**Bài toán định giá bất động sản:**

Để định giá được bất động sản, ta cần phải phân tích được các yếu tố liên quan đến bài toán. Câu hỏi là: “Giá trị của bất động sản sẽ phụ thuộc vào các đặc trưng nào?”

Các đặc trưng ảnh hưởng như:

* Vị trí.
* Diện tích.
* Mặt tiền.
* Độ rộng đường vào.
* Hướng nhà.
* Số phòng.
* Số toilet.
* Số tầng.

**1. Thu thập dữ liệu:**

Nơi thu thập dữ liệu: batdongsan.com.vn

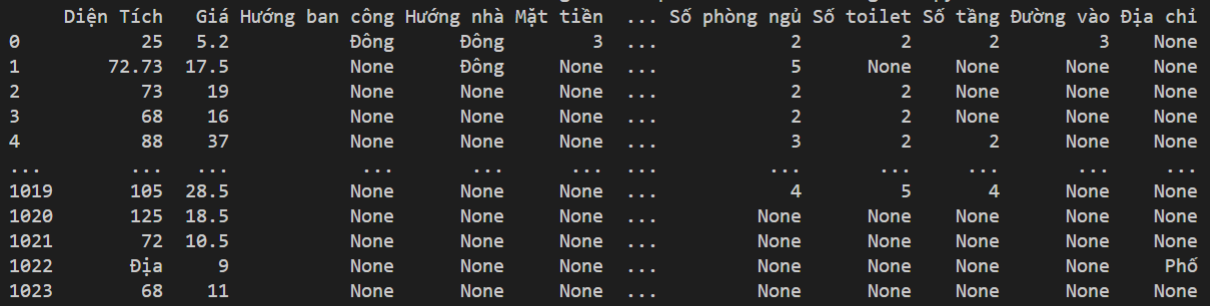
Thu thập từ trang batdongsan.com.vn:

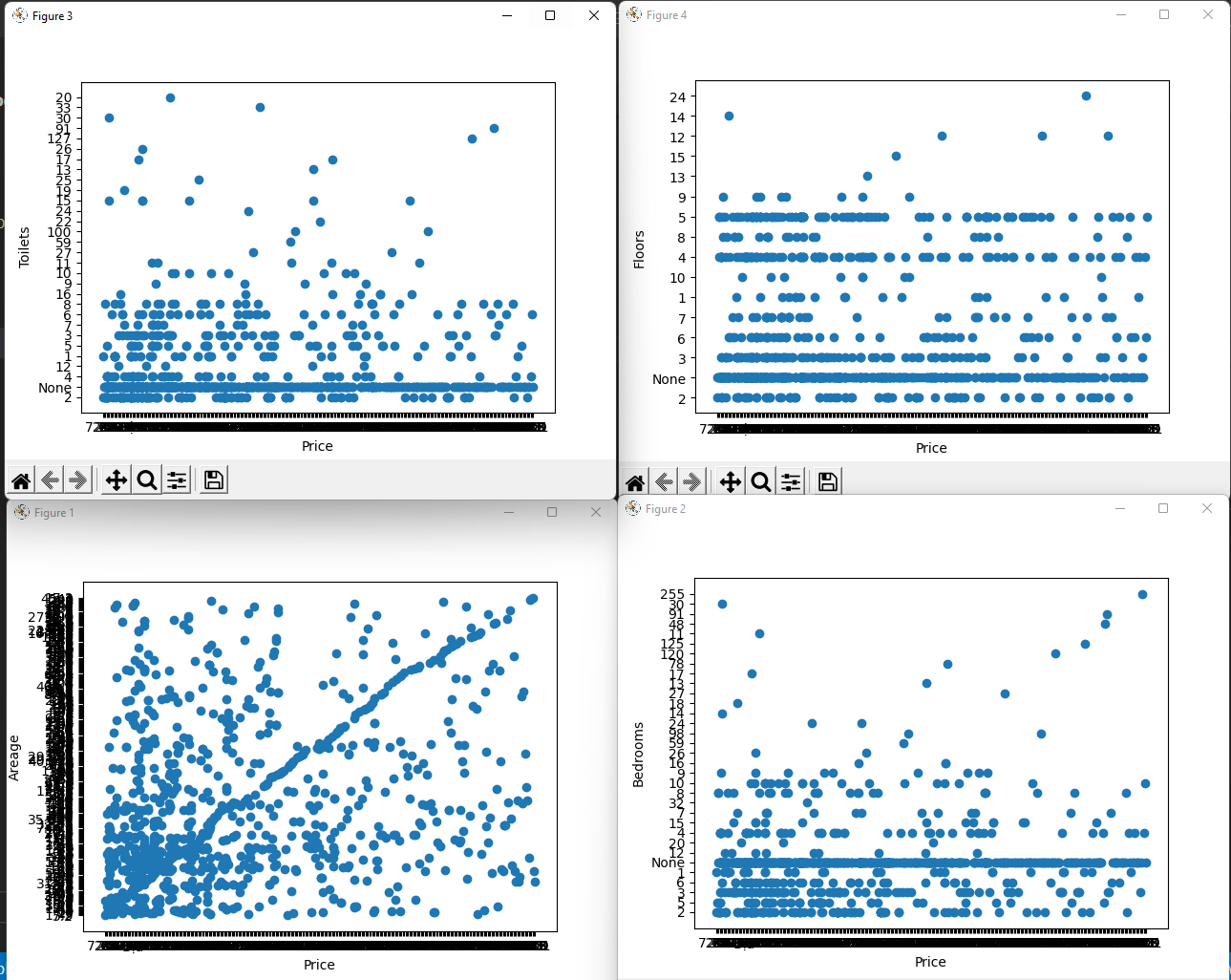
* Trang batdongsan.com.vn không có API, không thể lấy dữ liệu bằng cách gọi API.
* Thay vào đó phân tích cấu trúc html của website, tách từng thông tin để có được dữ liệu.
* Trang batdongsan.com.vn có bảo mật khá tốt, họ có thể chặn các request từ nhiều thư viện. Thư viện có thể dùng được là selenium.

