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

A. ***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.
-

B. ***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

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1460 entries, 0 to 1459
Data columns (total 81 columns):
 #    Column           Non-Null Count  Dtype
---   ------           --------------  -----
 0    Id               1460 non-null   int64
 1    MSSubClass       1460 non-null   int64
 2    MSZoning         1460 non-null   object
 3    LotFrontage      1201 non-null   float64
 4    LotArea          1460 non-null   int64
 5    Street           1460 non-null   object
 6    Alley            91 non-null     object
 7    LotShape         1460 non-null   object
 8    LandContour      1460 non-null   object
 9    Utilities        1460 non-null   object
 10   LotConfig        1460 non-null   object
 11   LandSlope        1460 non-null   object
 12   Neighborhood     1460 non-null   object
 13   Condition1       1460 non-null   object
 14   Condition2       1460 non-null   object
 15   BldgType         1460 non-null   object
 16   HouseStyle       1460 non-null   object
 17   OverallQual      1460 non-null   int64
 18   OverallCond      1460 non-null   int64
```


3 , Tổng quan về dữ liệu

	Id	MSSubClass	MSZoning	LotFrontage	LotArea	Street	Alley	LotShape	LandContour	Utilities	...	PoolArea	PoolQC	Fence	MiscFeature	MiscVal	MoSold	YrSold	SaleType	Sal
0	1	60	RL	65.0	8450	Pave	NaN	Reg	Lvl	AllPub	...	0	NaN	NaN	NaN	0	2	2008	WD	
1	2	20	RL	80.0	9600	Pave	NaN	Reg	Lvl	AllPub	...	0	NaN	NaN	NaN	0	5	2007	WD	
2	3	60	RL	68.0	11250	Pave	NaN	IR1	Lvl	AllPub	...	0	NaN	NaN	NaN	0	9	2008	WD	
3	4	70	RL	60.0	9550	Pave	NaN	IR1	Lvl	AllPub	...	0	NaN	NaN	NaN	0	2	2006	WD	
4	5	60	RL	84.0	14260	Pave	NaN	IR1	Lvl	AllPub	...	0	NaN	NaN	NaN	0	12	2008	WD	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
1455	1456	60	RL	62.0	7917	Pave	NaN	Reg	Lvl	AllPub	...	0	NaN	NaN	NaN	0	8	2007	WD	
1456	1457	20	RL	85.0	13175	Pave	NaN	Reg	Lvl	AllPub	...	0	NaN	MnPrv	NaN	0	2	2010	WD	
1457	1458	70	RL	66.0	9042	Pave	NaN	Reg	Lvl	AllPub	...	0	NaN	GdPrv	Shed	2500	5	2010	WD	
1458	1459	20	RL	68.0	9717	Pave	NaN	Reg	Lvl	AllPub	...	0	NaN	NaN	NaN	0	4	2010	WD	
1459	1460	20	RL	75.0	9937	Pave	NaN	Reg	Lvl	AllPub	...	0	NaN	NaN	NaN	0	6	2008	WD	

1460 rows × 81 columns

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)

	MSSubClass	MSZoning	LotFrontage	LotArea	Street	Alley	LotShape	LandContour	Utilities	LotConfig	...	PoolArea	PoolQC	Fence	MiscFe
0	60	RL	65.0	8450	Pave	NaN	Reg	Lvl	AllPub	Inside	...	0	NaN	NaN	
1	20	RL	80.0	9600	Pave	NaN	Reg	Lvl	AllPub	FR2	...	0	NaN	NaN	
2	60	RL	68.0	11250	Pave	NaN	IR1	Lvl	AllPub	Inside	...	0	NaN	NaN	
3	70	RL	60.0	9550	Pave	NaN	IR1	Lvl	AllPub	Corner	...	0	NaN	NaN	
4	60	RL	84.0	14260	Pave	NaN	IR1	Lvl	AllPub	FR2	...	0	NaN	NaN	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
1455	60	RL	62.0	7917	Pave	NaN	Reg	Lvl	AllPub	Inside	...	0	NaN	NaN	
1456	20	RL	85.0	13175	Pave	NaN	Reg	Lvl	AllPub	Inside	...	0	NaN	MnPrv	
1457	70	RL	66.0	9042	Pave	NaN	Reg	Lvl	AllPub	Inside	...	0	NaN	GdPrv	
1458	20	RL	68.0	9717	Pave	NaN	Reg	Lvl	AllPub	Inside	...	0	NaN	NaN	
1459	20	RL	75.0	9937	Pave	NaN	Reg	Lvl	AllPub	Inside	...	0	NaN	NaN	

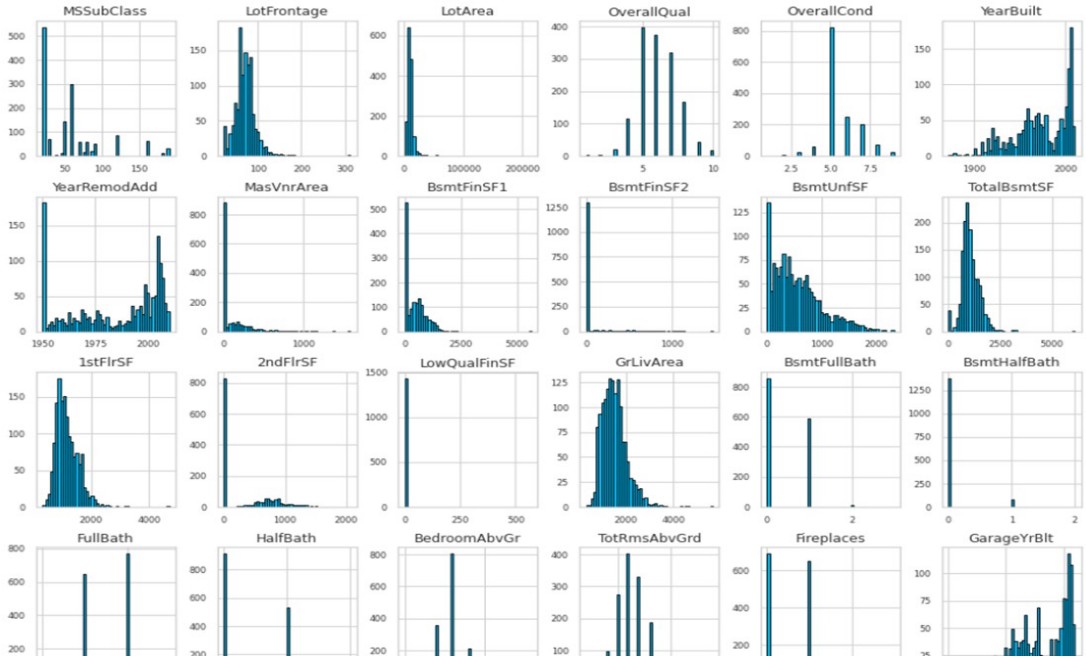
1460 rows × 80 columns

- 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 .

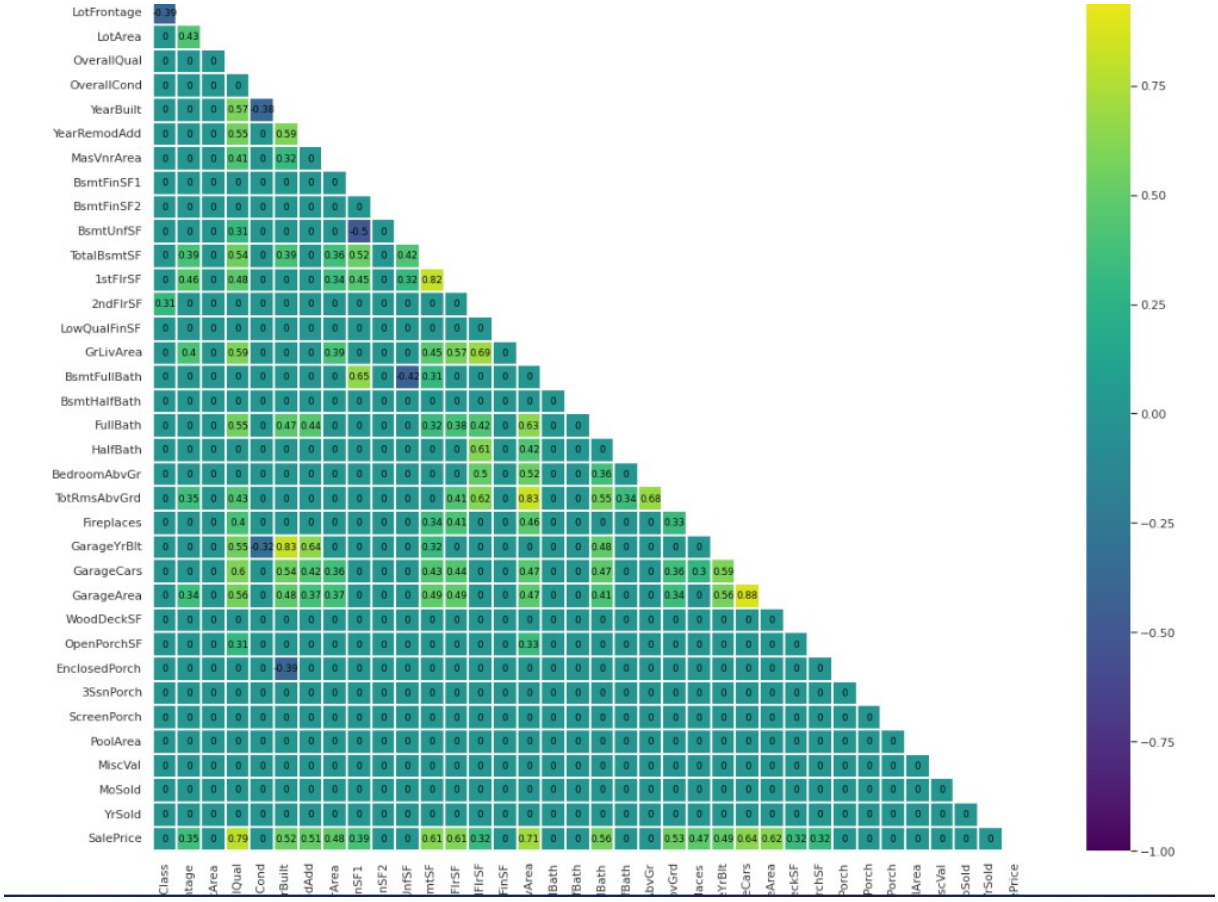
```
df_train_num = df.select_dtypes(exclude=["object"])
from sklearn.feature_selection import VarianceThreshold

