AI VIET NAM – WE ARE ONE TEAM – CLASS OF AIO2024

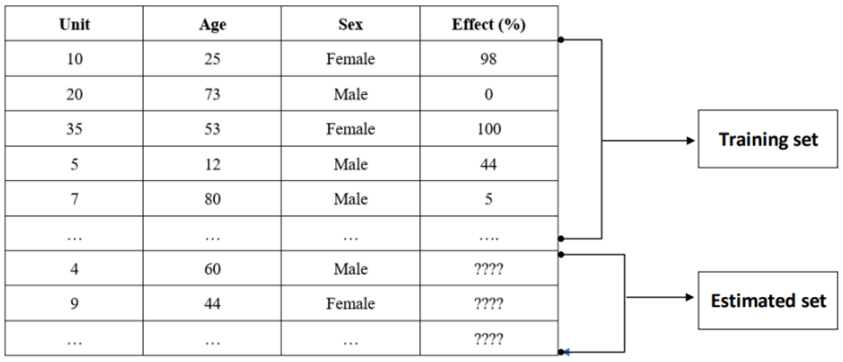
# SUMMARY
# of
# Decision Tree for Regression

Ngày 21 tháng 2 năm 2024

| | |
|---|---|
| **Date of publication:** | 15/02/2024 |
| **Authors:** | AIO |
| **Sources:** | |
| **Data sources (if any):** | |
| **Keywords:** | Decision Tree, Regression, Overfitting, Underfitting, Pruning Solution, Tree Complexity Penalty, Cross validation |
| **Summary by:** | Bùi Nhật Linh |

## I. Tổng quát:

1. Giới thiệu về Decision Tree for Regression:
   - Decision Tree for Regression là một mô hình máy học có giám sát (supervised learning machine algorithm) được sử dụng để huấn luyện (training), hồi quy (regresses) dữ liệu/ kết quả đầu ra liên tục (continuous outputs) hay các biến định lượng (regression taks) thành nhóm dữ liệu có cùng điều kiện.
   - Mô hình này cho ra kết quả dự báo có độ chính xác cao nhằm dự báo kết quả theo phạm vi giá trị cho trước của các outputs trong tập training set (range).




Hình 1: Ví dụ về kết quả mong muốn dự đoán

2. So sánh với Decision Tree for Classification:
   (a) Giống nhau:

- Về cấu trúc cây: Mô hình Decision Tree for Regression bao gồm cấu trúc Root-node, Terminal nodes, và Leaf nodes. Đồng thời, mô hình này bắt đầu bằng việc xác định cấu trúc tương tự như Mô hình Decision Tree for Classification
- Về mục tiêu: phân loại nhóm dữ liệu

(b) Khác nhau:

- Mô hình Decision Tree for Regression áp dụng cho dữ liệu/ kết quả đầu ra liên tục (continuous outputs).
- Để xác định các nodes, Mô hình này cần đánh giá lỗi/chênh lệch (Residual Error) từ đó tính tổng lỗi/chênh lệch theo Phương pháp Sum of Squared Error để so sánh lần lượt các outputs.
- Mô hình Decision Tree for Regression sử dụng Phương pháp chặt nhánh cây (Prunning Solution) nhằm hạn chế việc kết quả của mô hình toán áp dụng cho dữ liệu thực tế và đưa các kết quả có độ lệch lớn (Overfitting)
- Ngoài ra, mục tiêu của Mô hình Decision Tree for Regression mang tính chất dự báo trong khi Mô hình Decision Tree for Classification mang tính chất phân loại