**2. Quan sát, đánh giá dữ liệu:**

Quan sát, đánh giá dữ liệu là giai đoạn quan trọng nhằm hiểu được dữ liệu ta đang có để đưa ra các quyết định về lược bỏ dữ liệu, chọn model phù hợp để áp dụng, ...

Tổng số dữ liệu thu thập được là 1600 dòng. Có rất nhiều dữ liệu bị khuyết, dữ liệu lỗi.



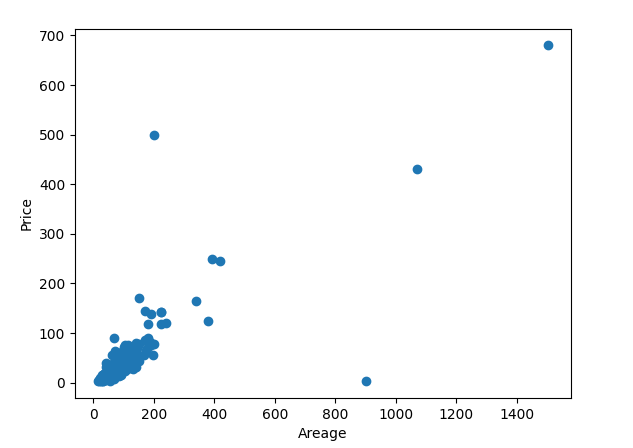
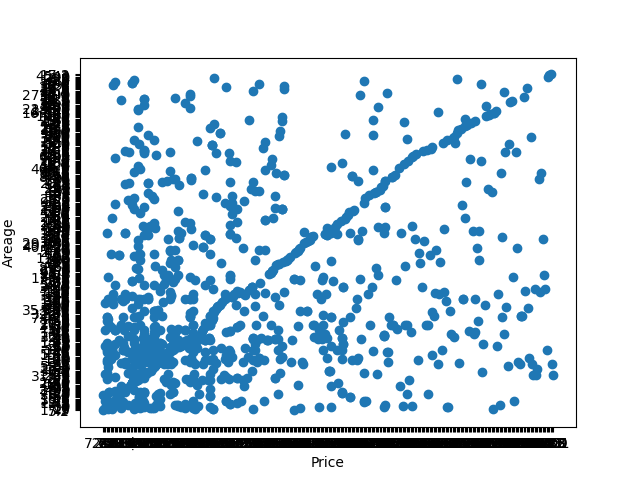


**3. Tiền xử lý dữ liệu:**

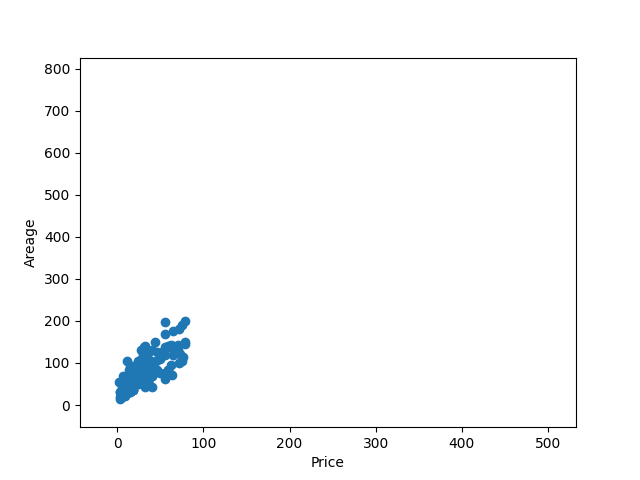
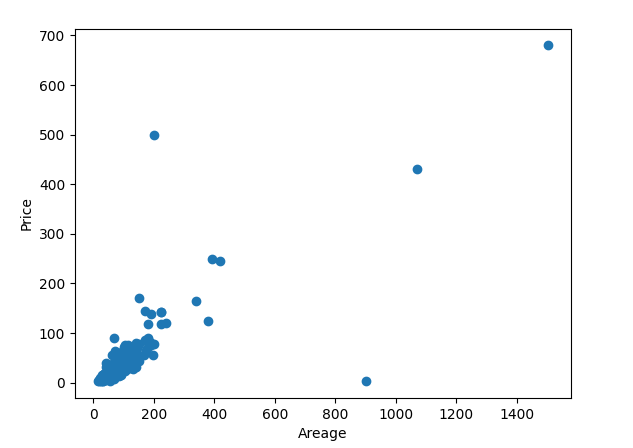
Tiền xử lý dữ liệu là giai đoạn rất quan trọng để giải quyết các bài toán machine learning. Khi thu thập dữ liệu, khả năng cao các dữ liệu sẽ không được đầy đủ, nhiều thuộc tính sẽ bị thiếu đi. Đặc biệt là khi lấy dữ liệu từ trang đăng tin mua bán, mỗi cá nhân sẽ có cách viết và cung cấp thông tin bất động sản theo cách khác nhau, việc này làm cho dữ liệu thu thập được không đồng nhất.

Các vấn đề đã gặp và kỹ thuật tiền xử lý dữ liệu:

* Dữ liệu bị khuyết: Các ô dữ liệu có giá trị NULL là các ô bị khuyết, có các phương pháp để xử lý vấn đề này:
* Giữ nguyên dữ liệu bị khuyết và sử dụng model có khả năng hiểu và xử lý phần khuyết.
* Xóa bớt dữ liệu bị khuyết, hoặc xóa bớt đặc trưng của dữ liệu.



* Xóa đi các dòng dữ liệu nhiễu.



* Điền dữ liệu bị khuyết bằng giá trị trung bình hoặc bằng số giá trị có tần suất xuất hiện lớn nhất.

