Phân tích và Dự đoán Giá Nhà với Hồi quy Nâng cao

Nguyễn Đăng Khoa, Lê Thị Trúc Ly, Lê Đoàn Kim Ngân, Lâm Tú Nhi

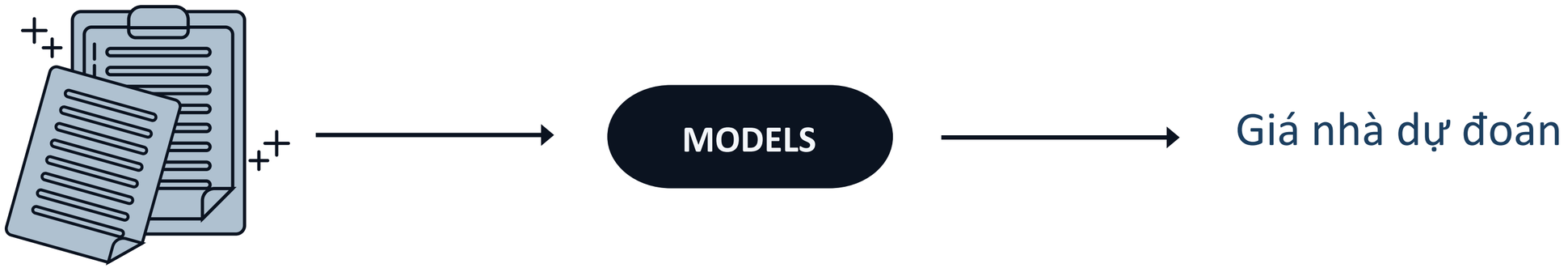
Nội dung trình bày

- ***Giới thiệu***
- ***Các công việc liên quan***
- **Phương pháp đề xuất**
- **Thực nghiệm và Thảo luận**
- **Kết luận**

1. Giới thiệu

Vấn đề cần giải quyết

- Dự đoán giá bán của một căn nhà ở thành phố Ames, Iowa
- Dựa trên các đặc điểm vật lý và thông tin liên quan
- Đây là một bài toán hồi quy
- **Input:** Tập dữ liệu gồm 79 biến đặc trưng mô tả căn nhà
- **Output:** Dự đoán giá bán (SalePrice) của các căn nhà trong tập kiểm tra



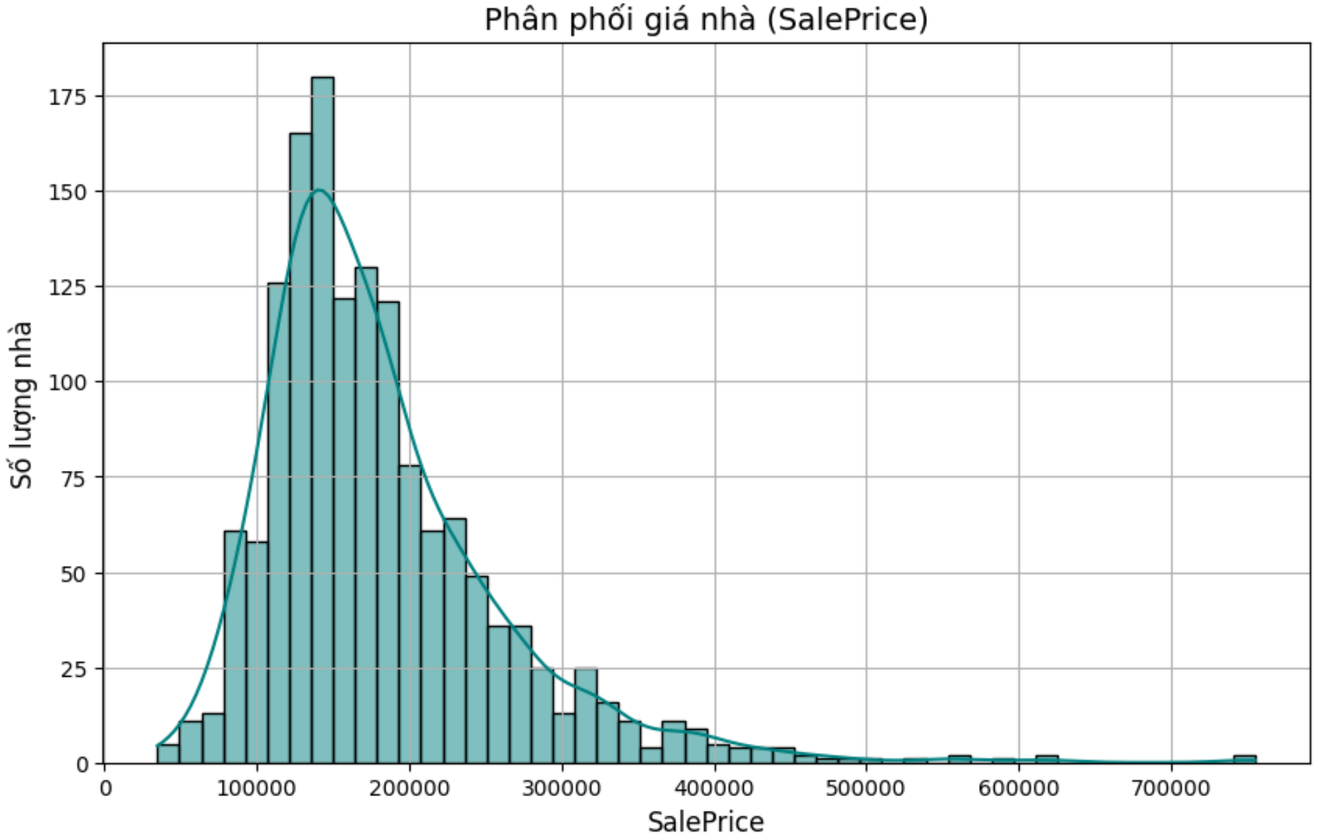
1. Giới thiệu

Dữ liệu

- Nguồn: Kaggle – House Prices Competition
- Thông tin chi tiết của các căn nhà tại thành phố Ames, bang Iowa (Mỹ)
- Bao gồm:
- Biến đặc trưng: 79 biến mô tả căn nhà

Nhóm thông tin chính

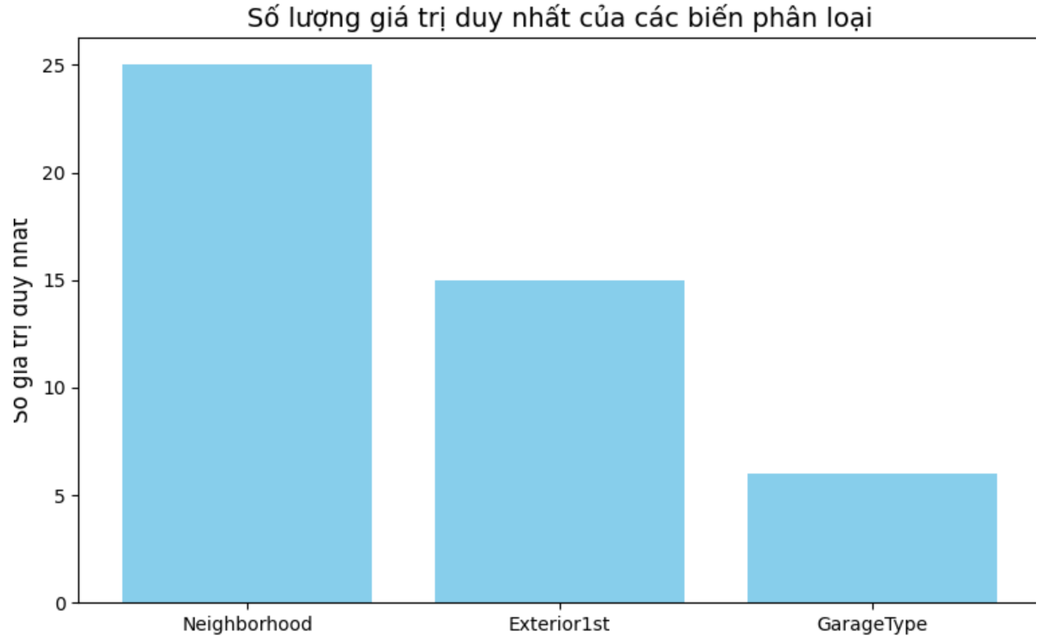
- **Vật lý:** Diện tích, số tầng, chất lượng
- **Thời gian:** Năm xây, năm sửa
- **Vị trí:** Khu phố, gần đường lớn
- **Tiện ích:** Gara, hồ bơi, tầng hầm,...



1. Giới thiệu

Thách thức

- Phân phối giá nhà bị lệch phải
- Dữ liệu thiếu và không đồng nhất
- Tương tác phi tuyến giữa các đặc trưng
- Nguy cơ overfitting với mô hình phức tạp

