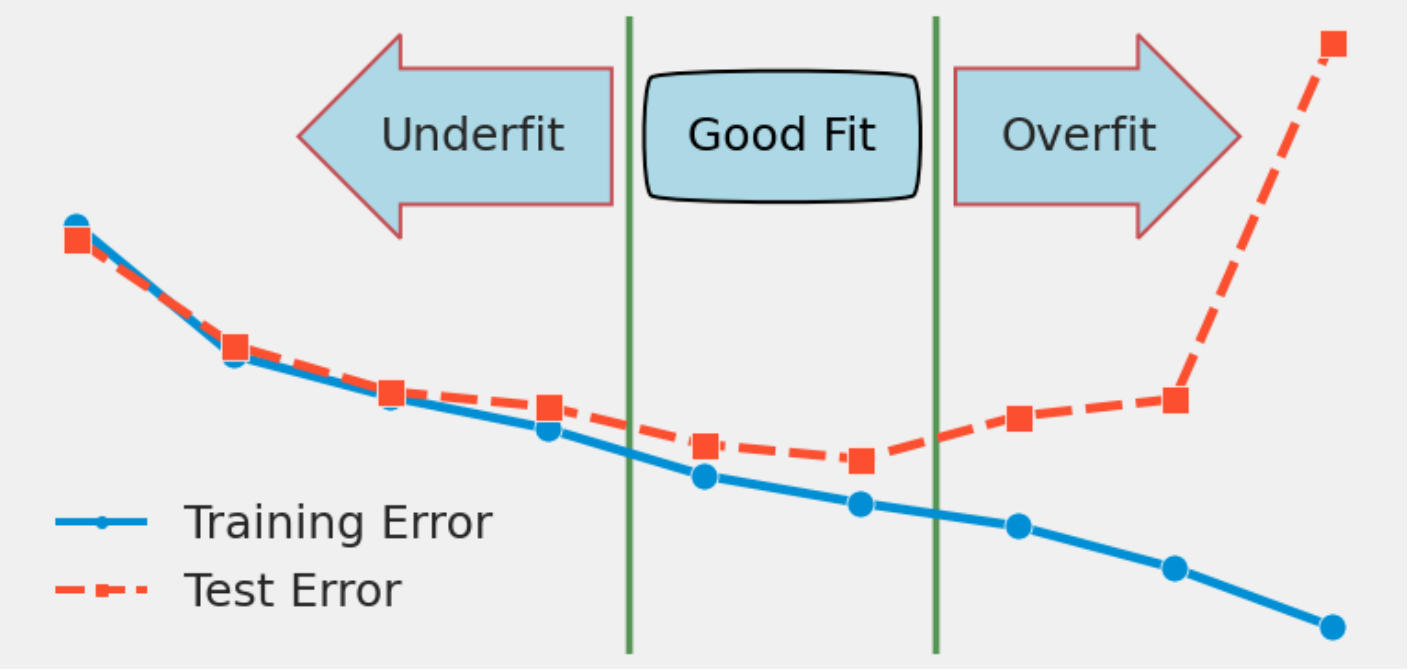
Unit (Đơn vị) vaccine



Should we stop?

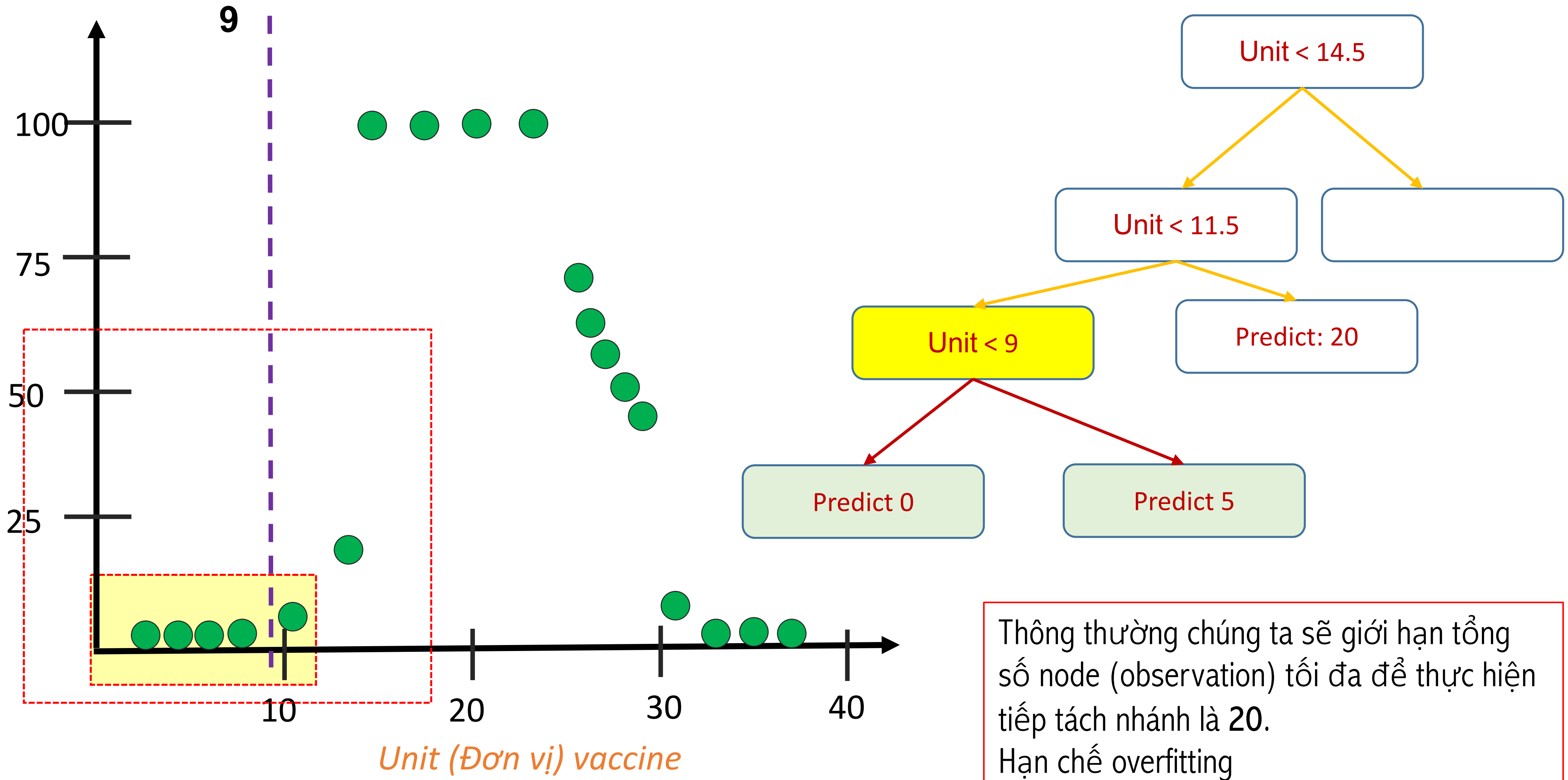
Overfitting Problem





Unit is a root node

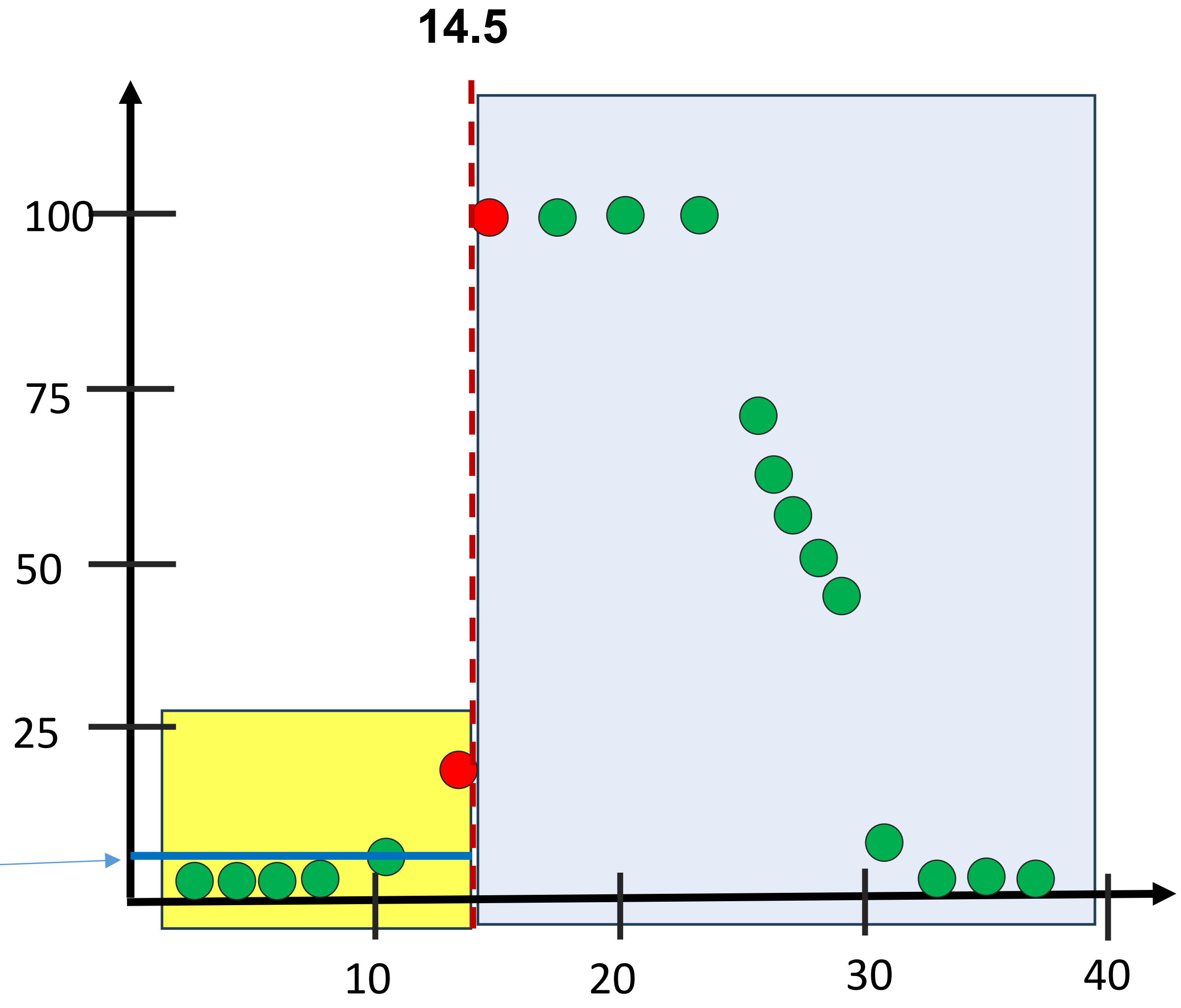
Effectiveness (Hiệu quả) (%)



Thông thường chúng ta sẽ giới hạn tổng số node (observation) tối đa để thực hiện tiếp tách nhánh là **20**.
Hạn chế overfitting

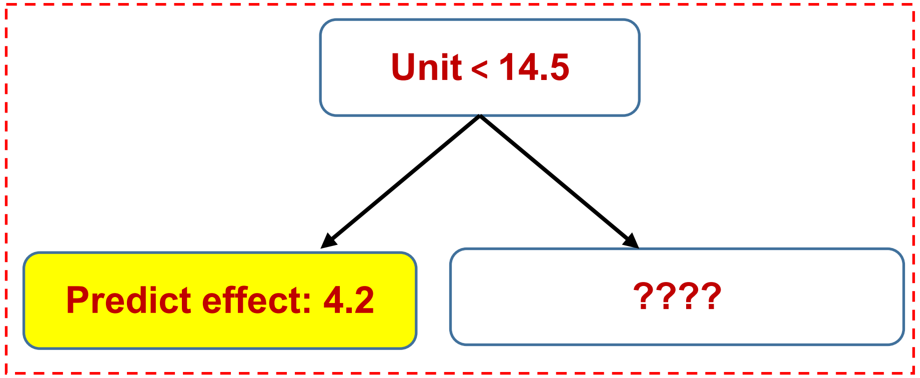
Unit is a root node

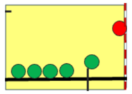
Effectiveness (Hiệu quả) (%)