# độ tương quan cao
sel = VarianceThreshold(threshold=0.05) # loại bỏ các cột trong đó 95% giá trị không đổi
sel.fit(df_train_num.iloc[:, :-1])
quasi_constant_features_list = [x for x in df_train_num.iloc[:, :-
                                                    1].columns if x not in df_train_num.iloc[:, :-1].columns[sel.get_support()]]
sel.fit(df_train_num.iloc[:, :-1])
df_train_num.drop(quasi_constant_features_list, axis=1, inplace=True)
```

Trực quan hóa các giá trị tuyệt đối



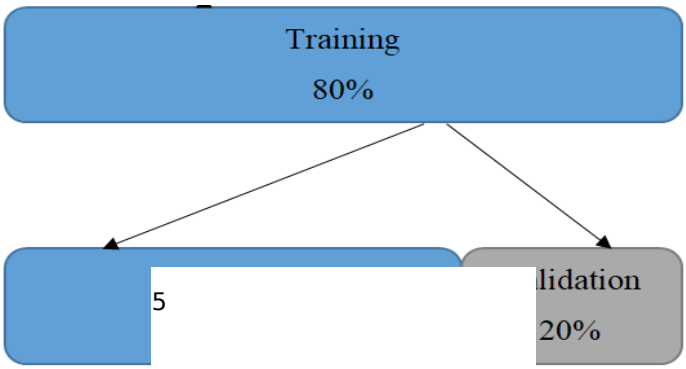
Xử dụng heatmap để đ quan của các đặc điểm với giá bán và tương quan giữa



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 phá

- ***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 tá

Gini Inde of class i

Thu thông tin: IG = G(paren định thì có nhiều cách phân

tách, dùng IG để lựa chọn phương thức tối ưu nhất)

2. **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.

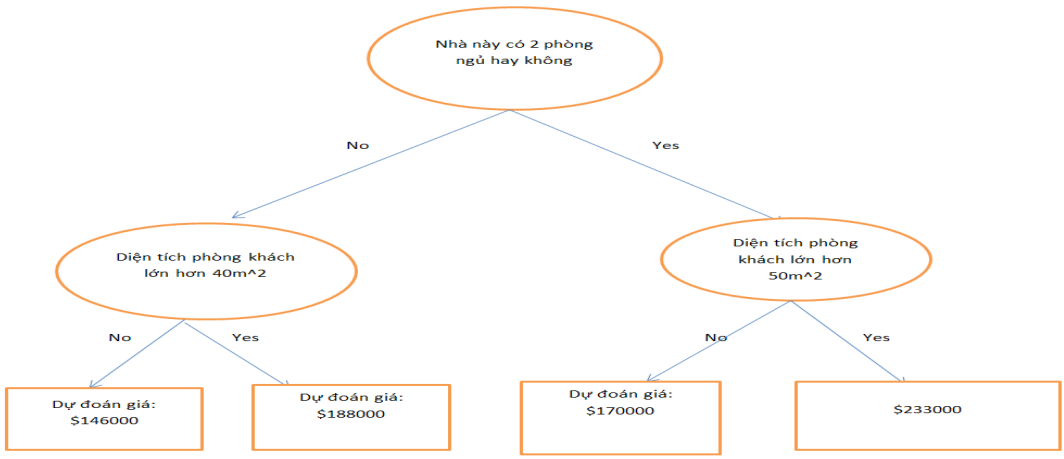
Var =

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.

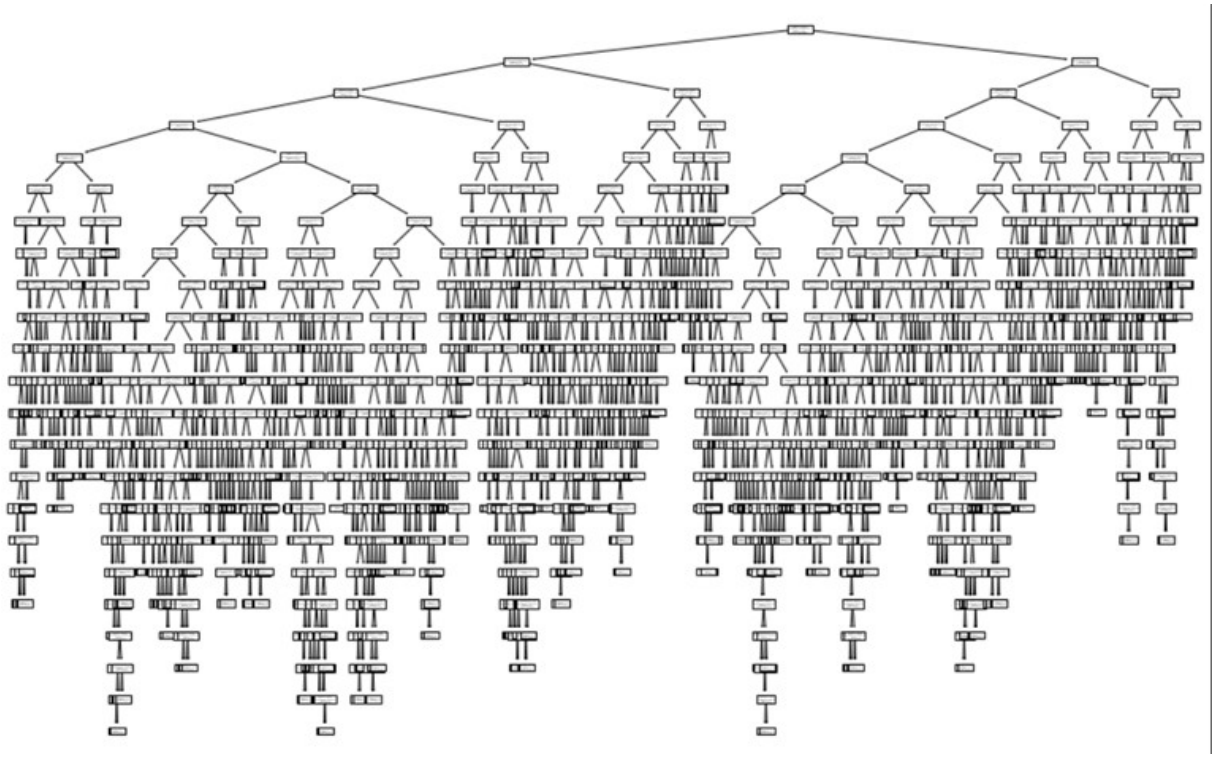
Id	LotArea	YearBuilt	1stFlrSF	2ndFlrSF	FullBath	BedroomAbvGr	TotRmsAbvGrd	SalePrice
1	8450	2003	856	854	2	3	8	208500
2	9600	1976	1262	0	2	3	6	181500
3	11250	2001	920	866	2	3	6	223500
4	9550	1915	961	756	1	3	7	140000
5	14260	2000	1145	1053	2	4	9	250000
6	14115	1993	796	566	1	1	5	143000
7	10084	2004	1694	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