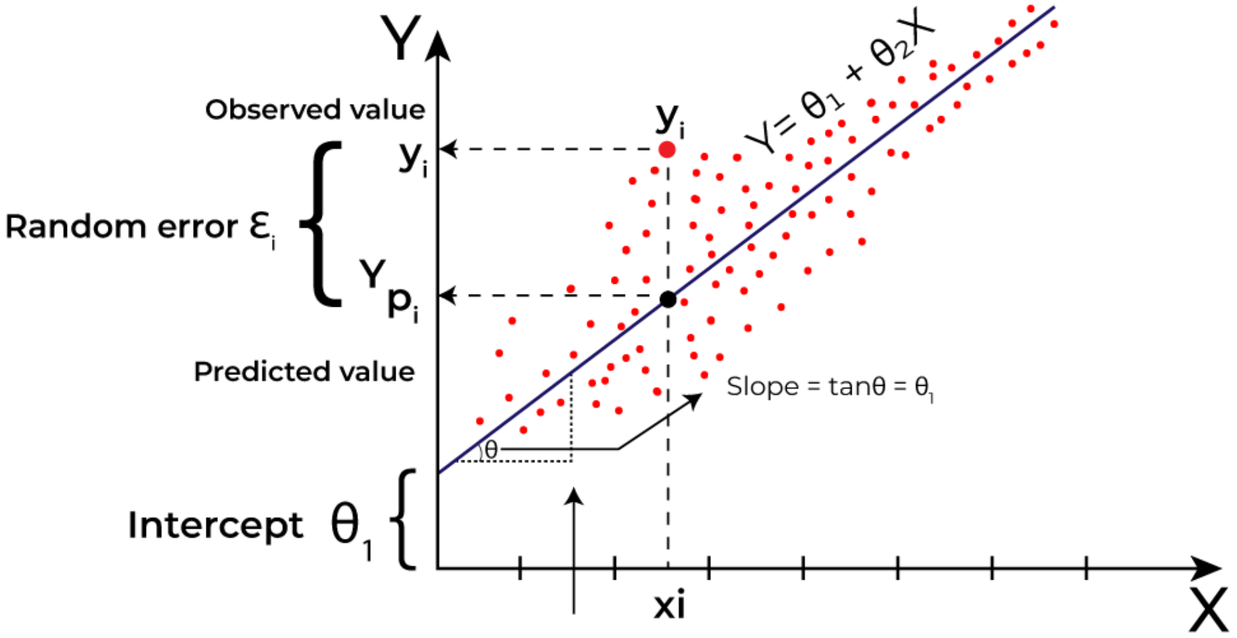


Các cột có giá trị bị thiếu

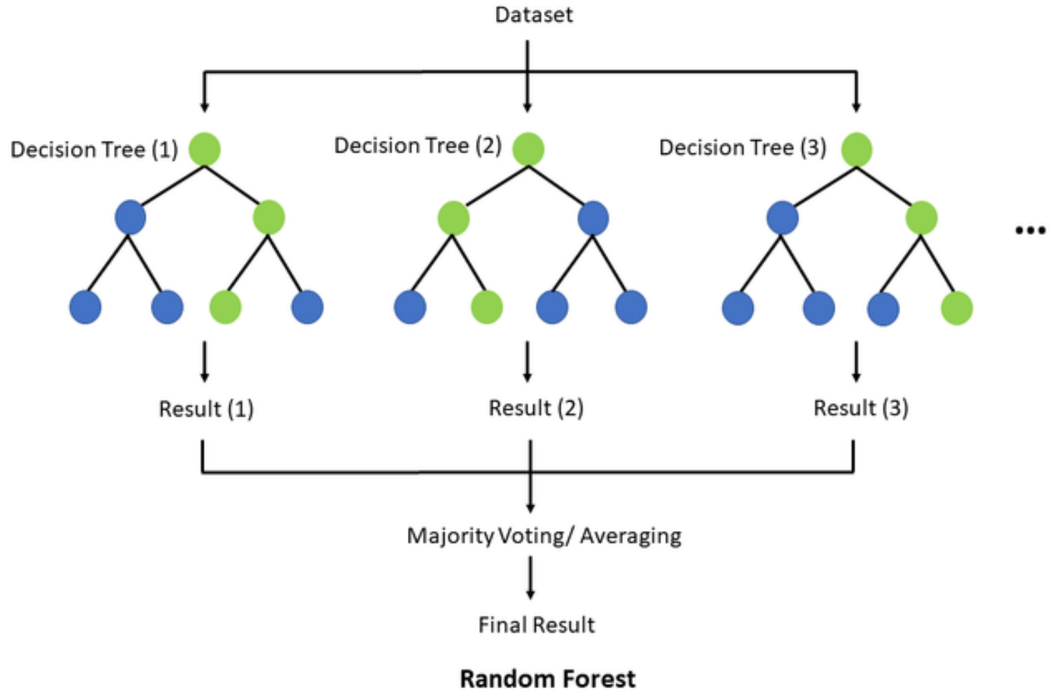
Tên cột	**Số lượng**
PoolQC	1453
MiscFeature	1406
Alley	1369
Fence	1179
MasVnrType	872
FireplaceQu	690
LotFrontage	259
GarageType	81

2. Các công việc liên quan

Các mô hình



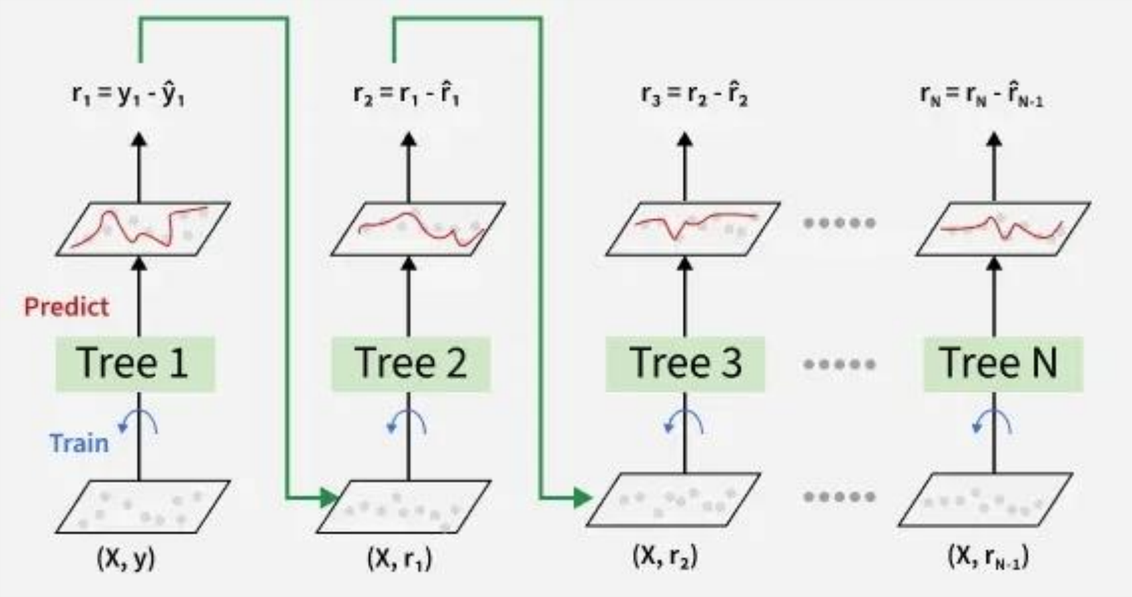
Linear Regression : Hồi quy tuyến tính cơ bản, dễ hiểu, dễ triển khai



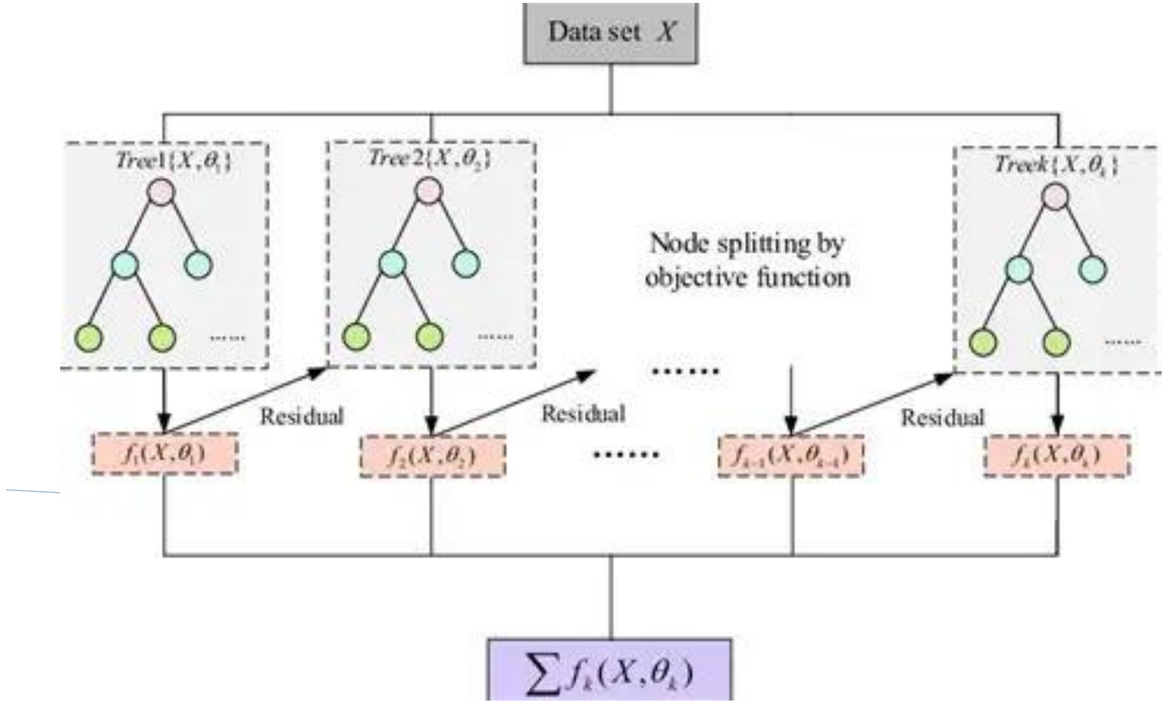
Random Forest: Ensemble từ nhiều cây quyết định, giảm overfitting