Unit (Đơn vị) vaccine

Kết quả dự đoán cho unit < 14.5



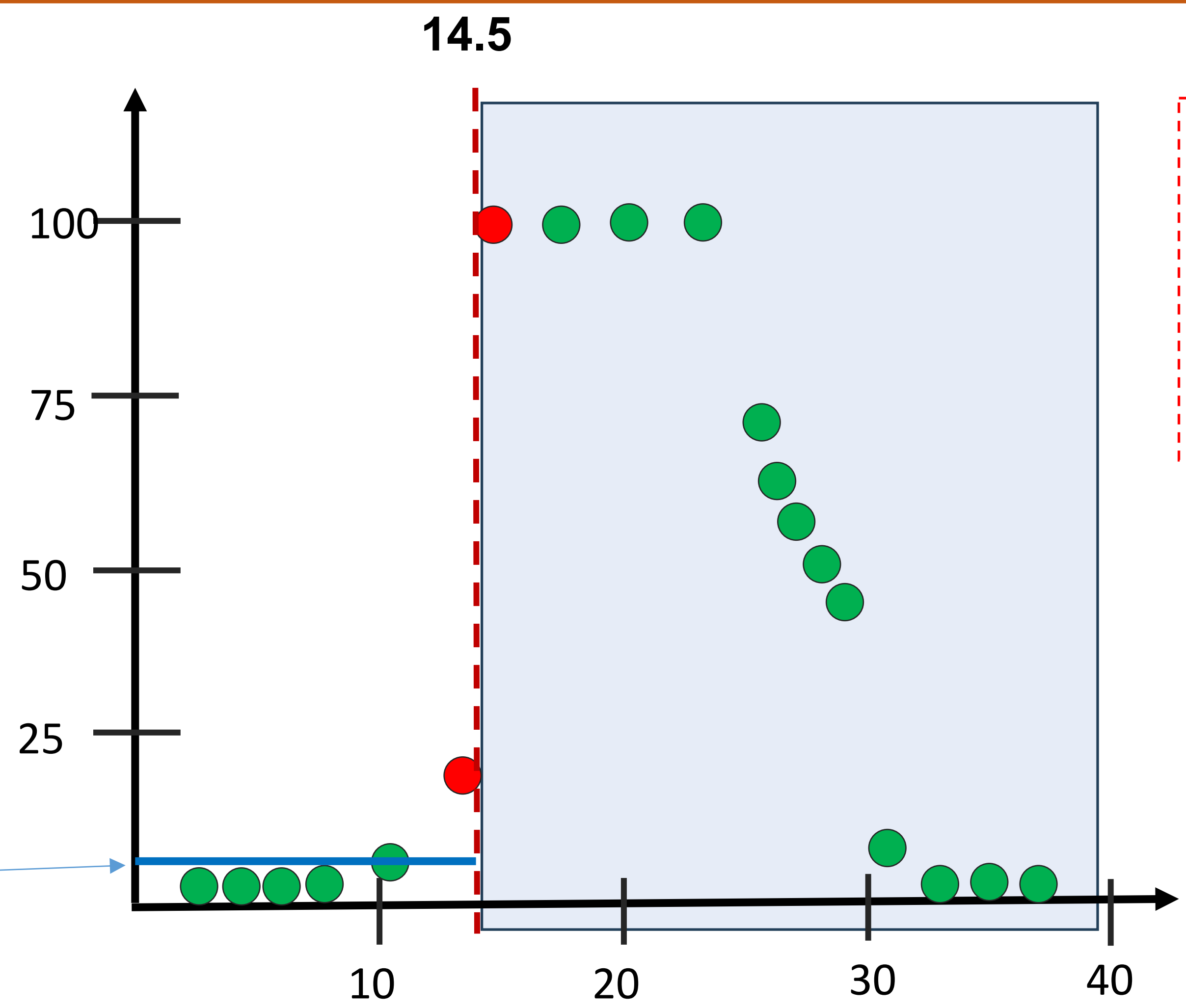
Average in effect.() = 4.2

Vì nhánh unit < 14.5 có tổng số nodes < 7 . Dừng triển khai tách nhánh

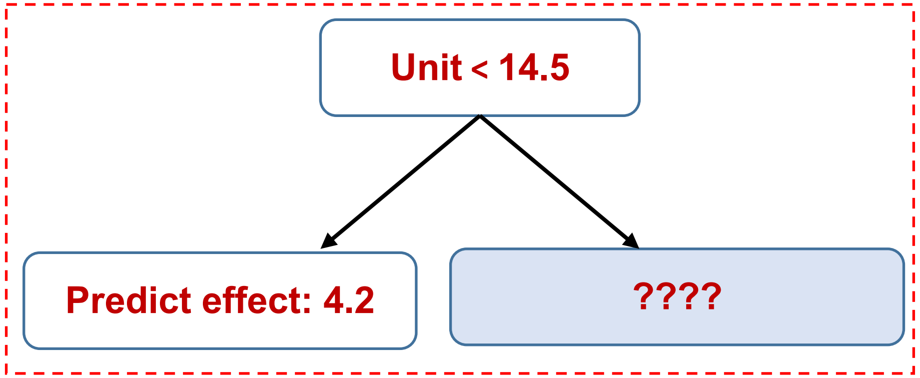
AI VIETNAM
All-in-One Course

Unit is a root node

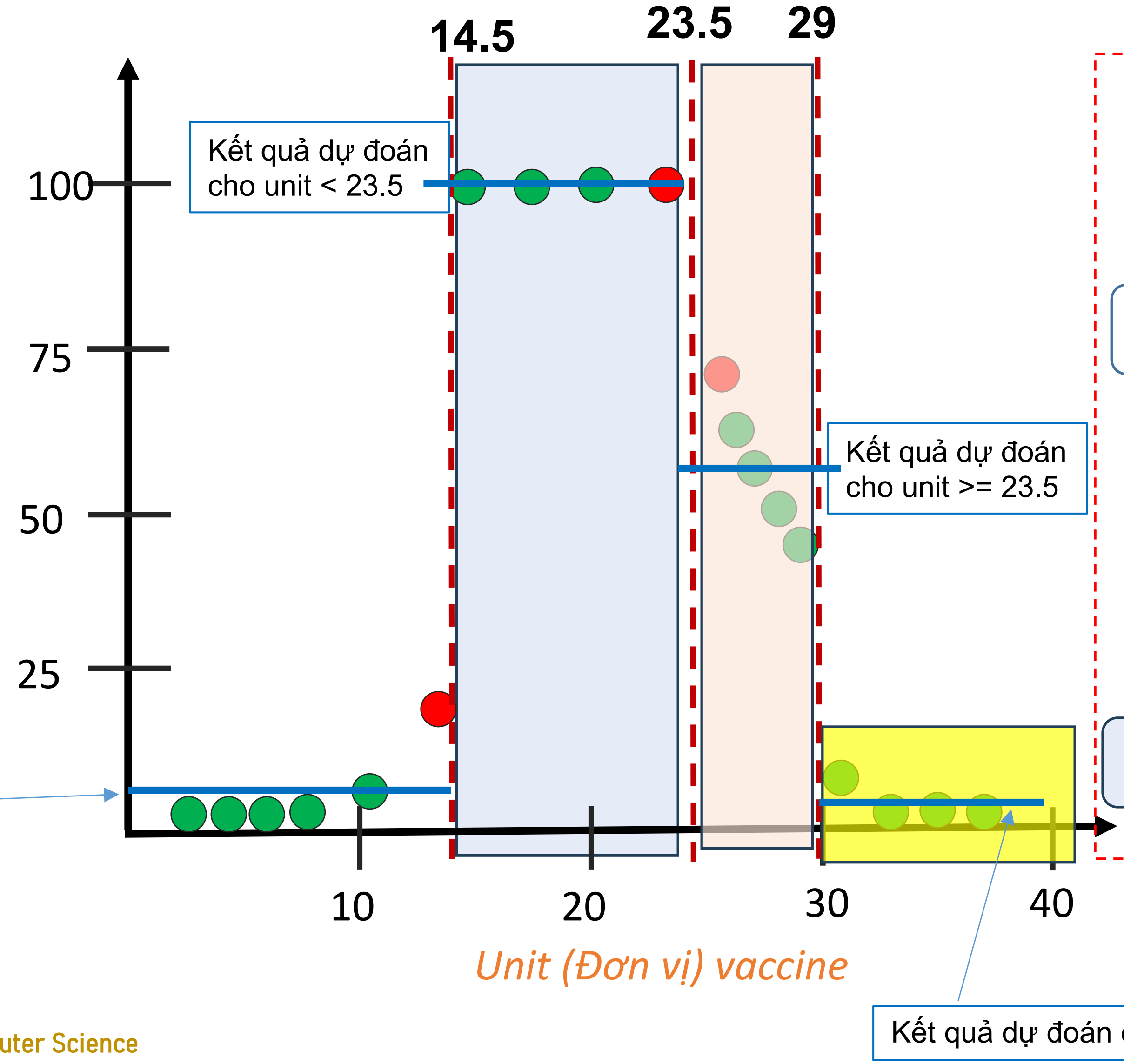
Effectiveness (Hiệu quả) (%)

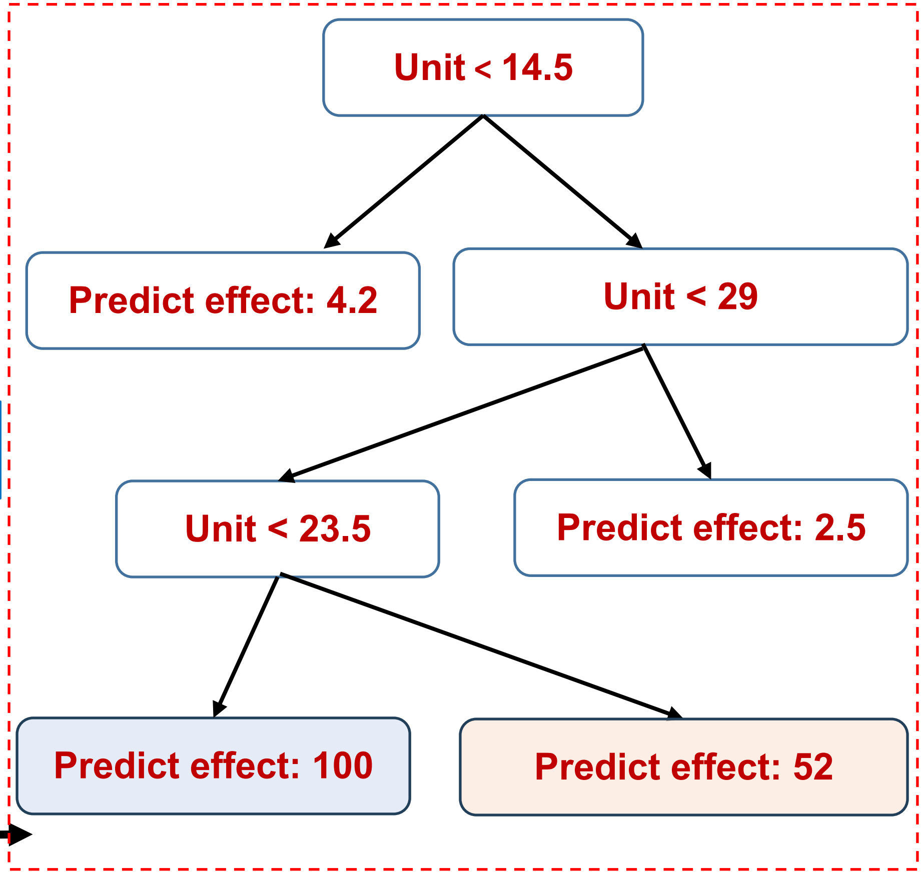


Unit (Đơn vị) vaccine

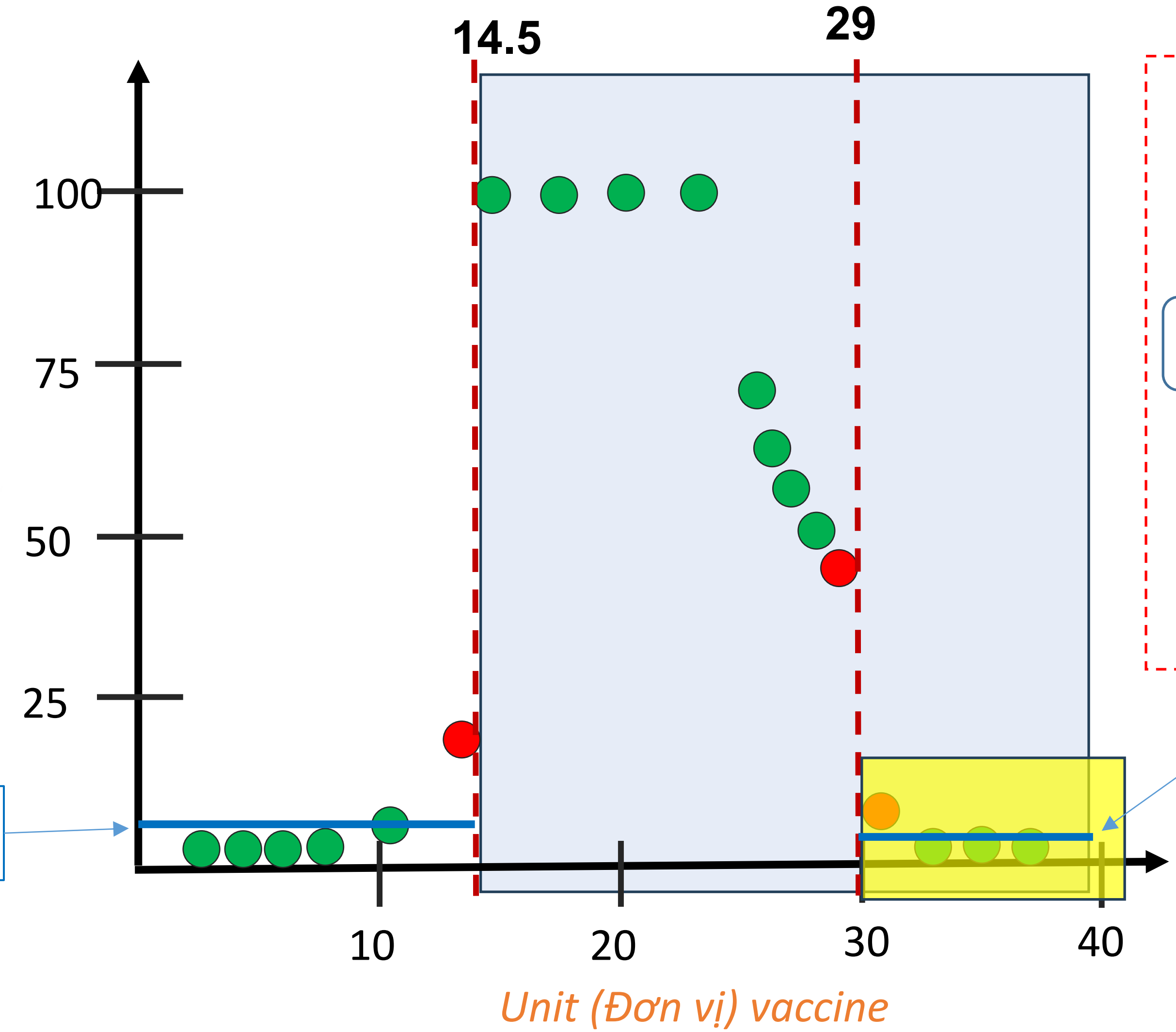


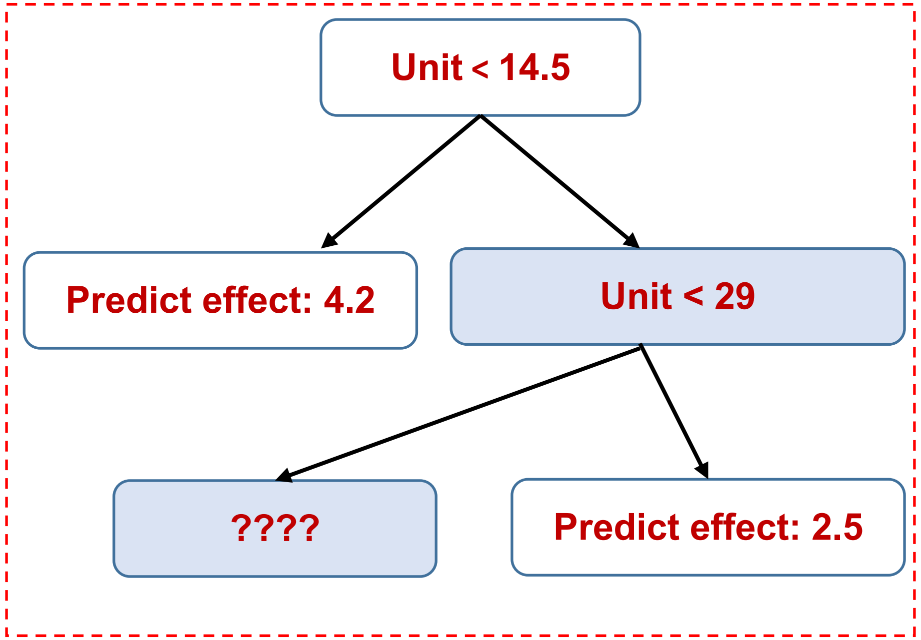
Unit is a root node





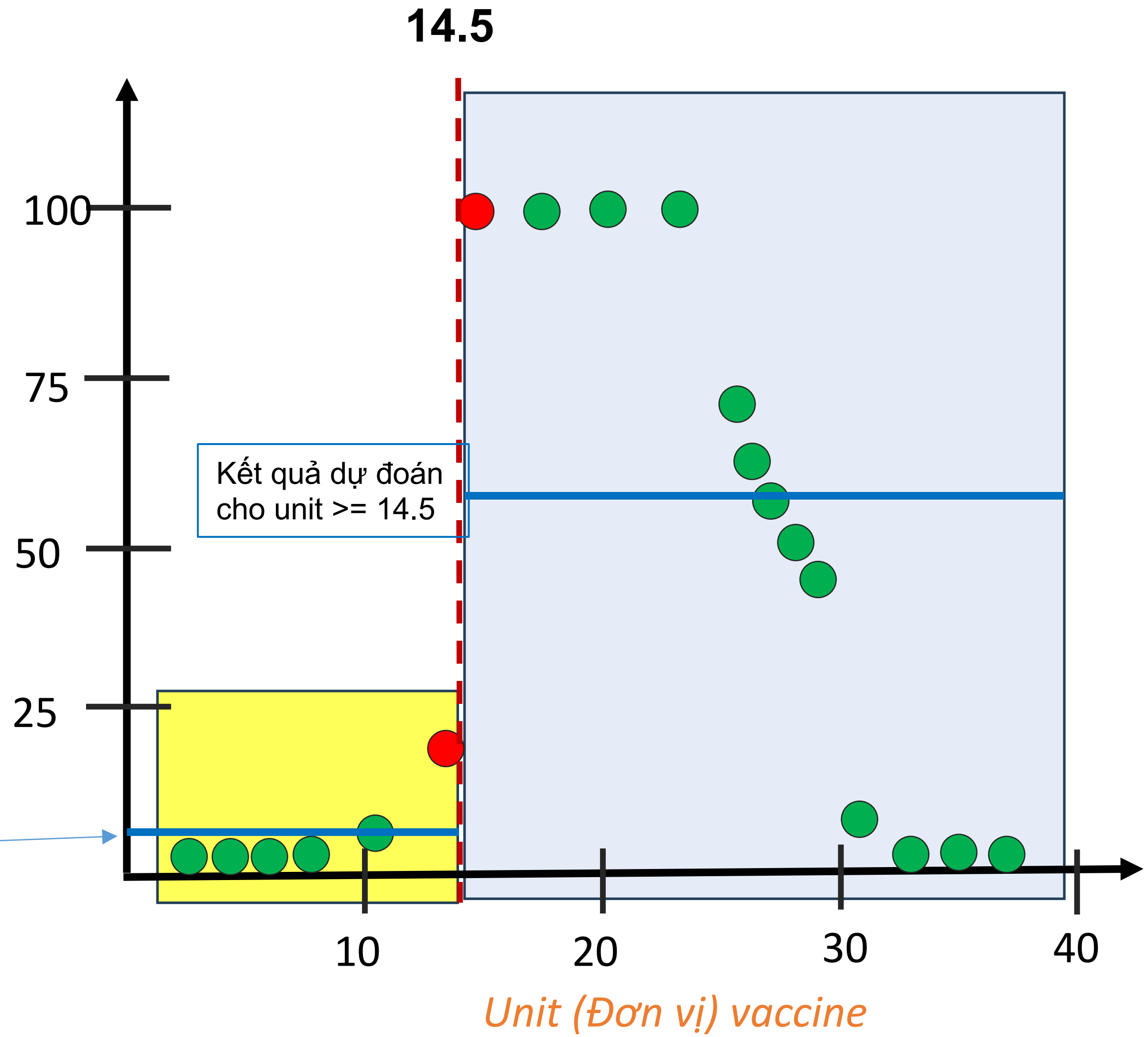
Unit is a root node

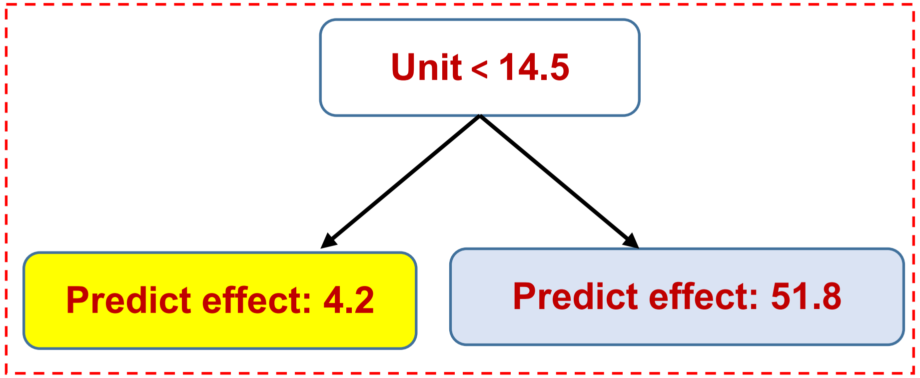




Bây giờ, chúng ta cài đặt mininum nodes cho tách nhánh là 20? What's happen?

Unit is a root node





Compute SSR for this case
SSR ~ 19.000

Case study

Unit	Age	Sex	Effect (%)
10	25	Female	98
20	73	Male	0
35	54	Female	100
5	12	Male	44
…	…	…	…

Tiêm 5 đơn vị vaccine, 12 tuổi, giới tính nam

Hiệu quả vaccine: 44%

Khi có 1 vaccine ra đời, chúng ta muốn dự đoán xem nó hiệu quả bao nhiêu % ứng với liều lượng dùng cố định, tuổi và giới tính của bệnh nhân.

Age node is a root?

