**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**  
 KHOA VIỄN THÔNG I



**BÁO CÁO BÀI TẬP LỚN**

**BỘ MÔN CẤU TRÚC DỮ LIỆU VÀ GIẢI THUẬT**

**Giảng viên giảng dạy: Nguyễn Minh Tuấn**

**Nhóm sinh viên thực hiện:**

**Nguyễn Thị Thu Trang - B19DCVT405**

**Nguyễn Hồng Đức - B19DCVT096**

**Trần Thành Trung - B19DCVT421**

**Bùi Trung Đức - B19DCVT090**

**Nguyễn Văn Nguyên – B19DCVT277**

**Lớp: D19-286**

*Hà Nội, ngày 18 tháng 4 năm 2022*

1. ***Mục tiêu***

***-*** Trình bày một cách xác định giá đất từ các dữ liệu đặc điểm của từng ngôi nhà dựa trên công cụ colab của google phát triên

- Giá trị của ngôi nhà mới (bài toán hồi quy).

- Công cụ: colab do google phát triển để phục vụ quá trình học tập và phát triển các mô hình machine learning và artificial intelligence.

-

1. ***Giới thiệu***

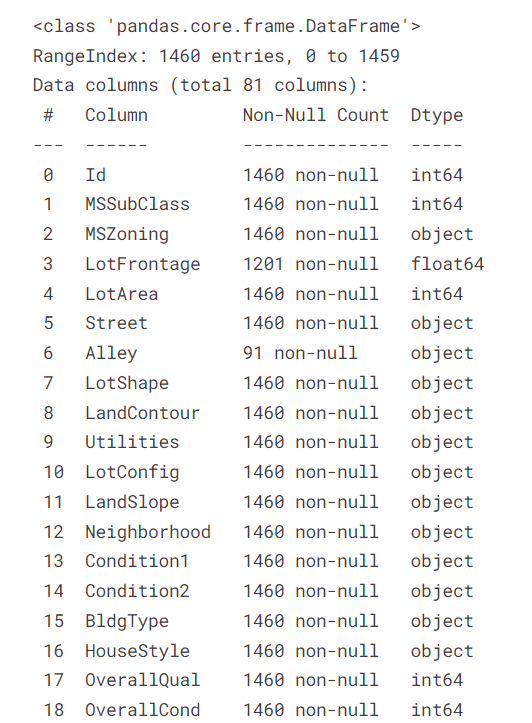
***I.Giới thiệu***

**1, Database**

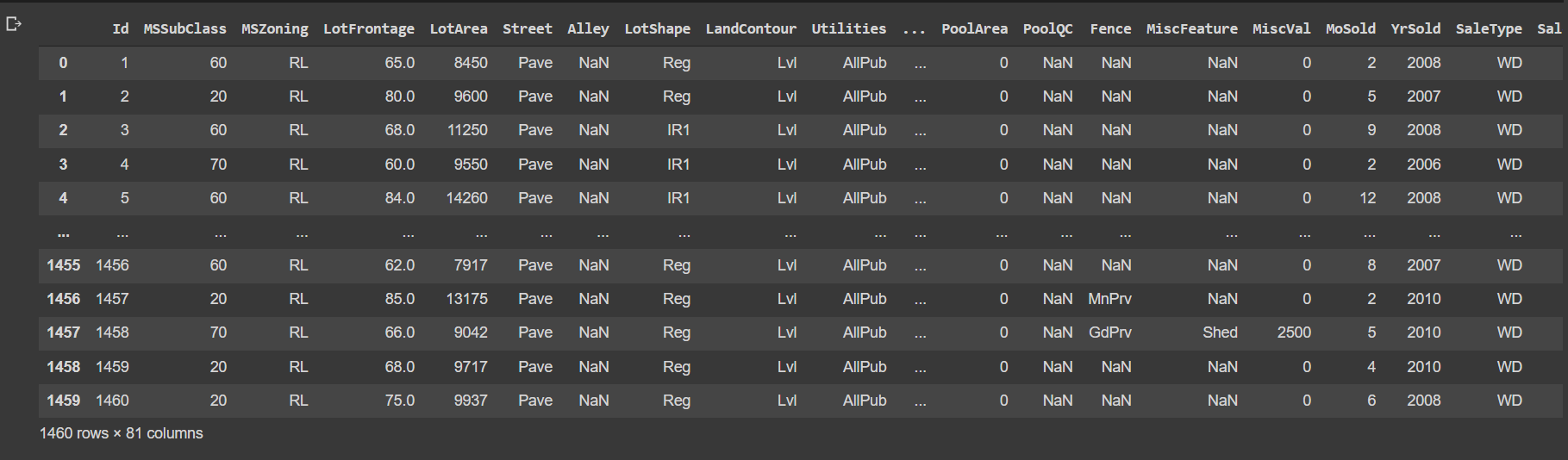
- Database: **House Prices - Advanced Regression Techniques by kaggle.com**

**-** data overview: Dữ liệu bao gồm các 81 đặc điểm của 1460 ngôi nhà

**2, Thông tinvề các đặc điểm**



**3 , Tổng quan về dữ liệu**



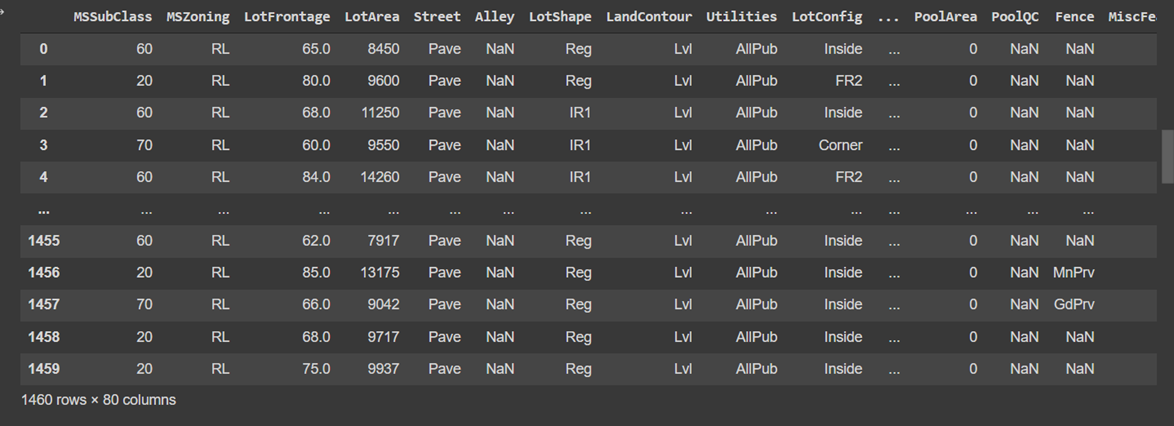
***II. Tiền xử lí tín hiệu***

- Xử lý dữ liệu bị khuyết(missing data), và các đặc điểm có độ tương quan cao.

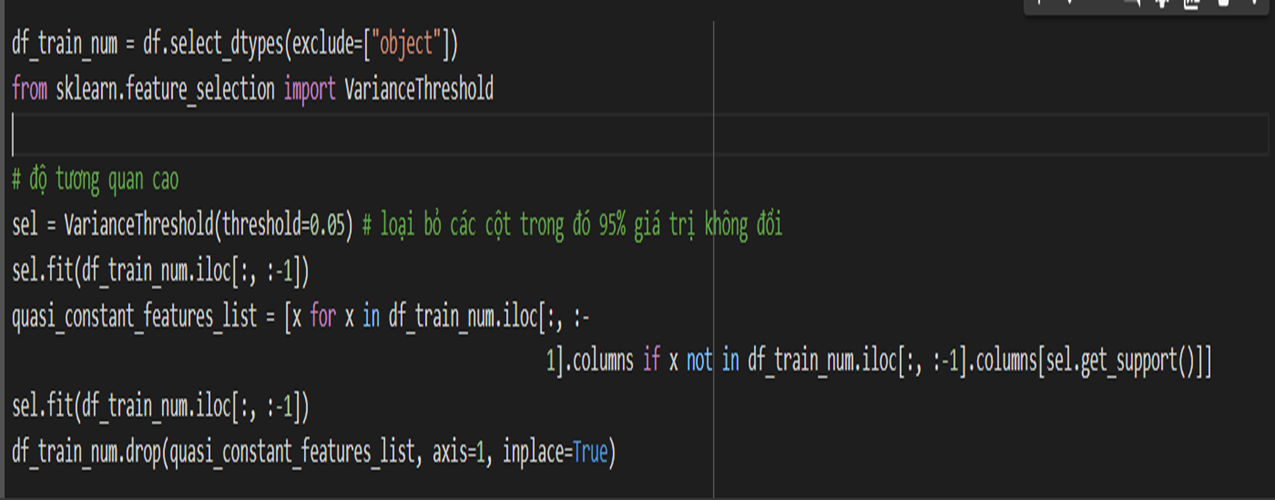
- Xử lý những column không cần thiết .