```
[26] from sklearn.tree import DecisionTreeRegressor
     dt_model = DecisionTreeRegressor(random_state=1)

     # Fit training data into model
     dt_model.fit(x_train,y_train)
     # Kiểm tra model
     y_preds = dt_model.predict(x_vali
     y_preds
```

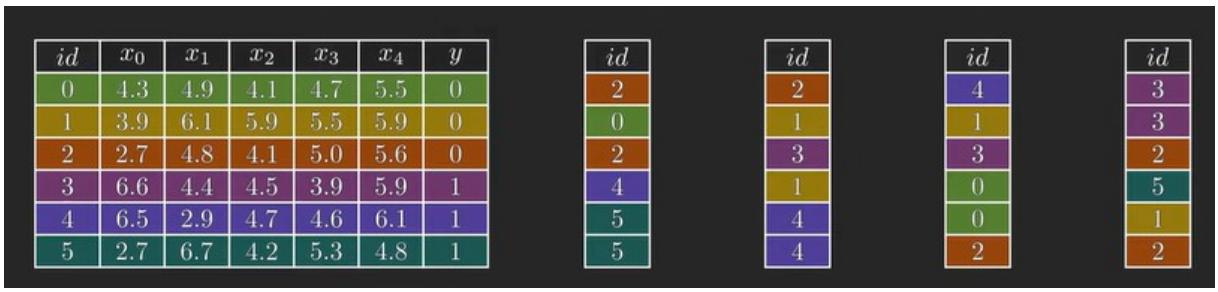
Thực tế bài toán



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.



id	x_0	x_1	x_2	x_3	x_4	y
0	4.3	4.9	4.1	4.7	5.5	0
1	3.9	6.1	5.9	5.5	5.9	0
2	2.7	4.8	4.1	5.0	5.6	0
3	6.6	4.4	4.5	3.9	5.9	1
4	6.5	2.9	4.7	4.6	6.1	1
5	2.7	6.7	4.2	5.3	4.8	1

id
2
0
2
4
5
5

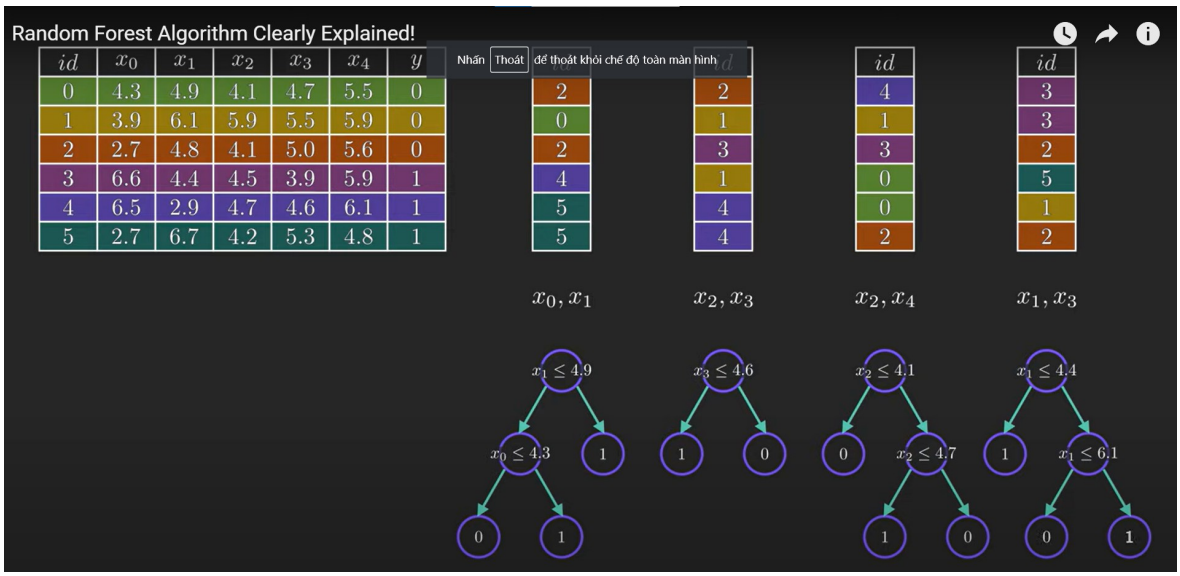
id
2
1
3
1
4
4

id
4
1
3
0
0
2

id
3
3
2
5
1
2