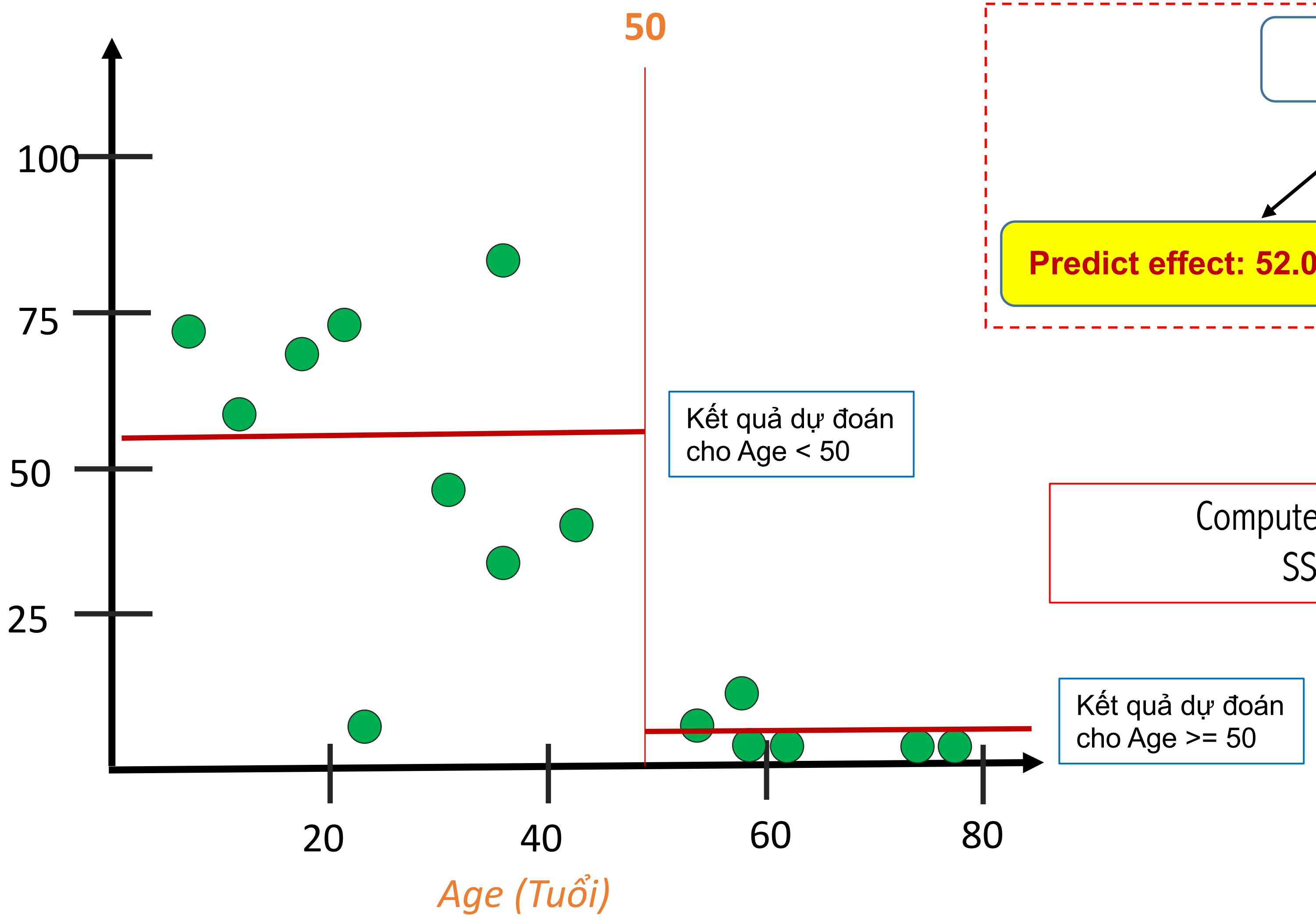
Age	Effect (hiệu quả) (%)
25	98
73	0
54	100
12	44
…	…

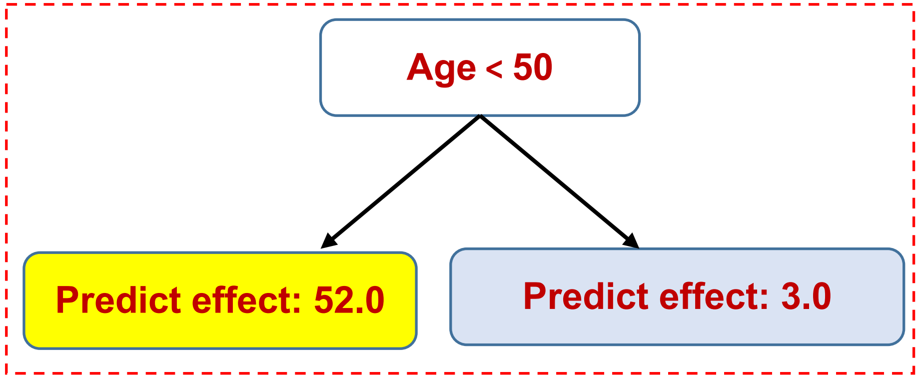
12 tuổi

Hiệu quả vaccine: 44%

Khi có 1 vaccine ra đời, chúng ta muốn dự đoán xem nó hiệu quả bao nhiêu % ứng với tuổi (age) của bệnh nhân.

Age is a root node





Compute SSR for this case
SSR ~ 12,000

Khi có 1 vaccine ra đời, chúng ta muốn dự đoán xem nó hiệu quả bao nhiêu % ứng với từng liều lượng dùng trên bệnh nhân.

Case study

Unit	Age	Sex	Effect (%)
10	25	Female	98
20	73	Male	0
35	54	Female	100
5	12	Male	44
…	…	…	…

Giới tính nam

Hiệu quả vaccine: 44%

Khi có 1 vaccine ra đời, chúng ta muốn dự đoán xem nó hiệu quả bao nhiêu % ứng với liều lượng dùng cố định, tuổi và giới tính của bệnh nhân.

Sex node is a root?

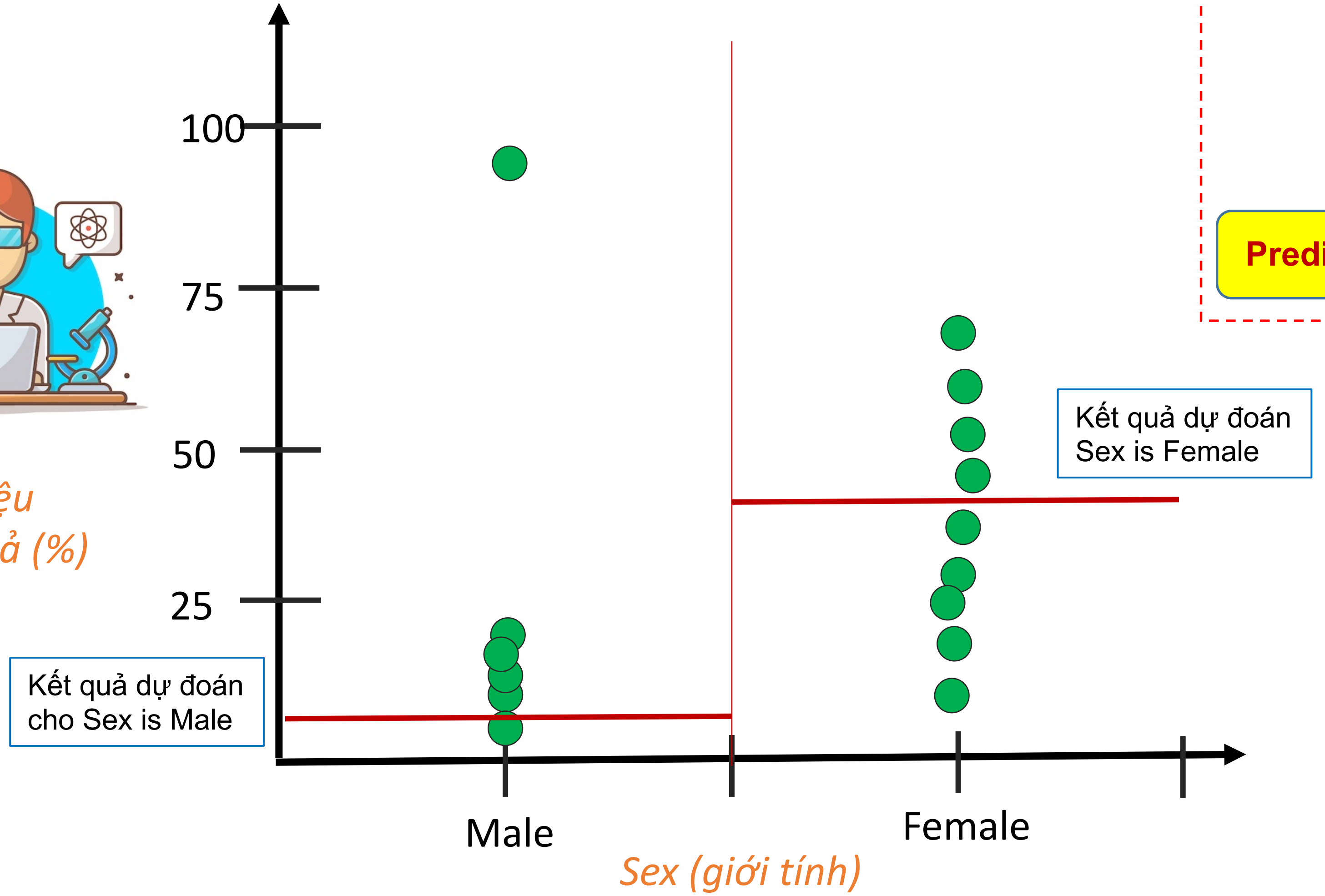
Sex	Effect (hiệu quả) (%)
Female	98
Male	0
Female	100
Male	44
…	…

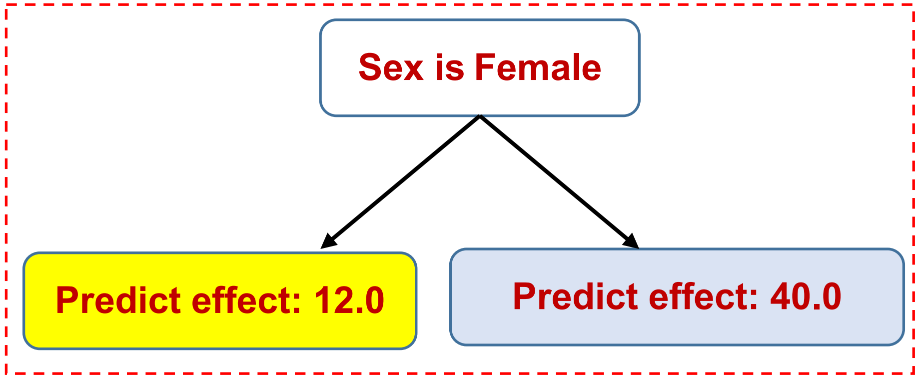
Giới tính Male

Hiệu quả vaccine: 44%

Khi có 1 vaccine ra đời, chúng ta muốn dự đoán xem nó hiệu quả bao nhiêu % ứng với giới tính (sex) của bệnh nhân.

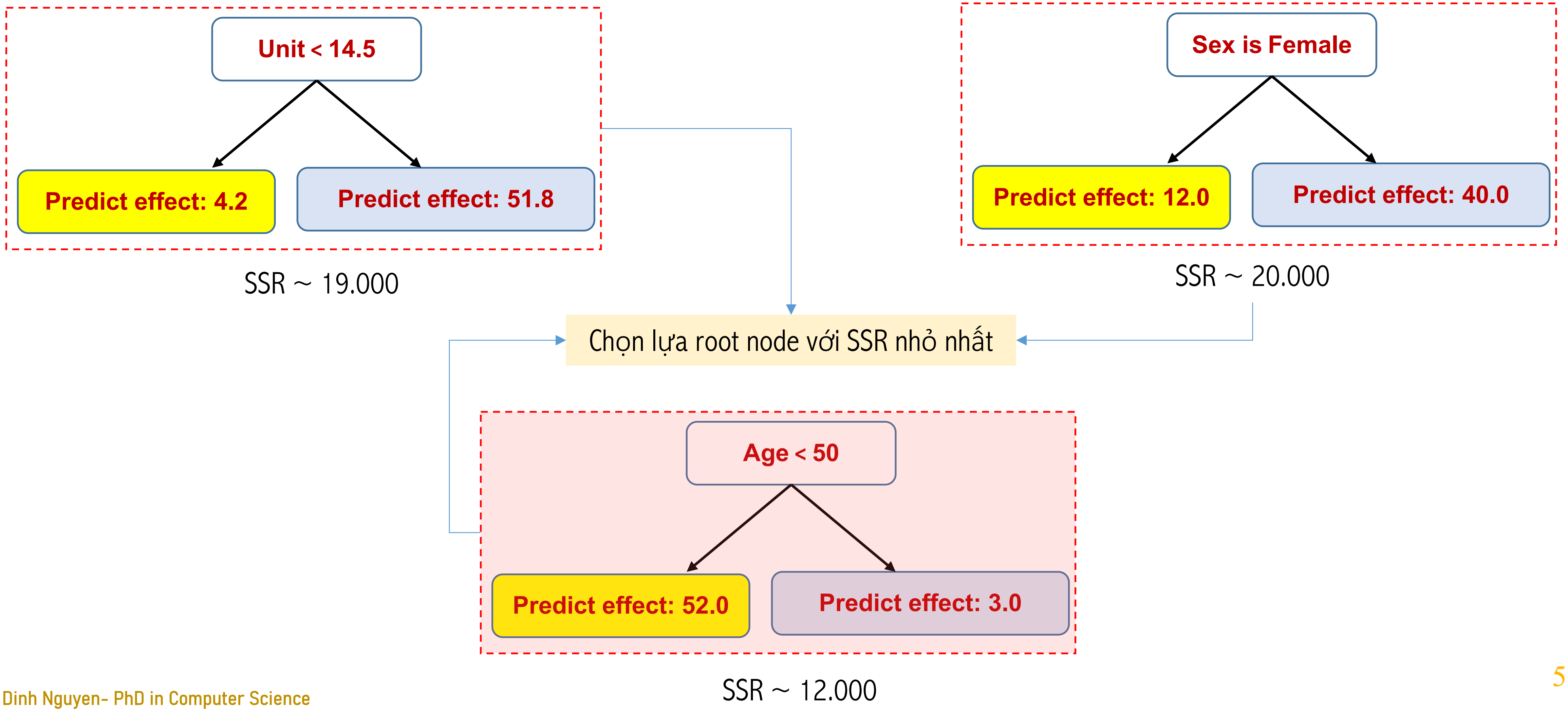
Sex is a root node





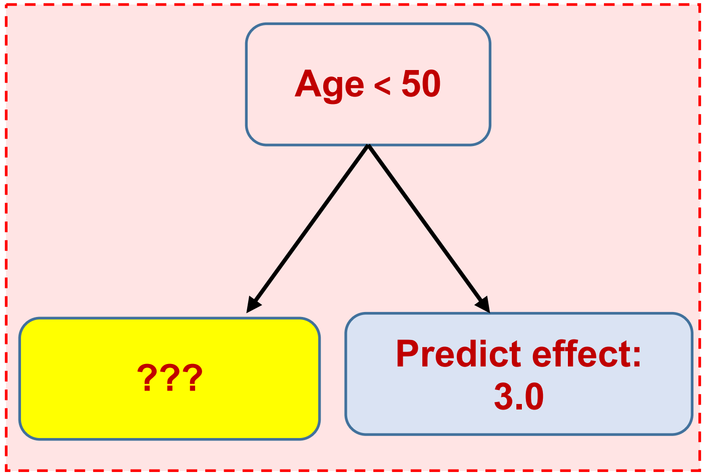
Compute SSR for this case
SSR ~ 20,000

Khi có 1 vaccine ra đời, chúng ta muốn dự đoán xem nó hiệu quả bao nhiêu % ứng với từng liều lượng dùng trên bệnh nhân.


Unit < 14.5
Predict effect: 4.2
Predict effect: 51.8
SSR ~ 19.000
Sex is Female
Predict effect: 12.0
Predict effect: 40.0
SSR ~ 20.000
Chọn lựa root node với SSR nhỏ nhất
Age < 50
Predict effect: 52.0
Predict effect: 3.0
SSR ~ 12.000

Case study

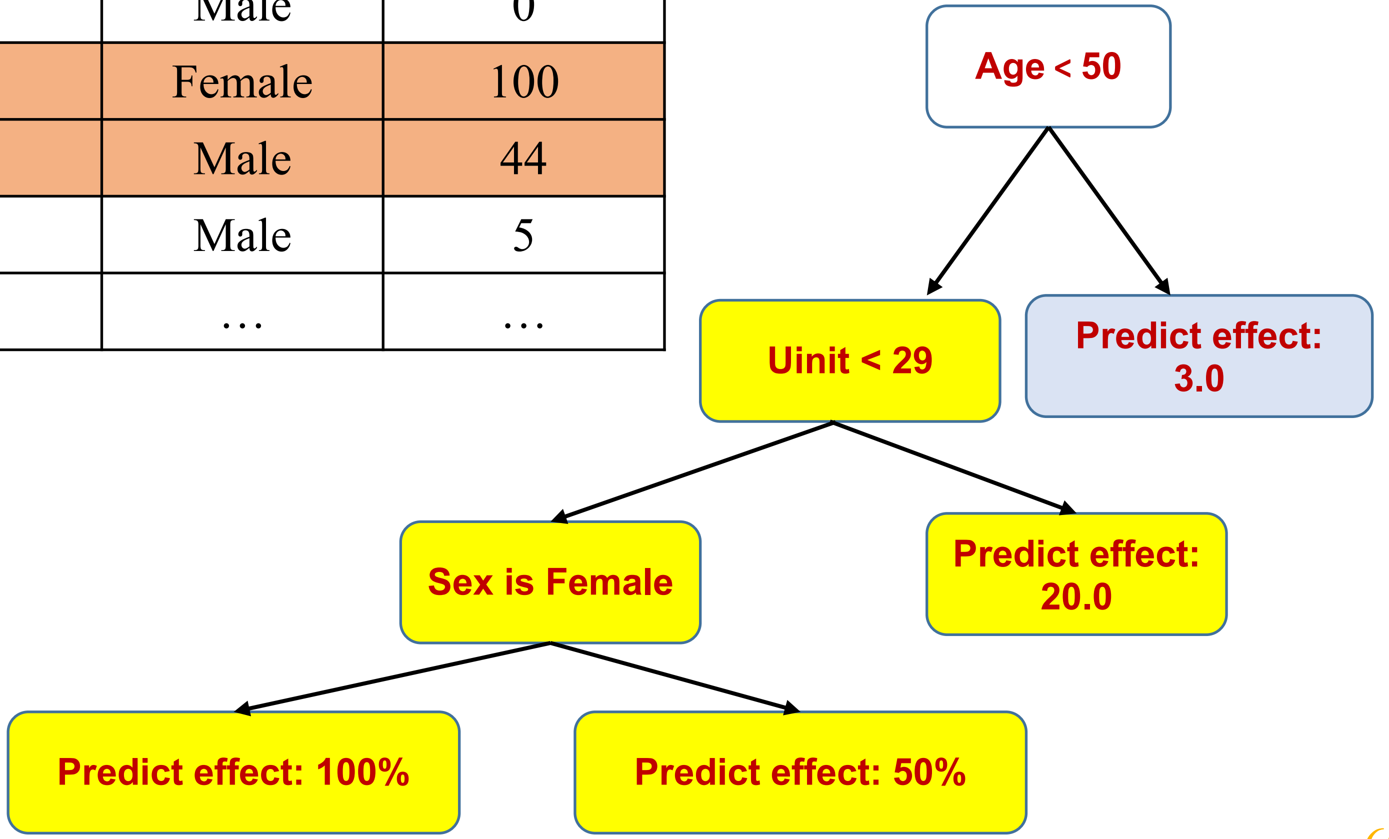
Unit	Age	Sex	Effect (%)
10	25	Female	98
20	73	Male	0
35	54	Female	100
5	12	Male	44
7	80	Male	5
…	…	…	…



Tiếp tục mở rộng cho trường hợp Age < 50
Unit hoặc **Sex** là node kế tiếp???