# Loại bỏ cột id

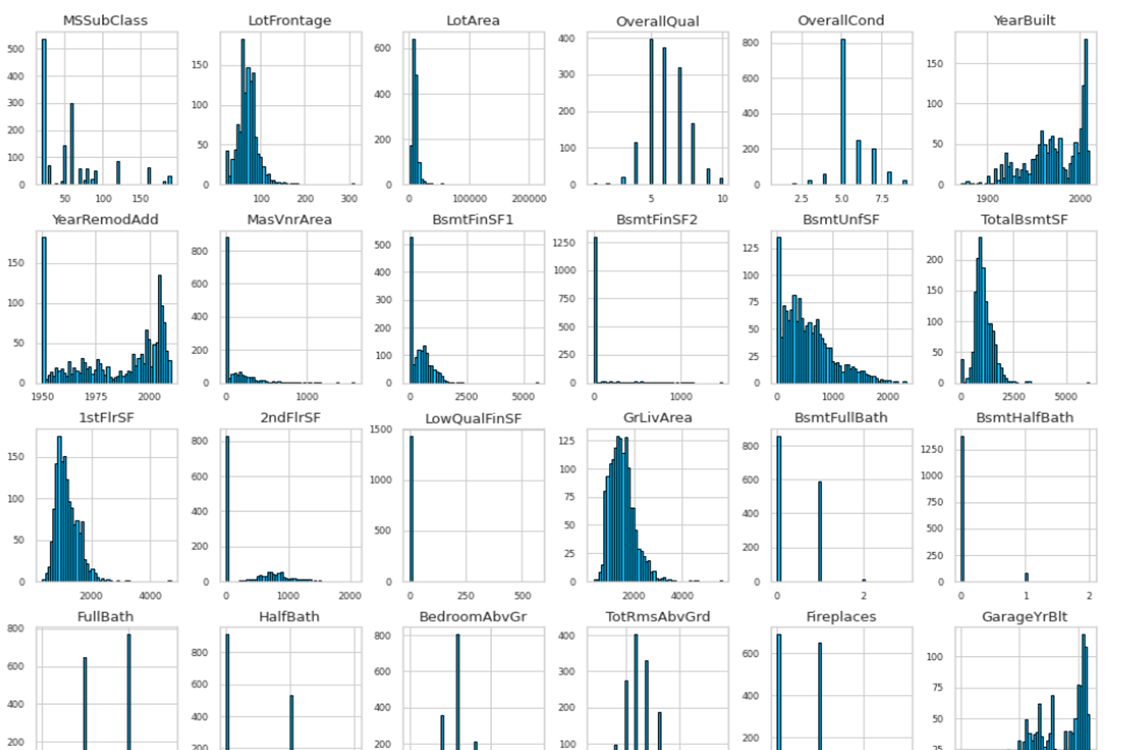
df.drop([“Id”], axis=1, inplace = True)

******

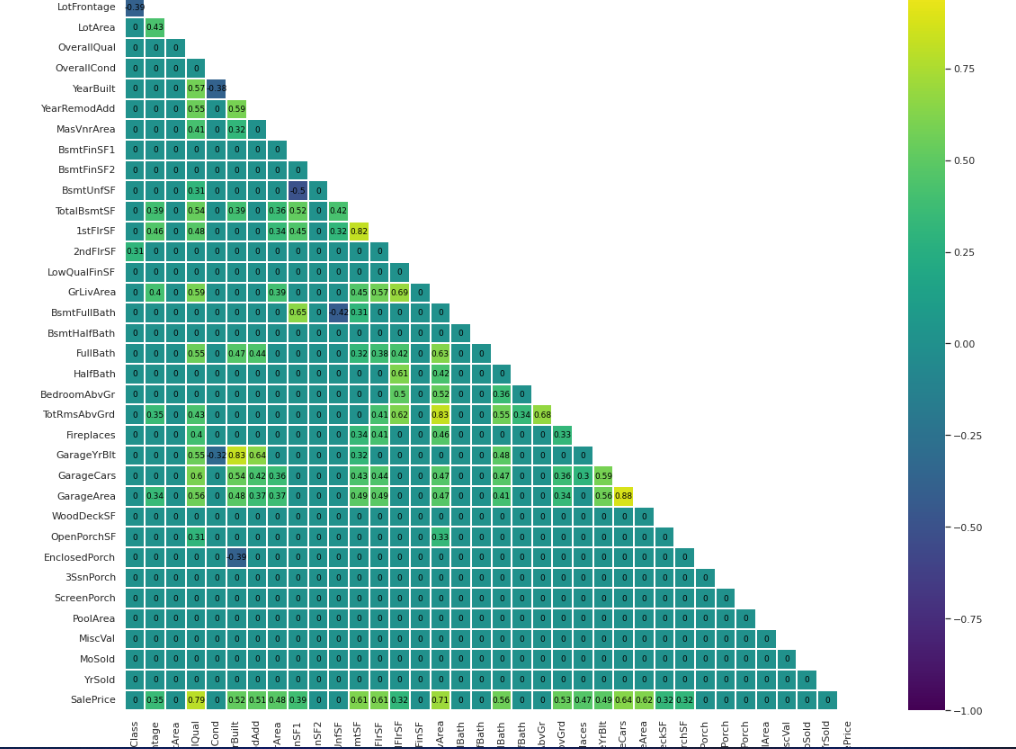
* Bỏ các tính năng gần như không đổi trong đó 95% giá trị là tương tự hoặc không đổi hay nói là chúng có độ tương quan cao .



Trực quan hóa các giá trị tuyệt đối



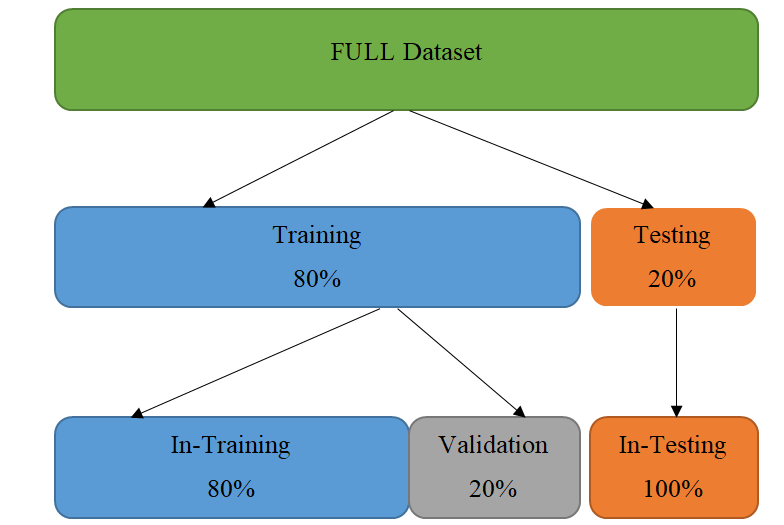
Xử dụng heatmap để đưa ra rõ hơn về độ tương quan của các đặc điểm với giá bán và tương quan giữa các đặc điểm với nhau .



**Feature selection :** lựa chọn tính năng

* **LotArea: Diện tích của cái căn nhà**
* **Yearbuilt: Năm xây nhà**
* **1stFlrSF: Diện tích tầng 1**
* **2stFlrSF: Diện tích tầng 2**
* **FullBath: Số phòng tắm**
* **BedroomAbvGr: Phòng ngủ đạt tiêu chuẩn**
* **TotRmsAbvGrd: Tổng số phòng đạt tiêu chuẩn**

**Tách tập dữ liệu**



**C , Method – Phương pháp**

* ***Decision Tree***
* ***Random Forest***

***I , Decision Tree***

Decision tree là cây nhị phân chia tách một cách đệ quy tập dữ liệu cho đến khi chúng ta chỉ còn các nút lá thuần túy. Nếu một mẫu dữ liệu thỏa mãn điều kiện tại một nút quyết định thì nó sẽ chuyển ***sang*** nút con bên trái, còn lại sang nút bên phải và cuối cùng đến nút lá nơi gán nhãn lớp cho nó.

1. **Classification Decision**

*Entropy* là thước đo lượng thông tin chứa trong một trạng thái. Entropy cao thì không chắc chắn về điểm được chọn ngẫu nhiên và chúng ta cần thêm bit để mô tả trạng thái bằng cách biểu thị xác suất của class thứ i.

Tìm mức tăng thông tin tương ứng với một phép tách, chúng ta cần trừ entropy kết hợp của các nút con khỏi entropy của nút cha

IG = E(parent) -

Mô hình so sánh mọi phân tách có thể và lấy cái nào tối đa hóa thông tin thu được để mô hình đi qua mọi tính năng có thể và giá trị tính năng để tìm kiếm tính năng tốt nhất và ngưỡng tương ứng.

Cây quyết định là một thuật toán tham lam, nó chọn tốt nhất hiện tại, tối đa hóa thông tin thu được, nó không quay lại và thay đổi phân tách trước đó. Vì vậy tất cả các phần tách sau sẽ phụ thuộc vào phần hiện tại và điều này không đảm bảo chúng ta có được bộ phân tách tối ưu nhất nhưng có sự tham lam tìm kiếm làm cho machine learning nhanh hơn nhiều.

Dùng GINI để tính toán mức tăng thông tin, chúng ta cần kiểm tra xem mức tăng thông tin hiện tại này có lớn hơn mức tăng thông tin tối đa hay không. Nếu lớn hơn thì chúng ta cần cập nhật phần tách tốt nhất của mình.