2. Các công việc liên quan

Các mô hình

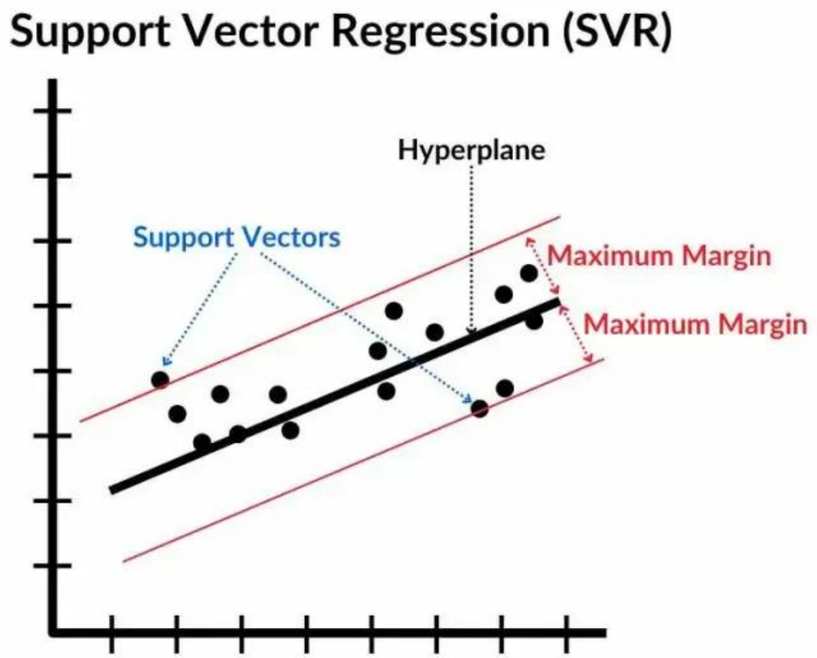


Gradient Boosting: Boosting tuần tự, học từ lỗi của cây trước

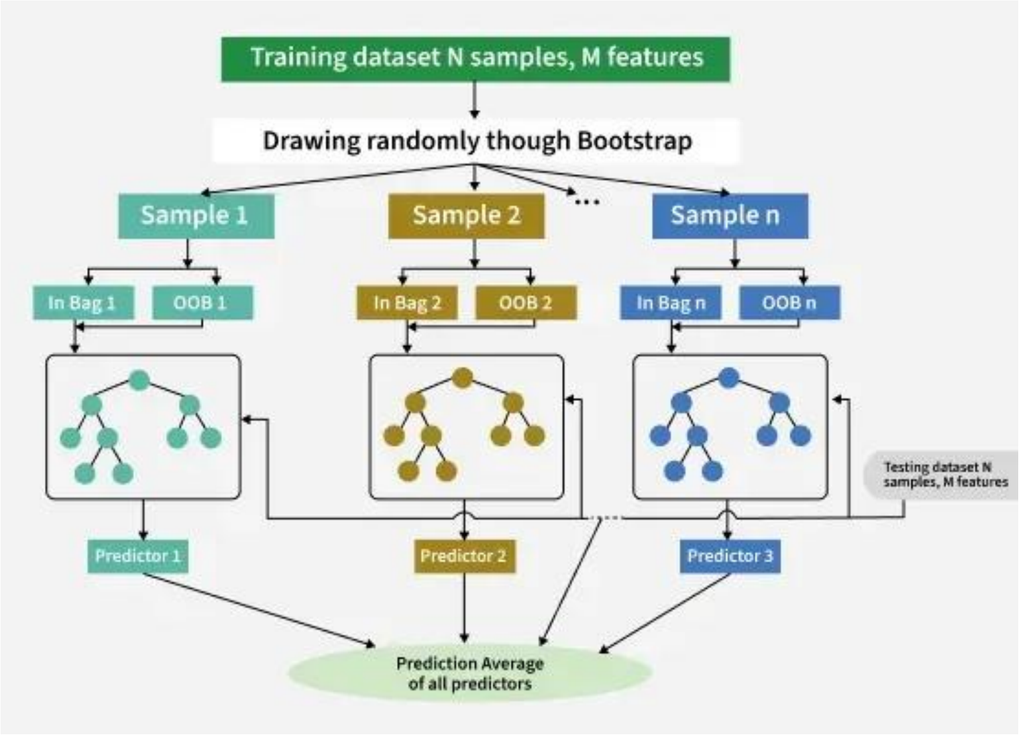


XGBoost: Tối ưu hóa từ GB, nhanh, có regularization, xử lý missing tốt

2. Các công việc liên quan



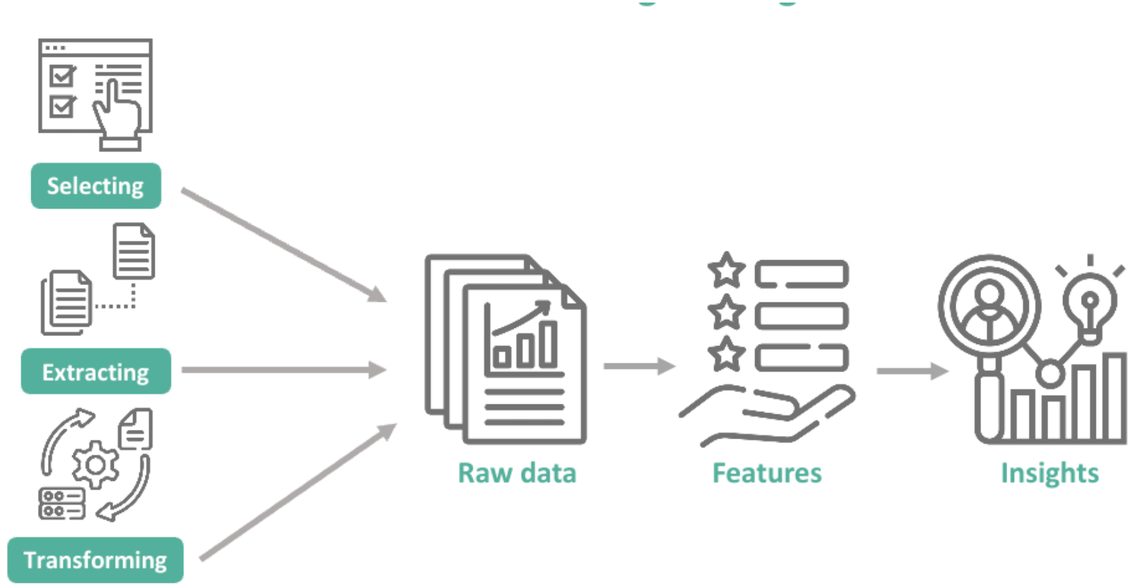
SVR: Biến thể hồi quy của SVM, dùng kernel để bắt quan hệ phi tuyến



CatBoost: Boosting tối ưu cho dữ liệu phân loại, tự xử lý categorical

2. Các công việc liên quan

Các kỹ thuật tiền xử lý dữ liệu



Mục tiêu:

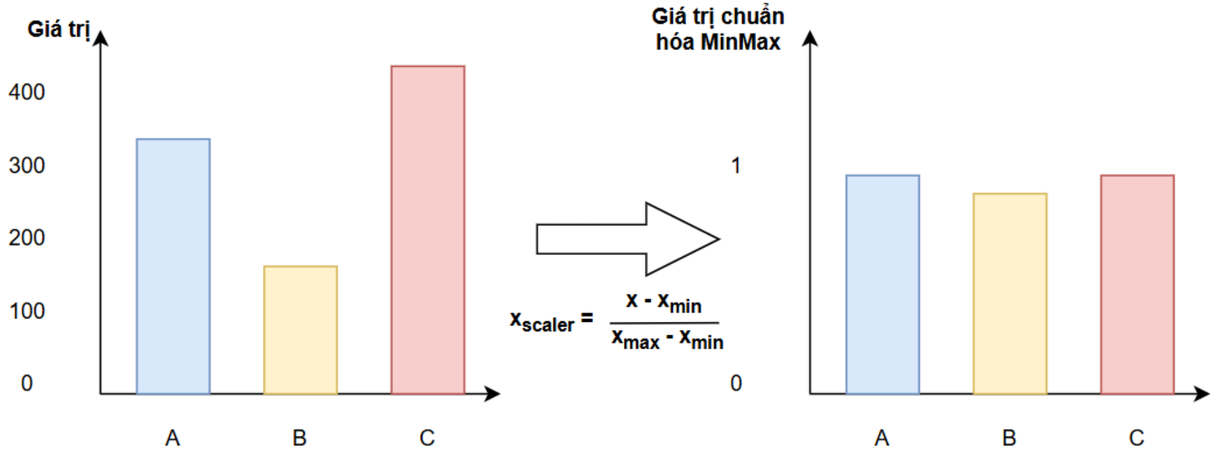
Xử lý các giá trị Missing trong data sao cho dữ liệu không còn giá trị thiếu (NaN) trước khi encode / scale.

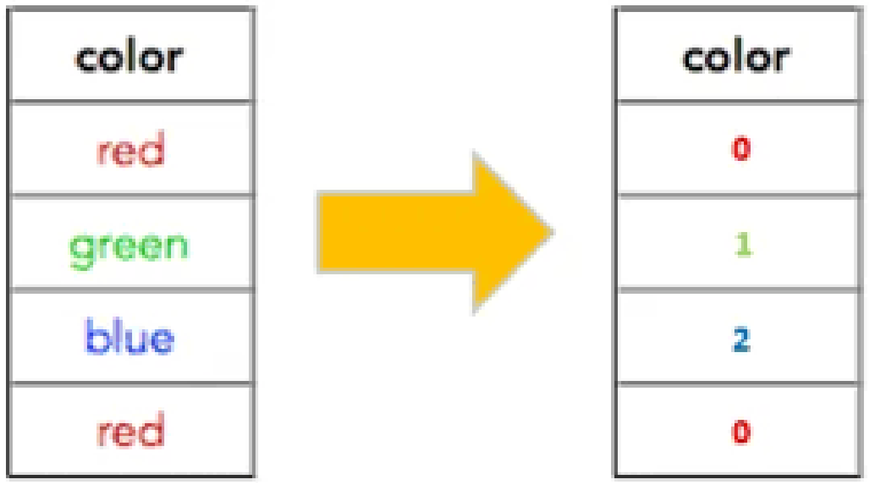
- Xử lý missing values theo ngữ nghĩa (None, 0, median)
- Tách nhóm biến số và phân loại để xử lý riêng
- Xây dựng pipeline xử lý dữ liệu

2. Các công việc liên quan

Các kỹ thuật tiền xử lý dữ liệu

- **Chuẩn hóa dữ liệu:** biến đổi các giá trị số trong cùng trường dữ liệu về cùng một thang đo, giúp mô hình ổn định hơn. Hai loại thường được sử dụng là **MinMaxScaler** và **StandardScaler**.

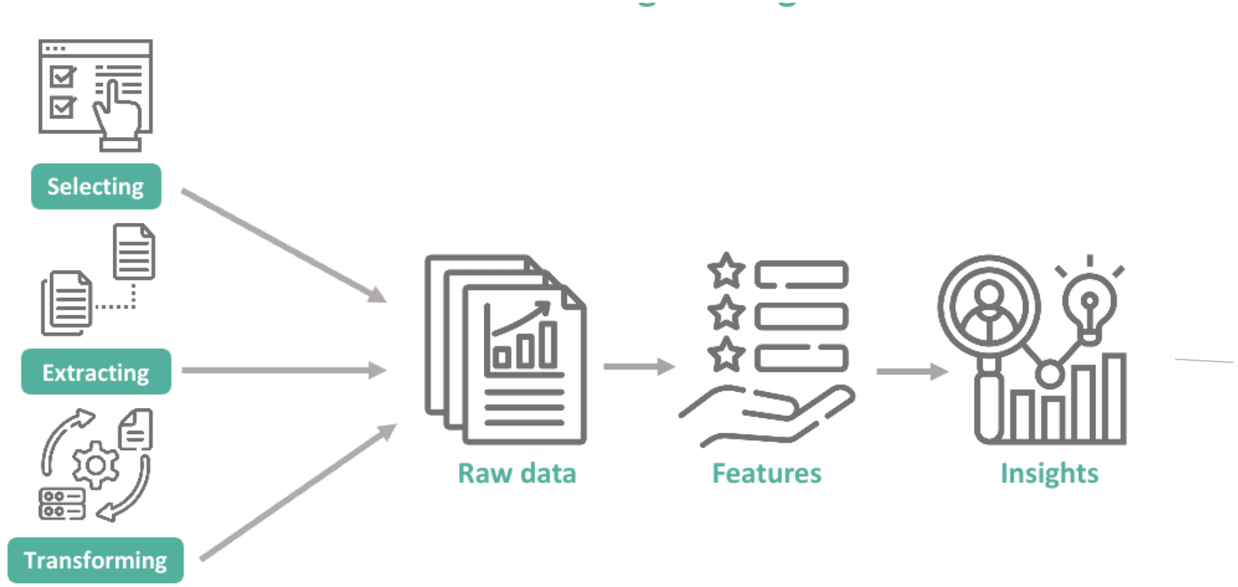




Mã hóa dữ liệu: Là kỹ thuật chuyển đổi dữ liệu định tính (**categorical**) thành dạng số, giúp mô hình học máy có thể hiểu và xử lý được. Hai phương pháp mã hóa phổ biến là **Label Encoding** và **One-Hot Encoding.**

2. Các công việc liên quan

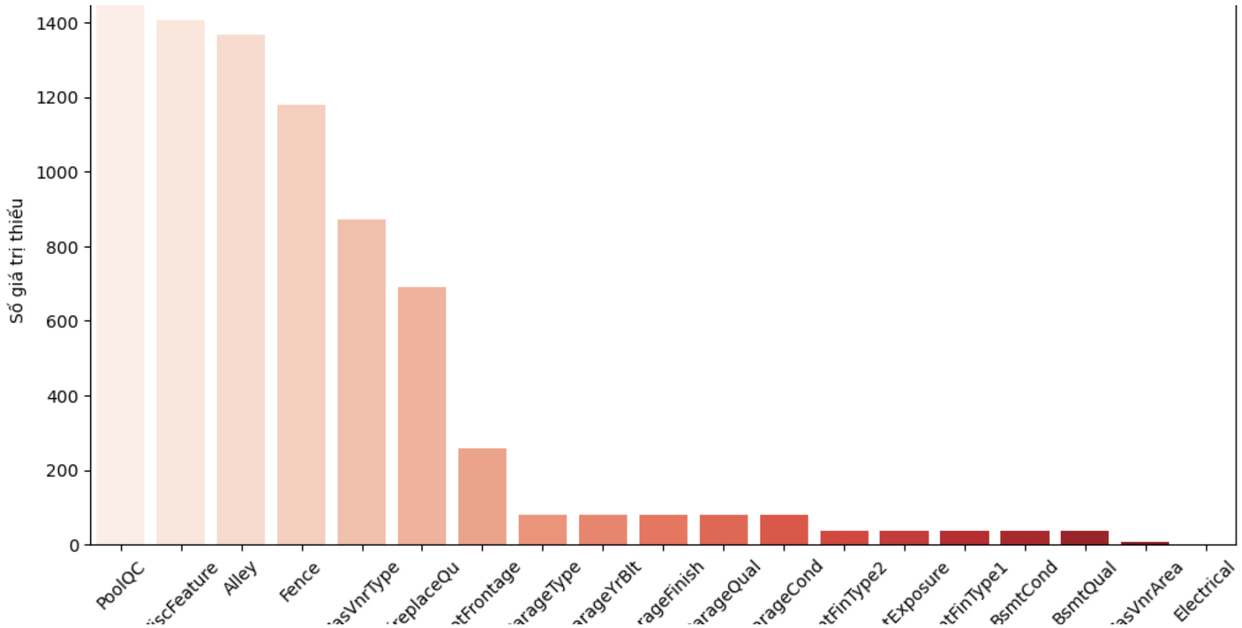
Feature Engineering Approaches



- Tạo các biến tổng hợp như TotalSF, TotalBath, TotalSpace để phản ánh quy mô và tiện nghi của căn nhà.
- Tạo các biến nhị phân (HasPool, HasGarage, v.v.) để mô hình nhận biết sự hiện diện của các tiện ích.