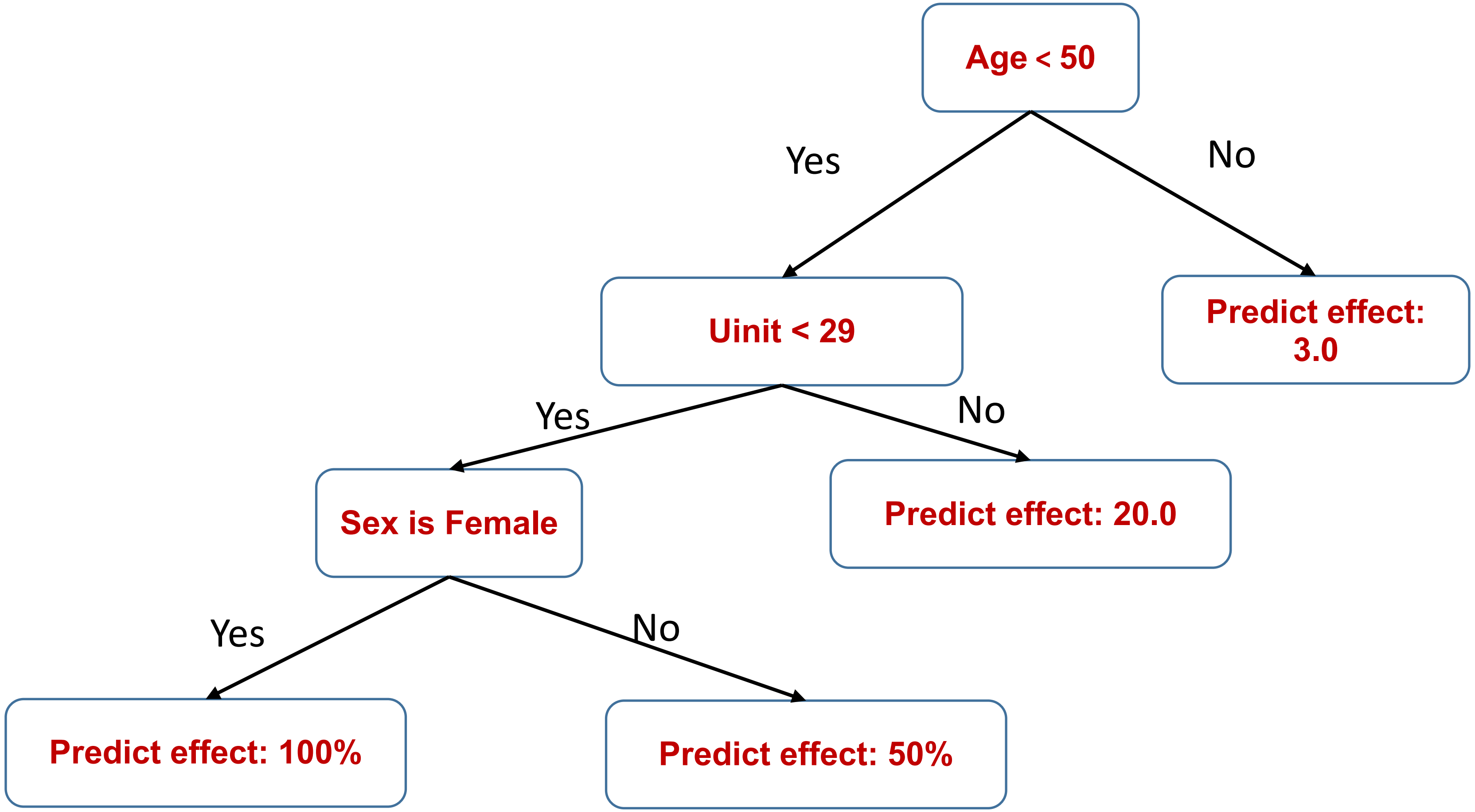
Case study

Unit	Age	Sex	Effect (%)
10	25	Female	98
20	73	Male	0
35	54	Female	100
5	12	Male	44
7	80	Male	5
…	…	…	…



AI VIETNAM
All-in-One Course

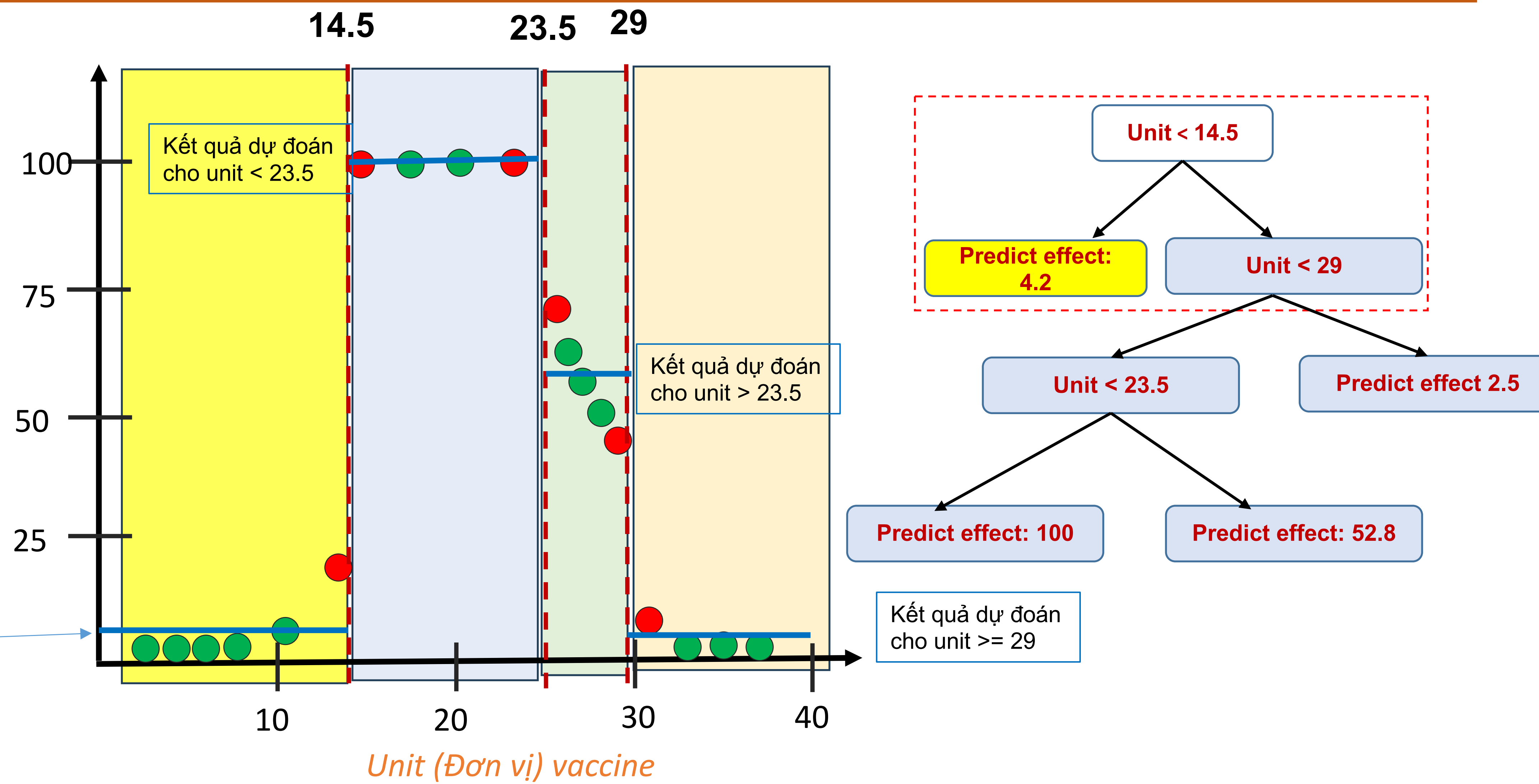
Final Tree



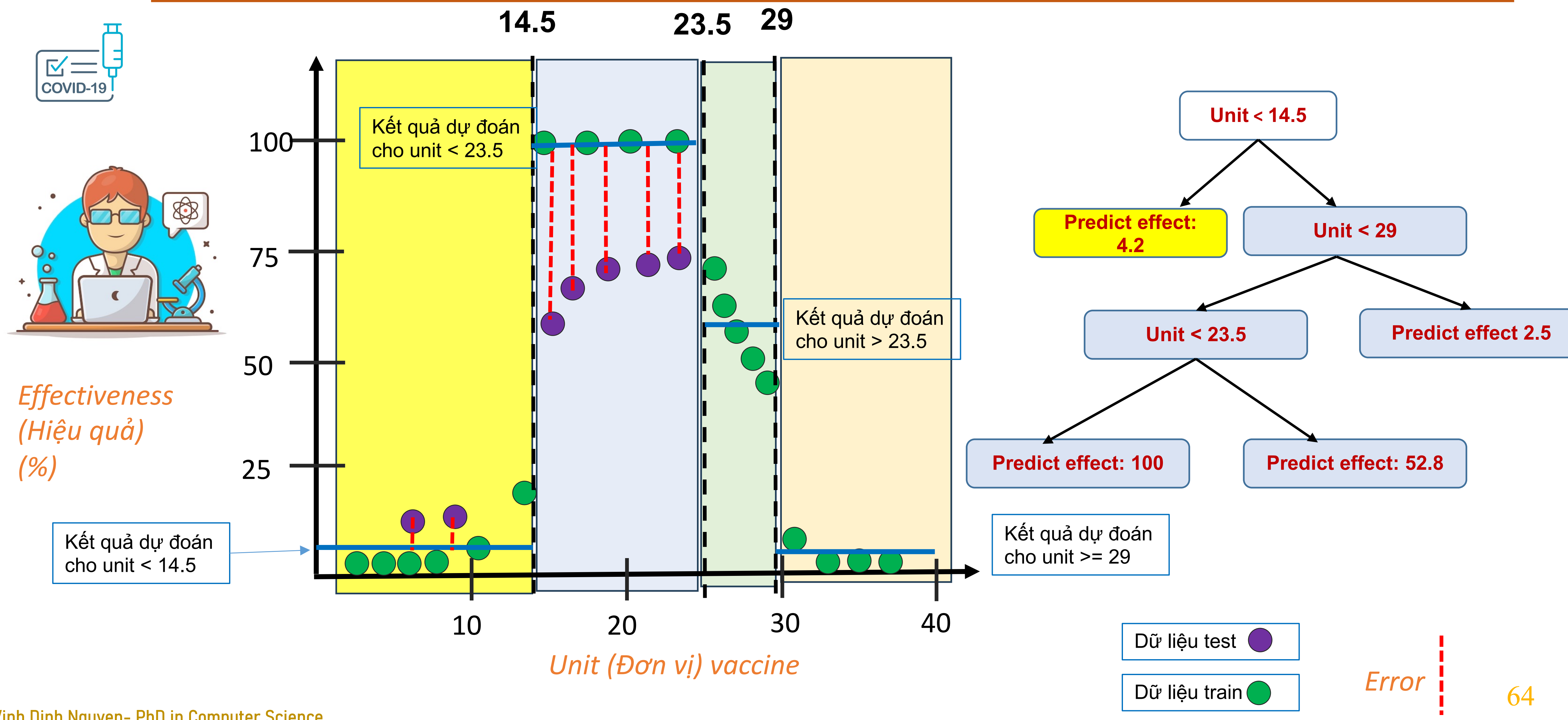
Outline

- Motivation for Regression Tree
- Regression Tree
- **Overfitting in Regression Tree**
- Case study