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 độ[8] ẫu nhiên một tập hợp con các đặc điểm cho từng cây và chỉ

Tiếp theo là xây dựng cây quy



id	x_0	x_1	x_2	x_3	x_4	y
0	4.3	4.9	4.1	4.7	5.5	0
1	3.9	6.1	5.9	5.5	5.9	0
2	2.7	4.8	4.1	5.0	5.6	0
3	6.6	4.4	4.5	3.9	5.9	1
4	6.5	2.9	4.7	4.6	6.1	1
5	2.7	6.7	4.2	5.3	4.8	1

id
2
0
2
4
5
5

id
2
1
3
1
4
4

id
4
1
3
0
0
2

id
3
3
2
5
1
2

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 c ể *dự đoán giá nhà* như sau:

Ứng dụng thực tế xây dự arning 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

nút trái = 164400

nút trái = 142350

Var nút trái = 433333

Var nút : 571542500

Id		
2	1976	181500
4	1915	140000
9	1931	129900
10	1939	118000

nút phải = 196500
Var nút phải = 2862250000

nút phải = 200500
Var nút phải = 1940166667

Var Red = Var(parent) -

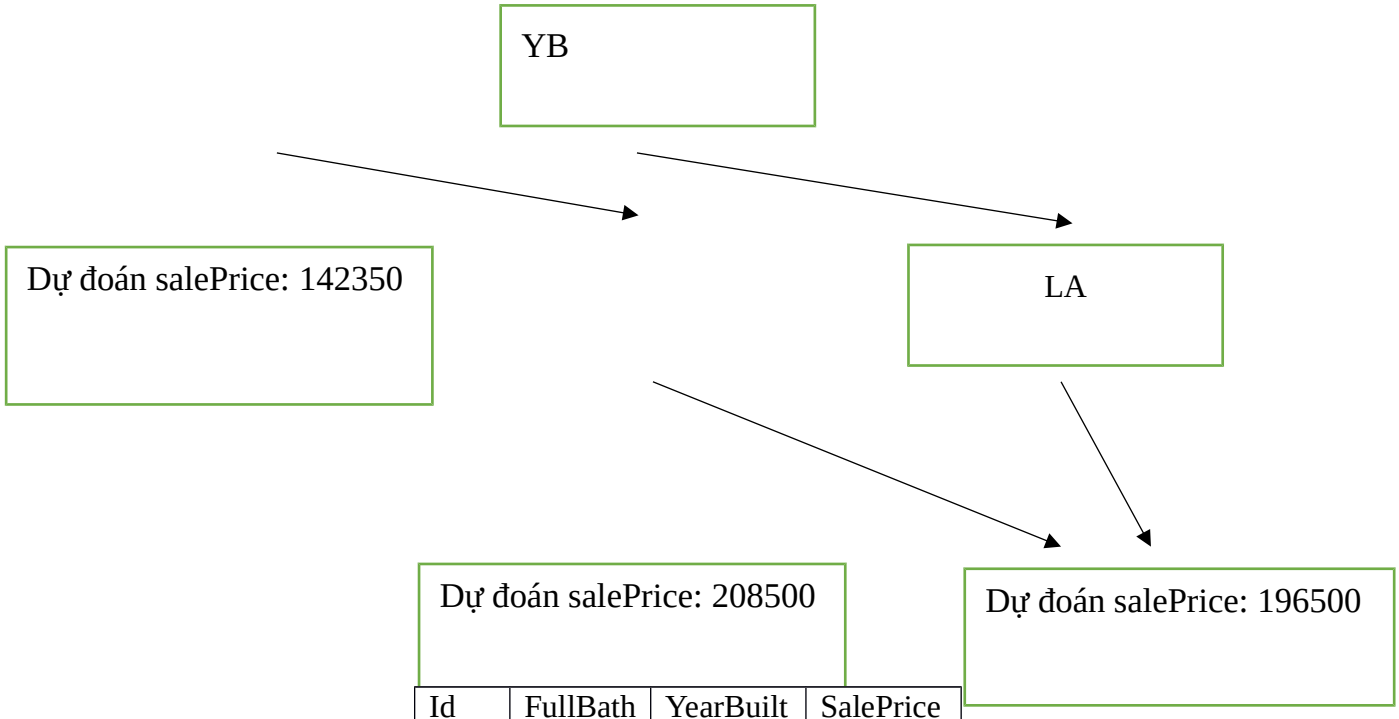
Var Red 1 = 2204258400 - *1353433333 - *2862250000 = 247298400

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.

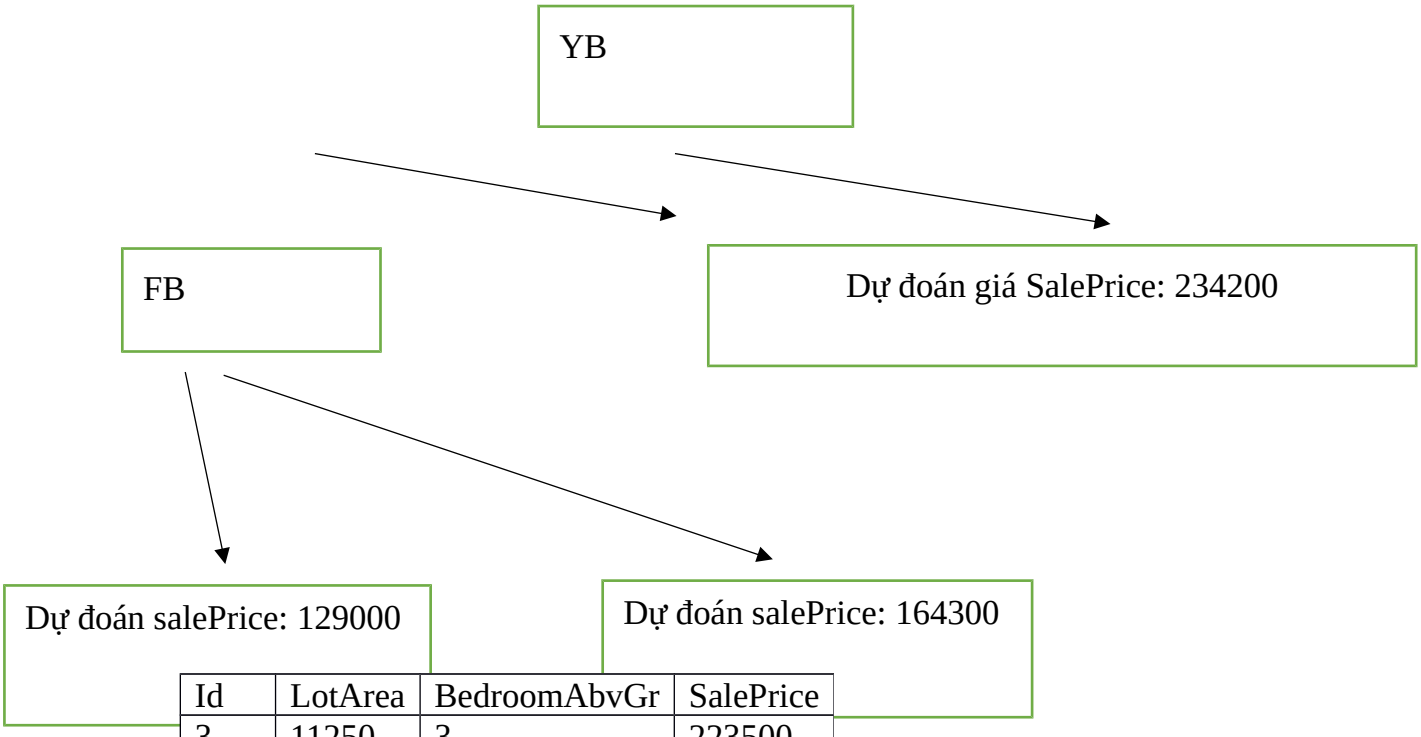
- ⇨ Xây dựng Decision Tree



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

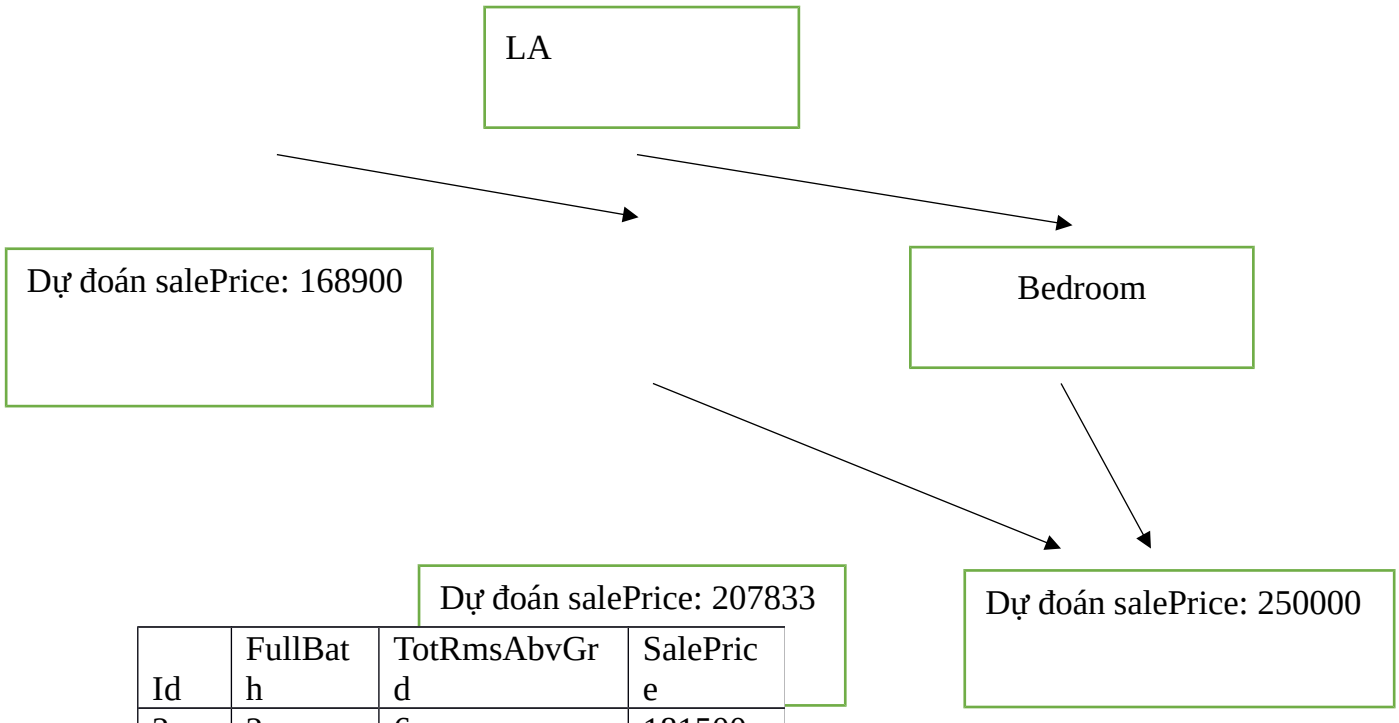
Thực hiện xử lý giống như dữ liệu 1, ta có được Decision Tree như sau:



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:

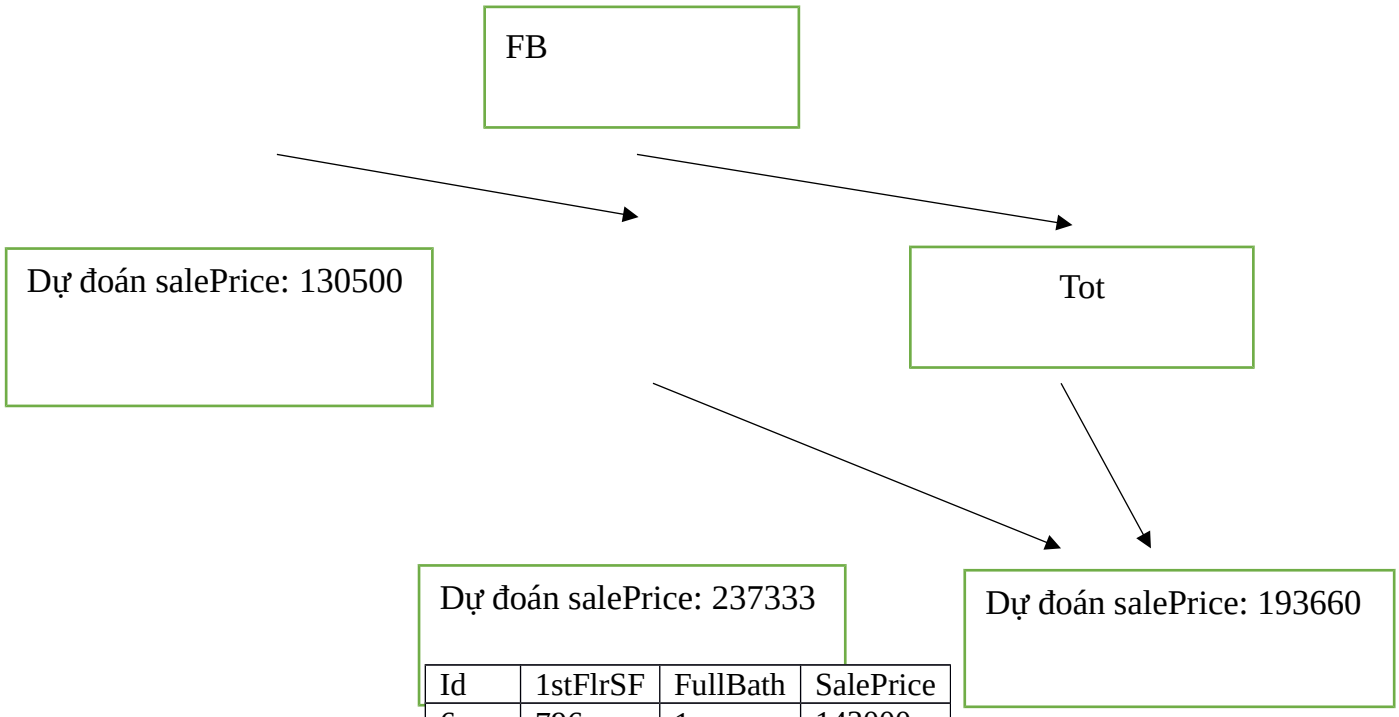
Thực hiện xử lý giống như dữ liệu 1, ta có được Decision Tree như sau:



ữ 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