3. Các vấn đề sau khi thực hiện áp dụng mô hình vào dữ liệu cần dự đoán:
   - Underfitting: là hiện tượng kết quả độ chênh lệch của mô hình được huấn luyện và kết quả độ chênh lệch của dữ liệu cần dự đoán đạt giá trị mức cao giống nhau, do mô hình chưa được huấn luyện đầy đủ. Cần xem lại cấu trúc của mô hình (tăng thêm độ phức tạp) để có thể huấn luyện các tập dữ liệu khó và tăng thêm dữ liệu huấn luyện để tăng hiệu suất của mô hình.
   - Overfitting: là hiện tượng kết quả của mô hình được huấn luyện quá tốt (độ chênh lệch thấp) nhưng khi áp dụng vào dữ liệu cần dự đoán thì mô hình đạt hiêu suất kém (độ chênh lệch cao) do mô hình đã học quát sát với dữ liệu huấn luyện và không có khả năng tổng quát hóa các dữ liệu cần dự đoán. Cần sử dụng một số các phương pháp tránh overfitting như tăng độ đa dạng của dữ liệu, giảm thiểu độ phức tạp của mô hình
4. Giải pháp tỉa cành cây – Pruning solution:
   - Pruning solution được áp dụng đối với trường hợp mô hình huấn luyện bị overfitting khi sử dụng mô hình Decision Tree bằng cách hạn chế kích thước, chiều sâu của mô hình này.
5. Độ phức tạp của cây – Tree complexity method:
   - Phương pháp Tree complexity như là phương pháp nomarlizing các kết quả SSR (trên thực tế có nhiều phương pháp normalization khác). Đồng thời, phương pháp cũng tính độ phức tạp của mô hình Decision Tree.
6. Cross validation – Tập huấn tập dữ liệu lẫn nhau:
   Là một phương pháp tập huấn tệp dữ liệu trong trường hợp tập dữ liệu (training set) không được đa dạng về số lượng. Phương pháp này áp dụng cho k số tập dữ liệu đã được xáo trộn.

## II. Thực hiện xác định mô hình Decision Tree:

1. Bối cảnh trường hợp áp dụng mô hình:
   - Các nhà khoa học nghiên cứu ra một loại vaccine và thử nghiệm trên các mẫu bệnh nhân bao gồm các yếu tố: Liều lượng dùng cố định (Unit), Tuổi (Age), Giới tính (Sex), và độ hiệu quả của vaccine (Effect).
     Câu hỏi: hãy thiết lập mô hình toán Decision Tree for Regression để dự đoán xem nếu áp dụng loại vaccine trên cho các mẫu bệnh nhân khác thì kết quả (Effect) như thế nào?
     *(Chú ý: Effect được coi là outputs, các data inputs của từng yếu tố Unit, Age, Giới tính được gọi là samples)*


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

Hình 2: Bối cảnh áp dụng thuật toán

2. Các trình tự thực hiện tạo lập mô hình Decision Tree:
   (a) Xác định nodes:
   - Bước 1: Giả định Root node:
     Giả định các trường hợp Root node lần lượt theo từng yếu tố, cụ thể: Liều lượng dùng cố định (Unit), Tuổi (Age), và Giới tính (Sex).
   - Xác định Terminal nodes theo giả định Root node:
     Sau khi xác định Root node, đặt các điều kiện cụ thể để phân nhánh sao cho đạt được hiệu quả cao nhất. Ví dụ, các terminal nodes nên bắt đầu từ 8 -> 20 nodes (samples) hoặc độ sâu (depth) của các cây <= 3
     Việc phân nhánh các Terminal nodes được thực hiện với lần lượt samples còn lại trong một điều kiện cụ thể. Đồng thời, để phân nhánh cần thực hiện việc tính tổng giá trị trung bình các outputs -> đánh giá lỗi/chênh lệch (Residual Error) giữa các samples -> tính tổng lỗi/chênh lệch theo công thức Sum of Squared Error của các lỗi/chênh lệch.
   - Bước 3: Xác định Final Root node:
     Sau khi tính tổng lỗi/chênh lệch các Terminal nodes theo từng giả định Root node, giả định Root node nào có giá trị lớn hơn sẽ được chọn làm Final Root node
   - Bước 4: Xác định Terminal node theo Final Root node:
     Việc xác định Terminal node theo Final Root node được xác định giống như các bước 2 -3. Trong đó, bước 3 sẽ xác định các Final Terminal Nodes
   - Bước 5: Xác định Leaf nodes: có thể được xác định khi không thể tách thêm

