Xác định giá nhà đất trên thị trường dựa trên mô hình máy học

Nhóm trình bày: 05

Nguyễn Thị Thu Trang - B19DCVT405

Nguyễn Hồng Đức - B19DCVT096

Trần Thành Trung - B19DCVT421

Bùi Trung Đức - B19DCVT090

Nguyễn Văn Nguyên – B19DCVT277

Các phương thức dự đoán giá nhà đất truyền thống

I. 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.

I. Introduction

- 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

Thông tin về 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
```

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

2. Tiền xử lý dữ liệu

- 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)
```

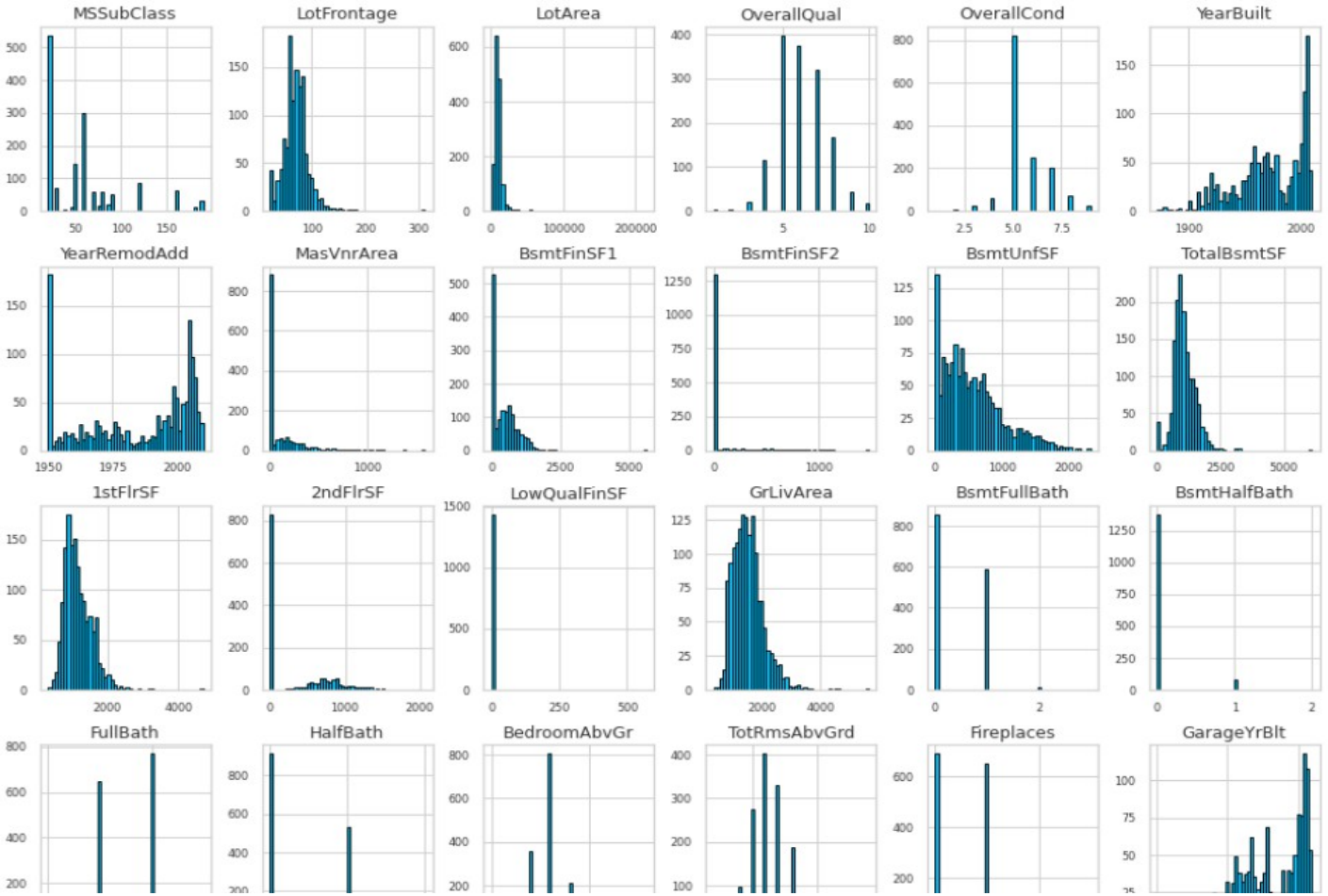
	MSSubClass	MSZoning	LotFrontage	LotArea	Street	Alley	LotShape	LandContour	Utilities	LotConfig	...	PoolArea	PoolQC	Fence	MiscFe
0	60	RL	65.0	8450	Pave	NaN	Reg	Lvl	AllPub	Inside	...	0	NaN	NaN	
1	20	RL	80.0	9600	Pave	NaN	Reg	Lvl	AllPub	FR2	...	0	NaN	NaN	
2	60	RL	68.0	11250	Pave	NaN	IR1	Lvl	AllPub	Inside	...	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