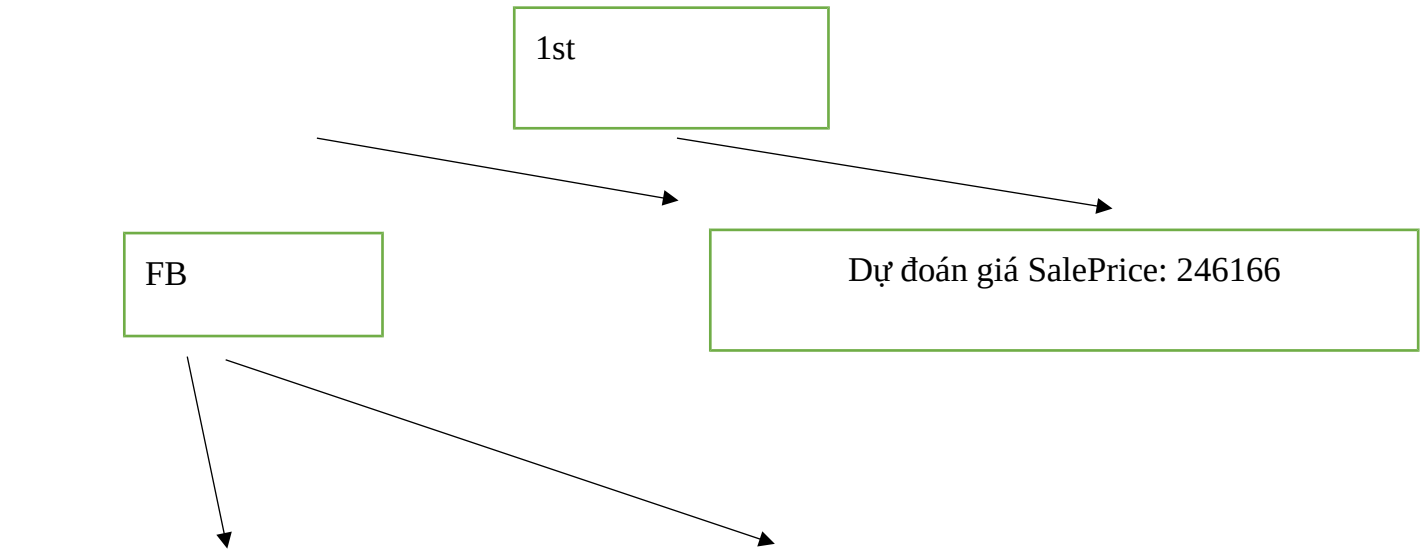
Thực hiện xử lý giống như dữ liệu 1, ta có được Decision Tree như sau:



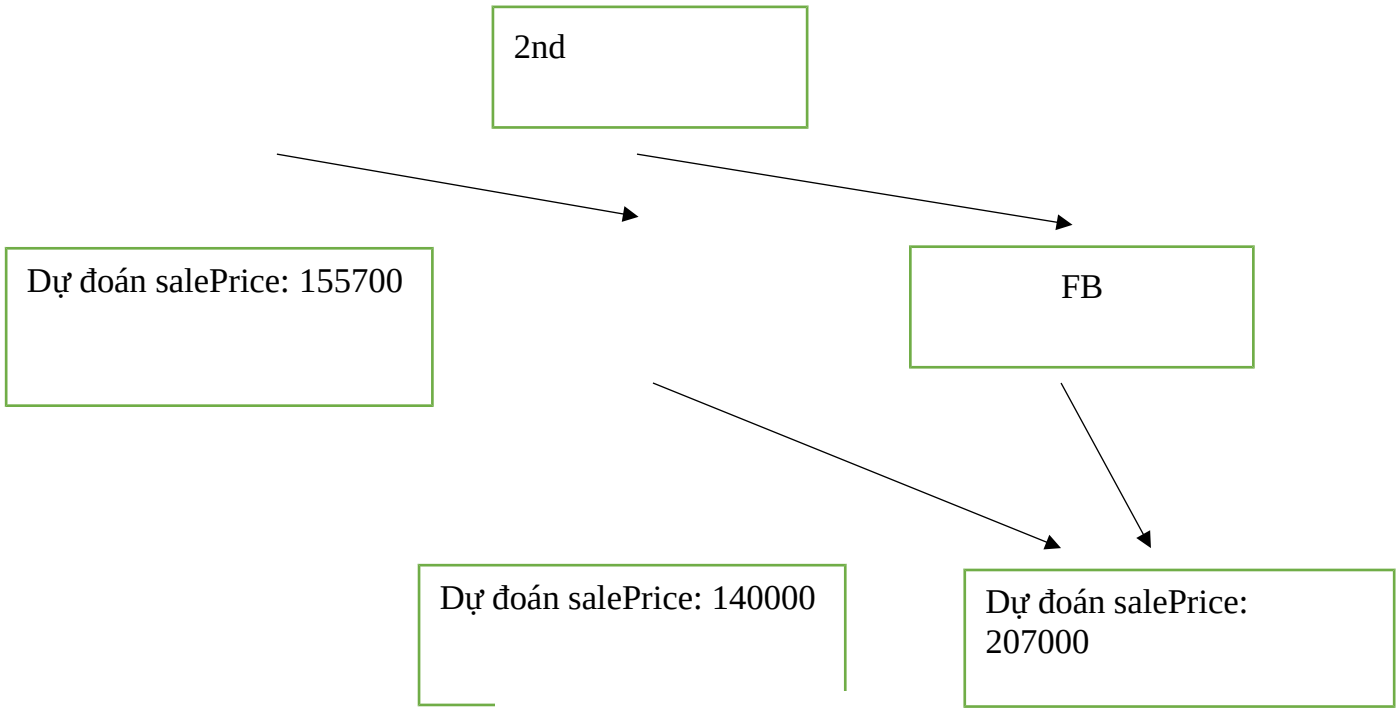
Dữ liệu 5:

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

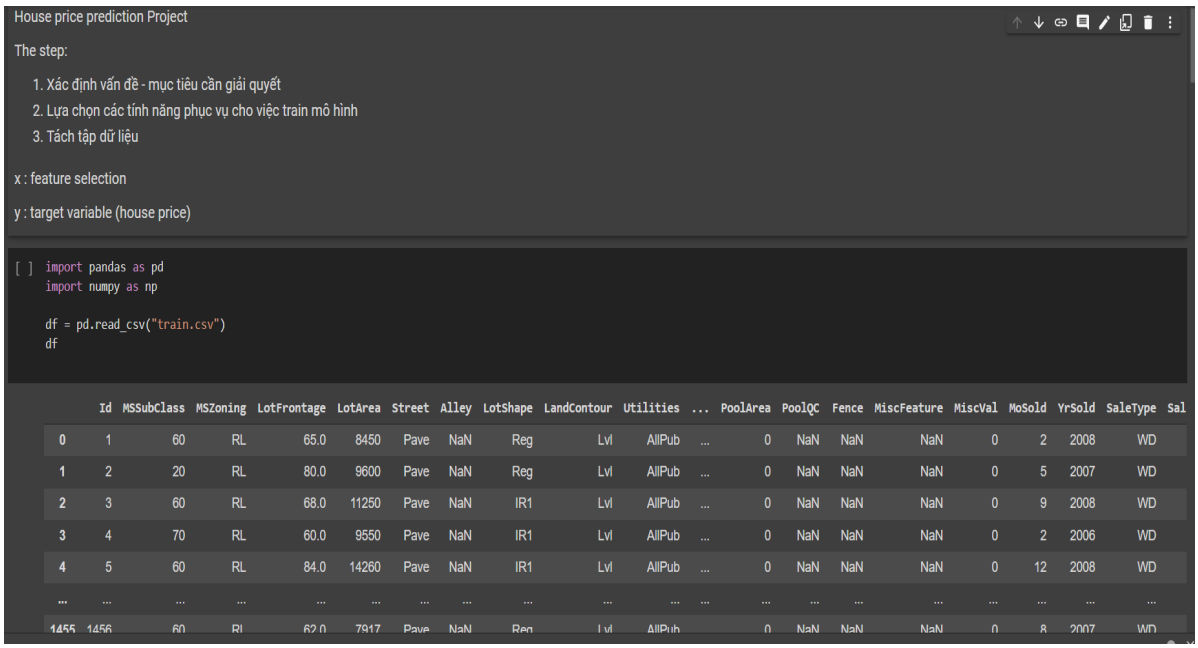


Ta đã có 1 forest với 6 cây.

Khi có một dữ liệu mới, ta ch ừng cây một và ghi lại các dự đoán. Sau đó kết hợp tất cả cá

I. 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.



House price prediction Project

The step:

1. Xác định vấn đề - mục tiêu cần giải quyết
2. Lựa chọn các tính năng phục vụ cho việc train mô hình
3. Tách tập dữ liệu

x : feature selection

y : target variable (house price)