**Ví dụ: ở dòng dữ liệu 1, số nhà vệ sinh là NULL**, 3 dòng còn lại lần lượt là 3, 2, 2 nhà vệ sinh. Ta chọn 1 trong 2 cách sau:

Điền dữ liệu khuyết bằng giá trị trung bình: [(3 + 3 + 2) / 3] = 2

Điền dữ liệu khuyết bằng tầng suất xuất hiện:

Giá trị 2 xuất hiện 2 lần. **(Chọn)**

Giá trị 3 xuất hiện 1 lần.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| STT | Giá | Diện Tích | Phòng Ngủ | Nhà Vệ Sinh | Tầng |
| 1 | 10 | 126 | NULL | NULL | 1 |
| 2 | 15 | 152 | 2 | 3 | 1 |
| 3 | 17 | 116 | 1 | 2 | 3 |
| 4 | 11 | 133 | 3 | 2 | 1 |

Sau khi sử lí, bảng sẽ thành:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| STT | Giá | Diện Tích | Phòng Ngủ | Nhà Vệ Sinh | Tầng |
| 1 | 10 | 126 | 2 | 2 | 1 |
| 2 | 15 | 152 | 2 | 3 | 1 |
| 3 | 17 | 116 | 1 | 2 | 3 |
| 4 | 11 | 133 | 3 | 2 | 1 |

* Dữ liệu phân bố rời rạc:
  + Lấy lại mẫu.
  + Áp dụng thuật toán phân cụm vào dữ liệu.

**4. Các thuật toán áp dụng:**

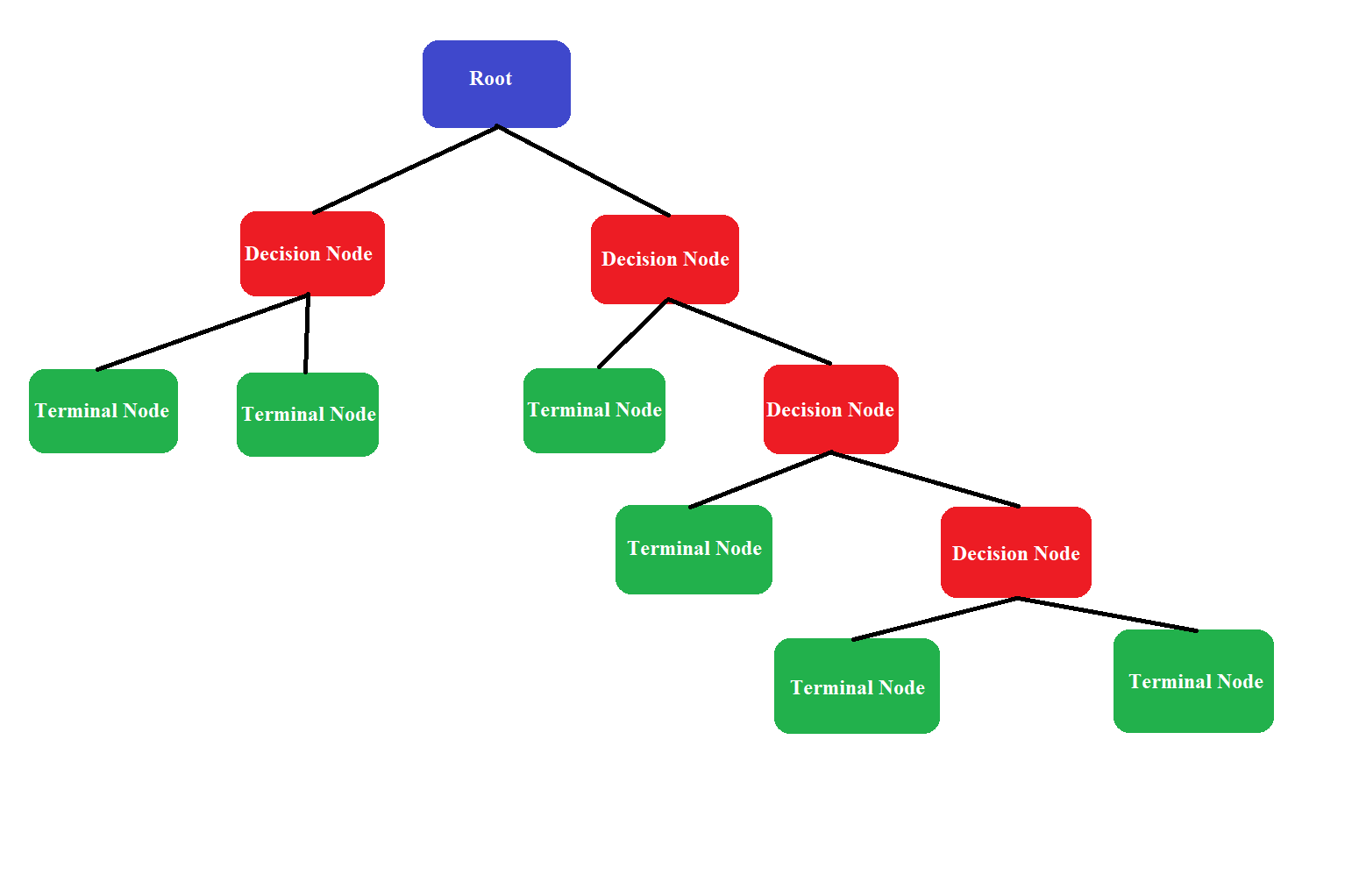
**4.1 Cây hồi quy (Regression Tree):** Cây hồi quy được dựa trên cây quyết định (decision tree).

**a/ Cây hồi quy là cây mà:**

* Là cây nhị phân.
* Root (gốc): là nút đại diện cho mẫu, là nút không có nút cha.
* Decision node (nút quyết định): là nút có nút con, hay nút được tách ra thành nút con từ hành động “splitting”.
* Leaf / Terminal node (lá, nút cuối): là nút không có nút con, nút lá, hay nút này ta không “splitting”.
* Lá con bên trái là “Đúng”, lá con bên phải là “Sai”.