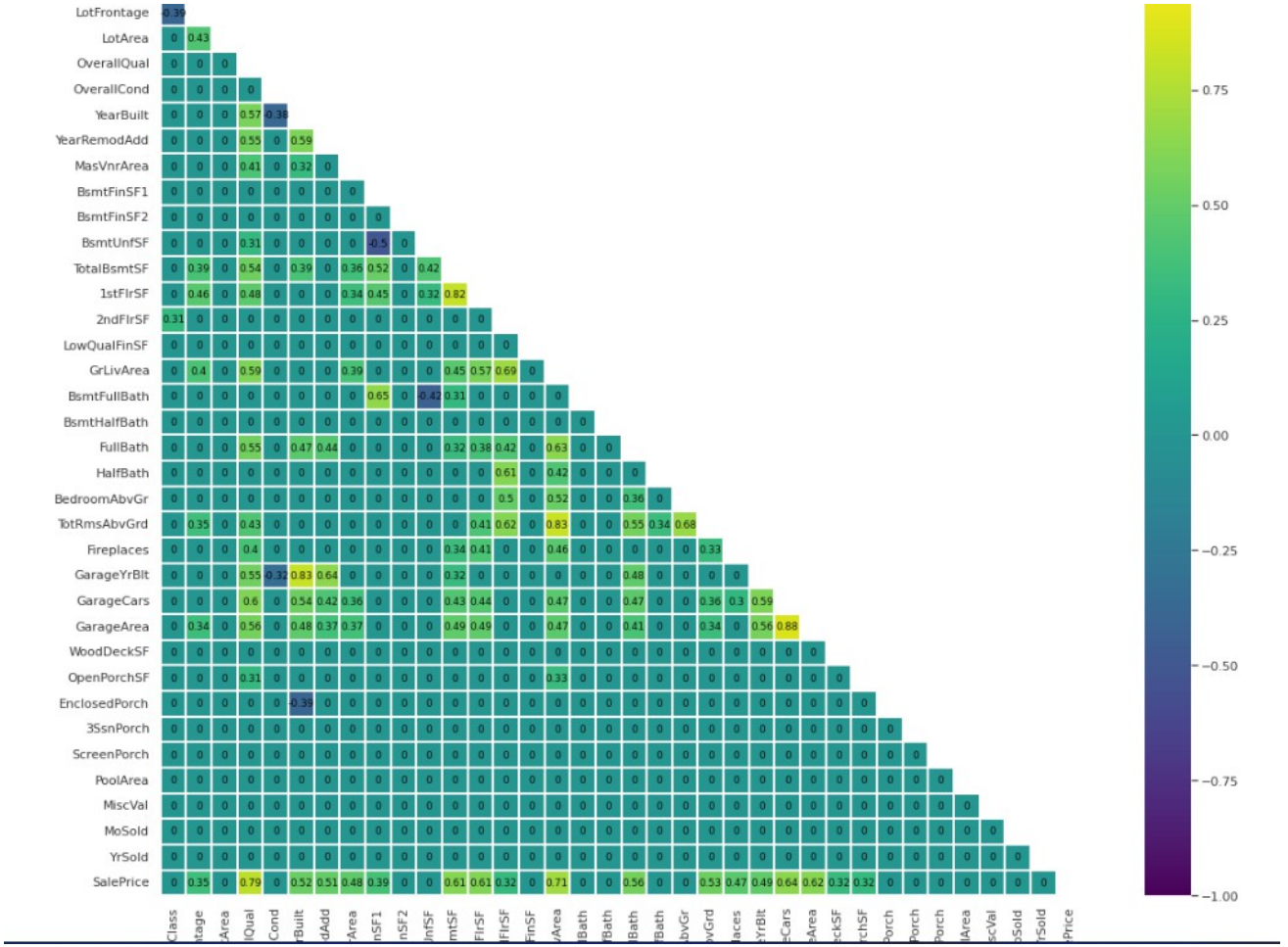
bỏ các tính năng gần như không đổi trong đó 95% giá trị là tương tự hoặc không đổi hay nói là chúng có độ tương quan cao

```
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 dữ liệu số



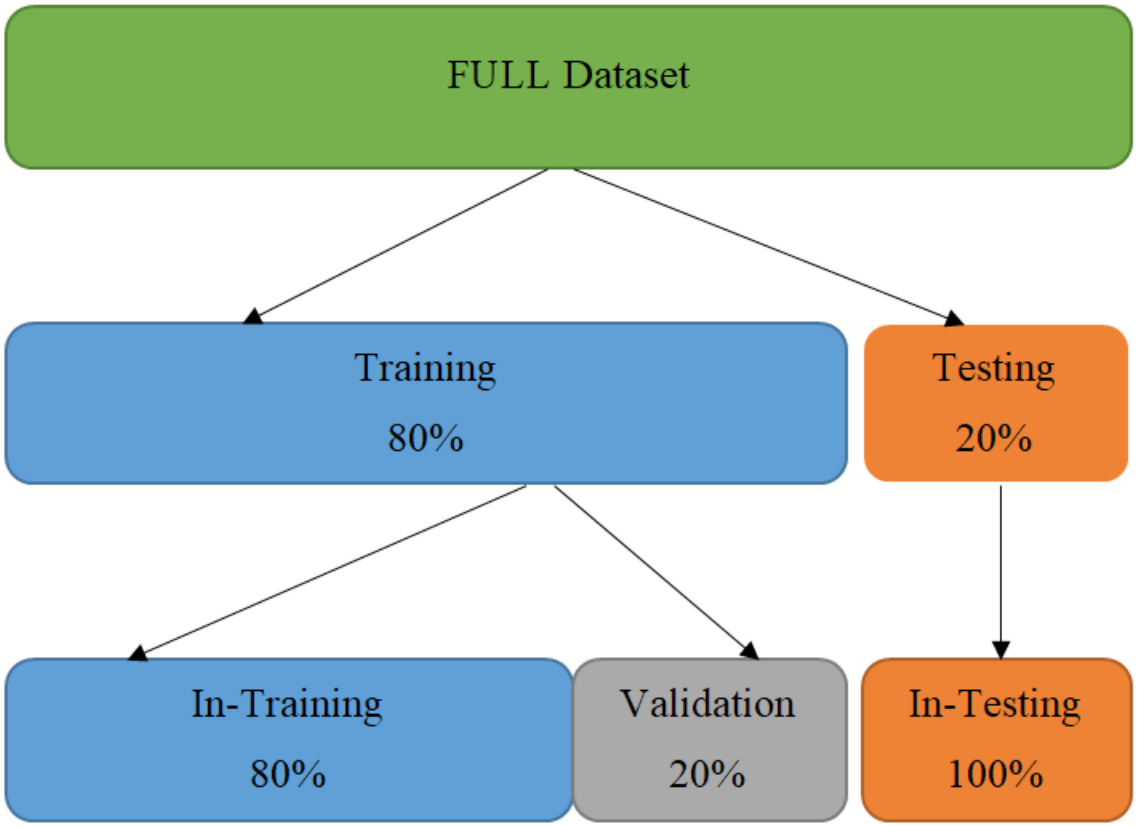
Xử dụng heatmap để đưa ra rõ hơn về độ tương quan của các đặc