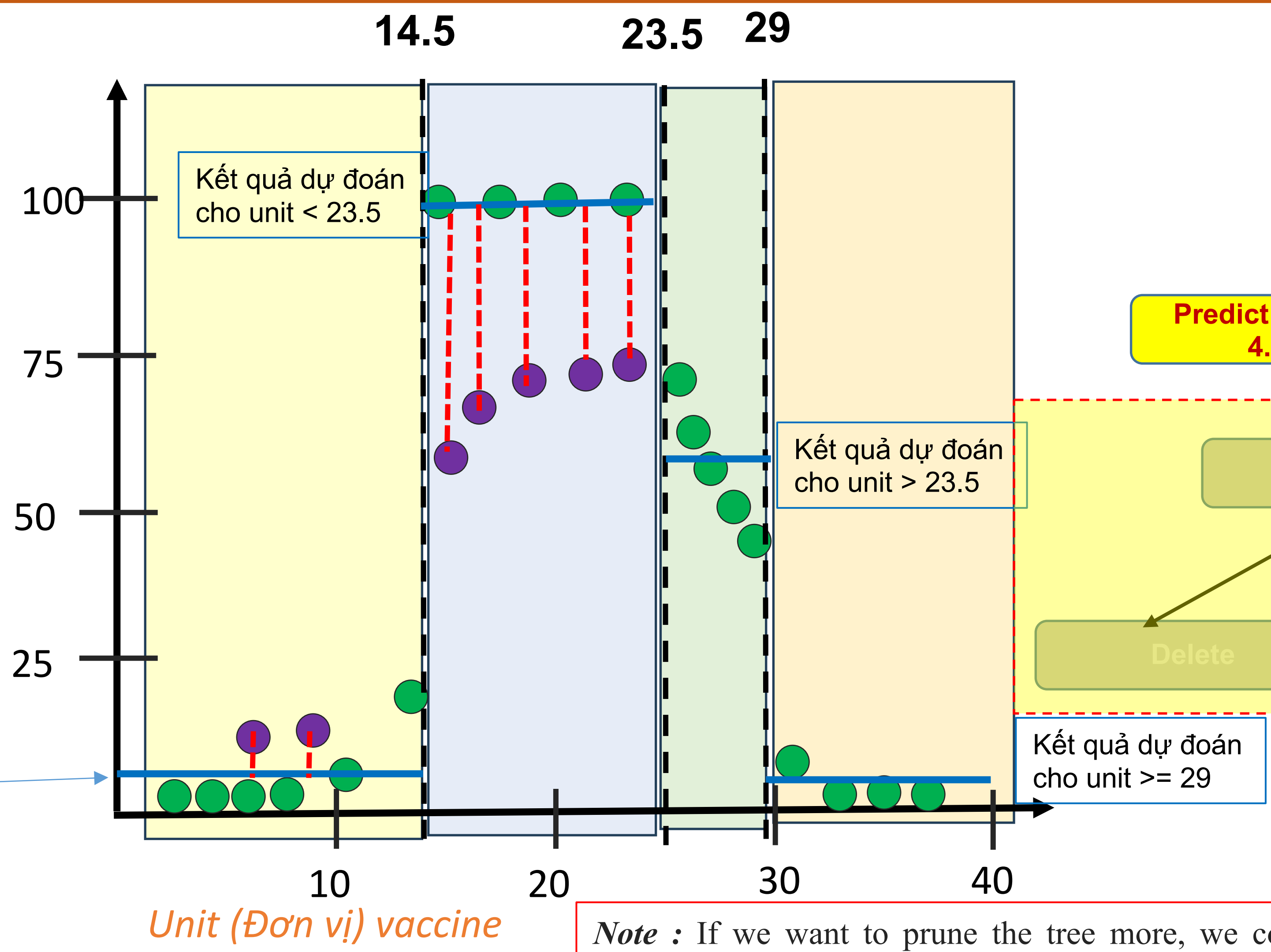
Overfitting Problem

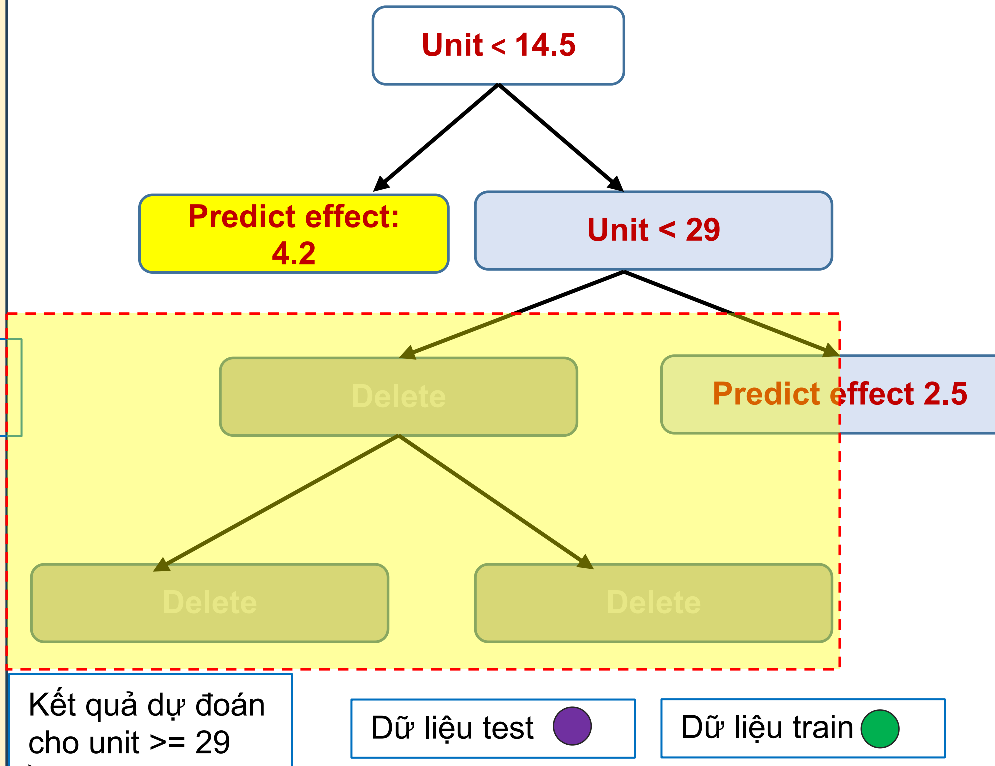


Overfitting Problem



Prunning Solution





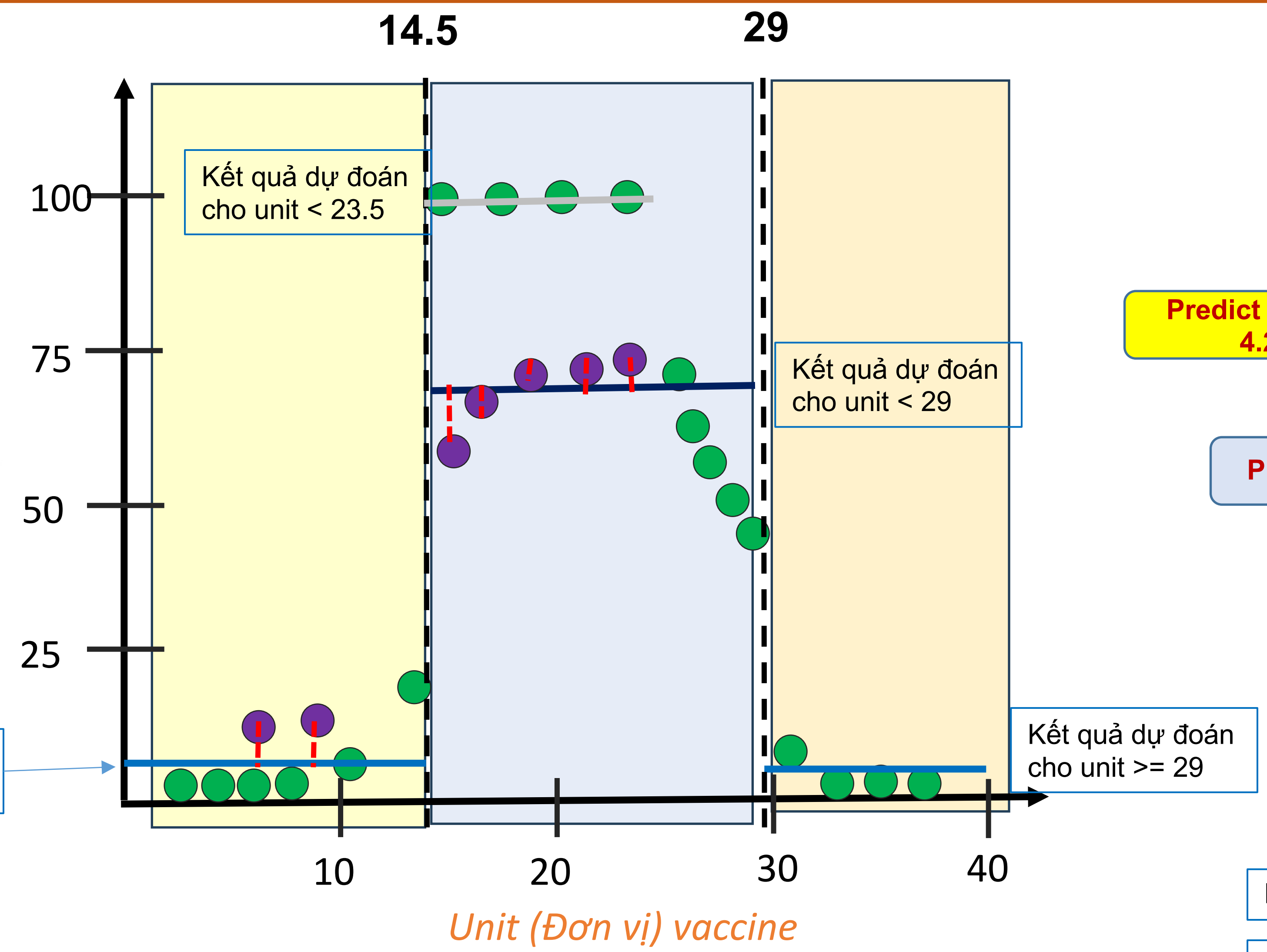
Note **:** If we want to prune the tree more, we could remove last two leaves and replace the split with a leaf that is the average of all of the observations

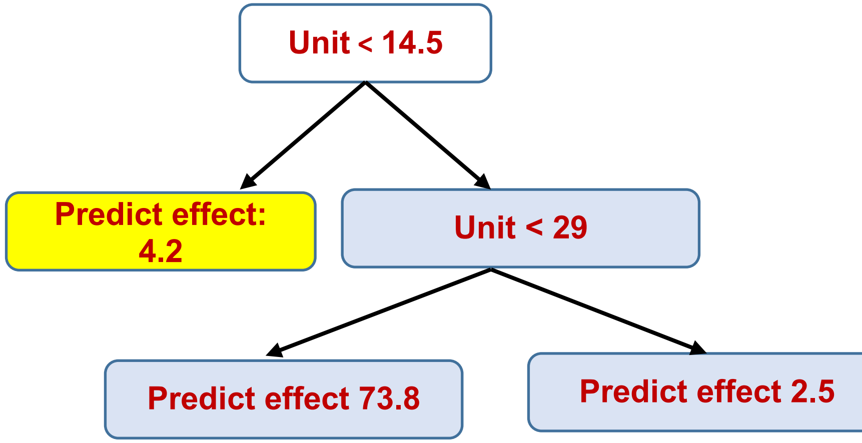
Error

Prunning Solution

Effectiveness
(Hiệu quả)
(%)