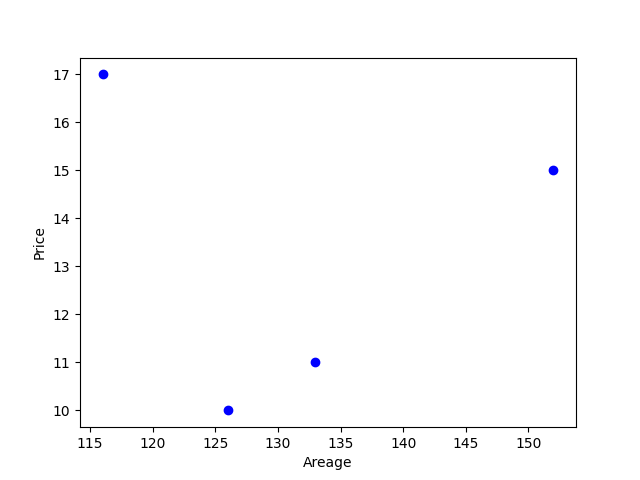
**b/ Các thao tác với cây hồi quy:**

* Splitting (tách, chẻ): là hành động khi chia 1 nút thành nhiều nút con.
* Pruning (cắt tỉa): là hành động khi ta xóa đi nút con của nút quyết định, ngược với hành động “splitting”.



Hình minh họa cây hồi quy

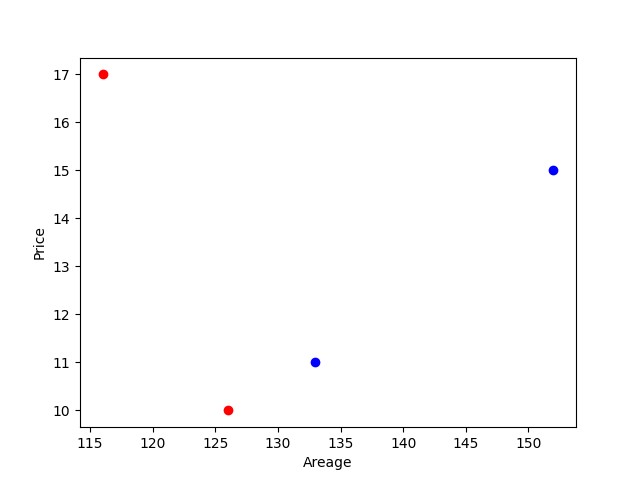
**c/ Dựng cây hồi quy:**

 Ví dụ ta có dữ liệu như sau:

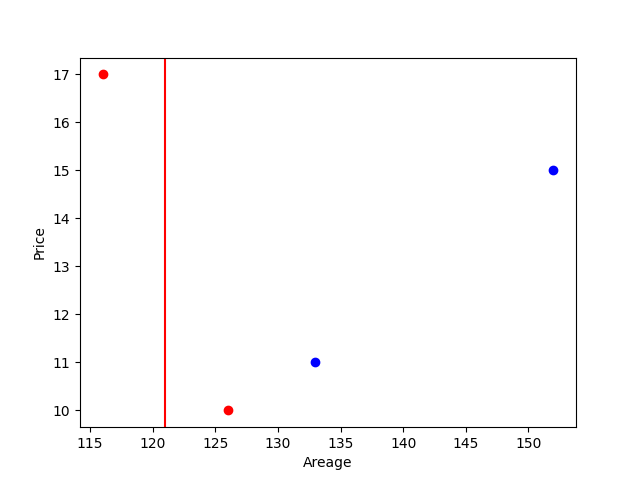
|  |  |
| --- | --- |
| Giá | Diện Tích |
| 10 | 126 |
| 15 | 152 |
| 17 | 116 |
| 11 | 133 |

**Ta sẽ phải tạo root cho cây (hay tìm ứng cử viên cho splitting), nhưng bằng cách nào?**

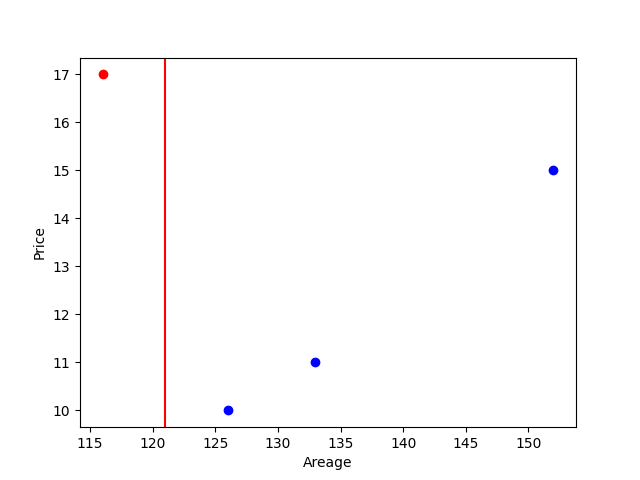
Ban đầu ta có 1 tập gồm 4 điểm (giá, diện tích) là (10;126) (11;133) (15;152) (17;116) đã được sắp xếp theo diện tích.



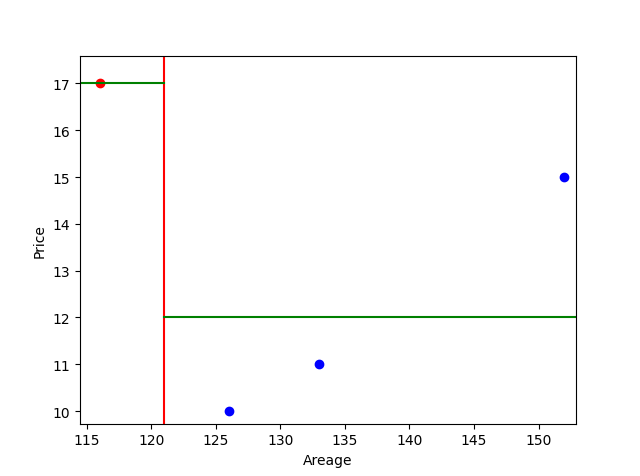
Ta chọn 2 điểm đầu tiên là (17;116) (10;126)