Gini Index = 1 - với pi = probability of class i

Thu thông tin: IG = G(parent) - (Trong một cây quyết định thì có nhiều cách phân tách, dùng IG để lựa chọn phương thức tối ưu nhất)

1. **Regressor Decision**

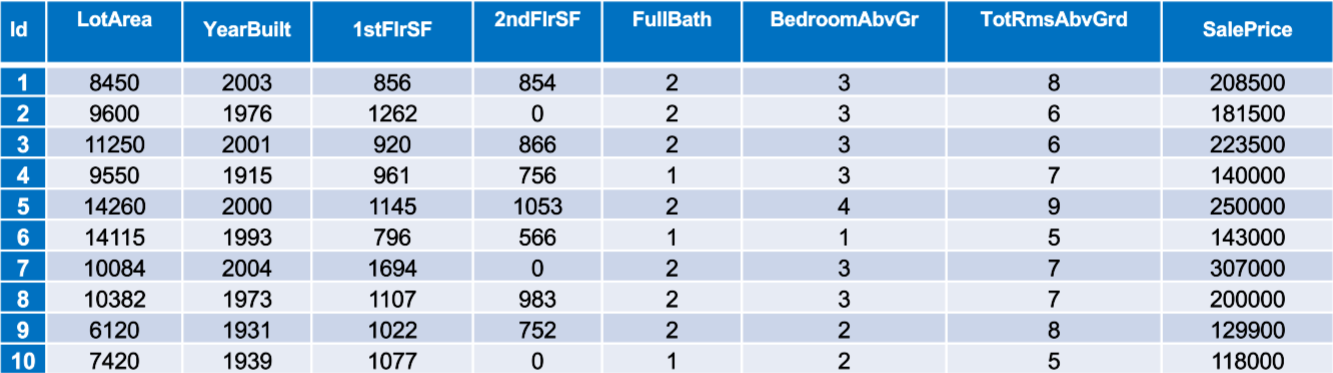
Trong hồi quy, ta sử dụng phương sai làm thước đo tạp chất giống như đã sử dụng chỉ số entropy hoặc gini trong bài toán phân loại.

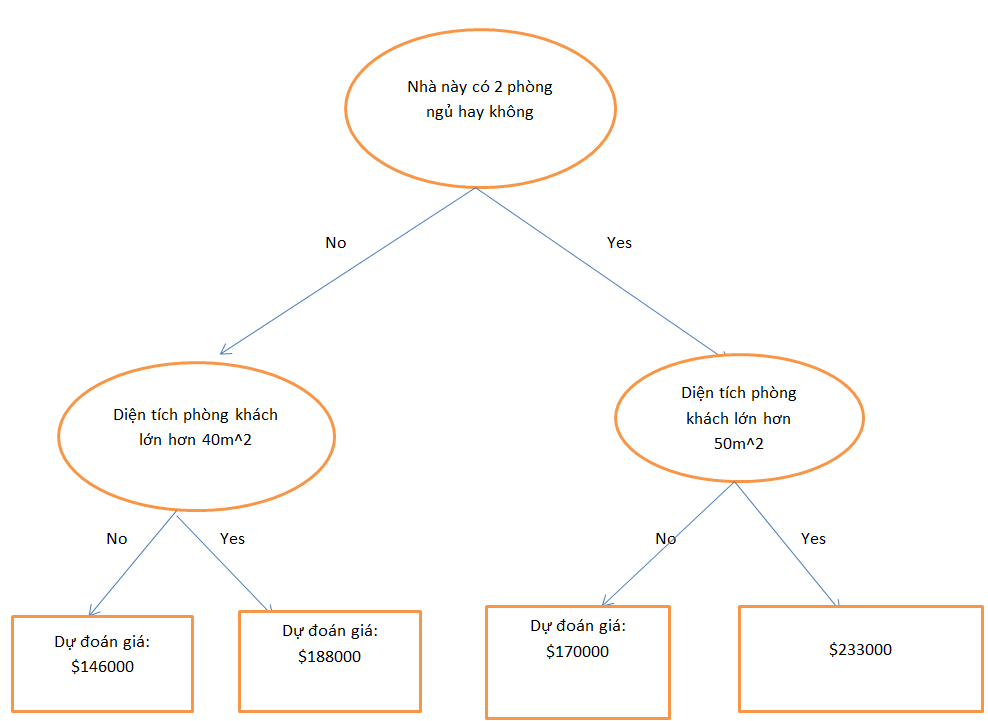
Var =

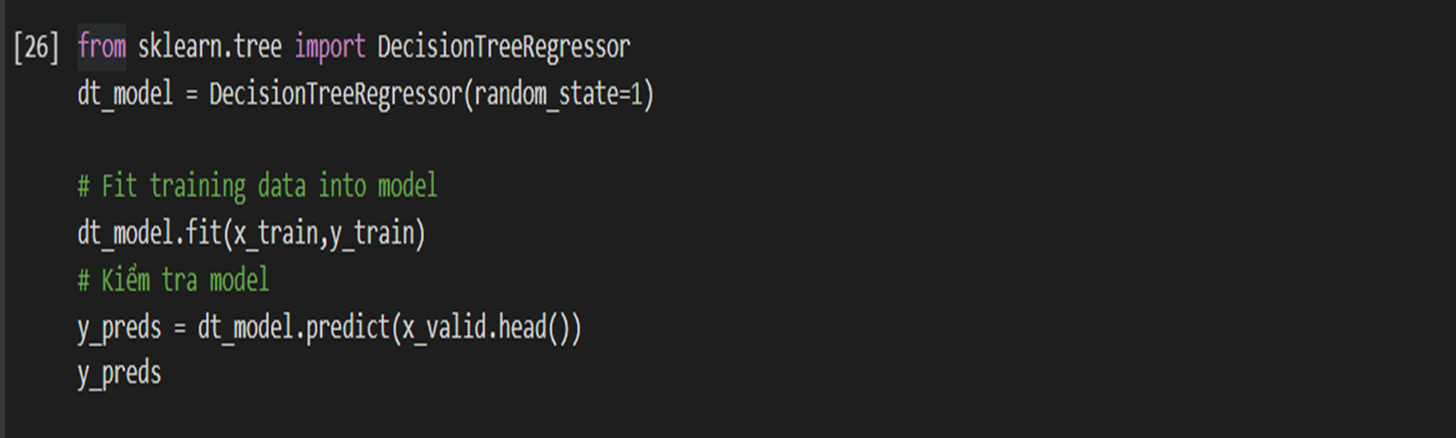
Phương sai cao hơn có nghĩa là tạp chất cao hơn.

Var Red = Var(parent) - (Varian reduction: độ giảm)

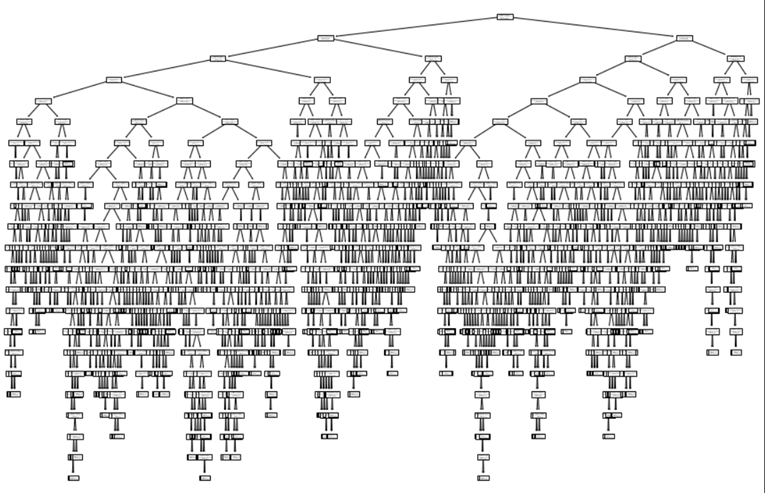
Trọng số chỉ là kích thước tương đối của nút con đối với nút cha. Dùng Var Red với mục đích tương tự IG.