Kết quả dự đoán
cho unit < 14.5



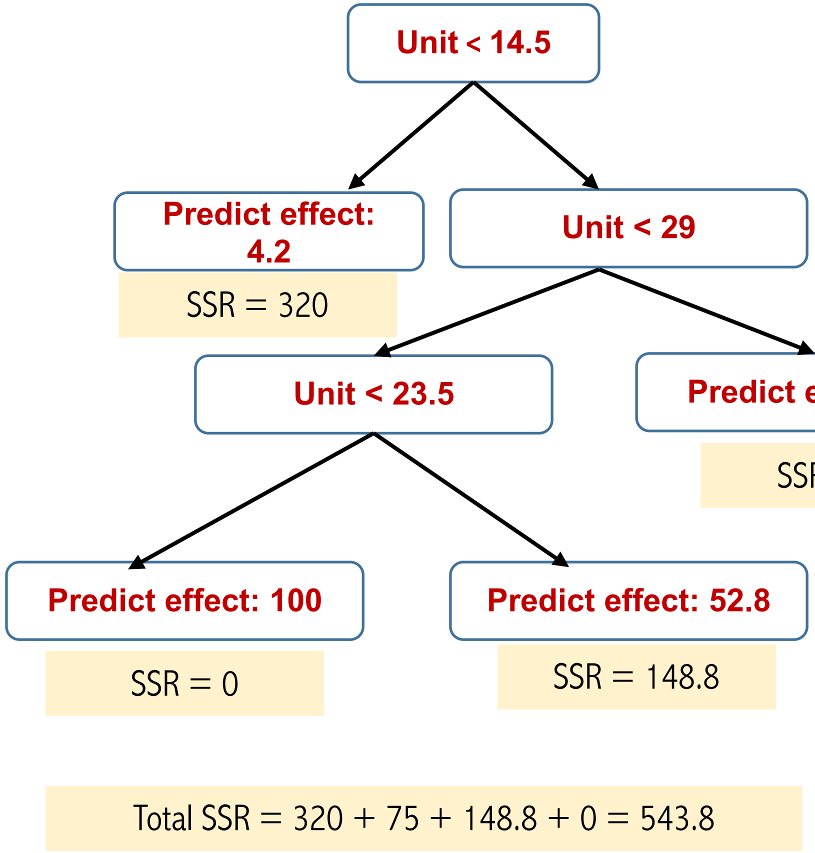


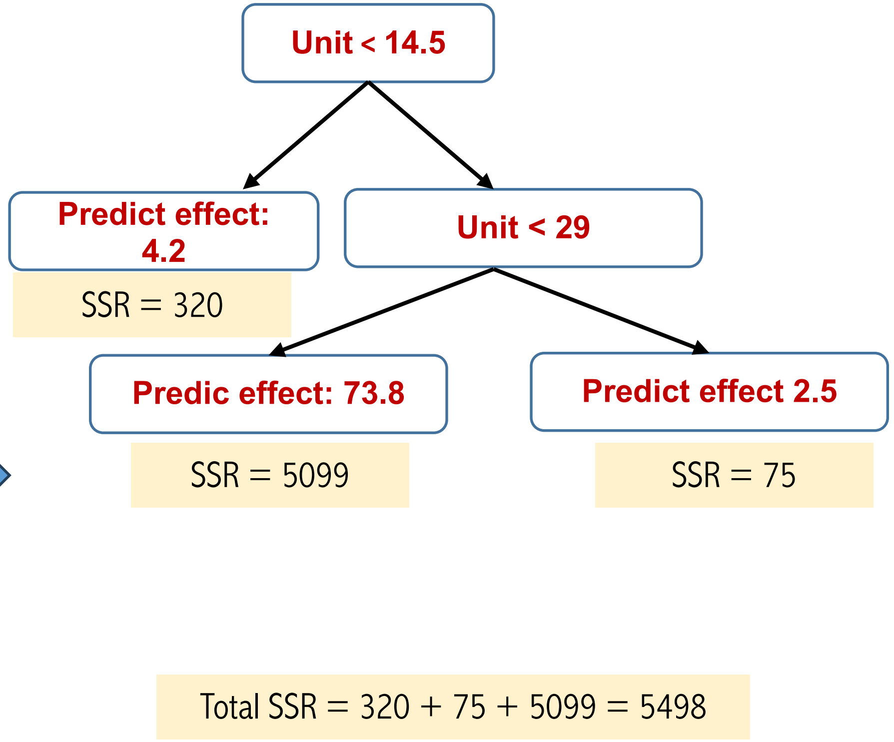
Dữ liệu test

Dữ liệu train

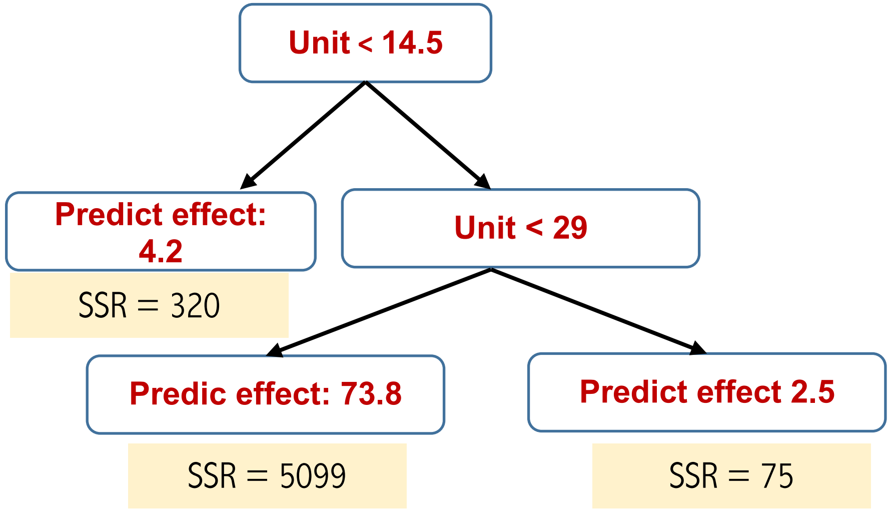
Error

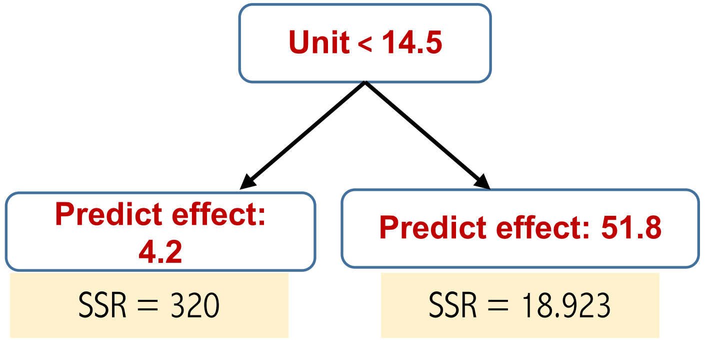
How to select an optimal Tree





How to select an optimal Tree





Total SSR = 320 + 18.923 = 19243

Total SSR = 320 + 75 + 5099 = 5498

How to select an optimal Tree

Unit < 14.5

Predict effect: 4.2

SSR = 320

Predict effect: 51.8

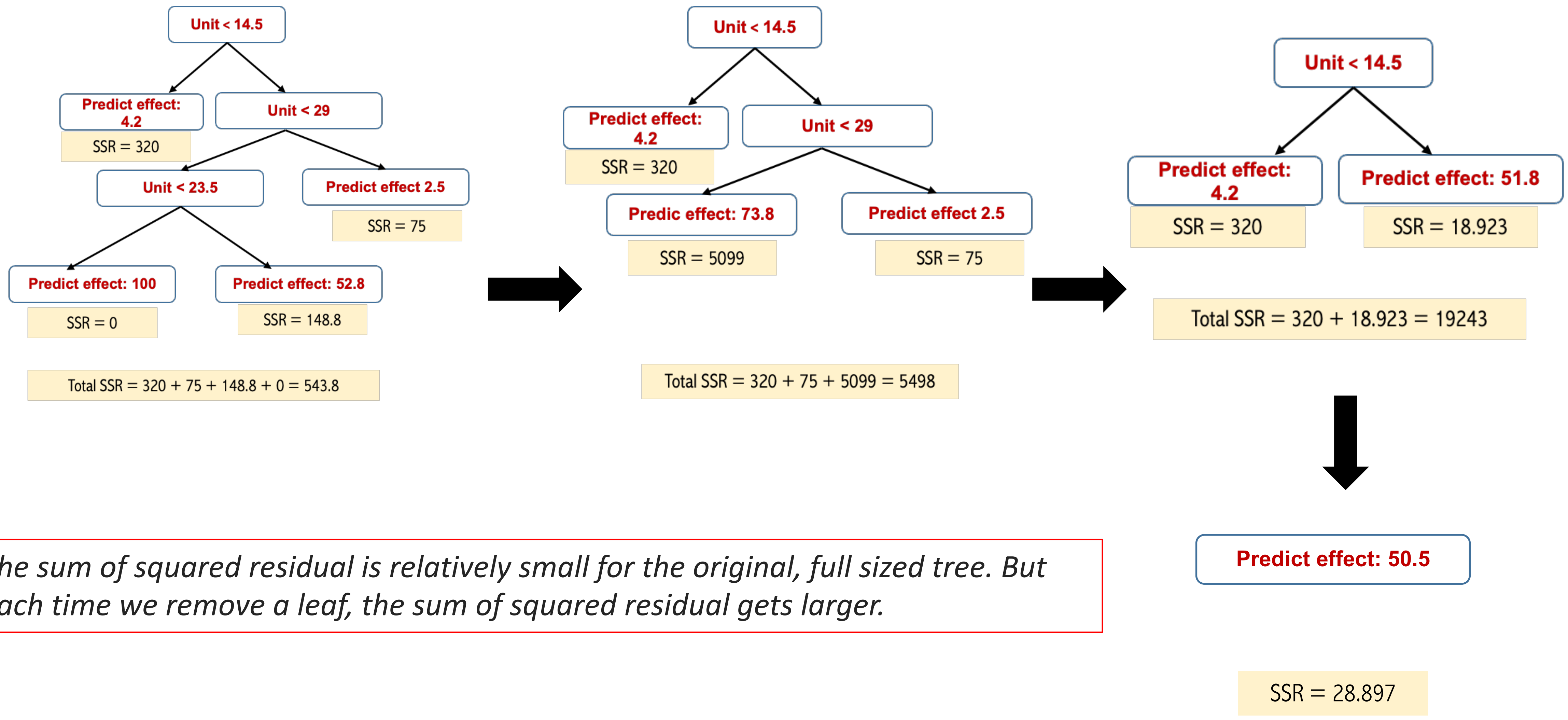
SSR = 18.923

Total SSR = 320 + 18.923 = 19243

Predict effect: 50.5

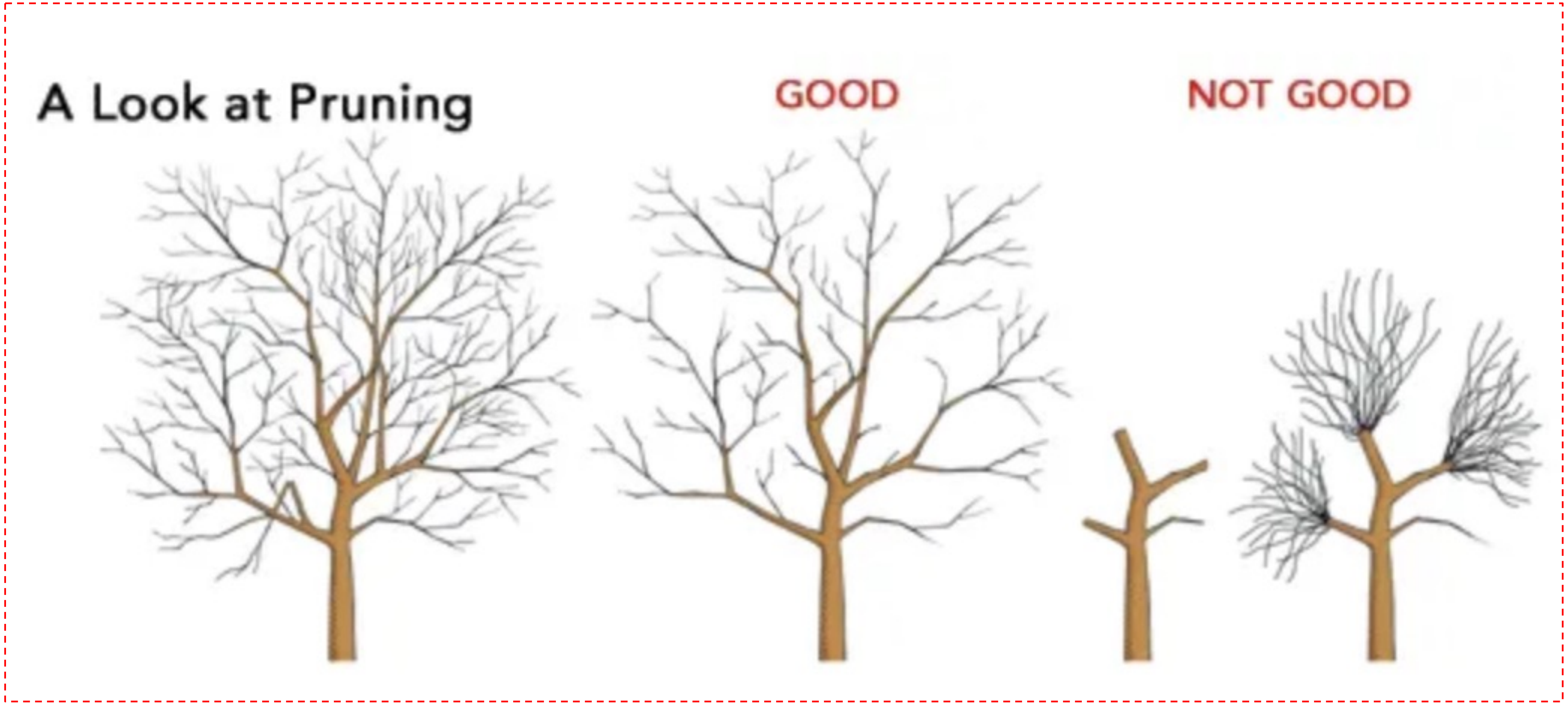
SSR = 28.897

How do we compare these trees?



The sum of squared residual is relatively small for the original, full sized tree. But each time we remove a leaf, the sum of squared residual gets larger.

Decision Tree-Pruning-Cost Complexity Method



Tree complexity penalty

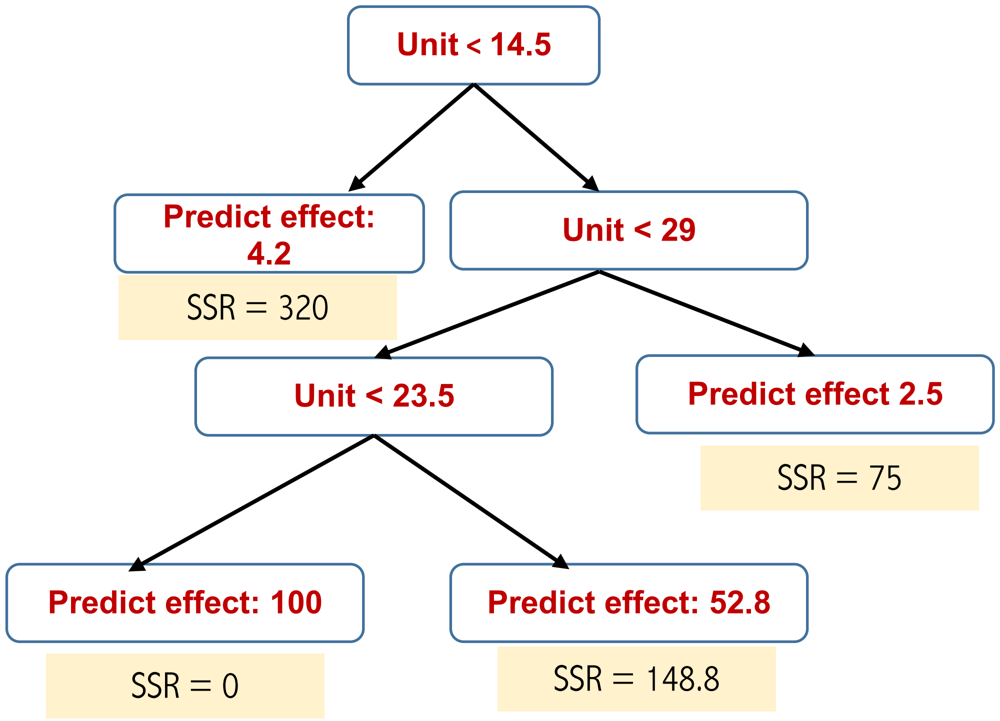
The tree complexity penalty compensates for the difference in the number of leaves.

$$\text{Tree Score} = \text{sum of squared residual} + \alpha T$$

α (alpha) is a tuning parameter that we finding using cross validation.
T is the total number of terminal nodes/the total number of leaves

For now, let's let $\alpha = 10{,}000$ and calculate tree score for each tree.

Tree Score



Total SSR = 320 + 75 + 148.8 + 0 = 543.8

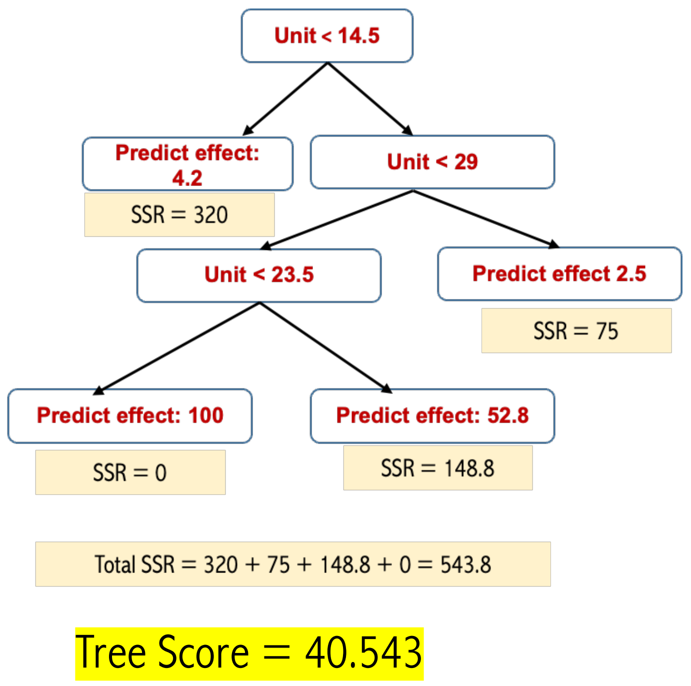
$\alpha = 10000, T = 4$

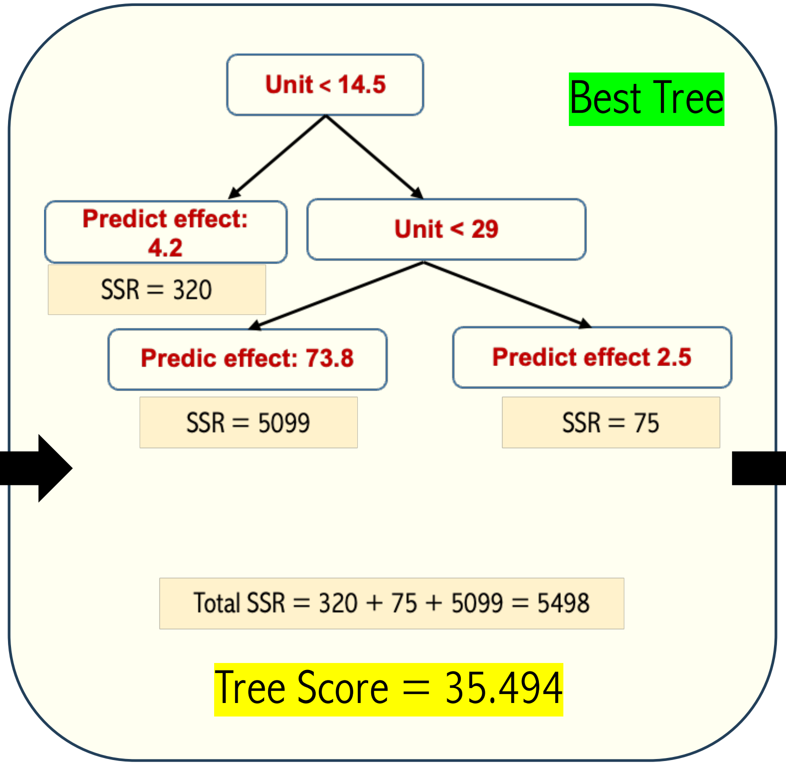
Tree Score = Total SSR+ αT

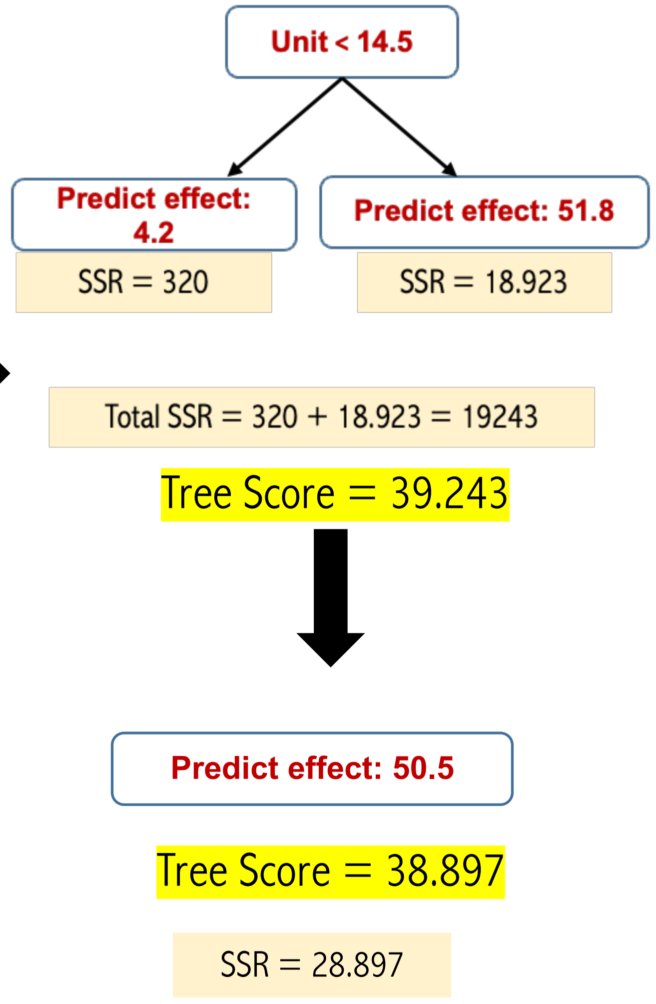
Tree Score = 543 + αT = 40.543

Tree Score

α = 10.000



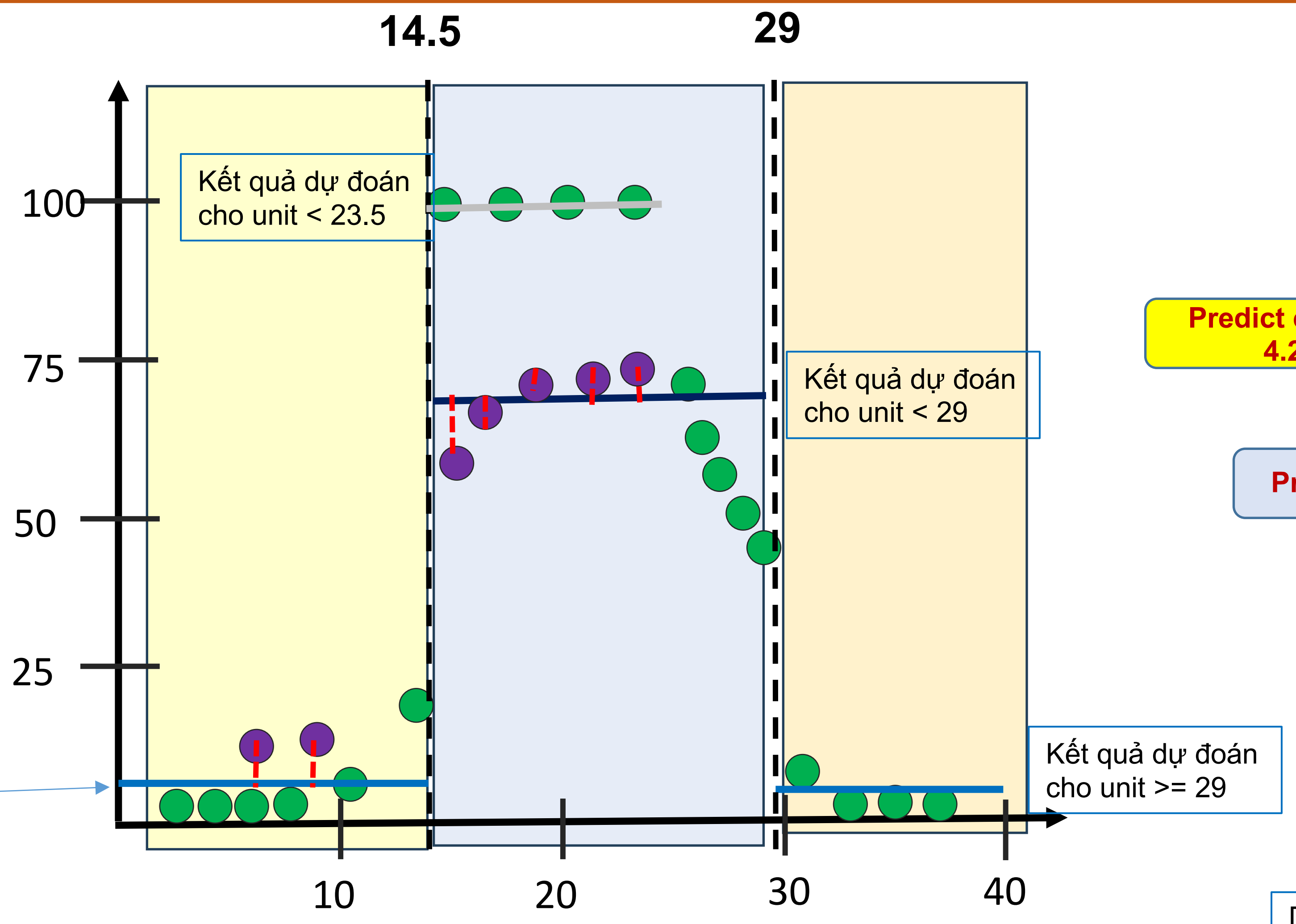




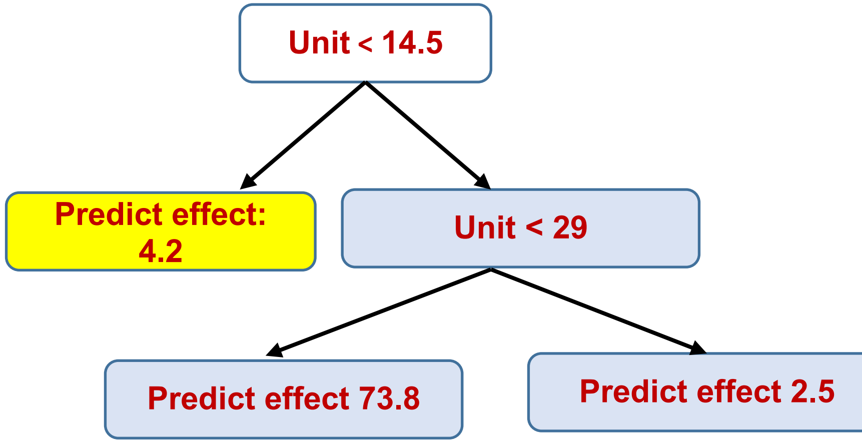
The sum of squared residual is relatively small for the original, full sized tree. But each time we remove a leaf, the sum of squared residual gets larger.