3. Các phương pháp đề xuất

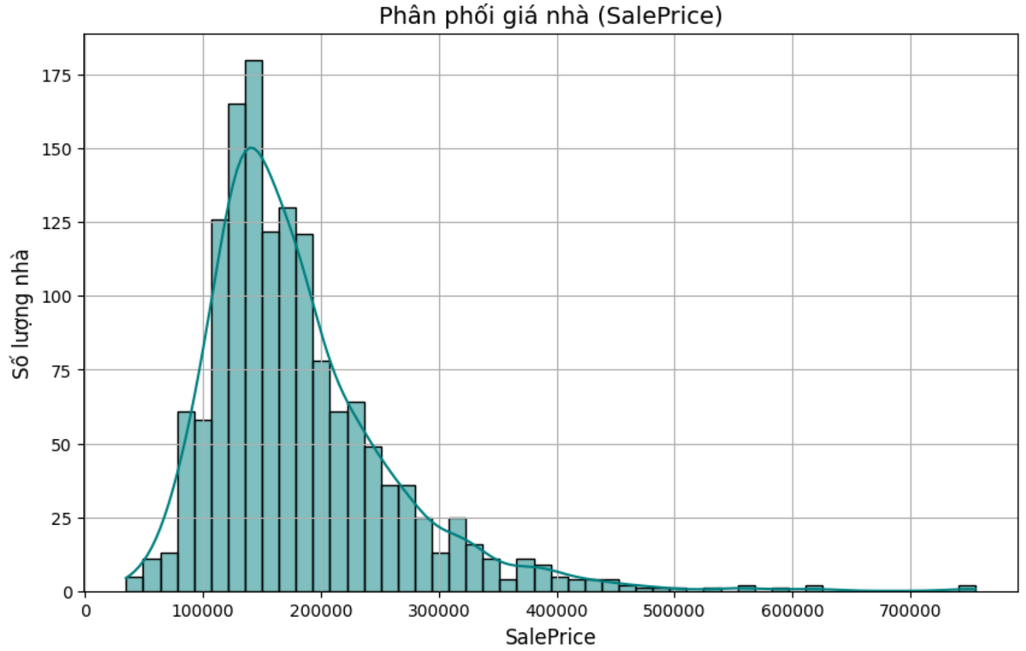
Exploratory data analysis (EDA)



Nhóm đặc trưng	Cột tiêu biểu	Ý nghĩa của giá trị thiếu
Không có tiện ích	Alley, PoolQC, Fence, FireplaceQu, MiscFeature	Nhà không có tiện ích đó → điền 'None'
Không có vật liệu	MasVnrT	Không có ốp ngoài → điền 'None' hoặc 0
Tầng hầm & Gara	BsmtQual, GarageType, GarageYrBlt, v.v.	Không có tầng hầm/gara → điền 'None' hoặc 0
Thông tin chưa ghi	LotFron, Electrical	Thiếu thông tin → điền median hoặc mode

3. Các phương pháp đề xuất

Exploratory data analysis (EDA)




```
# Tính skew
saleprice_skew = skew(train_df['SalePrice'])
print(f"Skewness của SalePrice: {saleprice_skew:.2f}")
```

```
Skewness của SalePrice: 1.88
```

Nhận xét

- Skewness = 1.88 → phân phối lệch phải mạnh
- Hầu hết nhà có giá thấp đến trung bình
- Một số nhà rất đắt tiền tạo "đuôi dài" → ảnh hưởng đến mô hình

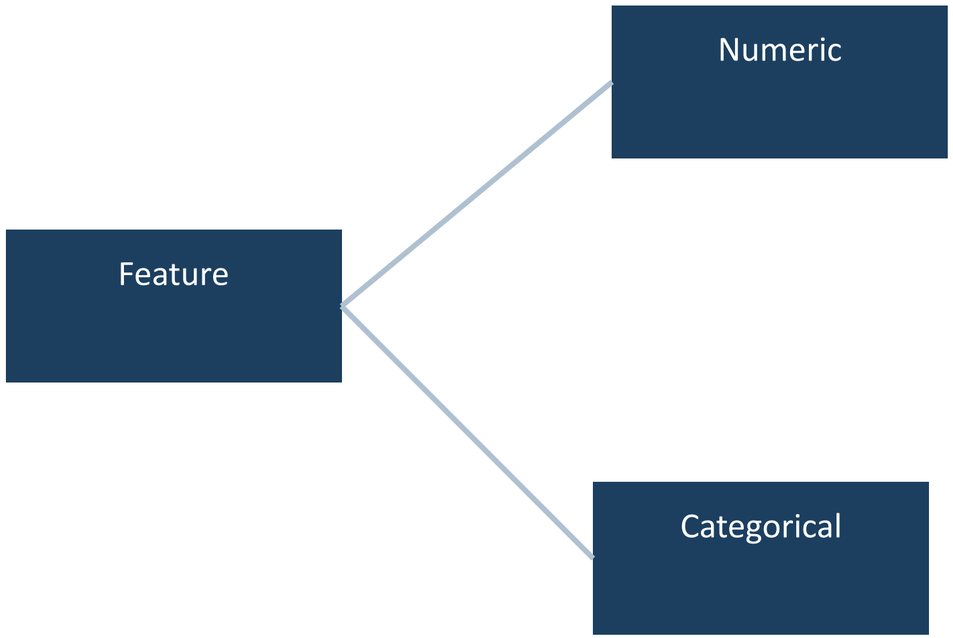
Giải pháp

- Áp dụng log transform: np.log1p(SalePrice)

→ Giúp phân phối gần chuẩn hơn, giảm ảnh hưởng outliers

3. Các phương pháp đề xuất

Exploratory data analysis (EDA)



Biến số - Numeric : 38 cột

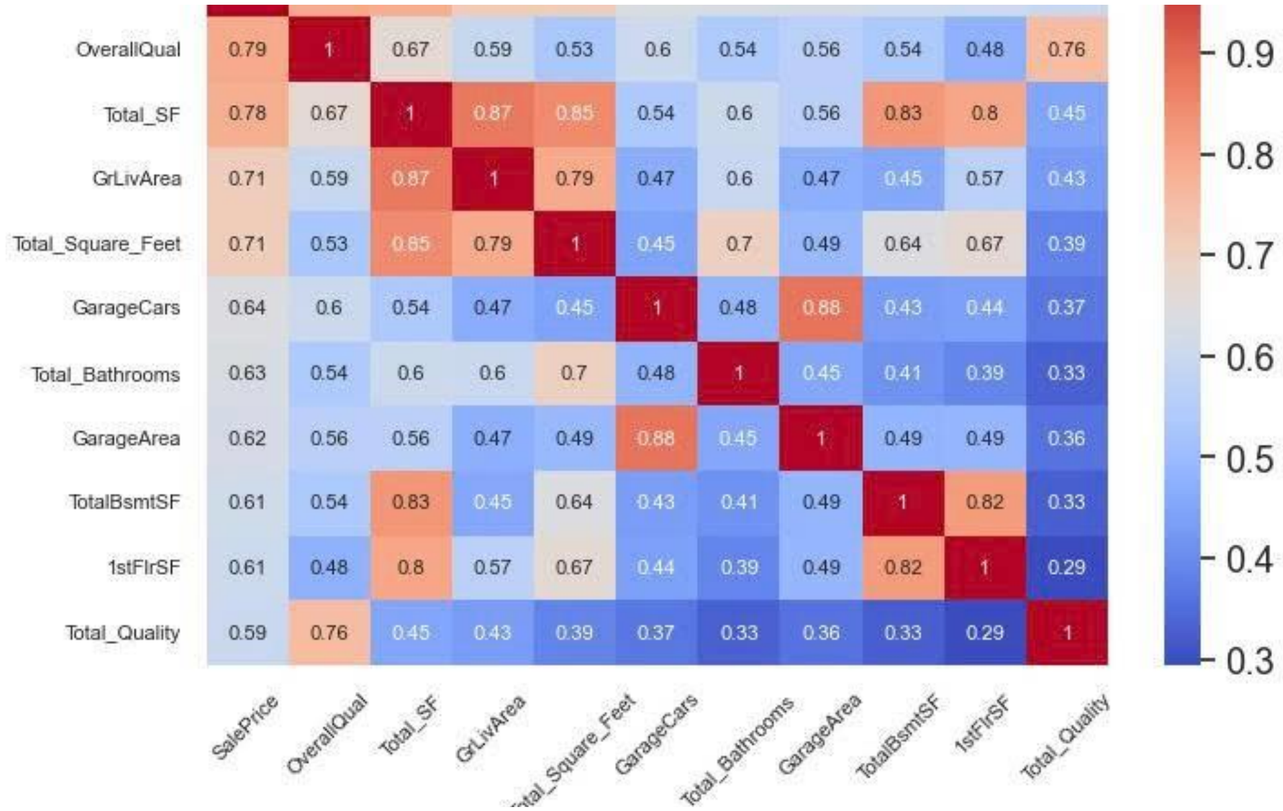
- **Liên tục:** diện tích, giá trị
- **Rời rạc:** số lượng phòng, năm xây, số tầng

Biến phân loại - Categorical : 43 cột

- **Vị trí & quy hoạch:** Neighborhood, MSZoning, Street
- **Kiến trúc và vật liệu:** HouseStyle, RoofStyle , Foundation v.v...
- **Tiện nghi & giao dịch:** FireplaceQu, PoolQC, SaleCondition, v.v.