Trung bình diện tích của 2 điểm này là (116 + 126) / 2 = 121 m2



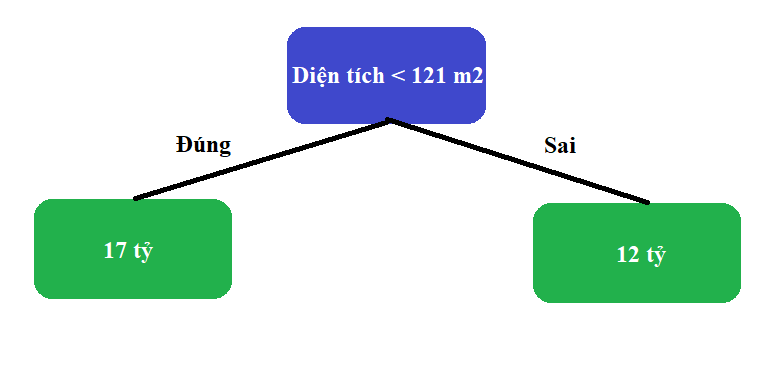
Chia 4 điểm thành 2 tập là đỏ S1 và xanh S2, tách nhau bởi đường thẳng x = 121



Trung bình “Giá” của tập S1 = 17/1 = 17 tỷ.

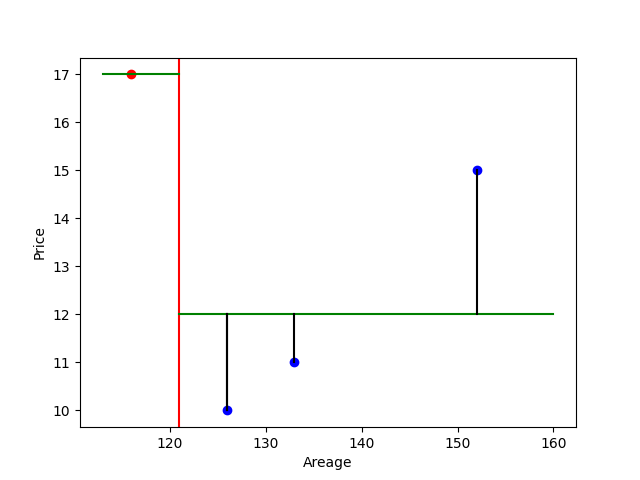
Trung bình “Giá” của tập S2 = (10+11+15)/3 = 12 tỷ.

Lúc này ta có cây là:



Nhưng đây liệu có là “root” tốt? Để tìm được root tốt, ta tính giá trị “residual”. Residual là khoảng cách thẳng đứng từ điểm đến đường hồi quy (đường màu xanh lá trên đồ thị).

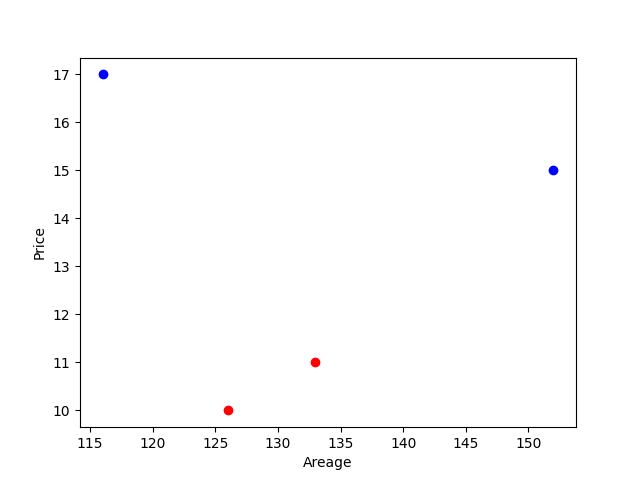
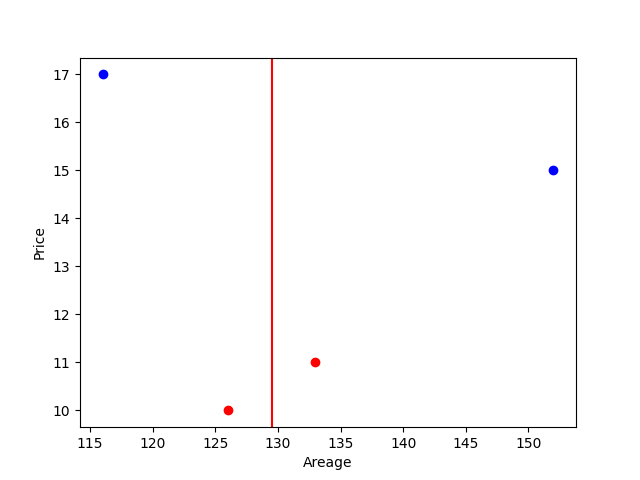
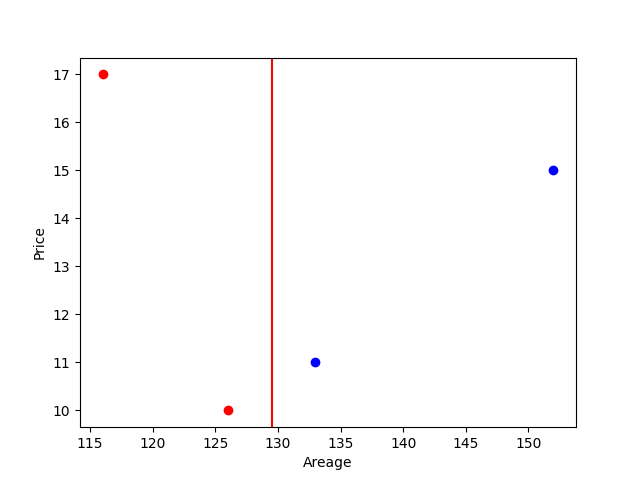
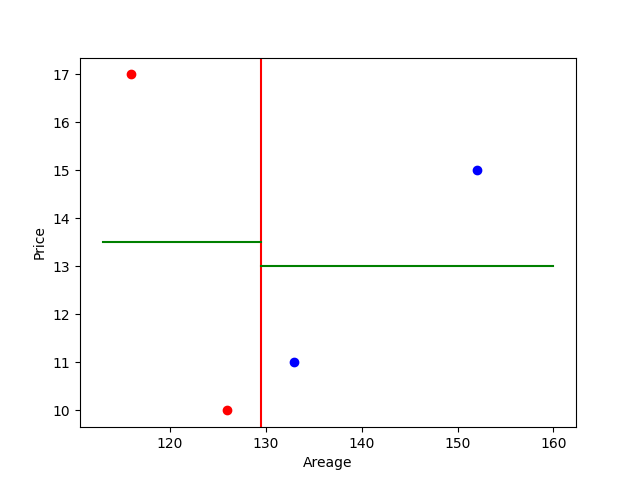
Ta tính tổng các residual khi chọn root là trung bình diện tích của 2 điểm đầu tiên.