Prunning Solution

Effectiveness
(Hiệu quả)
(%)



Unit (Đơn vị) vaccine

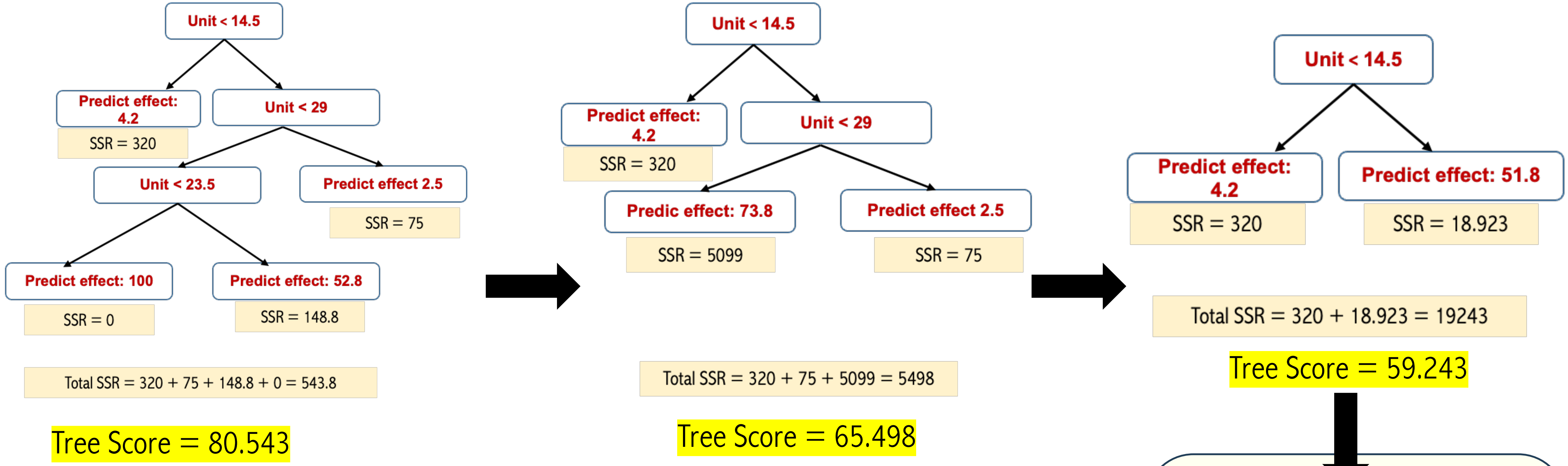


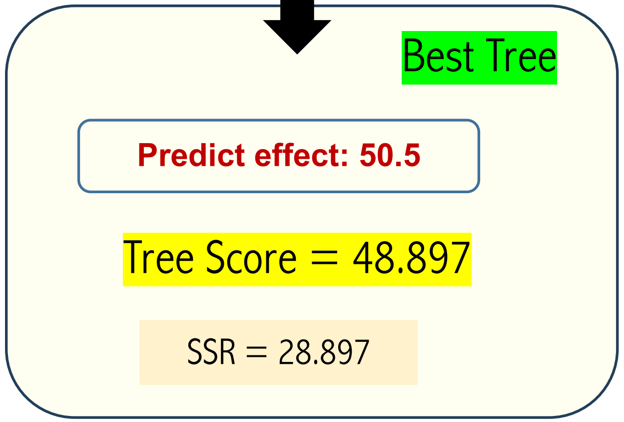
Dữ liệu test
Dữ liệu train

Error

Tree Score

$\alpha = 20.000$





The sum of squared residual is relatively small for the original, full sized tree. But each time we remove a leaf, the sum of squared residual gets larger.

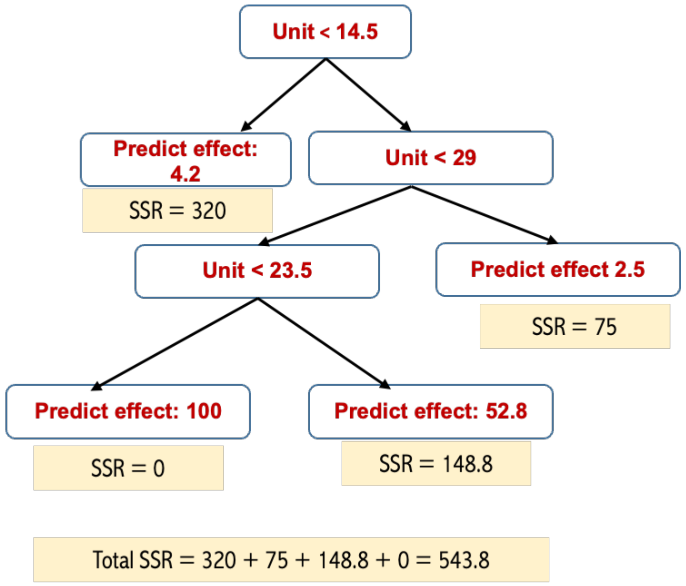
How to select α

1

Entire dataset

Unit	Age	Sex	Effect (%)
10	25	Female	98
20	73	Male	0
35	54	Female	100
5	12	Male	44
7	80	Male	5
…	…	…	…

Full size Tree



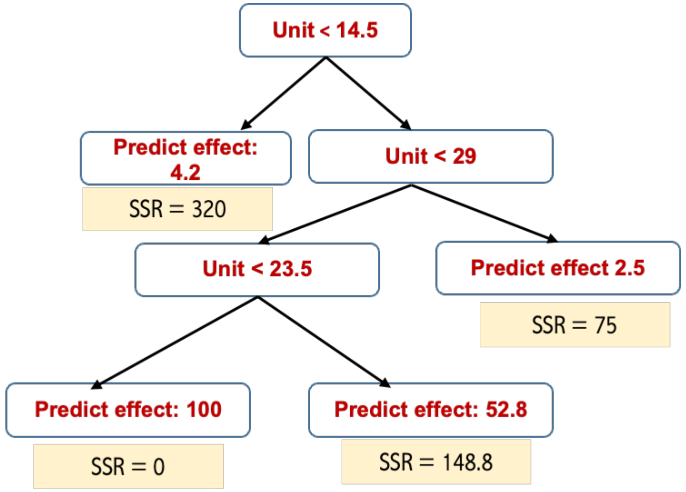
Tree Score = sum of squared residual + αT

This full size tree has lowest Tree Score when α = ??

How to select α

2

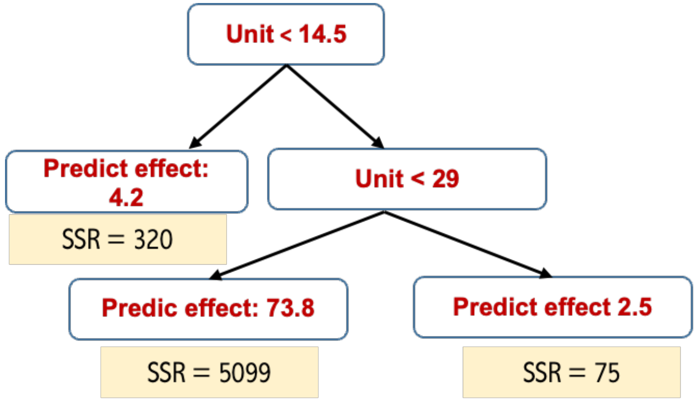
Full size Tree



Total SSR = 320 + 75 + 148.8 + 0 = 543.8

$\alpha = 0$

Prunning Tree



Total SSR = 320 + 75 + 5099 = 5498

$\alpha = 10.000$

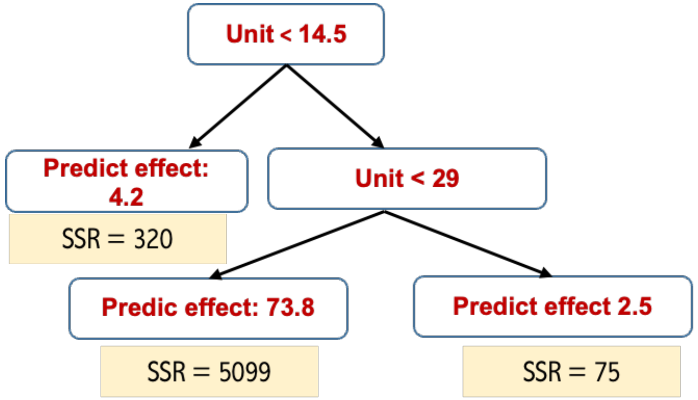
Tree Score = sum of squared residual + αT

Increase untill pruning leaves will give us a lower Tree Score

How to select α

3

Tree

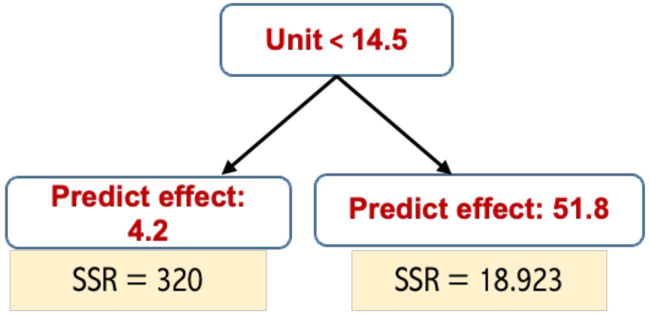


Total SSR = 320 + 75 + 5099 = 5498

$\alpha = 10.000$

Tree Score = sum of squared residual + αT

Prunning Tree



Total SSR = 320 + 18.923 = 19243

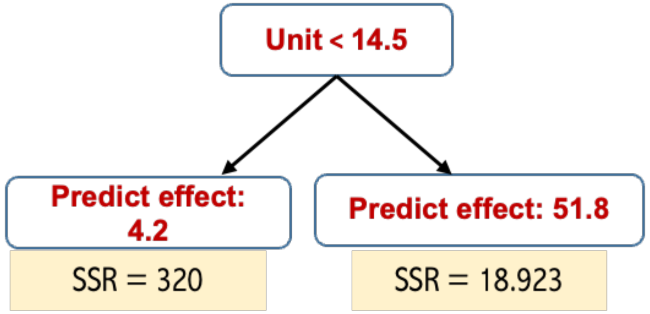
$\alpha = 15.000$

Increase untill pruning leaves will give us a lower Tree Score

How to select α

3

Tree



Total SSR = 320 + 18.923 = 19243

$\alpha = 15.000$

Tree Score = sum of squared residual + αT

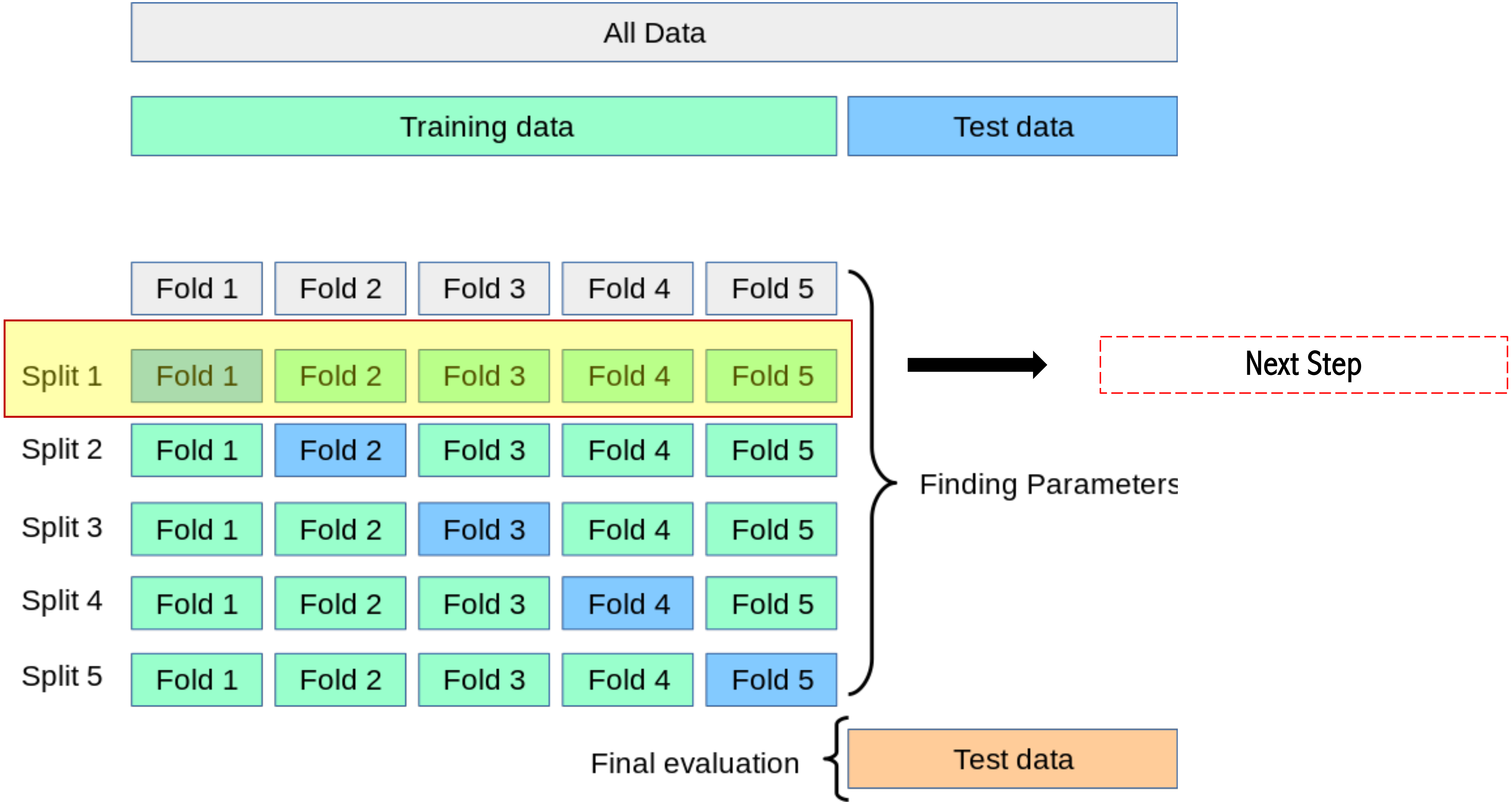
Leaf

Predict effect: 50.5

$\alpha = 20.000$

Increase untill pruning leaves will give us a lower Tree Score

How to select α



How to select α

For each Split

Entire dataset

Unit	Age	Sex	Effect (%)
10	25	Female	98
20	73	Male	0
35	54	Female	100
5	12	Male	44
7	80	Male	5
…	…	…	…

Training dataset

Unit	Age	Sex	Effect (%)
10	25	Female	98
20	73	Male	0
…	…	…	…

Build Tree with $\alpha = 0$, $\alpha = 10000$, $\alpha = 15000$, $\alpha = 20{,}000$

Testing dataset

Unit	Age	Sex	Effect (%)
5	12	Male	44
7	80	Male	5
…	…	…	…

Tree Score with $\alpha = 0$, $\alpha = 10000$, $\alpha = 15000$, $\alpha = 20{,}000$

How to select α

	α = 0	α = 10,000	α = 15000	α =20,000
Split 1	...	...	...	...
Split 2	...	...	...	...
Split 3	...	...	...	...
Split 4	...	...	...	...
Split 5	...	...	...	...
Average	50,000	5000	11,000	30,000

In this case, the optimal trees built with α = 10,000 had, on average, the lowest sum of square residuals. So α = 10,000 is our final value.

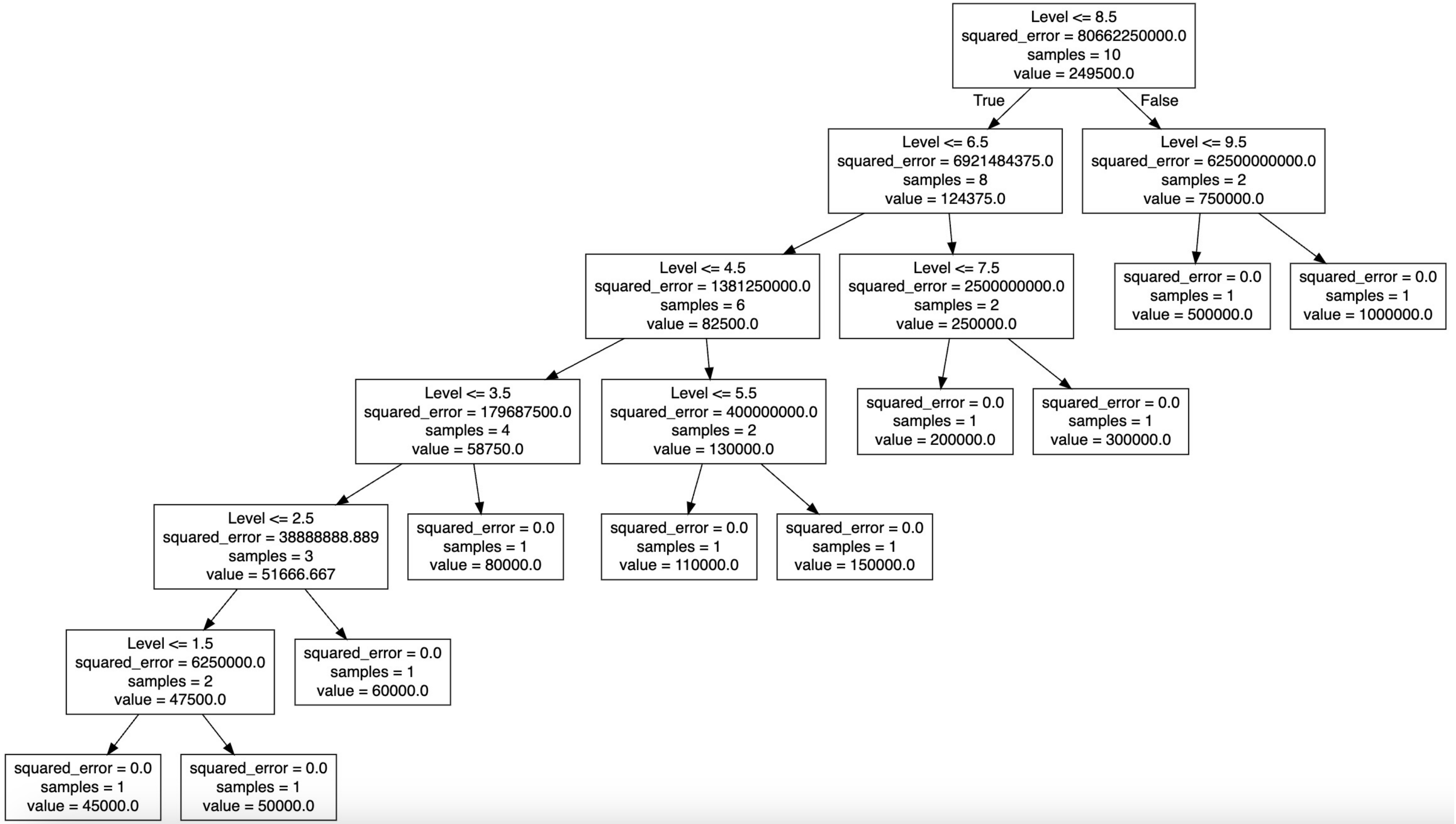
Outline

- Motivation for Regression Tree
- Regression Tree
- Overfitting in Regression Tree
- Case study