điểm với giá bán và tương quan giữa các đặc điểm với nhau



Feature selection

- 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



3. Method

- 1. Xử dựng thuật toán decision tree
- 2. Xử dụng thuật toán randomforest

1. Decision tree

- Cây quyết định là một thuật toán tham lam, nó chọn con đường tốt nhất, tối đa hóa thông tin thu được, nó sẽ không quay lại và thay đổi phân tách trước đó

- **ENTROPY**
- Là thước đo lượng thông tin chứa trong một trạng thái.
- Tìm mức tăng thông tin tương ứng với một phép tách, chúng ta cần trừ entropy k ết hợp của các nút con khỏi entropy của nút cha
-
-

$$IG = E(parent) - \sum \omega_i E(child_i)$$

GINI

- 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 thôn g tin tối đa hay không

Gini Index = $1 - \sum p_i^2$ với p_i = probability of class i

- 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

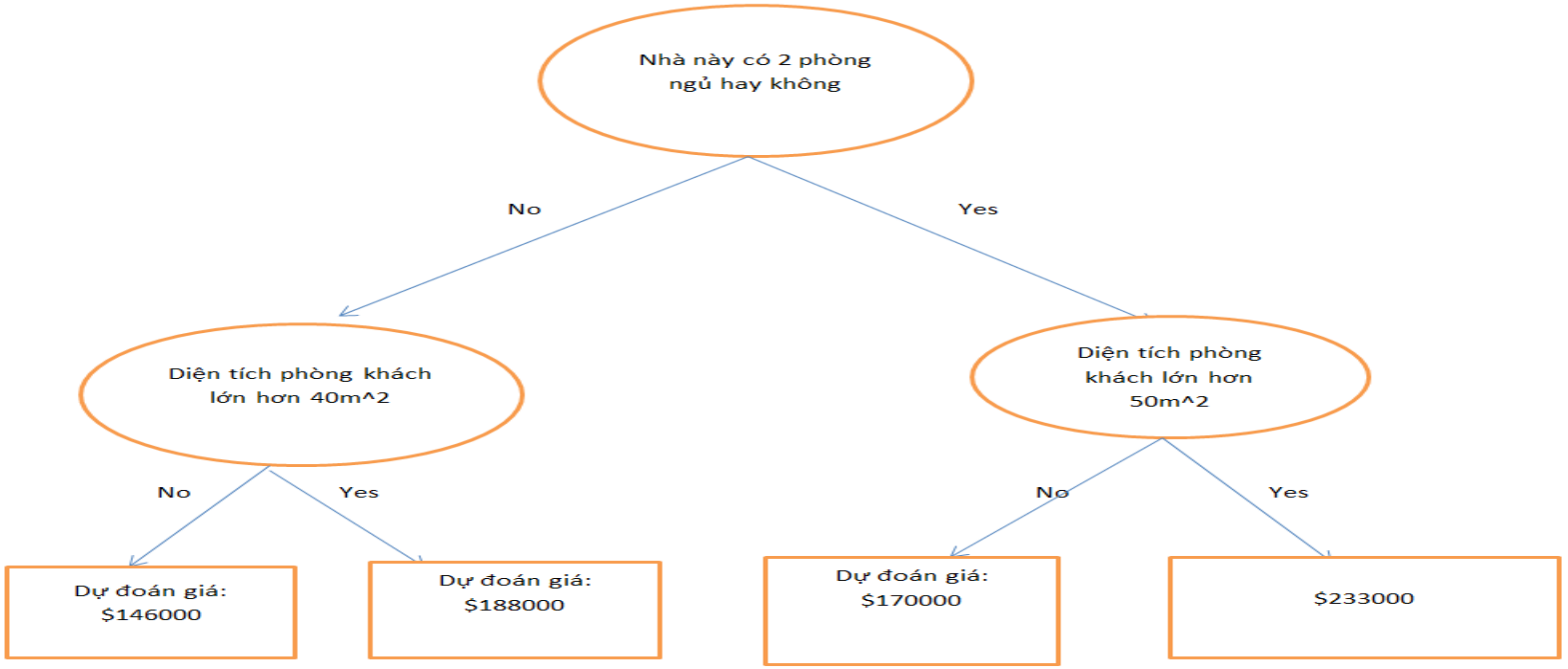
$$\text{Var} = \frac{1}{n}\sum(y_i - \bar{y})^2$$

- Phương sai cao hơn có nghĩa có tạp chất nhiều hơn