```
[ ] import pandas as pd
    import numpy as np

    df = pd.read_csv("train.csv")
    df
```

	Id	MSSubClass	MSZoning	LotFrontage	LotArea	Street	Alley	LotShape	LandContour	Utilities	...	PoolArea	PoolQC	Fence	MiscFeature	MiscVal	MoSold	YrSold	SaleType	Sal
0	1	60	RL	65.0	8450	Pave	NaN	Reg	Lvl	AllPub	...	0	NaN	NaN	NaN	0	2	2008	WD	
1	2	20	RL	80.0	9600	Pave	NaN	Reg	Lvl	AllPub	...	0	NaN	NaN	NaN	0	5	2007	WD	
2	3	60	RL	68.0	11250	Pave	NaN	IR1	Lvl	AllPub	...	0	NaN	NaN	NaN	0	9	2008	WD	
3	4	70	RL	60.0	9550	Pave	NaN	IR1	Lvl	AllPub	...	0	NaN	NaN	NaN	0	2	2006	WD	
4	5	60	RL	84.0	14260	Pave	NaN	IR1	Lvl	AllPub	...	0	NaN	NaN	NaN	0	12	2008	WD	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
1455	1456	60	RL	62.0	7917	Pave	NaN	Reg	Lvl	AllPub		0	NaN	NaN	NaN	0	8	2007	WD	

```
df.columns
```

```
Index(['Id', 'MSSubClass', 'MSZoning', 'LotFrontage', 'LotArea', 'Street',
       'Alley', 'LotShape', 'LandContour', 'Utilities', 'LotConfig',
       'LandSlope', 'Neighborhood', 'Condition1', 'Condition2', 'BldgType',
       'HouseStyle', 'OverallQual', 'OverallCond', 'YearBuilt', 'YearRemodAdd',
       'RoofStyle', 'RoofMatl', 'Exterior1st', 'Exterior2nd', 'MasVnrType',
       'MasVnrArea', 'ExterQual', 'ExterCond', 'Foundation', 'BsmtQual',
       'BsmtCond', 'BsmtExposure', 'BsmtFinType1', 'BsmtFinSF1',
       'BsmtFinType2', 'BsmtFinSF2', 'BsmtUnfSF', 'TotalBsmtSF', 'Heating',
       'HeatingQC', 'CentralAir', 'Electrical', '1stFlrSF', '2ndFlrSF',
       'LowQualFinSF', 'GrLivArea', 'BsmtFullBath', 'BsmtHalfBath', 'FullBath',
       'HalfBath', 'BedroomAbvGr', 'KitchenAbvGr', 'KitchenQual',
       'TotRmsAbvGrd', 'Functional', 'Fireplaces', 'FireplaceQu', 'GarageType',
       'GarageYrBlt', 'GarageFinish', 'GarageCars', 'GarageArea', 'GarageQual',
       'GarageCond', 'PavedDrive', 'WoodDeckSF', 'OpenPorchSF',
       'EnclosedPorch', '3SsnPorch', 'ScreenPorch', 'PoolArea', 'PoolQC',
       'Fence', 'MiscFeature', 'MiscVal', 'MoSold', 'YrSold', 'SaleType',
       'SaleCondition', 'SalePrice'],
      dtype='object')
```

Lựa chọn các tính năng cho mô hình (feature selection)

2. Lựa chọn tính năng (select feature)

LotArea: Diện tích của cái căn nhà

Yearbuilt: Năm xây nhà

1stFlrSF: Diện tích tầng 1

2stFlrSF: Diện tích tầng 2

2. Lựa chọn tính năng (select feature)

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