****

**Thực tế bài toán**



**II , Random Forest**

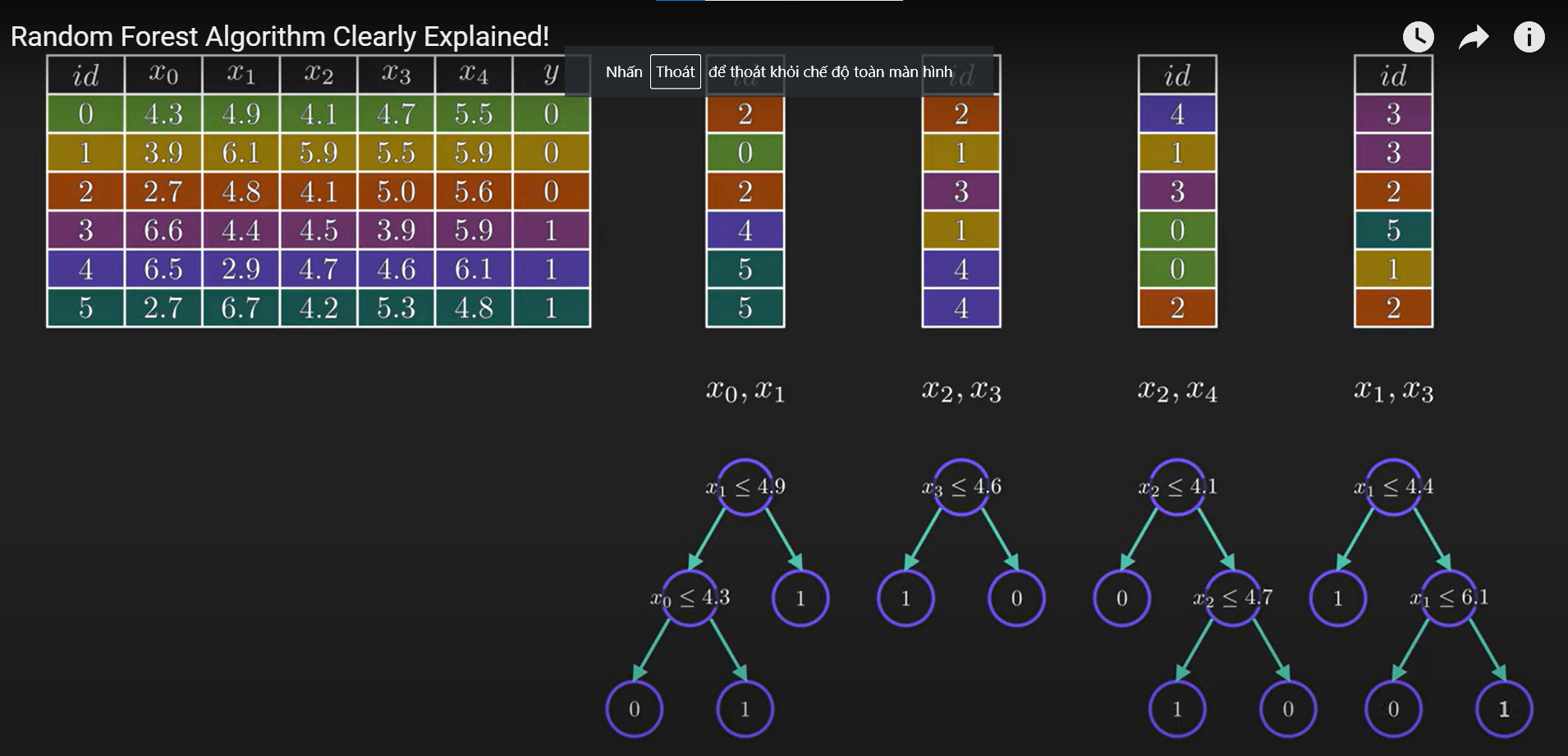
Là một tập hợp của nhiều cây quyết định ngẫu nhiên và nó ít nhạy cảm hơn nhiều với dữ liệu.

Bước đầu tiên là xây dựng bộ dữ liệu mới từ dữ liệu ban đầu. Để duy trì sự đơn giản thì chọn 4. Chọn ngẫu nhiên các hàng từ dữ liệu ban đầu để tạo tập dữ liệu mới của mình. Mọi tập dữ liệu sẽ chứa cùng số hàng với tập dữ liệu ban đầu.



Ví dụ có dữ liệu như hình và chọn ngẫu nhiên để tạo tập dữ liệu mới. Quá trình tạo dữ liệu mới được gọi là Bootstrapping. Bây giờ, ta sẽ đào tạo một cây quyết định trên từng bộ dữ liệu được khởi động 1 cách độc lập. Chọn ngẫu nhiên một tập hợp con các đặc điểm cho từng cây và chỉ sử dụng chúng để training.

Tiếp theo là xây dựng cây quyết định ngẫu nhiên



Ta đã có 1 forest với 4 cây.

Khi có một dữ liệu mới, ta chuyển điểm dữ liệu này qua từng cây một và ghi lại các dự đoán. Sau đó kết hợp tất cả các dự đoán, lấy đa số. Quá trình kết hợp các kết quả từ nhiều mô hình được gọi là tổng hợp.

*Vì sao gọi là ngẫu nhiên*? Vì đã sử dụng hai quy trình ngẫu nhiên, khởi động (bootstrapping) và lựa chọn tính năng (feature selection) ngẫu nhiên.

*Lý do việc khởi động và lựa chọn tính năng*? Bootstrapping đảm bảo rằng chúng ta không sử dụng cùng một dữ liệu cho mọi cây, vì vậy theo một cách nào đó, nó giúp mô hình của chúng ta ít nhạy cảm hơn với dữ liệu training ban đầu. Việc lựa chọn đặc điểm ngẫu nhiên giúp giảm bớt sự tương quan giữa các cây. Nếu sử dụng mọi tính năng thì hầu hết các cây sẽ có các nút quyết định giống nhau và sẽ hoạt động rất giống nhau. Điều đó làm tăng phương sai. Một số cây sẽ được đào tạo về những đặc điểm kém quan trọng hơn nên chúng sẽ đưa ra những dự đoán xấu nhưng cũng có những cây sẽ đưa ra những dự đoán xấu theo hướng ngược lại nên chúng sẽ cân bằng lại.

*Kích thước lý tưởng của tập hợp con tính năng là bao nhiêu*? Các nhà nghiên cứu nhận thấy rằng các giá trị gần với căn bậc 2 của tổng số đối tượng địa lý.

(2 đối tượng ~ căn 5)

Nhóm em thực hiện dự án dựa trên dữ liệu đầu vào để *dự đoán giá nhà* như sau: Ứng dụng thực tế xây dựng mô hình machine learning dựa trên thuật toán Decisiontree và Random Forest.

(Dưới đây chỉ là một đoạn dữ liệu đầu vào để dữ thực hiện xử lý demo Decision Tree và Random Forest)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Id | LotArea | YearBuilt | 1stFlrSF | 2ndFlrSF | FullBath | BedroomAbvGr | TotRmsAbvGrd | SalePrice |
| 1 | 8450 | 2003 | 856 | 854 | 2 | 3 | 8 | 208500 |
| 2 | 9600 | 1976 | 1262 | 0 | 2 | 3 | 6 | 181500 |
| 3 | 11250 | 2001 | 920 | 866 | 2 | 3 | 6 | 223500 |
| 4 | 9550 | 1915 | 961 | 756 | 1 | 3 | 7 | 140000 |
| 5 | 14260 | 2000 | 1145 | 1053 | 2 | 4 | 9 | 250000 |
| 6 | 14115 | 1993 | 796 | 566 | 1 | 1 | 5 | 143000 |
| 7 | 10084 | 2004 | 1694 | 0 | 2 | 3 | 7 | 307000 |
| 8 | 10382 | 1973 | 1107 | 983 | 2 | 3 | 7 | 200000 |
| 9 | 6120 | 1931 | 1022 | 752 | 2 | 2 | 8 | 129900 |
| 10 | 7420 | 1939 | 1077 | 0 | 1 | 2 | 5 | 118000 |

*Chọn dữ liệu ngẫu nhiên từ tập dữ liệu ban đầu*

Dữ liệu 1:

|  |  |  |  |
| --- | --- | --- | --- |
| Id | LotArea | YearBuilt | SalePrice |
| 1 | 8450 | 2003 | 208500 |
| 2 | 9600 | 1976 | 181500 |
| 1 | 8450 | 2003 | 208500 |
| 4 | 9550 | 1915 | 140000 |
| 5 | 14260 | 2000 | 250000 |
| 5 | 14260 | 2000 | 250000 |
| 6 | 10084 | 2004 | 143000 |
| 6 | 10084 | 2004 | 143000 |
| 9 | 6120 | 1931 | 129900 |
| 10 | 7420 | 1939 | 118000 |

= 177240

Var = => Var nút gốc = 2204258400

Với LA Với YB

|  |  |  |
| --- | --- | --- |
| Id | LotArea | SalePrice |
| 1 | 8450 | 208500 |
| 2 | 9600 | 181500 |
| 1 | 8450 | 208500 |
| 4 | 9550 | 140000 |
| 9 | 6120 | 129900 |
| 10 | 7420 | 118000 |

|  |  |  |
| --- | --- | --- |
| Id | YearBuilt | SalePrice |
| 2 | 1976 | 181500 |
| 4 | 1915 | 140000 |
| 9 | 1931 | 129900 |
| 10 | 1939 | 118000 |

nút trái = 164400 nút trái = 142350

Var nút trái = 1353433333 Var nút trái = 571542500

nút phải = 196500 nút phải = 200500

Var nút phải = 2862250000 Var nút phải = 1940166667

Var Red = Var(parent) -

Var Red 1 = 2204258400 - \*1353433333 - \*2862250000 = 247298400

Var Red 2 = 2204258400 - \*571542500 - \*1940166667 = 811541400

Var Red 2 > Var Red 1 => Chọn cách 2 do cách 2 làm giảm tạp chất hơn nhiều lần so với cách 1

**Ở đây chỉ so sánh 2 lần, trong thực tế, mô hình đánh giá phương sai giảm cho mỗi lần tách có thể và chọn tốt nhất. Quá trình này diễn ra 1 cách đệ quy trừ khi chúng ta đã đạt đến độ sâu mong muốn.**