3. Các phương pháp đề xuất

Exploratory data analysis (EDA)

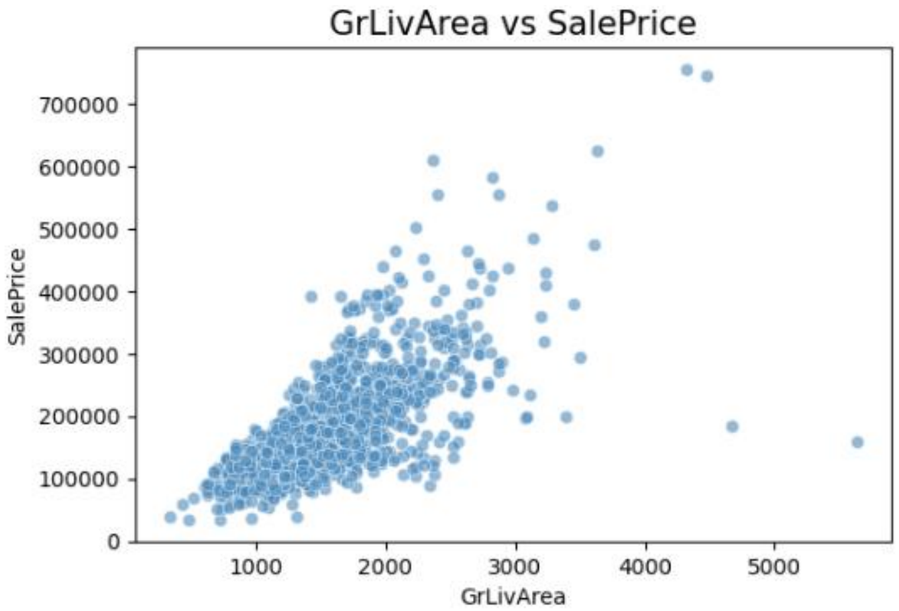


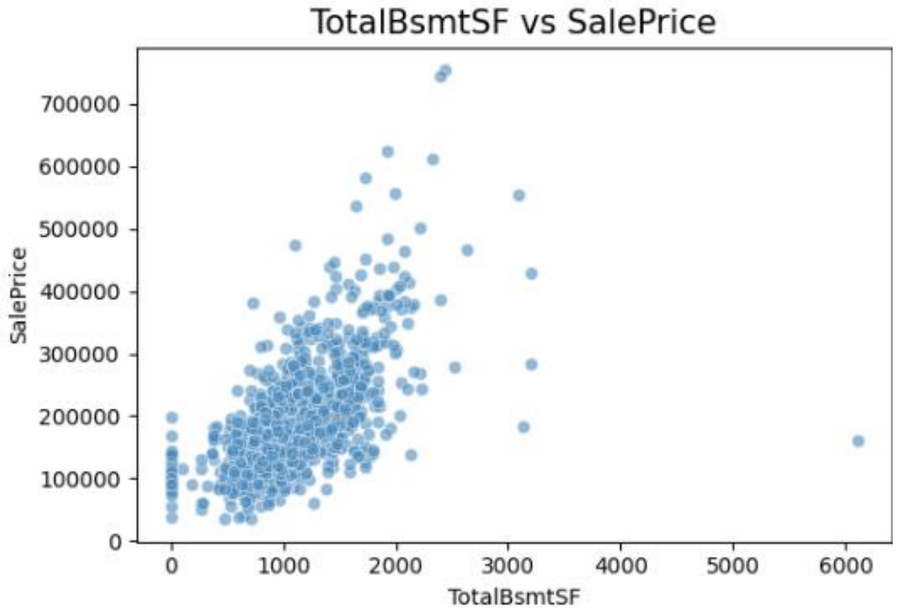
Top 10 đặc trưng có tương quan cao nhất

Biến số	**Hệ số tương quan**	Ý nghĩa
OverallQual	0.79	Chất lượng tổng thể nhà
Total_SF	0.78	Diện tích sinh sống
GrLivArea	0.71	Số chỗ đậu xe
Total_Square_Feet	0.71	Diện tích garage
GarageCars	0.64	Diện tích tầng hầm
Total_Bathrooms	0.63	Diện tích tầng 1
GarageArea	0.62	Số phòng tắm đầy đủ
TotalBsmtSF	0.61	Tổng số phòng
1stFirSF	0.61	**Năm xây nhà**
Total_Quality	0.59	Năm cải tạo nhà

3. Các phương pháp đề xuất

Exploratory data analysis (EDA)



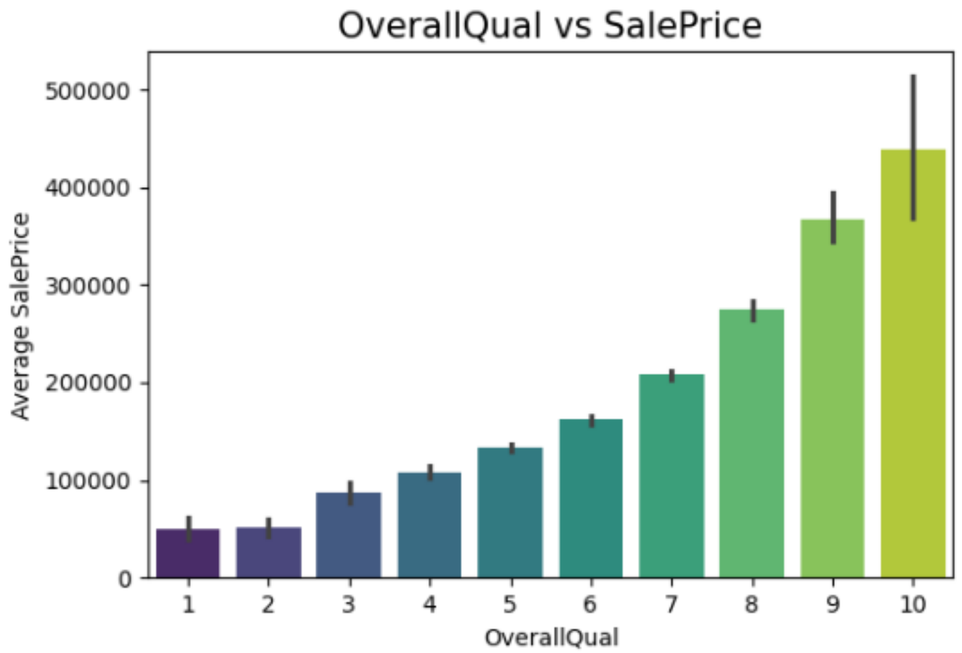


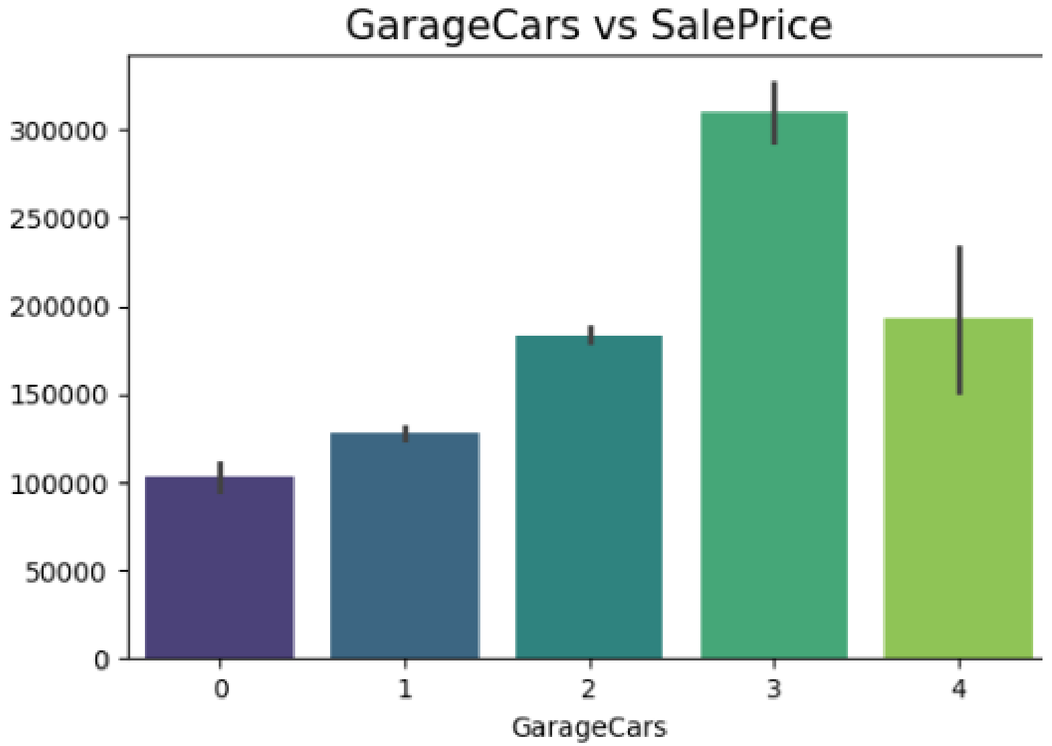
→ Diện tích sinh sống càng lớn → giá bán càng cao
→ Quan hệ gần tuyến tính, có vài outlier

→ Diện tích tầng hầm lớn → giá nhà tăng rõ rệt
→ Tầng hầm là yếu tố ảnh hưởng mạnh đến giá trị

3. Các phương pháp đề xuất

Exploratory data analysis (EDA)





Diện tích sinh sống (GrLivArea) càng lớn → giá bán (SalePrice) càng cao

Số lượng chỗ để xe trong garage (GarageCars) có ảnh hưởng rõ rệt đến giá bán nhà (SalePrice)

2. Các công việc liên quan

Tiêu chí đánh giá mô hình

1. K-Fold Cross-validation

KFold chia dữ liệu thành K phần bằng nhau

Lặp lại huấn luyện và kiểm tra K lần

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

Đo độ lệch trung bình giữa giá trị dự đoán và giá trị thực tế

RMSE càng nhỏ → mô hình càng tốt

Ý Nghĩa: Kiểm tra độ ổn định và khả năng tổng quát của mô hình

3. Các phương pháp đề xuất

Feature Engineering

Cột mới	Ý nghĩa
TotalSF	Tổng diện tích sàn
TotalPorchSF	Tổng diện tích hiên, ban công, sân thượng
TotalBath	Tổng số phòng tắm
TotalRooms	Tổng số phòng sử dụng (phòng + phòng tắm)
TotalSpace	Tổng không gian sử dụng (sàn + garage + porch)

3. Các phương pháp đề xuất

TotalSF: Tổng diện tích sàn

TotalBsmtSF	1stFlrSF	2ndFlrSF	TotalSF
856	856	854	2566
1262	1262	0	2524
920	920	866	2706

Phản ánh quy mô tổng thể của ngôi nhà
→ Nhà càng rộng, giá càng cao

TotalSpace: Tổng không gian sử dụng

TotalSF	GarageArea	TotalTorchSF	TotalSpace
2566	548	854	3968
2524	460	855	3839
2706	608	866	4180

Tổng diện tích có thể sử dụng → tương quan mạnh với SalePrice

3. Các phương pháp đề xuất

Feature Engineering

- **Biến nhị phân tiện ích: HasFireplace, HasGarage, HasPorch, HasPool, HasBsmt**