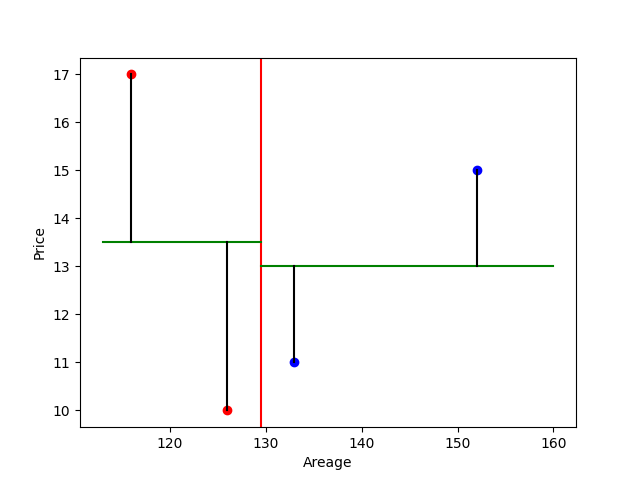


Tổng residual = (17 – 17)2 + (10 – 12)2 + (11 – 12)2 + (15 – 12)2 = 4 + 1 + 9 = 14

Tương tự, ta lần lượt chọn các cặp điểm tiếp theo, lần lượt chia thành 2 tập đỏ và xanh, vẽ đường hồi quy, tính tổng residual. Cặp nào có tổng residual bé nhất thì trung bình diện tích của cặp đó sẽ là root.

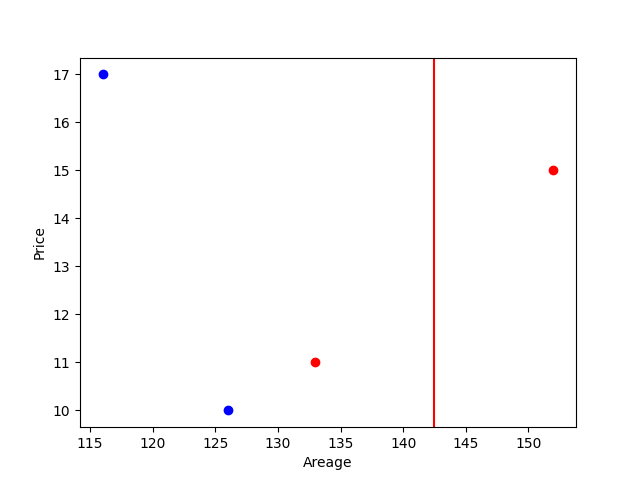
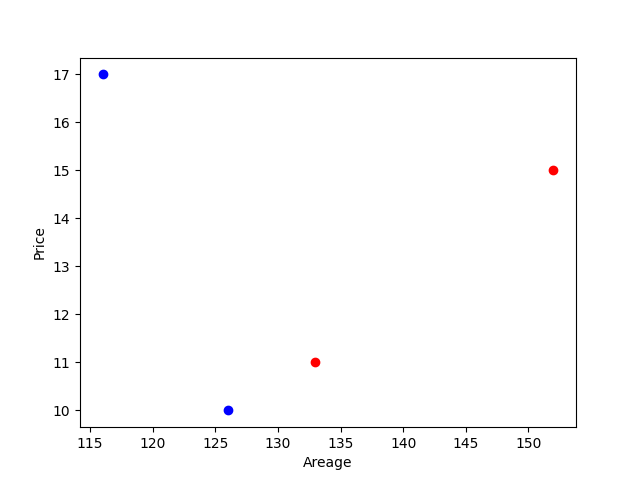
Cặp điểm tiếp theo (10,126) (11,133)

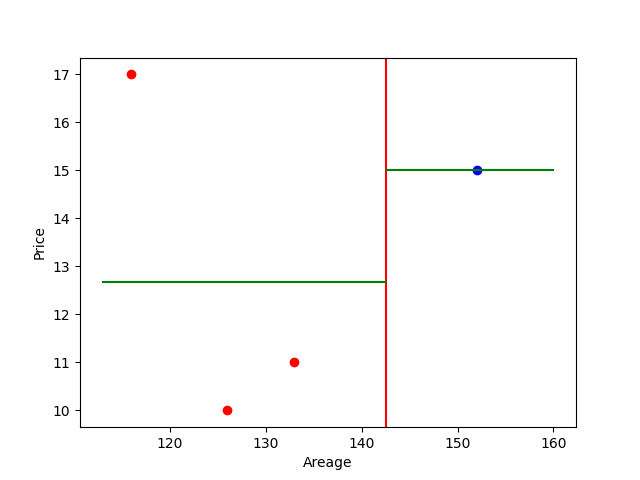
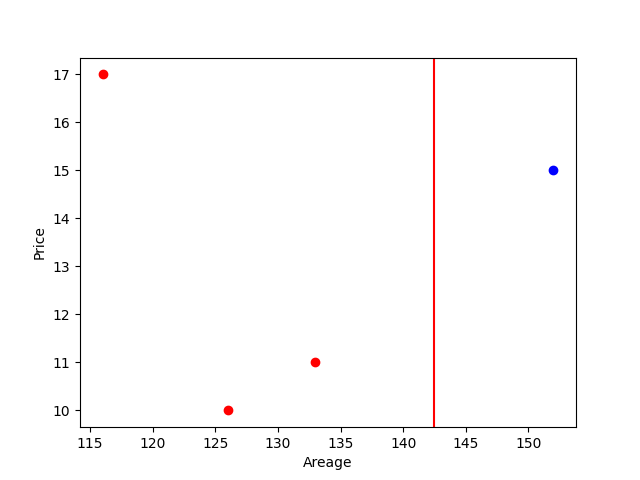
   