   (b) Phương thức tính toán:
   - Công thức tính trung bình:

$$\hat{Y} = \frac{\sum_{i=1}^{n} Xi}{n}$$

     – Ŷ Trung bình cộng
     – Xi: Giá trị các outputs trong 1 khoảng điều kiện
     – n: Số lượng các outputs trong 1 khoảng điều kiện
   - Công thức tính độ chênh lệch/lỗi:

$$ResidualError = (Yi - \hat{Y})^2$$

     – Residual Error: Độ chênh lệch/ lỗi giữa các outputs
     – Yi: Giá trị của outputs
     – Ŷ: Giá trị của average

- Công thức tính Tổng giá trị chênh lệch/lỗi:

$$SSR = \sum_{i=1}^{n} ResidualError$$

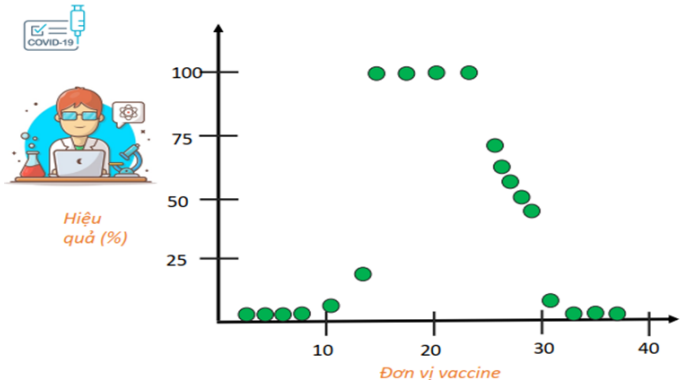
  - SSR: Tổng các chênh lệch/lỗi của các samples
  - Residual error: Độ chênh lệch/lỗi giữa các samples
- Công thức tính Tree score:

$$\text{Tree Score} = SSR + \alpha T$$

  - SSR: Tổng các chênh lệch/lỗi của các samples
  - $\alpha$: Tham số điều chỉnh được sử dụng thông qua phương pháp cross validation
  - T: Tổng số các Terminal nodes và Leaf nodes

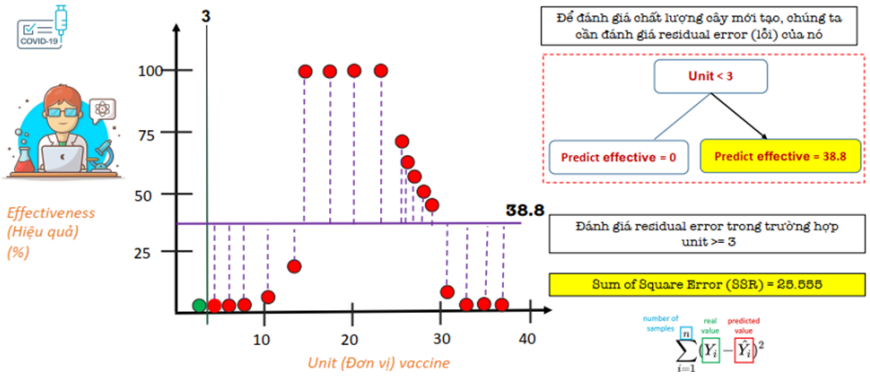
(c) Thực hiện áp dụng vào bối cảnh trường hợp:

- Giả định yếu tố Unit là Root node:



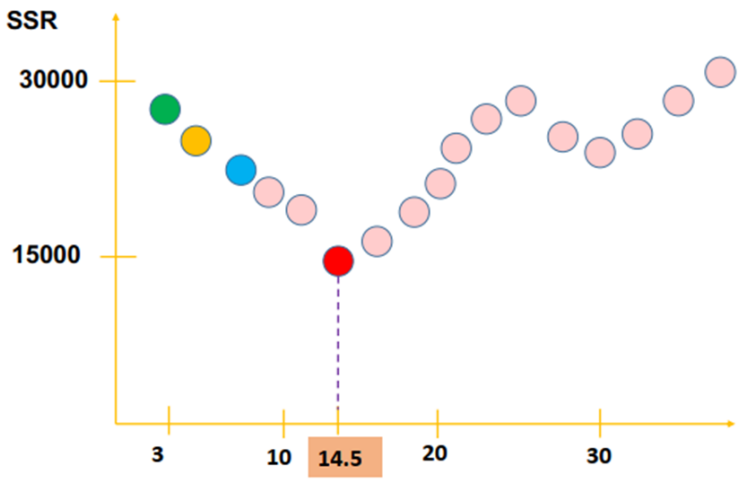

Hình 3: Biểu thị yếu tố Unit là trên biểu đổ

  - Tại đây lần lượt tính Residual Error và SSR giữa outputs với từng, cụ thể như sau:



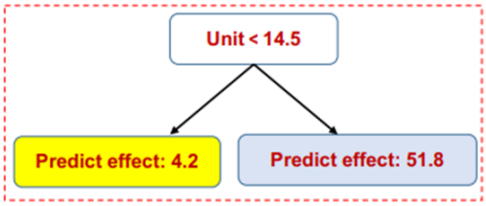

Hình 4: Tính Residual Error và SSR

  - Với hình trên, lựa chọn từng output từ trái sang phải để tính Residual Error và SSR. Sau khi đã tính hết các SSR, kết quả được biểu thị như sau:




Hình 5: Biểu thị các SSR lên biểu đồ

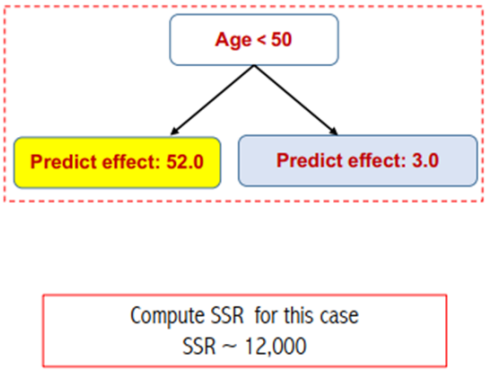
– Lựa chọn SSR có giá thấp nhất tương ứng với output có giá trị 14.5, tổng số node <= 7 và => 20, và độ sâu <= 3. Theo đó, lần lượt các bước, mô hình Decision Tree được xác định như sau:



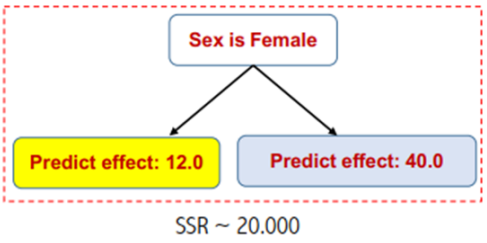

Hình 6: Biểu thị các SSR lên biểu đồ

Tính SSR trong tiêu chí này là 19.000

- Giả định Age/Sex là Root node:
  Thực hiện tương tự các bước như trường hợp của Unit, các mô hình và kết quả SSR được biểu thị như sau:

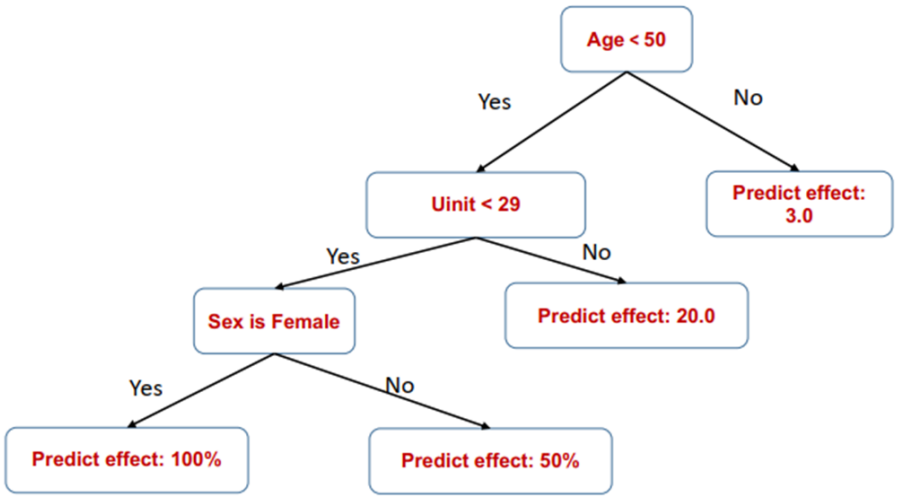



Hình 7: Decision Tree dựa theo tiêu chính Age




Hình 8: Decision Tree dựa theo tiêu chính Sex

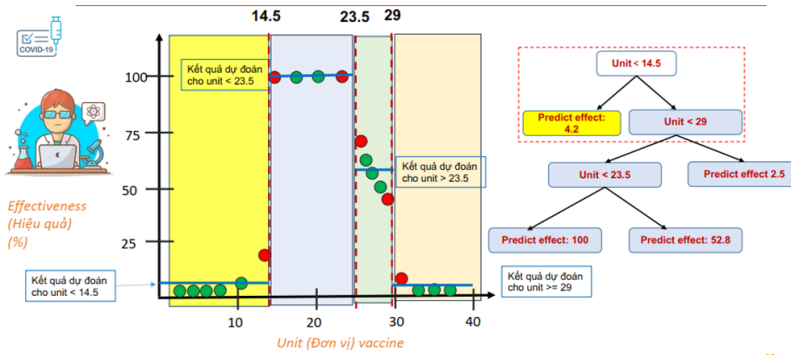
- Như vậy sau khi có các kết quả SSR ứng với từng yếu tố, lựa chọn Age là Root node
- Thực hiện các bước tương tự để xác định Terminal Nodes và Leaf node, kết quả cuối cùng được biểu thị như sau:




Hình 9: Final Decision Tree

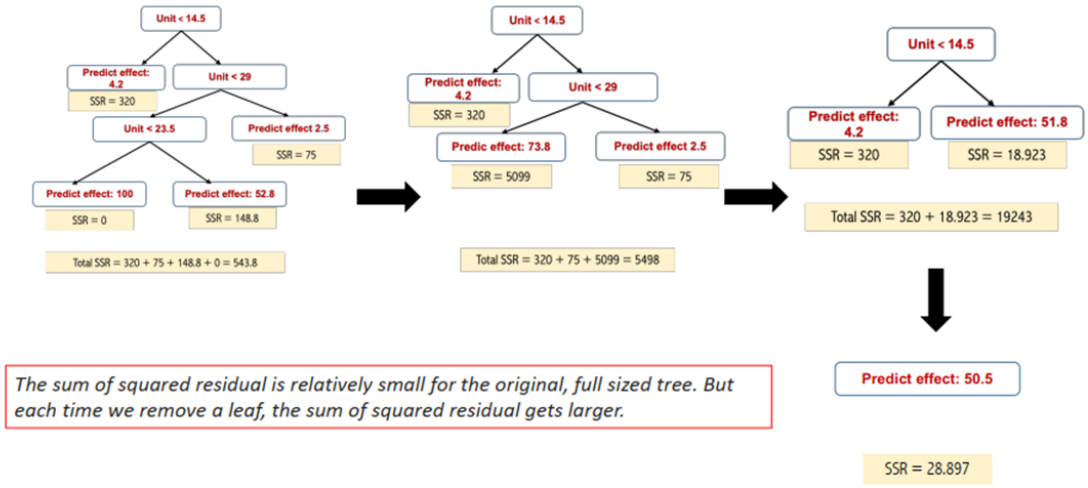
## III. Xử lý Overfitting trong mô hình Decision Tree:

1. Pruning Solution:
   Xét trường hợp Unit là Root node và có tạo ra cây Decision Tree bao gồm đầy đủ cấu trúc như sau:



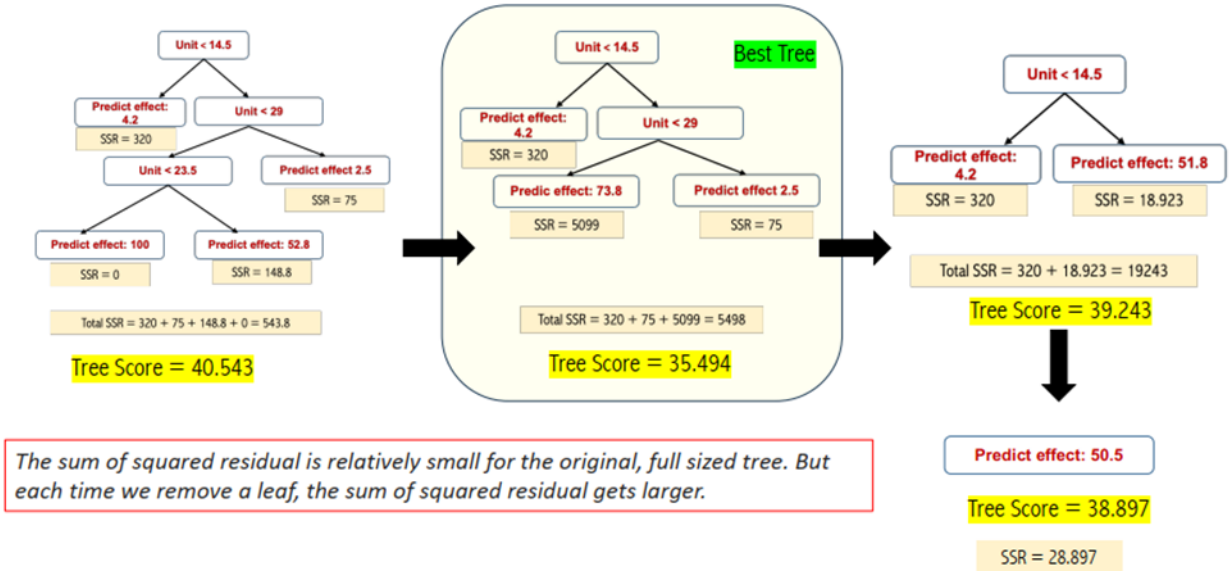

Hình 10: Mô hình Decision Tree trong trường hợp Unit là Root node

Sau khi áp dụng mô hình này lên một tập dữ liệu cần dự báo, nhận thấy độ chênh lệch ở mức cao. Do đó, giải pháp áp dụng Pruning Solution được tiến hành để giảm thiểu Overfitting lần lượt:




Hình 11: Các mô hình Decision Tree trước và sau khi Pruned

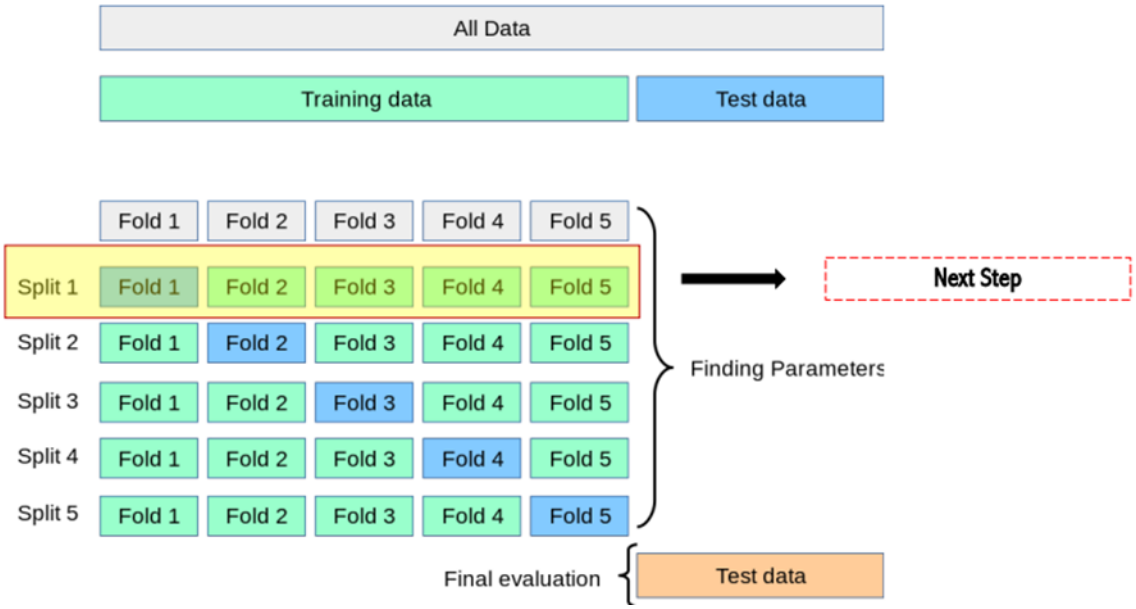
2. Độ phức tạp của cây (Tree complexity penalty):
   - Sau khi áp dụng giải pháp tỉa cành cây (Pruning solution), giải pháp sẽ cho ra kết quả SSR tương ứng với từng mô hình cây. Tuy nhiên, kết quả SSR không đủ để xác xác định mô hình cây nào phù hợp do càng tỉa cành cây kết quả SSR càng lớn. Do đó, việc áp dụng Tree complexity penalty được thực hiện như sau:
   - Giả sử $\alpha = 10.000$, tính giá trị Tree score tương ứng với các cây như sau:




Hình 12: Tree complexity score ứng với từng cây

Như vậy, cây có giá trị Tree score nhỏ nhất là 35.494 được sẽ được lựa chọn làm mô hình Decision Tree.

3. Cách tìm $\alpha$ Ở ví dụ trên đã set $\alpha$= 10.000 nhằm mục đích trực quan hóa cách áp dụng phương pháp Tree complexity penalty. Ở mục này, việc xác định $\alpha$ sẽ được thực hiện theo các bước như sau:
   - Bước 1: Đặt $\alpha$= 0 tương ứng với cây ban đầu
   - Bước 2: Lần lượt tăng $\alpha$ từ 0 cho tới một điểm giá trị $\alpha$i mà tại đó giá trị Tree score của cây ban đầu cao hơn cây tiếp theo
   - Bước 3: Tăng từ giá trị $\alpha$ tới giá trị $\alpha$i1,$\alpha$i2,$\alpha$i3,.. mà tại đó các giá trị Tree score của các cây tiếp theo cao hơn cây liền kề.
   - Bước 4: Tập hợp các giá trị $\alpha$ đã tính tại bước 2, bước 3
   - Bước 5: Áp dụng phương pháp Cross validation như sau:
     – Xáo trộn dữ liệu trong tập training set, chia dữ liệu thành các k-fold dữ liệu, và các k-splits. Mỗi split sẽ bao gồm: 1 fold testing (các fold testing sẽ được thay đổi lần lượt theo từng split) và k-1 fold training set.




Hình 13: Phương pháp Cross validation

   – Với mỗi Split, trên các tập k-1 fold, xác định mô hình Decision Tree tương ứng với từng $\alpha$ như đã tập hợp tại Bước 4. Với từng mô hình cây, tính giá trị các Tree score trên bộ fold testing theo từng $\alpha$
   – Tập hợp các giá trị Tree Score của từng Split. Tính giá trị trung bình theo từng $\alpha$ chọn $\alpha$ với giá trị trung bình nhỏ nhất

| | 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.

Hình 14: Lựa chọn $\alpha$