$$\text{Var Red} = \text{Var(parent)} - \sum \omega_i Var(child_i)$$

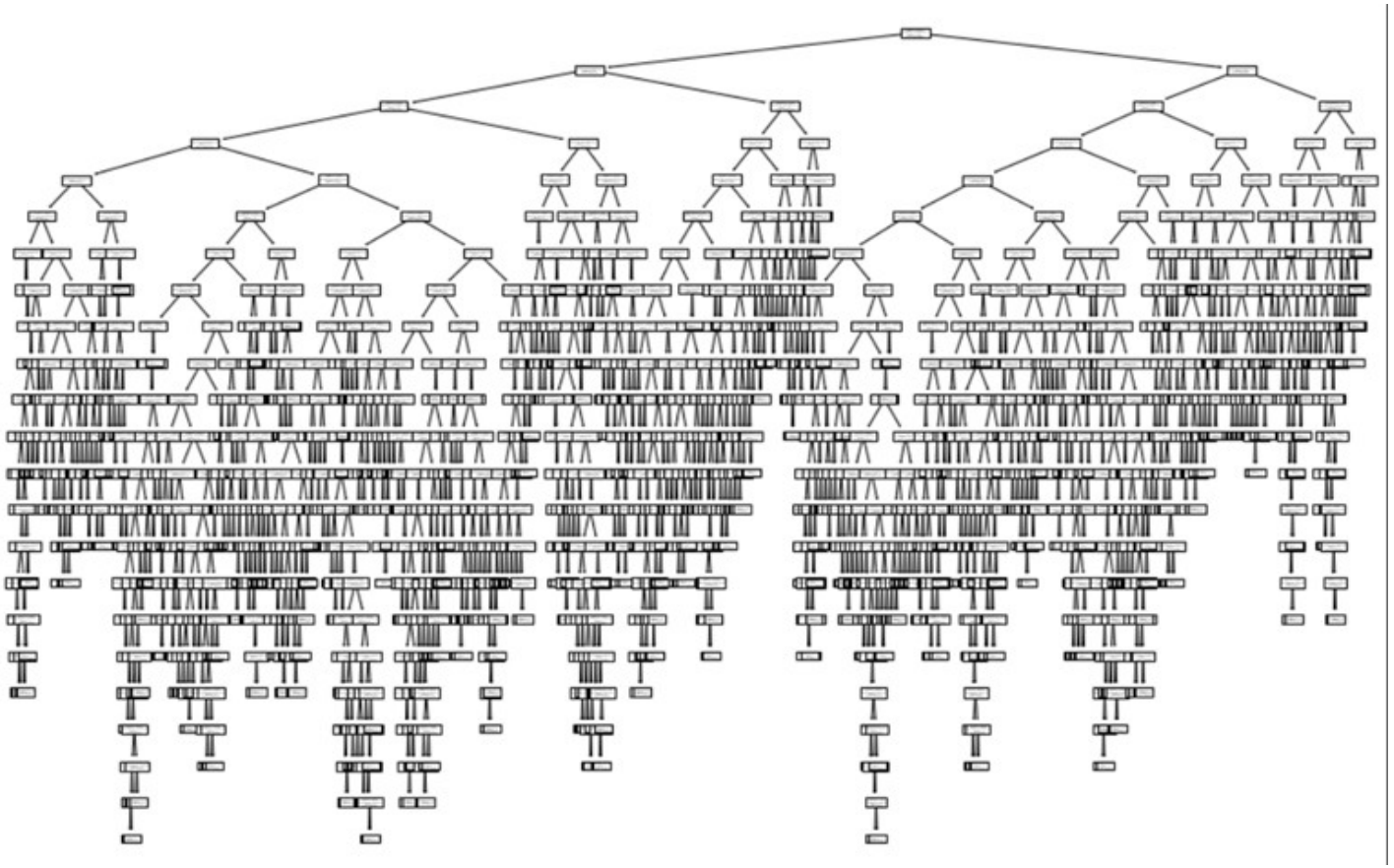
(Varian reduction: độ giảm)

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



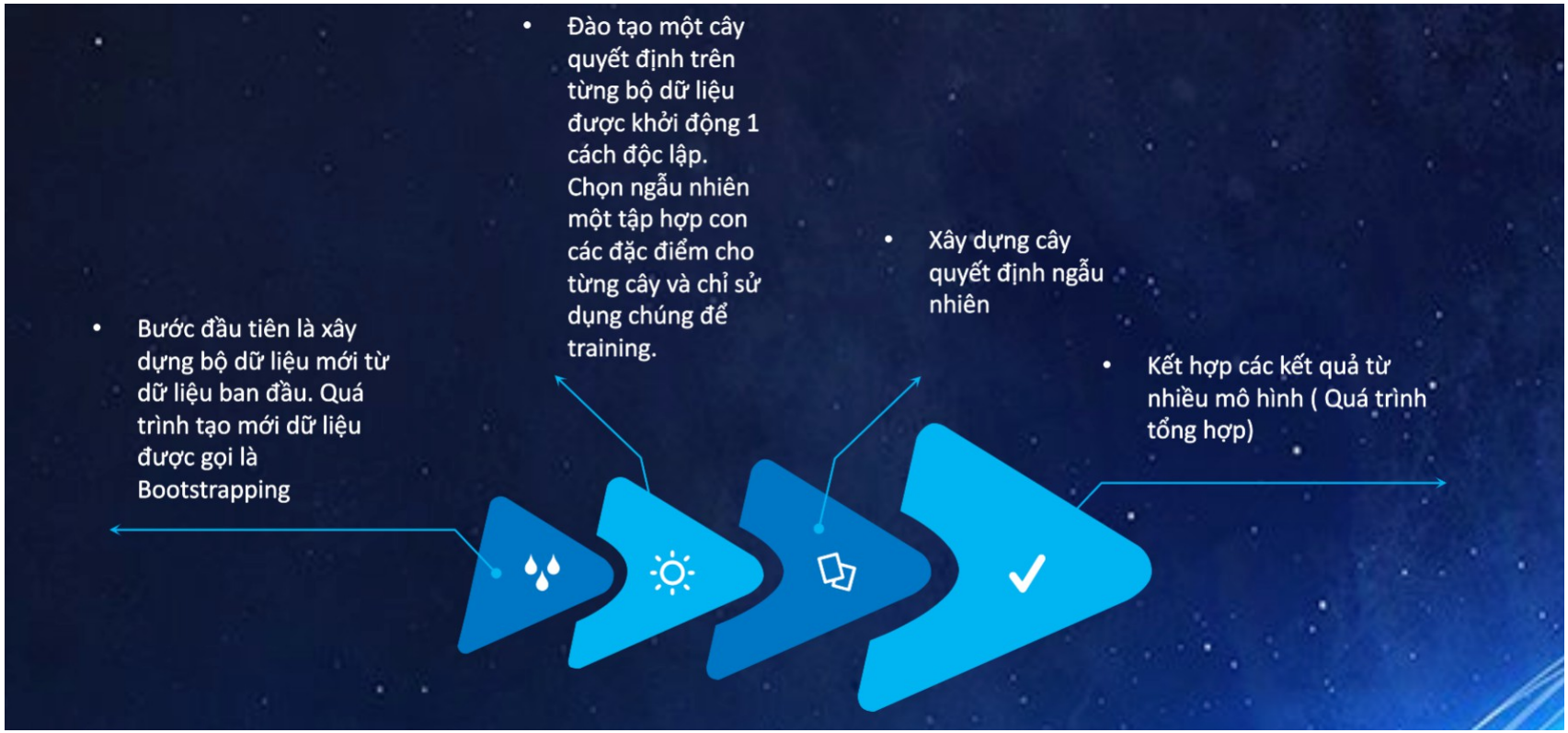
```
[26] from sklearn.tree import DecisionTreeRegressor
     dt_model = DecisionTreeRegressor(random_state=1)