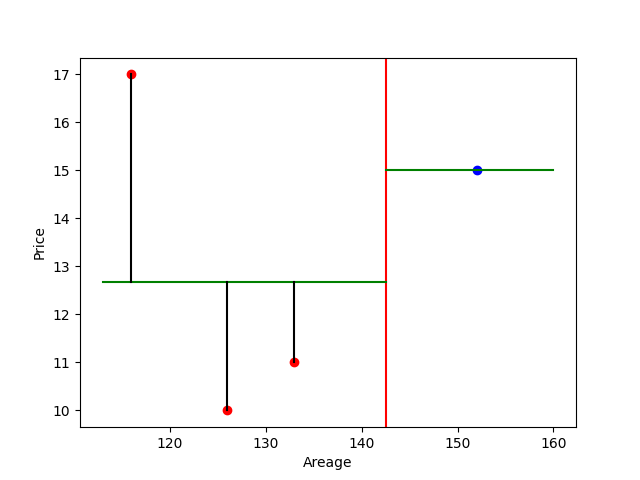


Tổng residual = (17 – 13.5)2 + (10 – 13.5)2 + (11 – 13)2 + (15 – 13)2 = 12.25 + 12.25 + 4 + 4 = **32.5**

Cặp tiếp theo (11, 133) (15, 152)

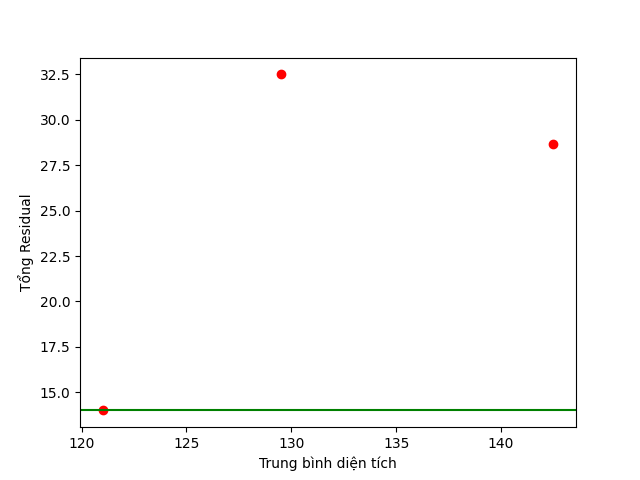




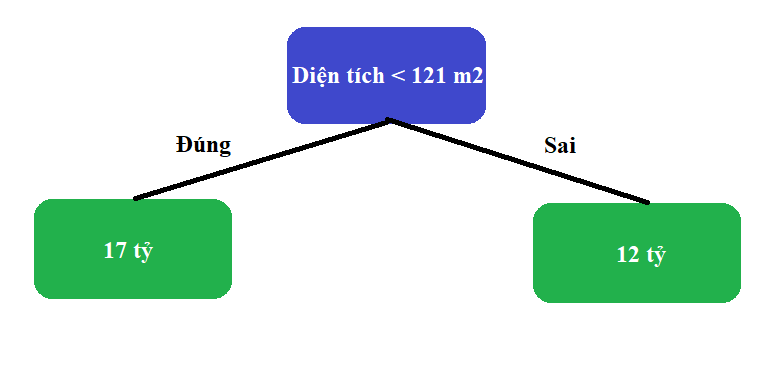


Tổng residual = (17 – 12.67)2 + (10 – 12.67)2 + (11 – 12.67)2 + (15 – 15)2 = 28.67

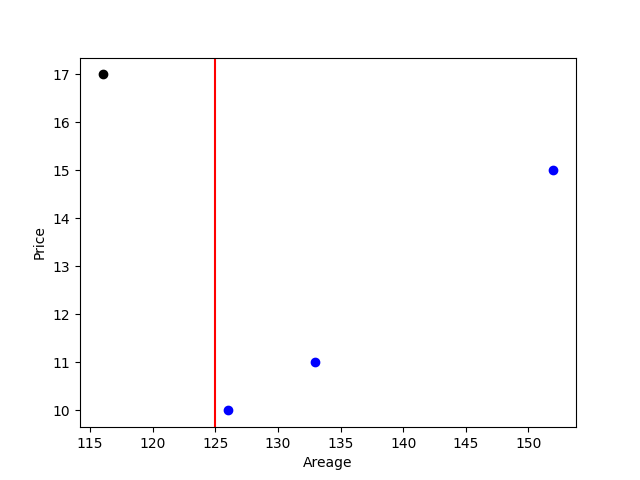
Từ 3 cặp điểm trên ta có:



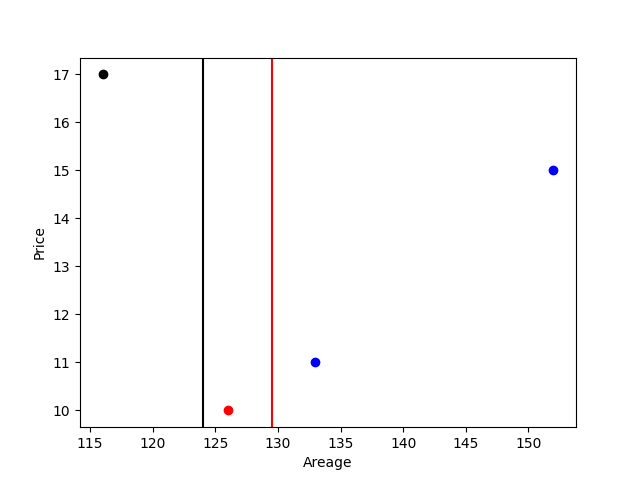
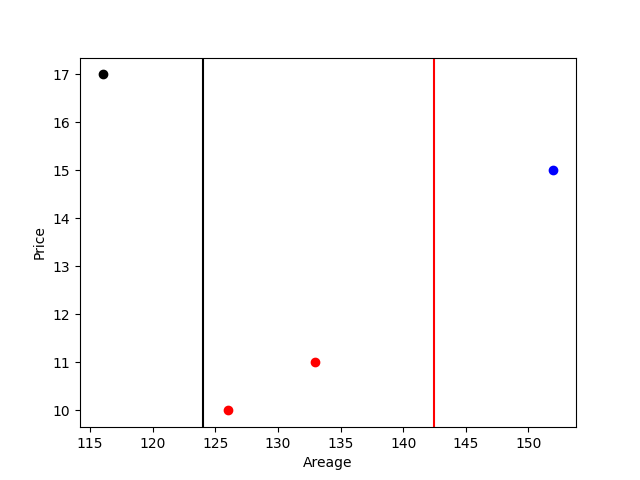
Trung bình diện tích của cặp điểm có tổng residual là 14 (bé nhất) sẽ là root (giống như kết quả ban đầu)



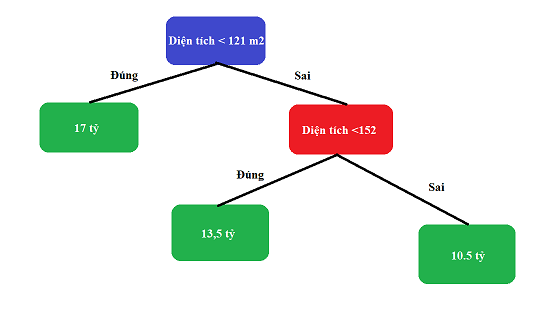
Kết quả ở đây ta chỉ mới có 3 node và với diện tích >= 121 m2, cây hiện tại cho ra kết quả 12 tỷ. Cây vẫn chưa “ổn”. Ta tiến hành “splitting” nút lá 12 tỷ.



Tập điểm với diện tích >= 121m2 gồm 3 điểm màu xanh. Ta sẽ tiến hành chọn lần lượt từng cặp điểm màu xanh rồi tính trung bình diện tích, chia 2 tập đỏ và xanh, tính tổng residual như trên.

Ta được cây là:

****

Ta có thể split các nút, mỗi lần split sẽ tăng 1 đơn vị chiều cao của cây. Split quá ít làm cây không hoàn thiện, nhưng split quá nhiều cũng không tốt, có thể gây ra overfitting. Nên việc lựa chọn độ cao phù hợp cho cây rất quan trọng.

Một trong những phương pháp hạn chế split: khi tập dữ liệu được chia đủ nhỏ, ta ngưng không cho cây split nữa.

Với bộ dữ liệu có nhiều đặc tính hơn, không chỉ mỗi diện tích:

* Với mỗi đặc tính, tìm root/ứng cử viên tốt nhất.
* Sắp xếp các ứng cử viên và split cây theo tổng residual của root/ứng cử viên tốt nhất giữa các đặc tính.

**Hồi quy tuyến tính (Linear Regression)**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| STT | Giá | Diện Tích | Phòng Ngủ | Nhà Vệ Sinh | Tầng |
| 1 | 10 | 126 | 2 | 2 | 1 |
| 2 | 15 | 152 | 2 | 3 | 1 |
| 3 | 17 | 116 | 1 | 2 | 3 |
| 4 | 11 | 133 | 3 | 2 | 1 |

**Giới thiệu:**

***Ký hiệu:***

* Biến phụ thuộc, giá trị thực: Giá (y).
* Biến độc lập, đặc trưng: Diện tích (x1), Phòng Ngủ (x2), Nhà Vệ Sinh (x3), Tầng (x4).
* Kết quả dự đoán của mô hình: y’

***Mục tiêu:***

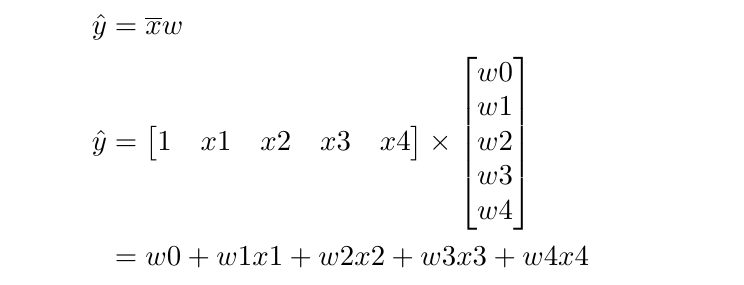
* Kết quả dự đoán gần nhất với giá trị thực: y ≈ y’

***Phương trình hồi quy tuyến tính:***

y' = f(x) = w1x1 + w2x2 + w3x3 + w4x4 + w0

- Vì hàm số f(x) có dạng đường thẳng, nên được gọi là tuyến tính.

- **w** = {w0, w1, w2, w3, w4} là các hằng số, w0 gọi là bias. Là các số cần phải tìm.



**Bài toán:** Tìm **w** = {w0, w1, w2, w3, w4} sao cho sai số giữa y’ và y gần 0 nhất:



Để giá trị không âm, ta có 2 cách:

* Lấy giá trị tuyệt đối.
* Bình phương. (Ta chọn cách này, vì phương trình bậc 2 luôn có đạo hàm).



Vì bước sau sẽ tiến hành đạo hàm nên ta nhân thêm ½ để kết quả đẹp hơn.