* Xây dựng Decision Tree

Dữ liệu 2:

|  |  |  |  |
| --- | --- | --- | --- |
| Id | FullBath | YearBuilt | SalePrice |
| 7 | 2 | 2004 | 307000 |
| 2 | 2 | 1976 | 181500 |
| 1 | 2 | 2003 | 208500 |
| 9 | 2 | 1931 | 129900 |
| 4 | 1 | 1915 | 140000 |
| 3 | 2 | 2001 | 223500 |
| 3 | 2 | 2001 | 223500 |
| 10 | 1 | 1939 | 118000 |
| 2 | 2 | 1976 | 181500 |
| 1 | 2 | 2003 | 208500 |

Thực hiện xử lý giống như dữ liệu 1, ta có được Decision Tree như sau:

|  |  |  |  |
| --- | --- | --- | --- |
| Id | LotArea | BedroomAbvGr | SalePrice |
| 3 | 11250 | 3 | 223500 |
| 10 | 7420 | 2 | 118000 |
| 1 | 8450 | 3 | 208500 |
| 4 | 9550 | 3 | 140000 |
| 8 | 10382 | 3 | 200000 |
| 8 | 10328 | 3 | 200000 |
| 9 | 6120 | 2 | 129900 |
| 5 | 14260 | 4 | 250000 |
| 1 | 8450 | 3 | 208500 |
| 1 | 8450 | 3 | 208500 |

Dữ liệu 3:

Thực hiện xử lý giống như dữ liệu 1, ta có được Decision Tree như sau:

Dữ liệu 4:

|  |  |  |  |
| --- | --- | --- | --- |
| Id | FullBath | TotRmsAbvGrd | SalePrice |
| 2 | 2 | 6 | 181500 |
| 9 | 2 | 8 | 129900 |
| 3 | 2 | 6 | 223500 |
| 9 | 2 | 8 | 129900 |
| 5 | 2 | 9 | 250000 |
| 7 | 2 | 7 | 307000 |
| 5 | 2 | 9 | 250000 |
| 6 | 1 | 5 | 143000 |
| 1 | 2 | 8 | 208500 |
| 10 | 1 | 5 | 118000 |

Thực hiện xử lý giống như dữ liệu 1, ta có được Decision Tree như sau:

Dữ liệu 5:

|  |  |  |  |
| --- | --- | --- | --- |
| Id | 1stFlrSF | FullBath | SalePrice |
| 6 | 796 | 1 | 143000 |
| 7 | 1694 | 2 | 307000 |
| 5 | 1145 | 2 | 250000 |
| 4 | 961 | 1 | 140000 |
| 2 | 1262 | 2 | 181500 |
| 3 | 920 | 2 | 223500 |
| 9 | 1022 | 2 | 129900 |
| 1 | 856 | 2 | 208500 |
| 1 | 856 | 2 | 208500 |
| 1 | 856 | 2 | 208500 |

Thực hiện xử lý giống như dữ liệu 1, ta có được Decision Tree như sau:

|  |  |  |  |
| --- | --- | --- | --- |
| Id | 2ndFlrSF | FullBath | SalePrice |
| 4 | 756 | 1 | 140000 |
| 5 | 1053 | 2 | 250000 |
| 2 | 0 | 2 | 181500 |
| 9 | 752 | 2 | 129900 |
| 3 | 866 | 2 | 118000 |
| 2 | 0 | 2 | 181500 |
| 1 | 854 | 2 | 208500 |
| 1 | 854 | 2 | 208500 |
| 5 | 1053 | 2 | 250000 |
| 9 | 752 | 2 | 129900 |

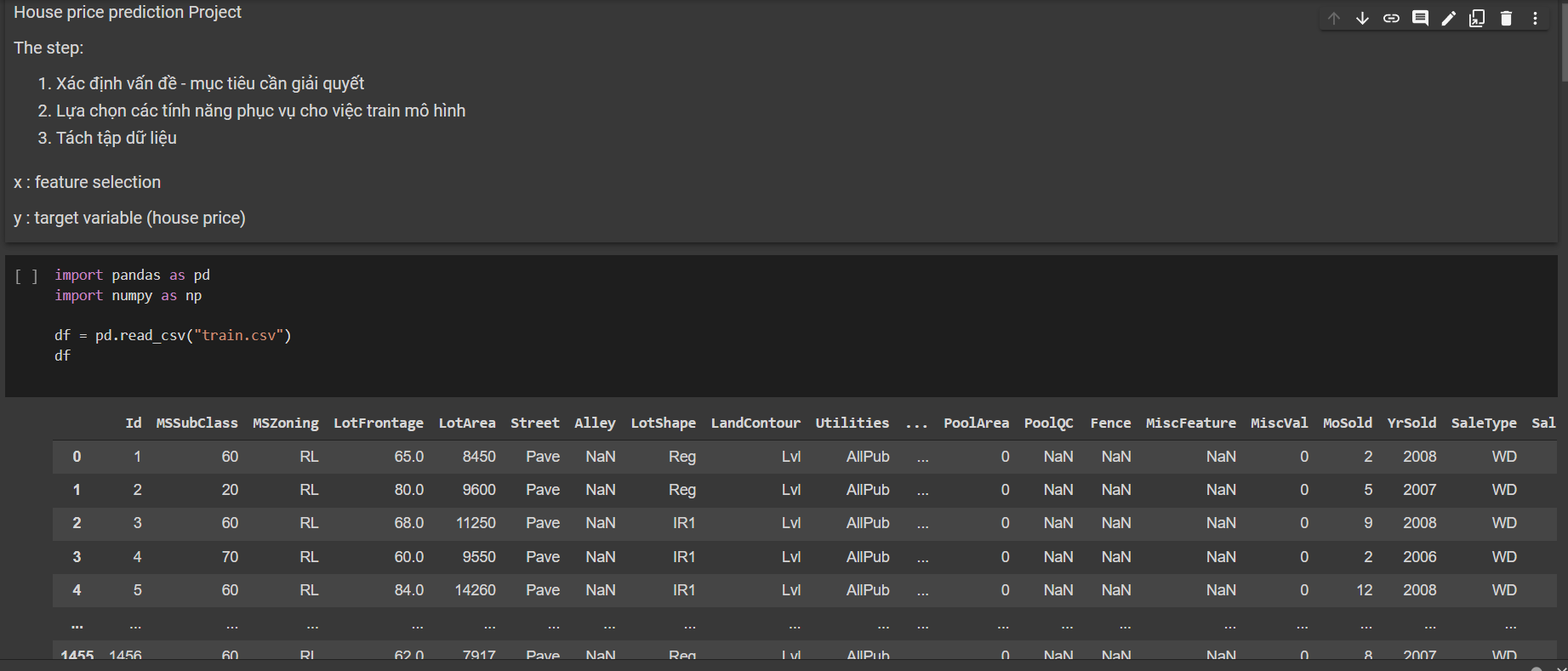
Dữ liệu 6:

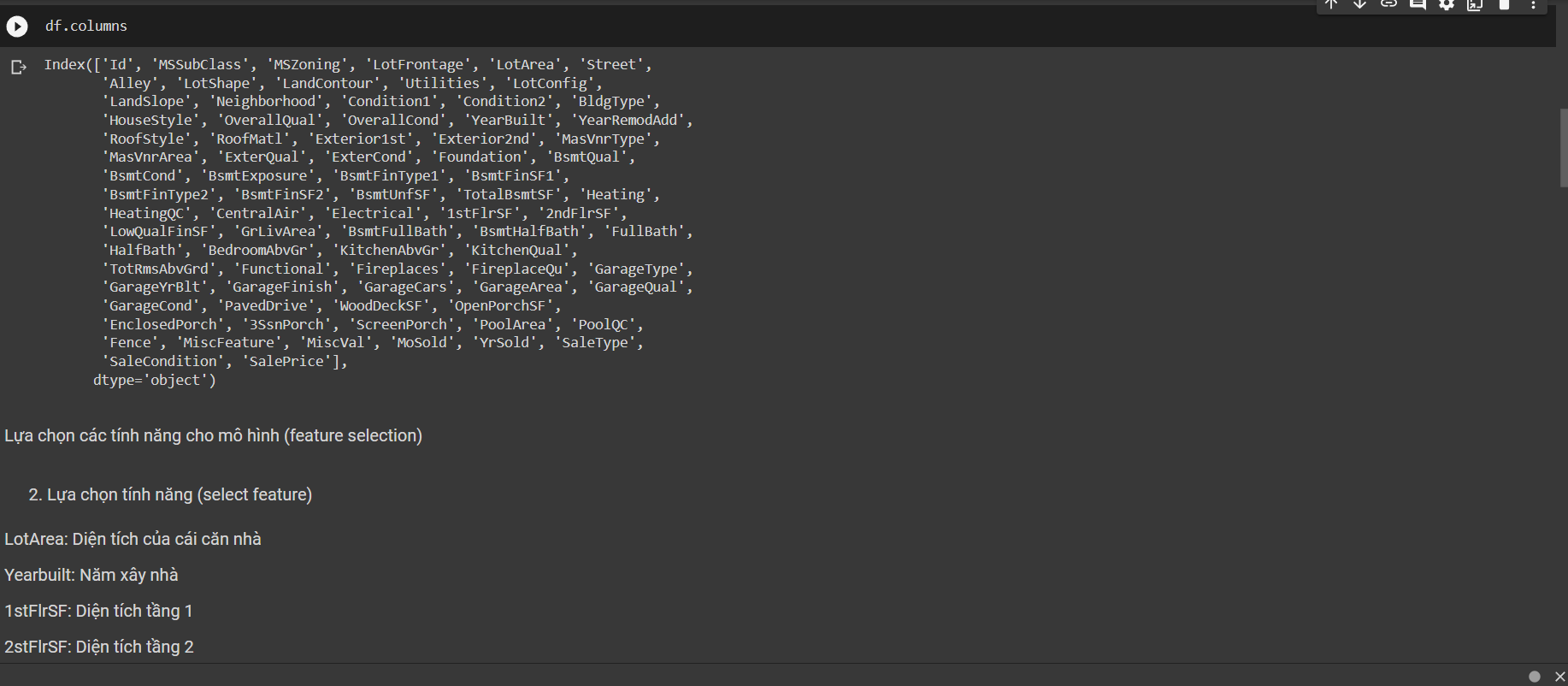
Ta đã có 1 forest với 6 cây.