Case study

Position_Salaries

Position	Level	Salary
Business Analyst	1	45000
Junior Consultant	2	50000
Senior Consultant	3	60000
Manager	4	80000
Country Manager	5	110000
Region Manager	6	150000
Partner	7	200000
Senior Partner	8	300000
C-level	9	500000
CEO	10	1000000



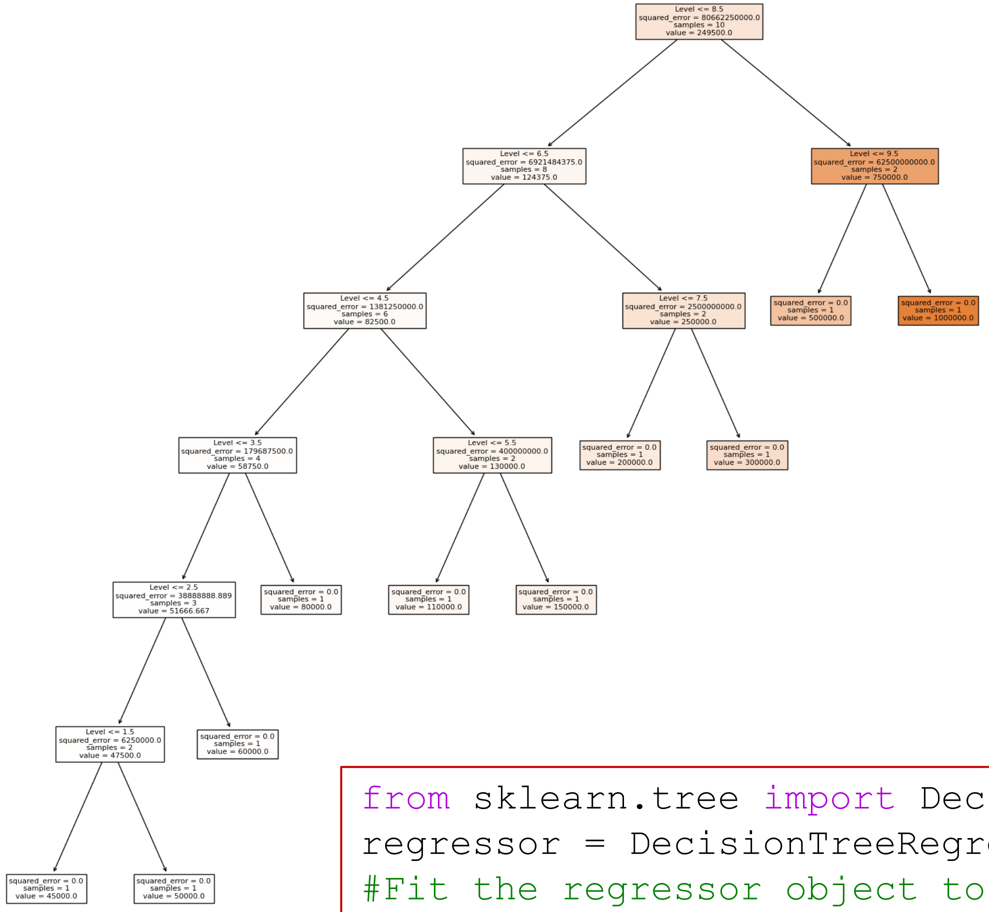
http://www.webgraphviz.com/

```
export_graphviz(regressor, out_file ='tree.dot',
feature_names =["Level"])
```

Case study

Position_Salaries

Position	Level	Salary
Business Analyst	1	45000
Junior Consultant	2	50000
Senior Consultant	3	60000
Manager	4	80000
Country Manager	5	110000
Region Manager	6	150000
Partner	7	200000
Senior Partner	8	300000
C-level	9	500000
CEO	10	1000000



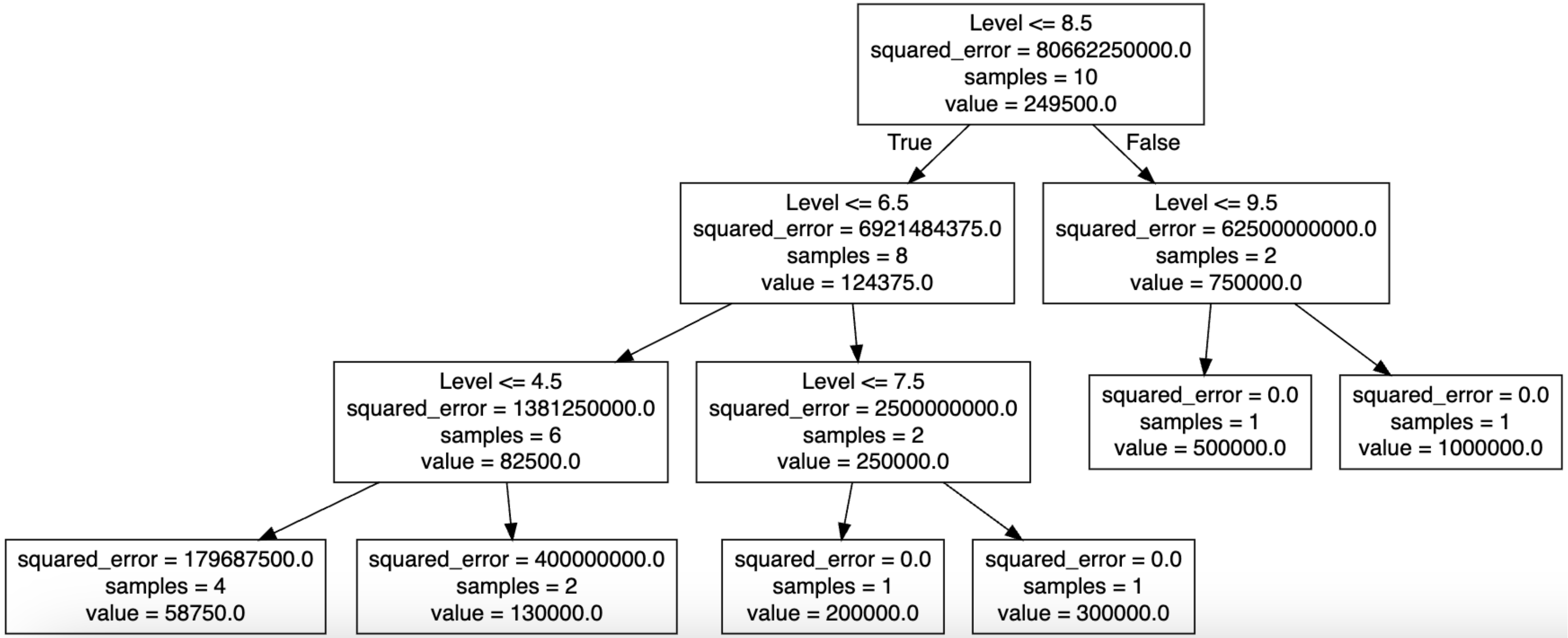

```
from sklearn.tree import DecisionTreeRegressor
regressor = DecisionTreeRegressor(random_state=0)
#Fit the regressor object to the dataset.
regressor.fit(X,y)
```

```
tree.plot_tree(regressor, ax=ax, feature_names = ["Level"],
               filled=True)
```

Case study

Position_Salaries

Position	Level	Salary
Business Analyst	1	45000
Junior Consultant	2	50000
Senior Consultant	3	60000
Manager	4	80000
Country Manager	5	110000
Region Manager	6	150000
Partner	7	200000
Senior Partner	8	300000
C-level	9	500000
CEO	10	1000000

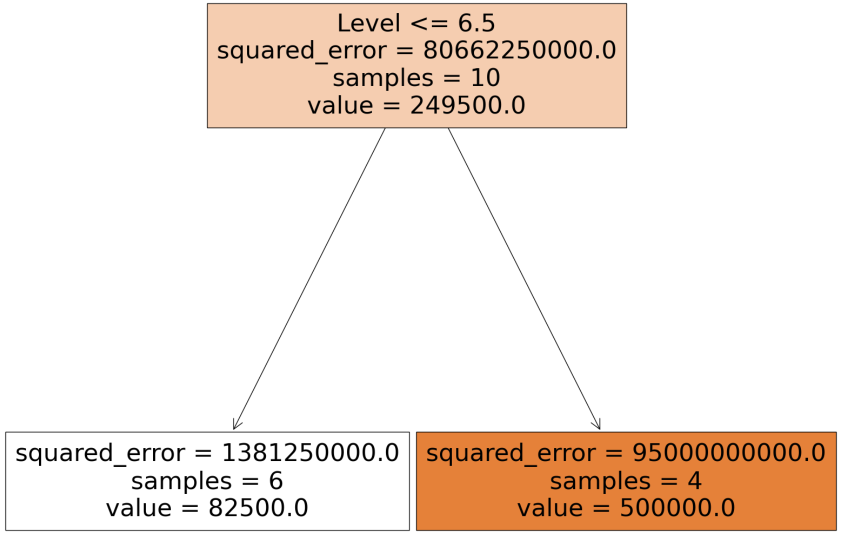



```
regressor = DecisionTreeRegressor(random_state=0,
max_depth=3)
```

Case study

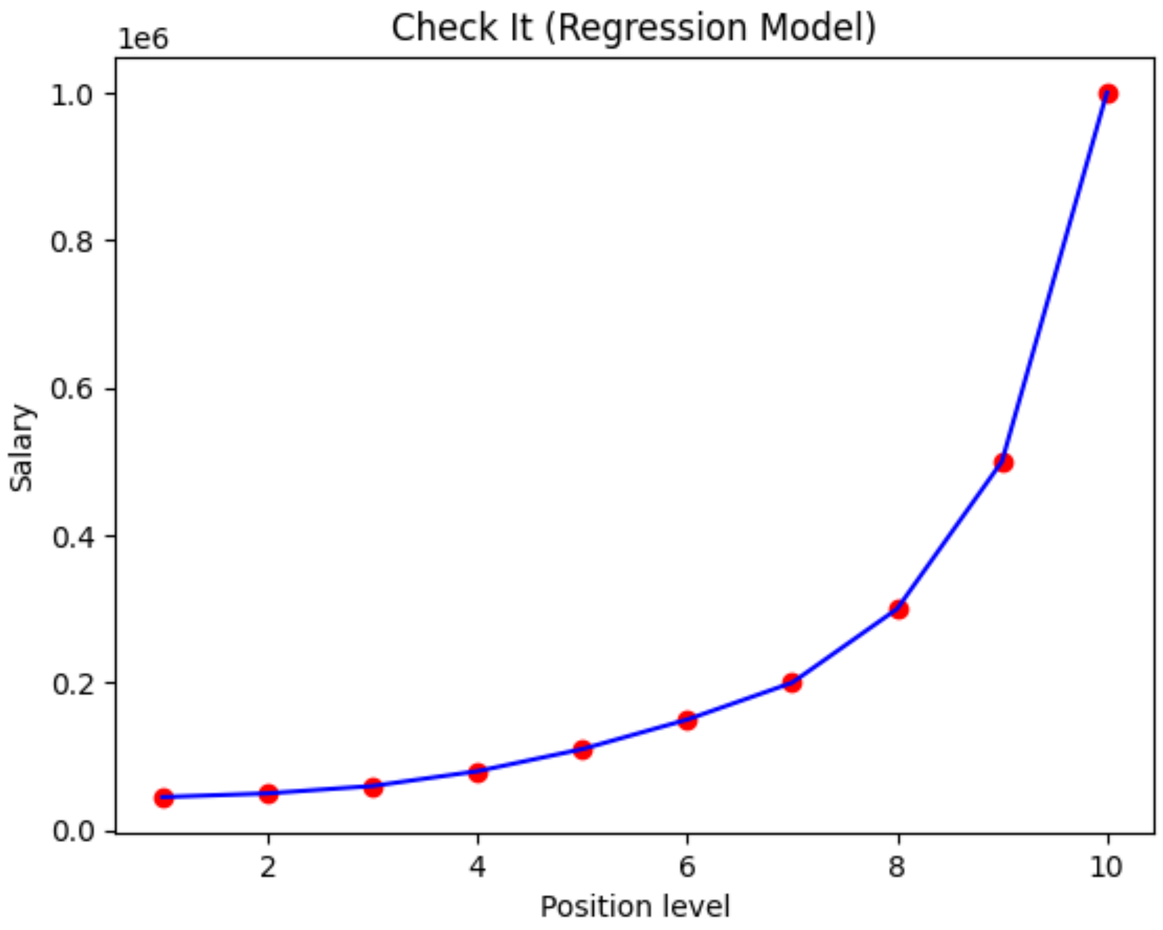
Position_Salaries

Position	Level	Salary
Business Analyst	1	45000
Junior Consultant	2	50000
Senior Consultant	3	60000
Manager	4	80000
Country Manager	5	110000
Region Manager	6	150000
Partner	7	200000
Senior Partner	8	300000
C-level	9	500000
CEO	10	1000000



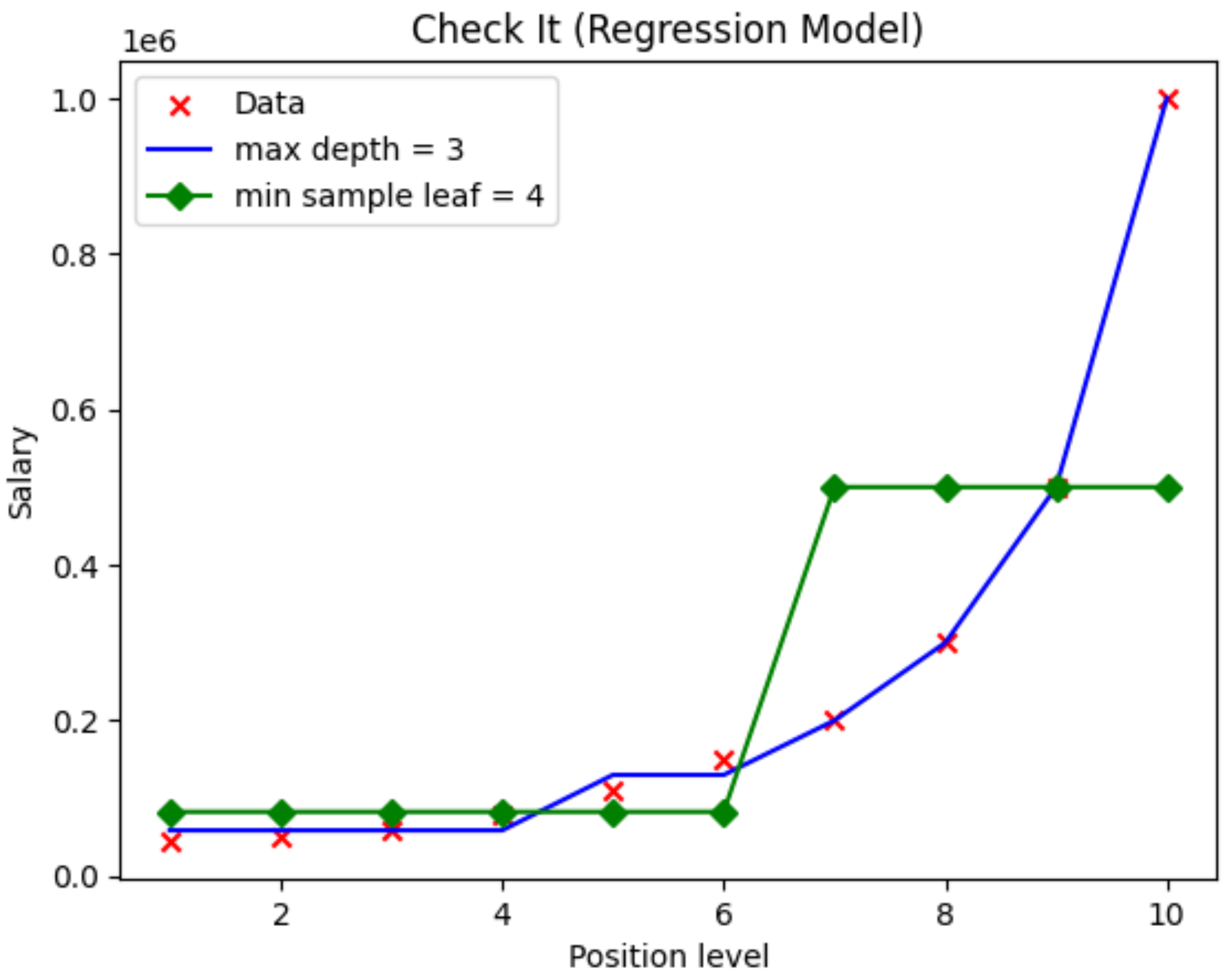

```
regressor = DecisionTreeRegressor(random_state=0,
min_samples_leaf=4)
```

Case study




```
Visualising the Decision Tree Regression results
plt.scatter(X, y, color = 'red')
plt.plot(X, regressor.predict(X), color = 'blue')
plt.title('Check It (Regression Model)')
plt.xlabel('Position level')
plt.ylabel('Salary')
plt.show()
```

Case study


Check It (Regression Model)
1e6
1.0
0.8
0.6
0.4
0.2
0.0
Salary
Data
max depth = 3
min sample leaf = 4
2
4
6
8
10
Position level


THANK
YOU