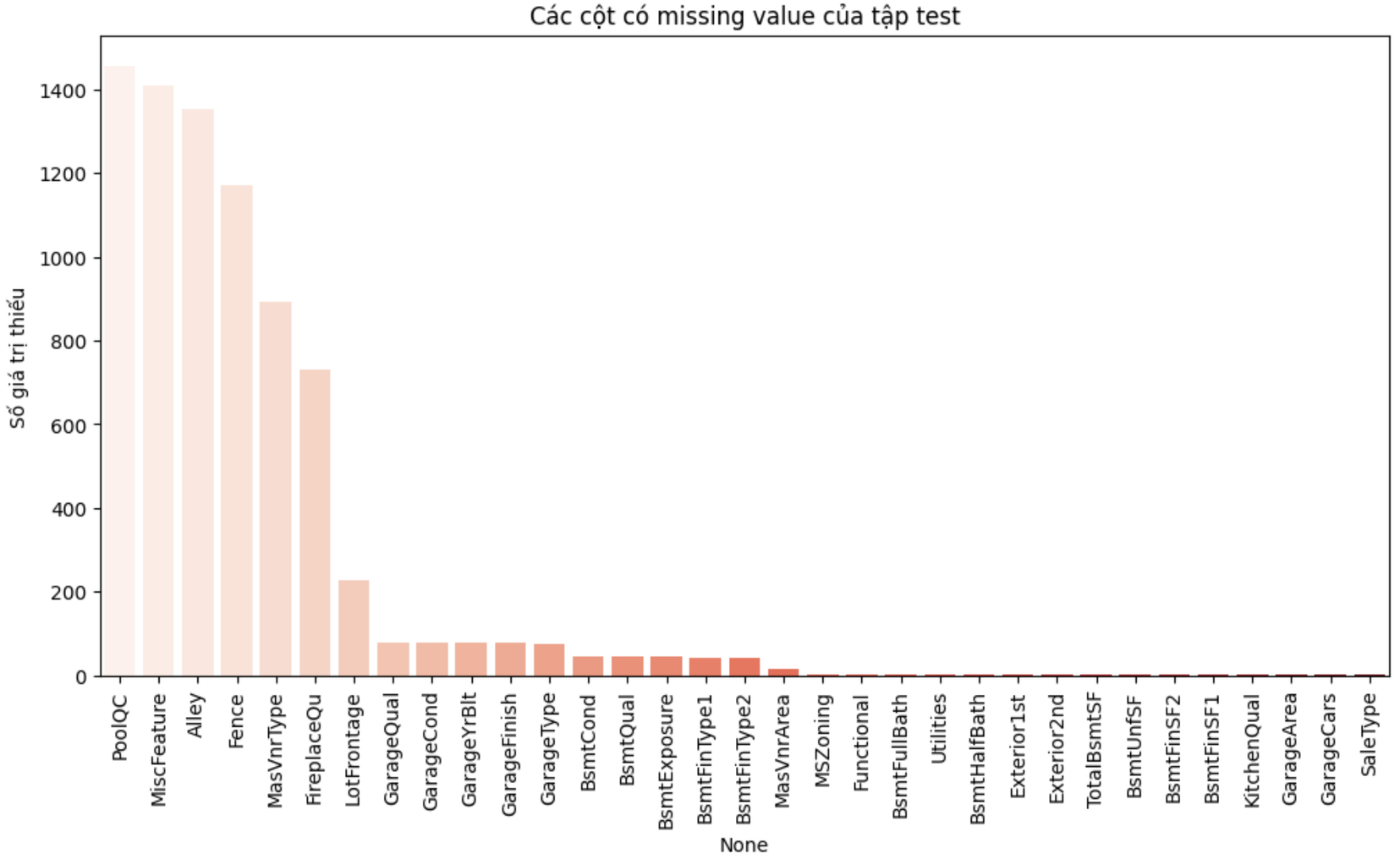
Cột mới	Ý nghĩa
HasPool	Có hồ bơi không
HasFirePlace	Có lò sưởi không
HasGarage	Có garage không
HasBsmt	Có tầng hầm không
HasPorch	Có hiên/ban công không

Ý nghĩa: Có tiện ích → tăng giá trị sống → tăng giá bán

→ Giúp mô hình hiểu rõ hơn về sự hiện diện của các tiện ích lớn ảnh hưởng đến giá

3. Các phương pháp đề xuất

Tiền xử lí dữ liệu



Dữ liệu gốc

↓

Xử lý NA

↓

Mã hóa biến phân loại

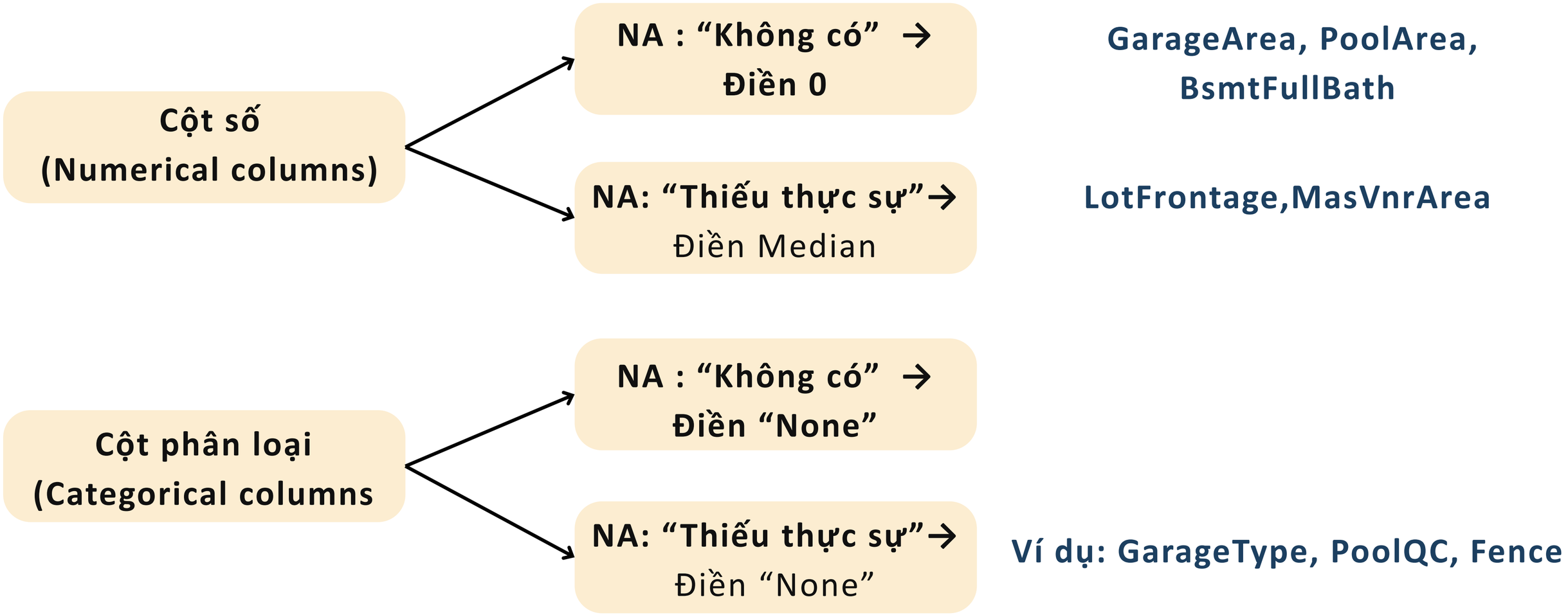
↓

Chuẩn hóa biến số

3. Các phương pháp đề xuất

Tiền xử lí dữ liệu

Xử lí các giá trị bị thiếu- Missing Value Imputation



3. Các phương pháp đề xuất

Mã hóa dữ liệu - OneHot Encoder

- Các đặc trưng phân loại: Neighborhood, HouseStyle, GarageType, ExterQual, KitchenQual, ...

Dùng OneHotEncoder để chuyển thành biểu diễn nhị phân (0/1)

Lý do: Tránh việc mô hình hiểu sai thứ tự giữa các giá trị phân loại (ví dụ: 'Poor' < 'Fair' < 'Good' < 'Excellent')

CentralAir
Y
N
Y

CentralAir_	CentralAir_
0	1
1	0
0	1

3. Các phương pháp đề xuất

Chuẩn hóa dữ liệu

Áp dụng chuẩn hóa cho các biến số:

- GrLivArea, TotalBsmtSF, LotArea, GarageArea, MasVnrArea, ...
- Dùng StandardScaler để đưa về phân phối chuẩn:

$$z = \frac{x - \mu}{\sigma}$$

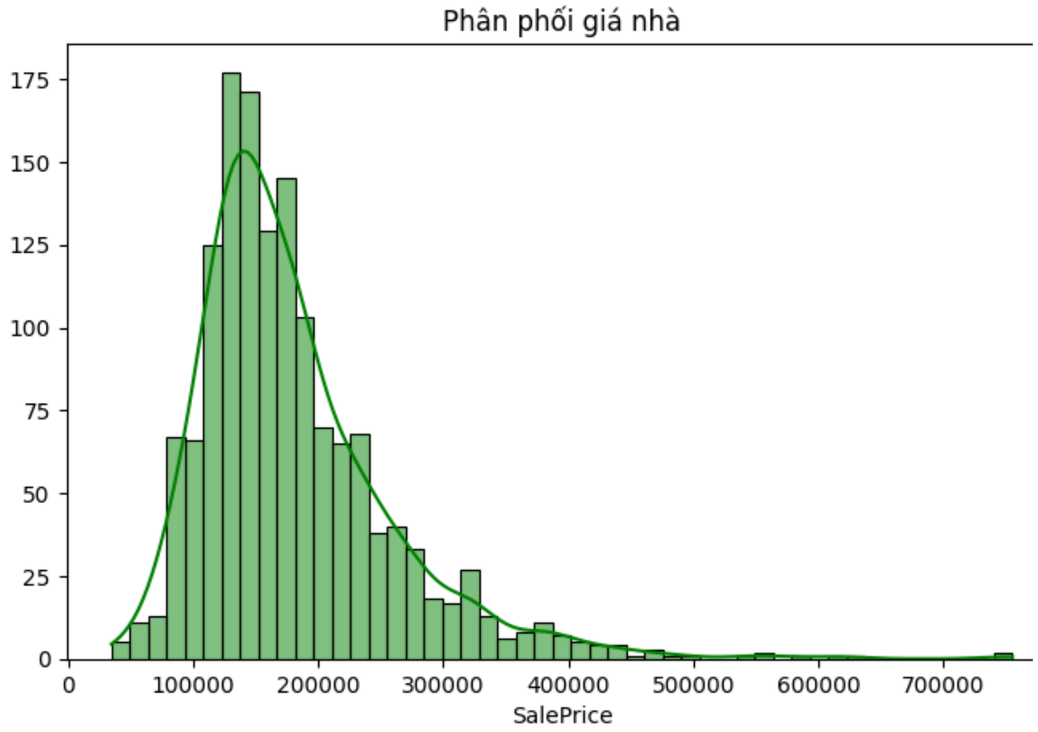
σ : Độ lệch chuẩn

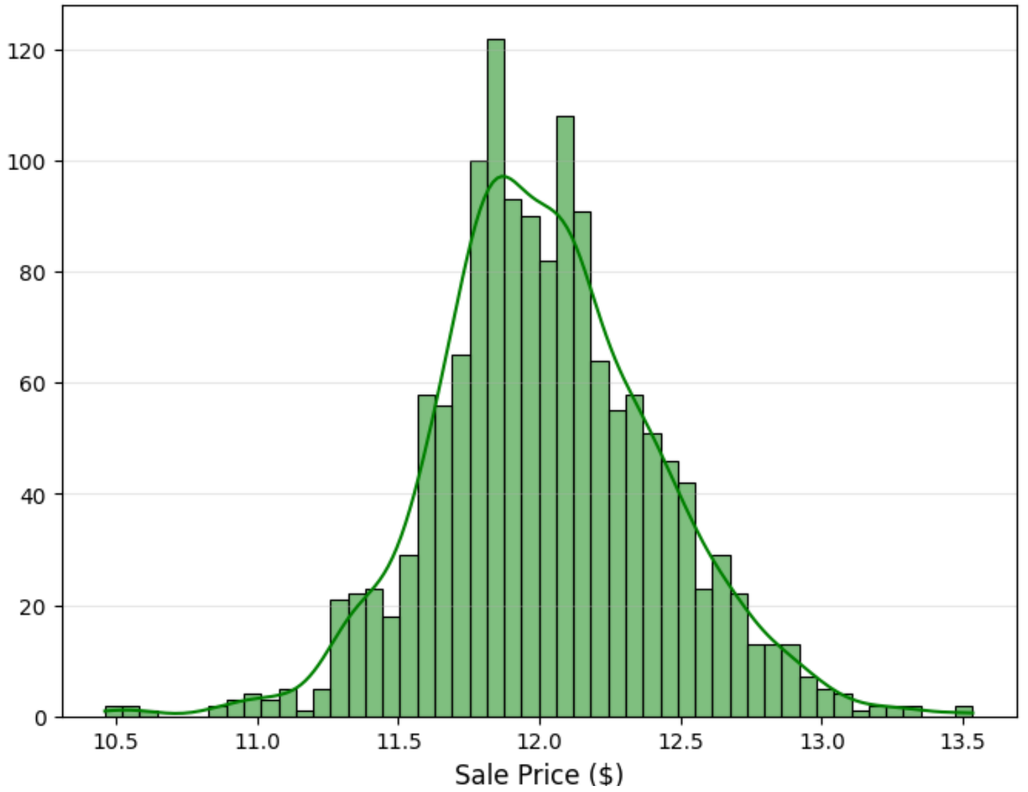
μ: Trung bình

3. Các phương pháp đề xuất

Chuẩn hóa dữ liệu

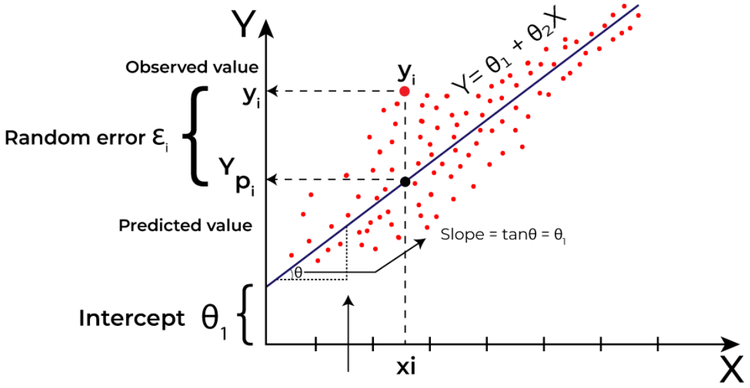
Áp dụng Log Transformation (log (1 + x) cho Fare và Age → giảm độ lệch phải (skewness) → dữ liệu ổn định hơn

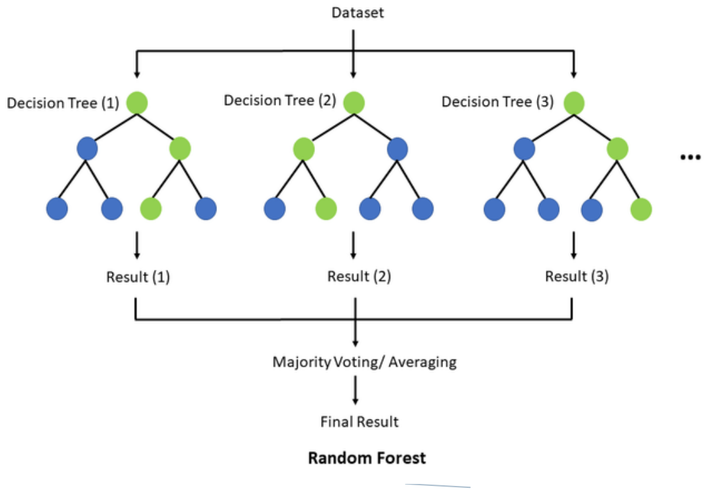


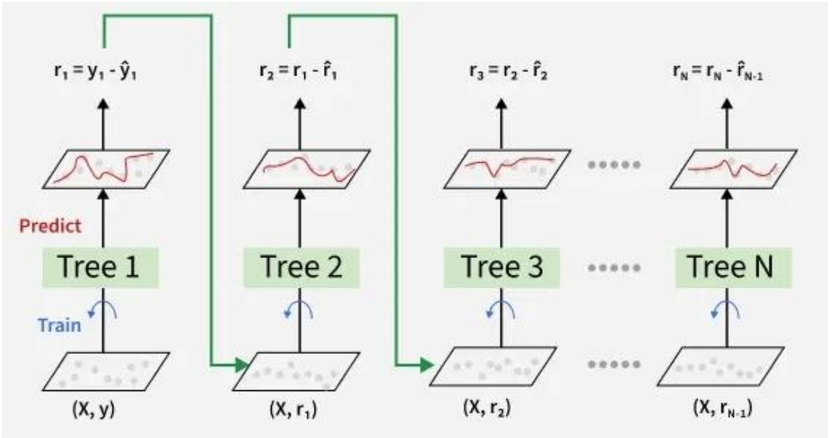


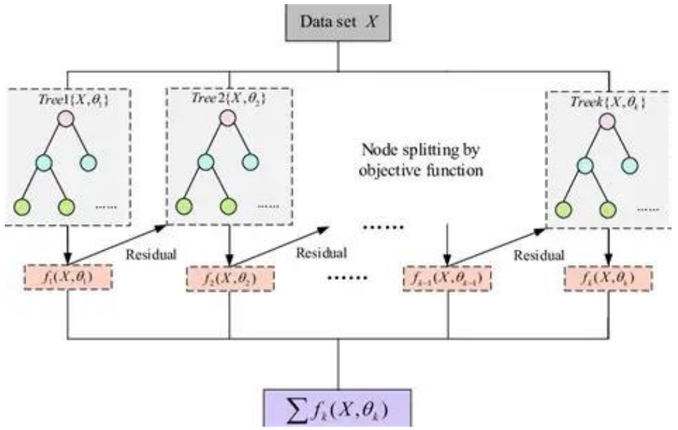
3. Các phương pháp đề xuất

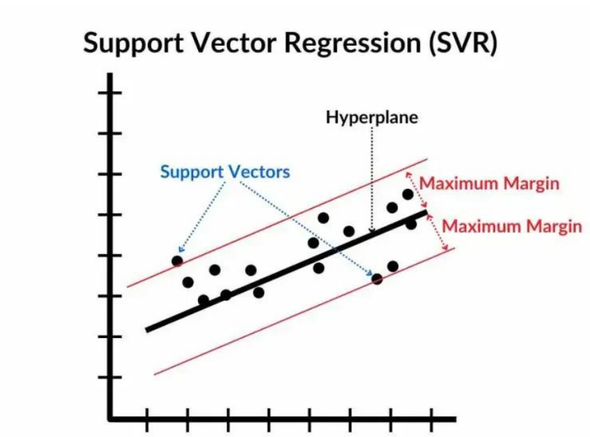
Model training

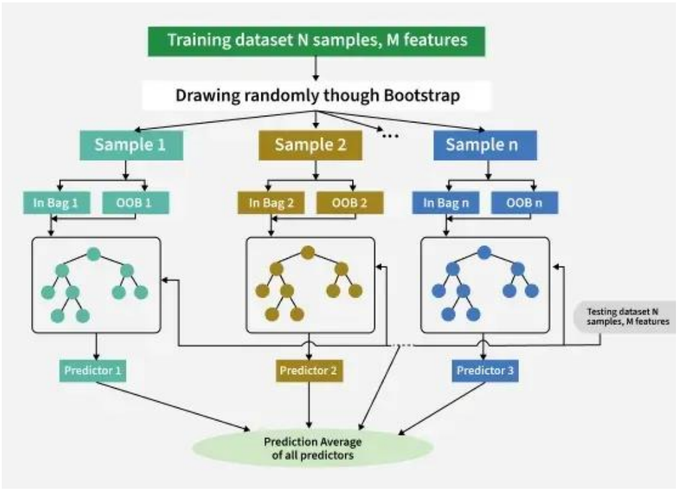












3. Các phương pháp đề xuất

Model evaluation:

Đánh giá bằng K-Fold

- Chia tập huấn luyện thành 5 phần bằng nhau
- Lặp lại huấn luyện và kiểm tra 5 lần với các fold khác nhau
- Tính sai số trung bình:

3. Các phương pháp đề xuất

Model evaluation:

RMSE – Root Mean Squared Error

- Đo độ lệch trung bình giữa giá trị dự đoán và thực tế
- RMSE càng nhỏ → mô hình càng tốt

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

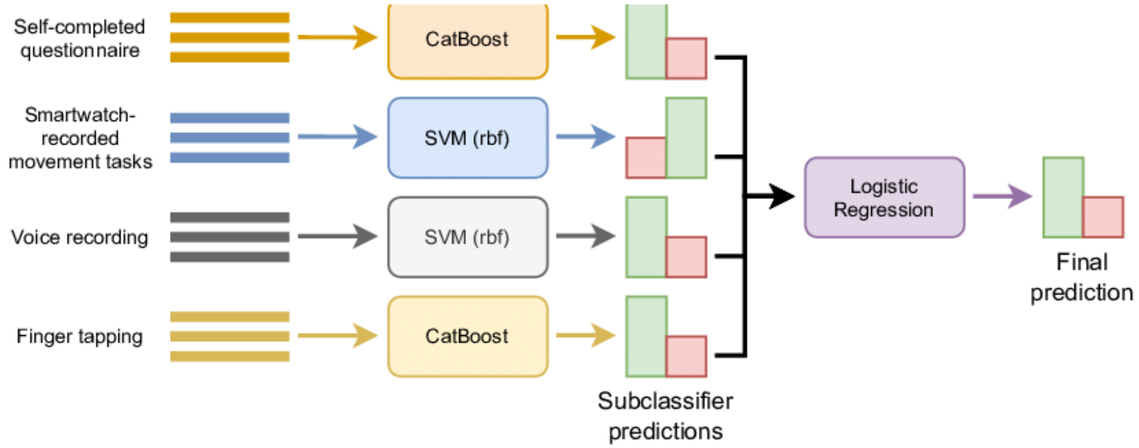
Nhận xét:

- K-Fold giúp kiểm tra độ ổn định mô hình trên nhiều tập dữ liệu khác nhau → tránh overfitting
- RMSE là thước đo chính để so sánh các mô hình → chọn mô hình có RMSE thấp nhất

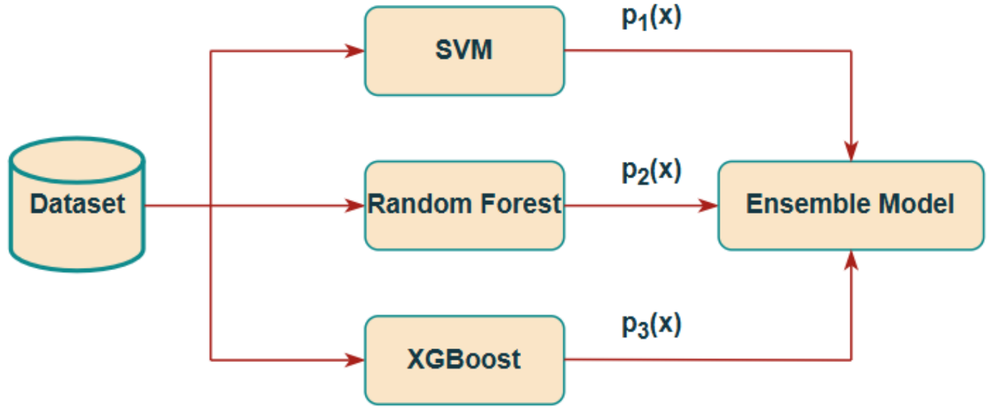
3. Các phương pháp đề xuất

Ensemble learning

Stacking classifier: Kết hợp nhiều mô hình "học cơ sở" và dùng một mô hình meta → tổng hợp kết quả từ chúng



Weighted Average Ensemble: kết hợp dự đoán của nhiều mô hình bằng cách gán trọng số khác nhau cho từng mô hình tùy theo độ chính xác của chúng.