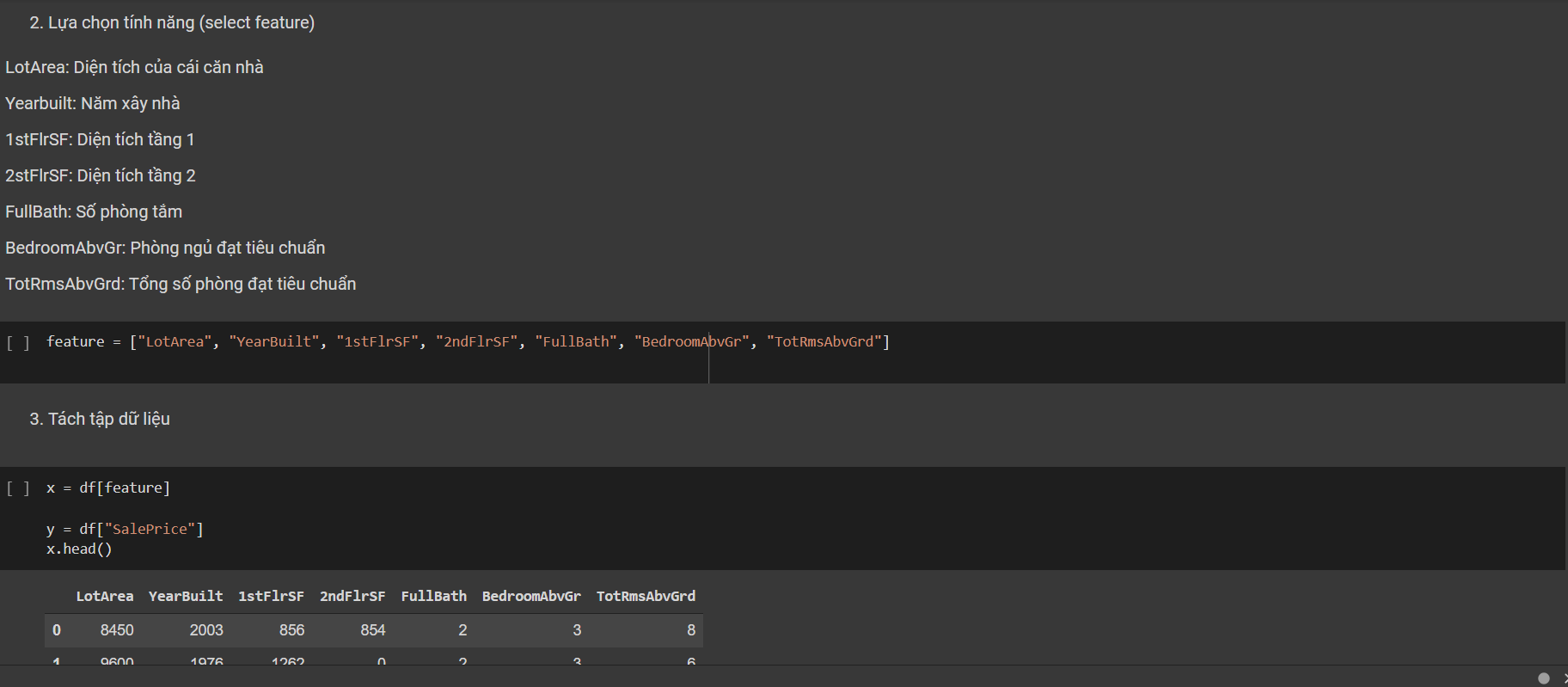
Khi có một dữ liệu mới, ta chuyển điểm dữ liệu này qua từng cây một và ghi lại các dự đoán. Sau đó kết hợp tất cả các dự đoán, lấy đa số.

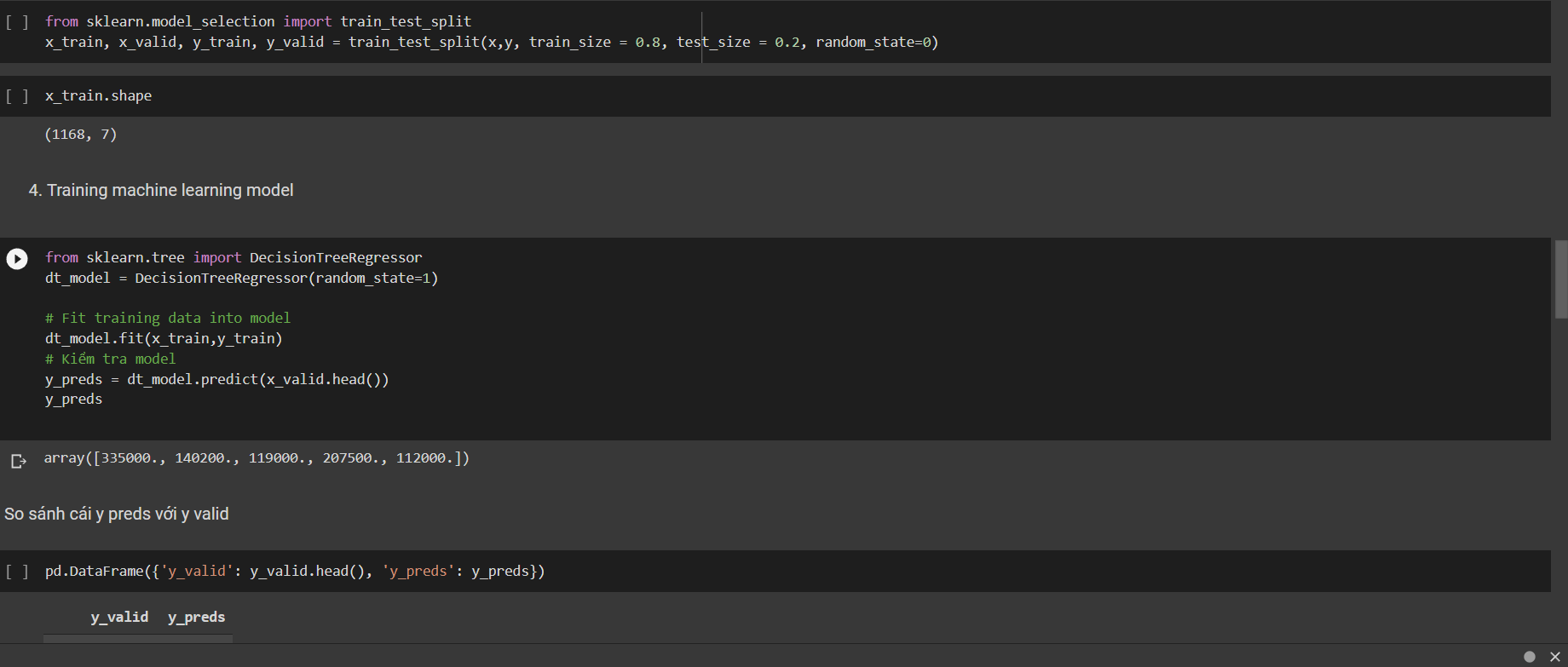
1. **Mô phỏng (Source code Python)**

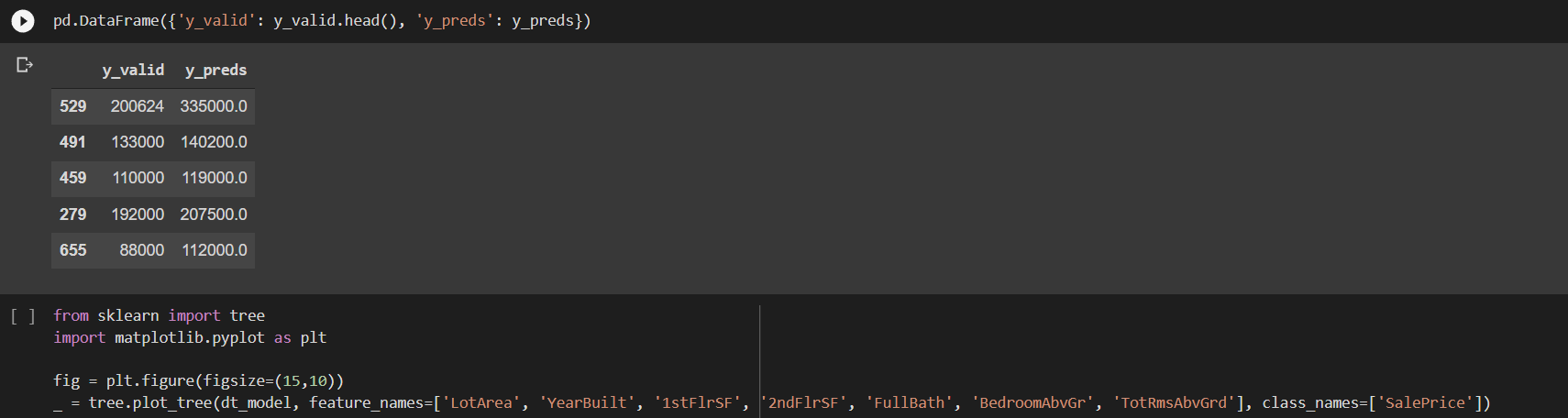
Nhóm em sẽ xây dựng một mô hình machine learning đơn giản dựa trên 2 thuật toán đó là Decision Tree và Randomforest đó là dựa đoán giá nhà dựa trên các tập dữ liệu có sẵn trên trang dữ liệu kaggle.