```
[ ] feature = ["LotArea", "YearBuilt", "1stFlrSF", "2ndFlrSF", "FullBath", "BedroomAbvGr", "TotRmsAbvGrd"]
```

3. Tách tập dữ liệu

```
[ ] x = df[feature]

    y = df["SalePrice"]
    x.head()
```

	LotArea	YearBuilt	1stFlrSF	2ndFlrSF	FullBath	BedroomAbvGr	TotRmsAbvGrd
0	8450	2003	856	854	2	3	8
1	9600	1976	1262	0	2	3	6

```
[ ] from sklearn.model_selection import train_test_split
    x_train, x_valid, y_train, y_valid = train_test_split(x,y, train_size = 0.8, test_size = 0.2, random_state=0)

[ ] x_train.shape

    (1168, 7)
```

4. Training machine learning model

```
from sklearn.tree import DecisionTreeRegressor
dt_model = DecisionTreeRegressor(random_state=1)

# Fit training data into model
dt_model.fit(x_train,y_train)
# Kiểm tra model
y_preds = dt_model.predict(x_valid.head())
y_preds
```

```
array([335000., 140200., 119000., 207500., 112000.])
```

So sánh cái y preds với y valid

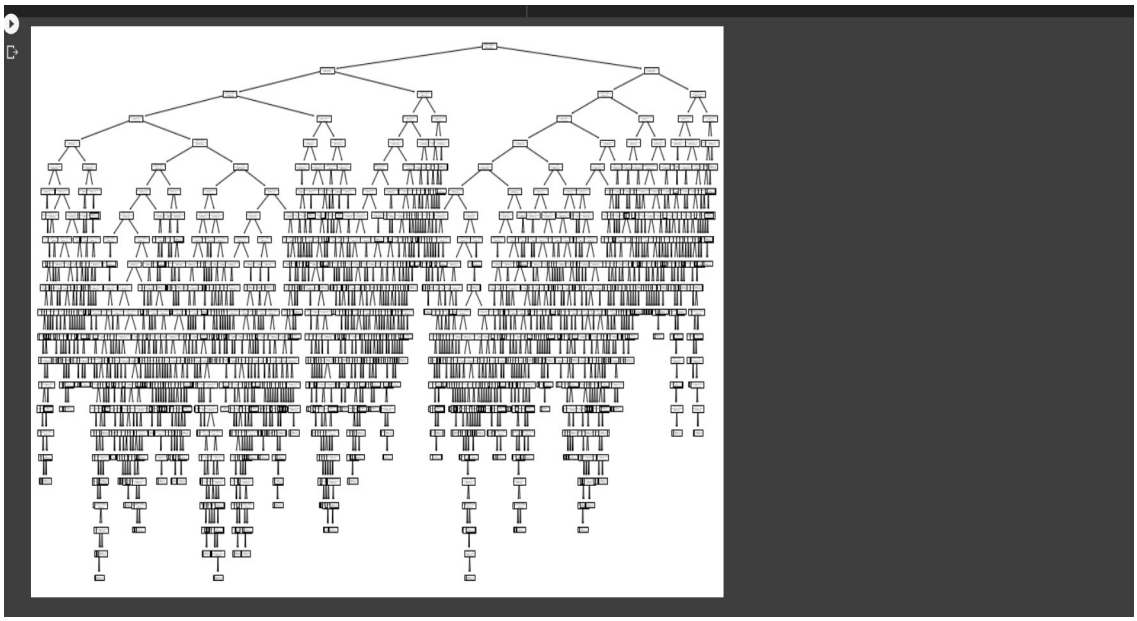
```
[ ] pd.DataFrame({'y_valid': y_valid.head(), 'y_preds': y_preds})
```

	y_valid	y_preds
529	200624	335000.0
491	133000	140200.0
459	110000	119000.0
279	192000	207500.0
655	88000	112000.0

```
[ ] from sklearn import tree
    import matplotlib.pyplot as plt

    fig = plt.figure(figsize=(15,10))
    _ = tree.plot_tree(dt_model, feature_names=['LotArea', 'YearBuilt', '1stFlrSF', '2ndFlrSF', 'FullBath', 'BedroomAbvGr', 'TotRmsAbvGrd'], class_names=['SalePrice'])
```


Mô hình Decision t : tế:



Sử dụng random forest để huấn luyện mô hình:

Use Random Forest

```
from sklearn.ensemble import RandomForestRegressor, GradientBoostingRegressor
# Tạo model use Random Forest
rf_model = RandomForestRegressor(random_state=1)
# Đưa dữ liệu vào model
rf_model.fit(x_train, y_train)

# Đưa dữ liệu vào để model dự đoán

rf_pre = rf_model.predict(x_valid)
x_valid.head()
```

	LotArea	YearBuilt	1stFlrSF	2ndFlrSF	FullBath	BedroomAbvGr	TotRmsAbvGrd
529	32668	1957	2515	0	3	4	9
491	9490	1941	958	620	1	3	5
459	7015	1950	979	224	1	3	5
279	10005	1977	1156	866	2	4	8
655	1680	1971	525	567	1	3	6

Đánh giá mô hình

```
# evolution
pd.DataFrame({'y_valid': y_valid, 'y_preds': rf_pre})
```

	y_valid	y_preds
529	200624	271690.00
491	133000	155039.00
459	110000	122024.00
279	192000	188915.00
655	88000	91147.00
...	...	...
326	324000	275931.87
440	555000	478954.15
1387	136000	176038.00
1323	82500	78096.00
61	101000	81104.00

292 rows × 2 columns

```
[ ] # Đưa một dữ liệu bất kì cho mô hình dự đoán

rf_model.predict([[6969, 2021, 1000, 800, 4, 5, 8]])
```


D, Kết quả mô phỏng v

So sánh kết quả

- Decision tree

	y_valid	y_preds
529	200624	335,000.00
491	133000	140,200.00
459	110000	119,000.00
279	192000	207,500.00
655	88000	112,000.00

Randomforest

	y_valid	y_preds
529	200624	271690.00
491	133000	155039.00
459	110000	122024.00
279	192000	188915.00
655	88000	91147.00

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 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 [illegible] được nhắc đến khá nhiều gần đây, mà tiêu biểu nhất như t[illegible] của google. Không chỉ trong lĩnh vực trí tuệ nhân tạo, mà [illegible]cũng đang có sự tham gia của ML, và kiến thức về ML thì r[illegible] trong khuôn khổ bài viết này, tôi sẽ chỉ nhắc tới một vài kh[illegible]e 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.

a) *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 thực 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 k24 facebook, phân loại spam email án y khoa, phát hiện thẻ tín dụng ứng khoán, dự đoán kết quả trận đ ân 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 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.