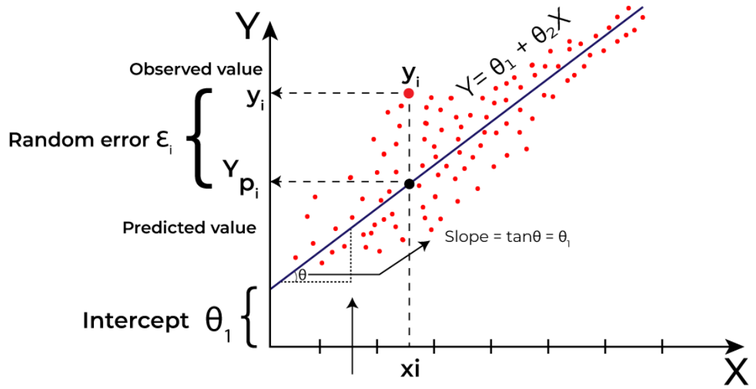


4. Thực nghiệm và trao đổi

Baseline

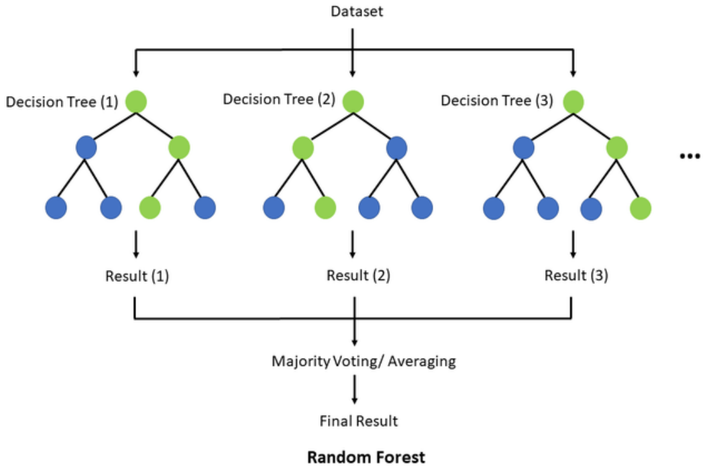
Linear Regression setup:

- fit_intercept: True
- n_jobs: None
- positive: False



Random Forest setup:

- n_estimators: 300
- Max depth: None
- random state: 42

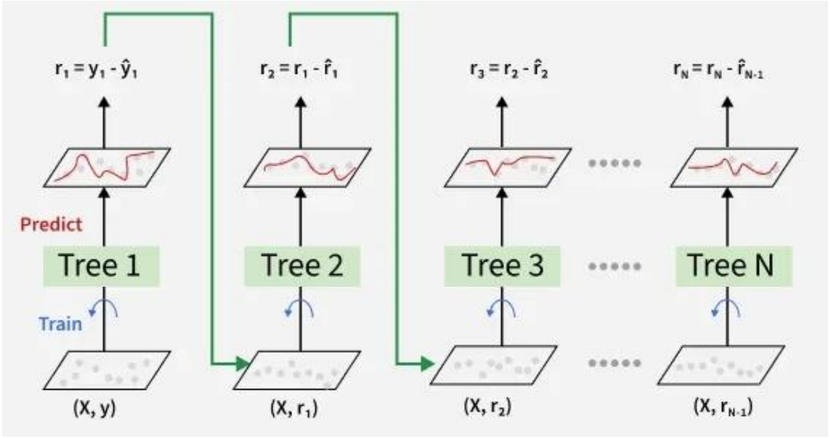


4. Thực nghiệm và trao đổi

Baseline

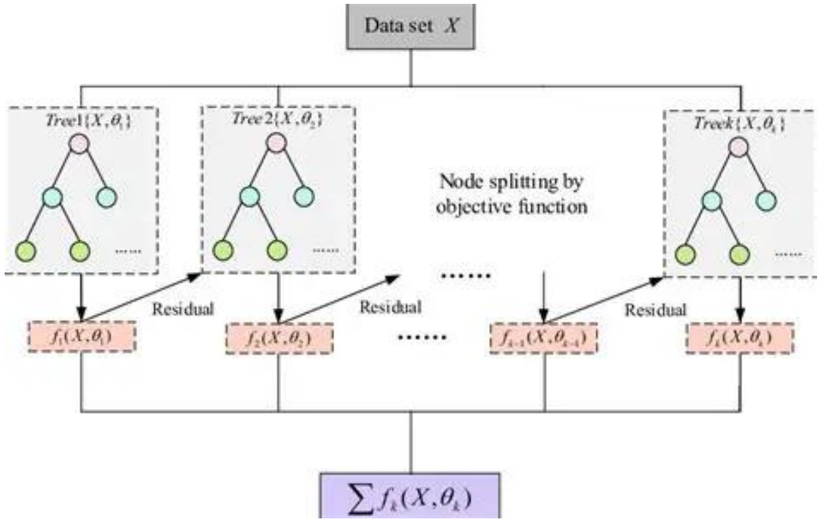
Gradient Boosting setup:

- n_estimators : 300
- learning rate: 0.05
- max depth: 3
- random state: 42



XGBoost setup:

- n_estimators : 1000
- learning rate: 0.05
- max depth: 3
- colsample_bytree : 0.7
- reg_lambda: 1
- subsample: 0.7



4. Thực nghiệm và trao đổi

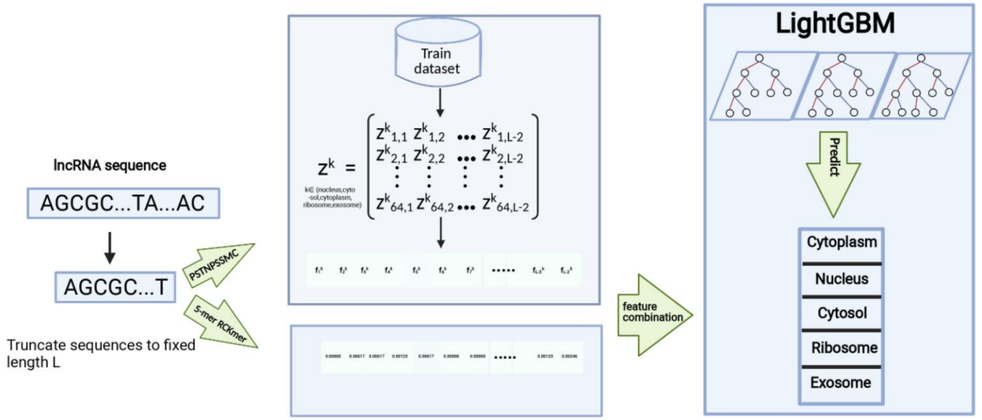
Baseline

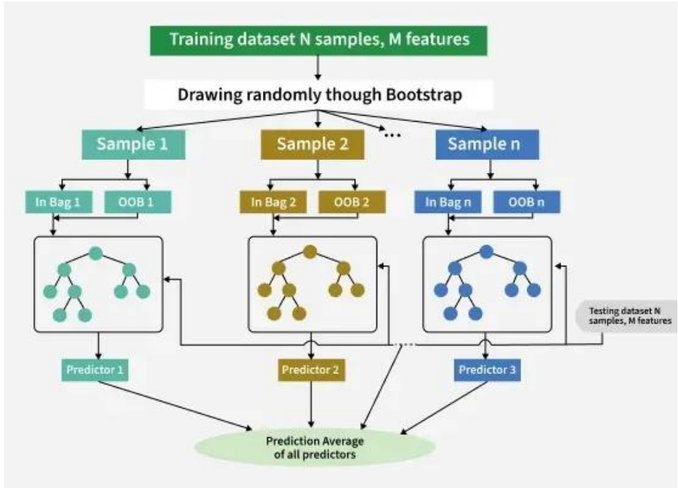
LightGBM setup:

- n_estimators : 1000
- learning rate: 0.05
- max depth: 3
- num_leaves: 30
- subsamples: 0.7
- colsample_bytree: 0.7

CatBoost setup:

- iterations: 1000
- learning rate: 0.05
- depth: 6
- verbose: False





4. Thực nghiệm và trao đổi

Kết quả của baseline

Model	RSME	Kaggle score
XGBoost	**0.1287 ± 0.0176**	**0.12364**
CatBoost	0.1255 ± 0.0174	0.12416
Gradient Boosting	0.1324 ± 0.0211	0.12862
LightGBM	0.1335 ± 0.0192	0.12896
Random forest	0.1451 ± 0.0195	0.14444
Linear regression	0.1720 ± 0.0351	0.1526

Nhận xét:

- XGBoost và CatBoost, `LightGBM dẫn đầu với RMSE thấp và điểm Kaggle cao
- `Gradient Boosting` và `Random Forest` có hiệu năng khá tốt nhưng huấn luyện chậm hơn một chút.

→ **Kết luận: Các mô hình boosting (như là `XGBoost`, `LightGBM`, `CatBoost`) nên được ưu tiên tinh chỉnh trong bước tiếp theo để cải thiện điểm Kaggle.**

4. Thực nghiệm và trao đổi

Kết quả của dữ liệu sau khi feature engineering

Model	RSME	Kaggle score
XGBoost	0.1299 ± 0.0155	0.1232
CatBoost	**0.1251 ± 0.0162**	**0.12188**
Gradient Boosting	0.1299 ± 0.0155	0.1232
LightGBM	0.1343 ± 0.0208	0.12885
Random forest	0.1415 ± 0.0182	0.14009
Linear regression	0.1299 ± 0.0155	0.1232

Nhận xét:

- CatBoost cải thiện rõ rệt, đạt RMSE = 0.1251 ± 0.0162
- Các mô hình boosting như Gradient Boosting, LightGBM cũng có hiệu suất tốt nhưng chưa vượt qua CatBoost

4. Thực nghiệm và trao đổi

Kết quả của ensemble learning

Model	RSME	Kaggle score
Stacking CB+LGB+KNN+SVR+RF+GB	0.1262 ± 0.0185	0.12203
Stacking CB+RF+GB	0.1255 ± 0.0186	0.12247
Weight CB+XGB+LGB	**0.1256 ± 0.0179**	**0.12091**
Weight LGB+XGB+GB	0.1279 ± 0.0190	0.12318

Nhận xét:

- Weight CB+XGB+LGB là mô hình có RMSE thấp nhất và điểm Kaggle cao nhất → hiệu suất tốt nhất trong nhóm ensemble
- Các mô hình có thêm KNN, SVR, RF không cải thiện rõ rệt → có thể gây nhiễu khi kết hợp

5. Kết luận

- Weight CB+XGB+LGB là mô hình cho thấy hiệu suất mạnh mẽ và ổn định nhất, đạt RMSE thấp nhất và điểm Kaggle cao nhất trên tập validation

→ **Là lựa chọn tối ưu cho bài toán dự đoán giá nhà.**

- Quá trình xử lý dữ liệu và tạo đặc trưng đóng vai trò quan trọng, giúp mã hóa các yếu tố ảnh hưởng đến giá trị bất động sản thành các đặc trưng rõ ràng và có ý nghĩa với mô hình học máy

THANK YOU