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

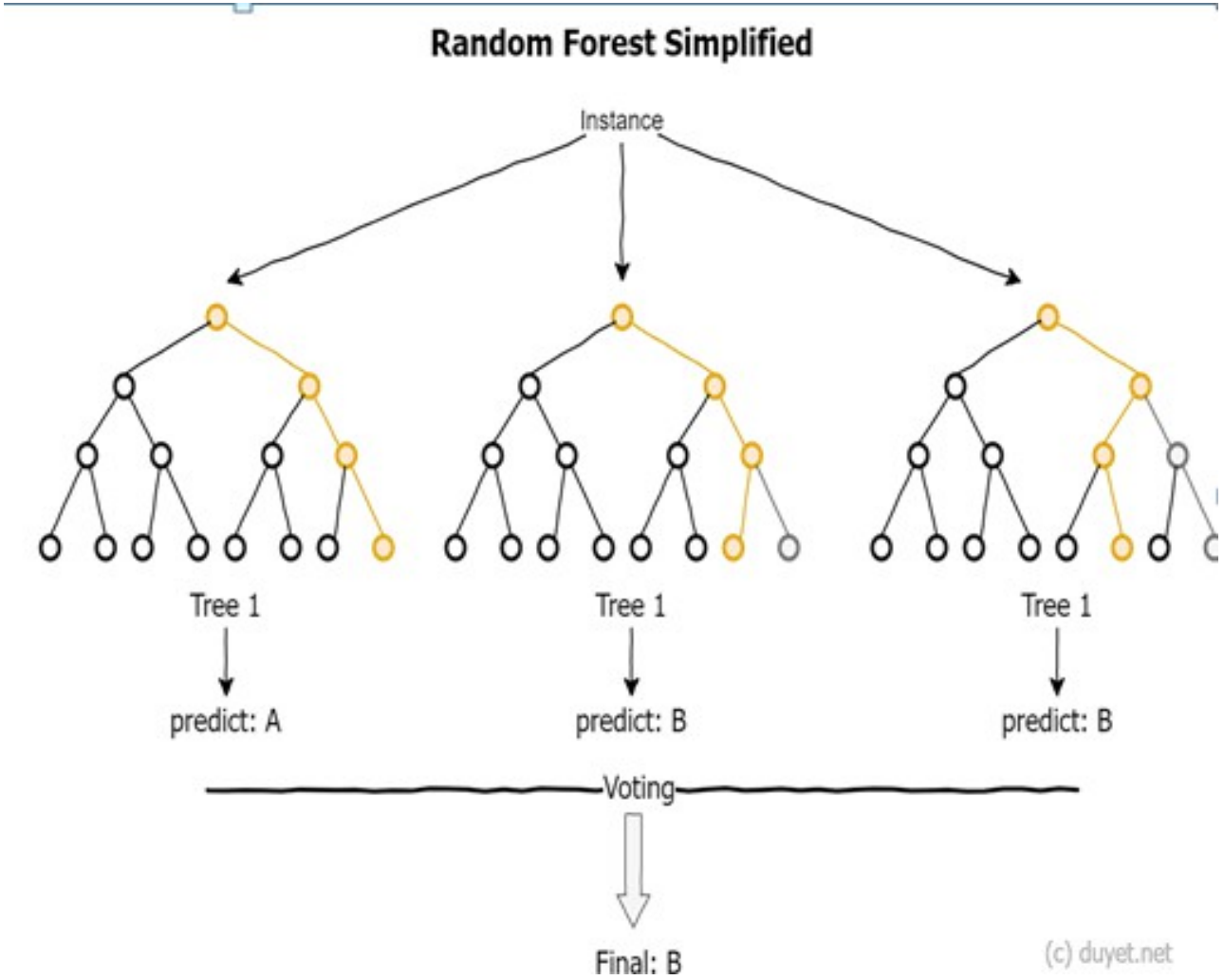
ThựC tế bài toán



2. Random forest

- Là một tập hợp của nhiều cậy quyết định ngẫu nhiên
- Và nó được tối ưu hơn so với mô hình cây quyết định




Random Forest Simplified
Instance
Tree 1
Tree 1
Tree 1
predict: A
predict: B
predict: B
Voting
Final: B
(c) duyet.net

```python
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()
```

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.