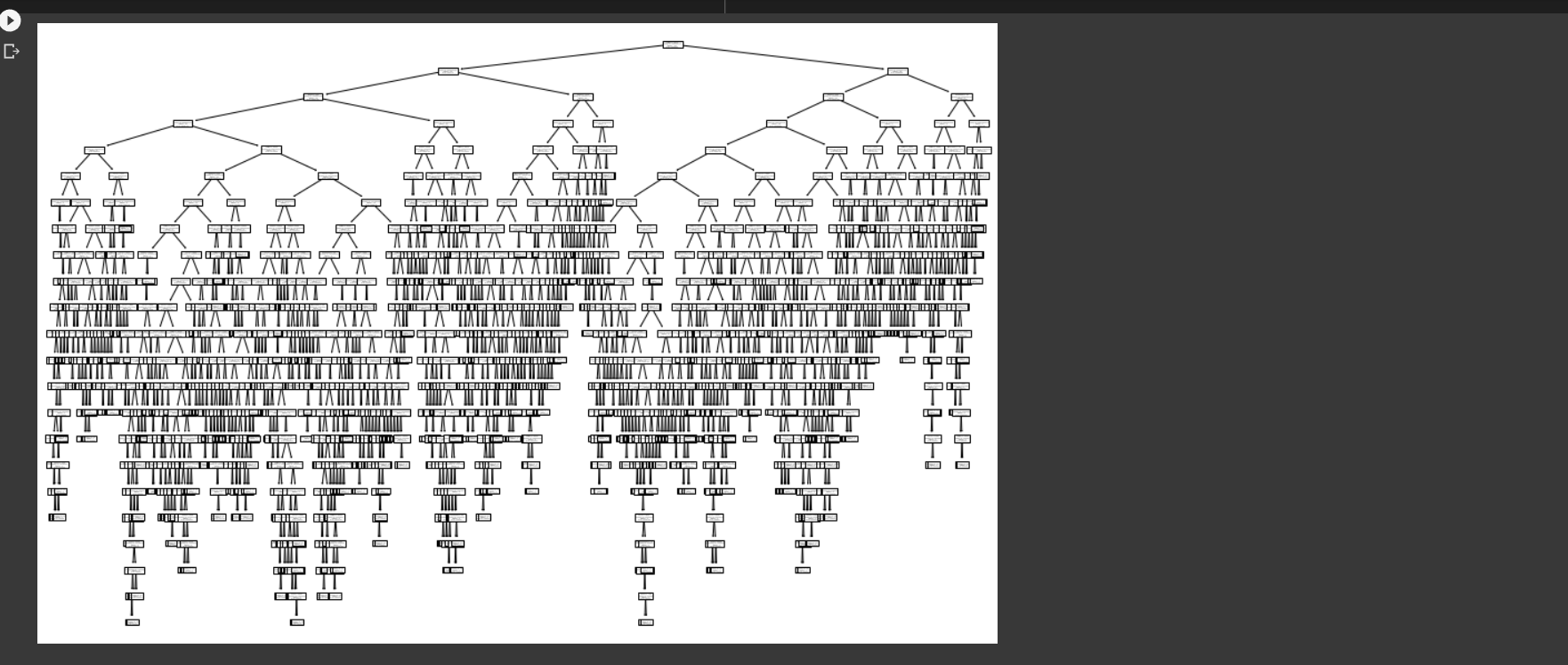




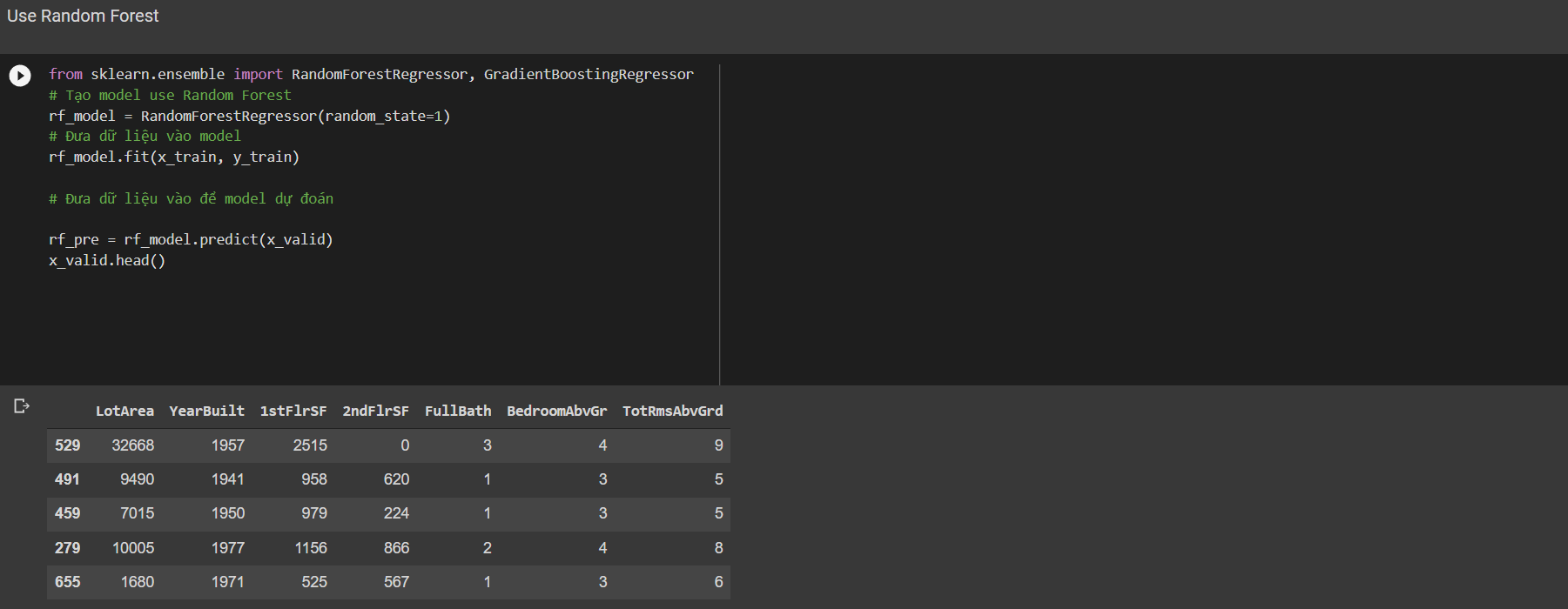




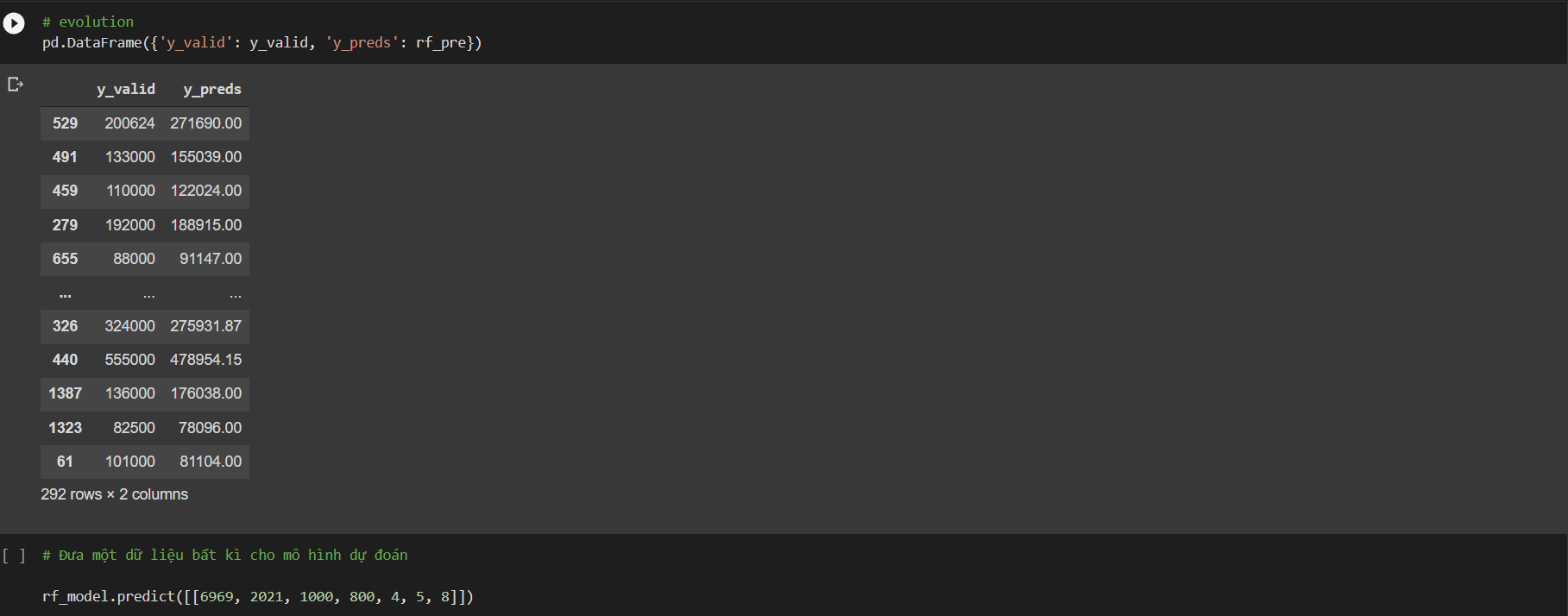
Mô hình Decision tree trên tập huấn luyện thực tế:



Sử dụng random forest để huấn luyện mô hình:

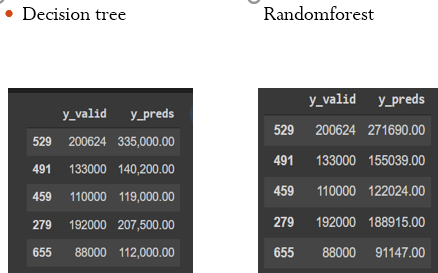


Đánh giá mô hình



***D, Kết quả mô phỏng và thảo luận***

***So sánh kết quả***

******

***Kết luận đánh giá***

Từ kết quả dự đoán thực tế của model cho thấy random forest cho ra kết quả quan hơn so với decision tree do nó được xây dựng trên decision tree nhưng khắc phục được những điểm yếu.

***E , Làm lại kết luận – Conclusive remakes***

***Một số ghi chú :***

* Tăng số lượng bài test , chia thành nhiều phần để chạy Traning cho chương trình .

***# Mở rộng :***

**Data là gì?**

Data hay còn được gọi là dữ liệu, là tập hợp thông tin bao gồm các số, từ hoặc hình ảnh, được chia làm dữ liệu thô và dữ liệu đã được xử lý.

Trong đó, dữ liệu thô là các số, ký tự, hình ảnh, ký hiệu, đại lượng vật lý và thường được tiếp tục xử lý bởi con người hoặc đưa vào máy tính.

Dữ liệu trong máy tính được lưu trữ và xử lý tại chỗ hoặc được chuyển (output) cho người hoặc máy tính khác xử lý. Dữ liệu thô mang tính tương đối vì dữ liệu đã được xử lý ở bước này có thể được gọi là dữ liệu thô ở bước tiếp theo.

* *Big Data*

Big data là tập hợp dữ liệu có khối lượng lớn và phức tạp mà các phần mềm xử lý dữ liệu truyền thống không thể thu thập, quản lý và xử lý trong một khoảng thời gian ngắn.

Bao gồm dữ liệu có cấu trúc, không có cấu trúc và bán cấu trúc, có thể được khai thác để tìm hiểu insights của khách hàng.

Đặc trưng của Big data:

* Volume: Khối lượng dữ liệu lớn
* Variety: Đa dạng các loại dữ liệu
* Velocity: Tốc độ xử lý và phân tích dữ liệu

1. **Khoa học dữ liệu là gì?**

Mỗi một cuộc cách mạng công nghệ đều sẽ mang đến một bước ngoặt lớn với cách thức chúng ta sản xuất, lao động, hãy nhìn lại thế giới xung quanh bạn đang thay đổi từng ngày như thế nào: chúng ta có các sản phẩm trí tuệ nhân tạo mô phỏng được các hoạt động y hệt con người, thậm chí là giỏi hơn khi AlphaGo của google đã đánh bại Lee Sedol, kì thủ cờ vây hàng đầu thế giới, rồi chụp x quang 3 chiều giúp phát hiện sớm ung thư, công nghệ nano giúp chữa trị ung thư cho con người, công nghệ thực tế ảo trong pokemon go từng gây sốt cho toàn thế giới, ... Thế giới đang đi những bước dài mỗi ngày, góp một phần không nhỏ trong đó chính là công nghệ thông tin, và cụ thể hơn, một trong các công nghệ góp phần vào bước phát triển của công nghệ thông tin, chính là machine learning.

Machine Learning (ML) là một trong các cụm từ được nhắc đến khá nhiều gần đây, mà tiêu biểu nhất như tôi đã nêu ở trên, AlphaGo của google. Không chỉ trong lĩnh vực trí tuệ nhân tạo, mà rất, rất nhiều lĩnh vực khác cũng đang có sự tham gia của ML, và kiến thức về ML thì rất nhiều và rất khó học nên trong khuôn khổ bài viết này, tôi sẽ chỉ nhắc tới một vài khái niệm cơ bản của Machine Learning, áp dụng của ML trong thực tế. Những khái niệm nâng cao hơn, đi sâu hơn về ML sẽ được đem đến trong những bài viết sau của series.

* 1. *Khái niệm*

Thực chất thì tới thời điểm hiện tại, vẫn chưa có một định nghĩa thống nhất cho ML, nhưng đa phần khi tìm tài liệu trên mạng, chúng ta sẽ thấy định nghĩa về machine learning như thế này:

Machine learning is the subfield of computer science that gives computers the ability to learn without being explicitly programmed."

Định nghĩa này do Arthur Samuel đưa ra năm 1959, tạm dịch là "Maching learning là một ngành học thuộc khoa học máy tính, giúp máy tính có khả năng tự học mà không phải lập trình một cách rõ ràng"

Hoặc theo Tom Mitchell:

A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P if its performance at tasks in T, as measured by P, improves with experience E

Định nghĩa này có vẻ khó hiểu hơn cái trước, tạm hiểu là Tom Mitchell coi Machine Learning như 1 chương trình, nhiệm vụ của nó là thưc hiện 1 task T nào đó, khi thực hiện xong, ta thu được experience E. Nhờ vào việc học hỏi experience E, ta có thể thay đổi (hoặc không) để tiến tới thực hiện task T+1, và nhằm cải thiện hiệu suất P. Lấy ngay ví dụ là AlphaGo, T chính là chơi mỗi ván cờ với các người chơi khác, E chính là kinh nghiệm thu được sau khi chơi các ván đó, còn P chính là xác suất AlphaGo thắng ván tiếp theo, nhờ vào việc liên tục chơi (thực hiện task T) và cập nhật kinh nghiệm E để nâng cao P.

* Machine Learning có sự gắn bó chặt chẽ với khá nhiều ngành khác, ví dụ như Big Data, AI, Statistics Learnig, đã và đang ứng dụng sâu rộng vào cuộc sống hàng ngày: trí tuệ nhân tạo AlphaGo, nhận diện khuôn mặt, gợi ý bạn bè từ faceboook, phân loại spam email từ google mail, chuẩn đoán y khoa, phát hiện thẻ tín dụng giả, phân tích thị trường chứng khoán, dự đoán kết quả trận đấu, nhận dạng giọng nói, phân loại các chuẩn DNA, ...
* Machine learning Algorithm được chia làm 2 loại chính là: Supervised Learning (Học có giám sát) và Unsupervised Learning (Học không giám sát). Ngoài ra còn 1 vài loại khác như SemiSupervised Learning, Reinforcement Learning, Learning to Learn, Developmental Learning, ... Trong bài viết hôm nay tôi sẽ chỉ tập trung vào Supervised Learning và Unsupervised Learning. Bài viết tham khảo chính từ khóa học [Stanford Machine Learning](https://www.coursera.org/learn/machine-learning) nên nếu các bạn đã từng học ML ở đây thì nên chuyển sang một bài viết khác.

*b) Lý do Machine learning là cần thiết với đời sống*

**-** Nhu cầu toàn cầu lớn**:** Hiện nay, nhu cầu Machine learning đang dần trở nên bùng nổ trên toàn thế giới. Và mức lương nhập cảnh của nó đang được bắt đầu trong khoảng từ $100k – $150k. Chính vì vậy, các nhà khoa học dữ liệu, kỹ sư phần mềm và nhà phân tích kinh doanh đều nhận được nhiều lợi ích nếu như biết đến Machine learning.

- Dữ liệu là sức mạnh**:** Dữ liệu của Machine learning đang dần có các bước

tiến biến đổi những thứ xung quanh chúng ta. Chính vì vậy, các tổ chức từ những công ty khởi nghiệp cho đến các doanh nghiệp khổng lồ công nghệ đều đang chạy đua để khai thác dữ liệu từ nó.

- Machine learning thú vị**:** Machine learning được xem là tuyệt vời nhờ sở hữu sự pha trộn độc đáo giữa sự khám phá, ứng dụng và kỹ thuật kinh doanh độc nhất. Nhờ vậy, bạn có thể sẽ thu được rất nhiều niềm vui rất phong phú và sôi động từ nó.

END.