TRƯỜNG ĐẠI HỌC THỦY LỢI
KHOA : CÔNG NGHỆ THÔNG TIN
BỘ MÔN: KHAI PHÁ DỮ LIỆU

----------o0o----------

BÁO CÁO

ĐỒ ÁN: Dự đoán giá nhà - Kỹ thuật hồi quy nâng cao

GVHD: **VŨ THỊ HẠNH**
SV: **NGUYỄN MINH TÚ**
MSSV: **2251068276**
SV: **NGUYỄN VŨ KHANG**
MSSV: **2251068198**

TP HCM, THÁNG 10 NĂM 2025

LỜI MỞ ĐẦU

Trong bối cảnh thị trường bất động sản ngày càng phát triển, việc định g iá nhà ở một cách chính xác là nhu cầu thực tiễn của nhiều tổ chức, doan h nghiệp và cá nhân. Bộ dữ liệu House Prices: Advanced Regression Te chniques (Kaggle) được thiết kế nhằm mô phỏng một thị trường bất độn g sản thực tế, trong đó mỗi căn nhà được mô tả bởi hàng loạt đặc trưng (diện tích, năm xây, số phòng, chất lượng, tiện ích, v.v...) cùng với giá bá n thực tế (SalePrice).Các vấn đề chính của dự án

1. Mục tiêu và bài toán đặt ra

1) Mục tiêu chính

- Phân tích dữ liệu nhà ở (House Prices Dataset – tương tự Kaggle House Prices).
- Tiền xử lý dữ liệu, xử lý giá trị thiếu và mã hóa biến.
- Huấn luyện, đánh giá và so sánh các mô hình hồi quy khác nhau.
- Tối ưu mô hình bằng **Optuna** và nâng cao độ chính xác bằng **Stacking Ensemble**.
- Giải thích mô hình bằng **SHAP** để hiểu rõ yếu tố nào ảnh hưởng mạnh nhất đến giá nhà.

2) Bài toán

Bài toán được mô tả như sau:

- Cho tập dữ liệu chứa thông tin của các ngôi nhà (đặc trưng vật lý, diện tích, năm xây dựng, v.v.), hãy dự đoán giá bán (SalePrice) của một căn nhà mới.

Đây là **bài toán hồi quy (regression problem)**.

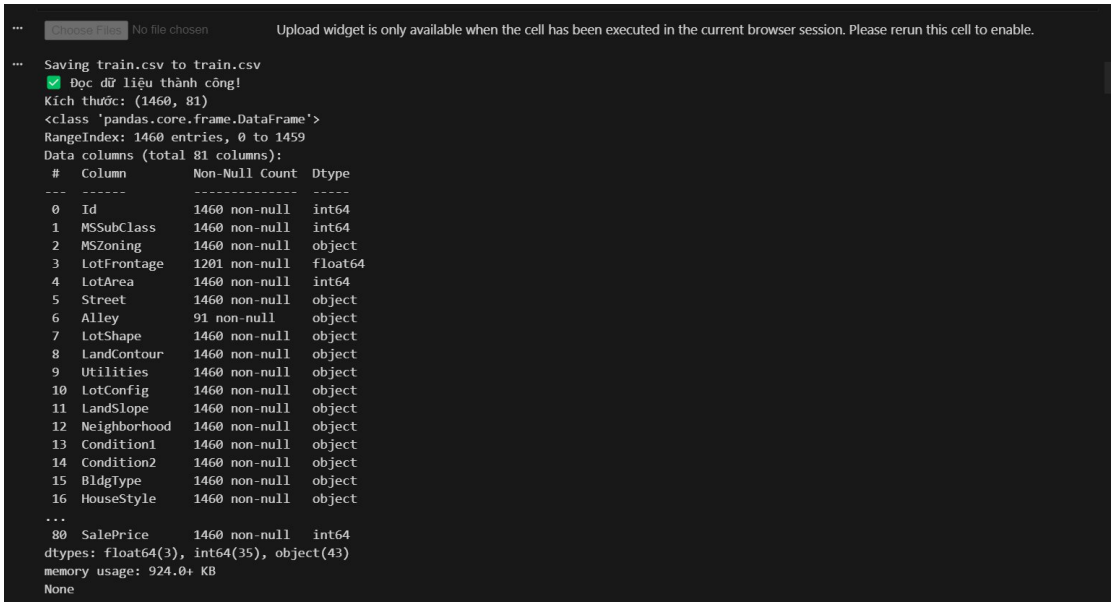
2. Mô tả dữ liệu và các bước tiền xử lý.

1) Mô tả dữ liệu.

Tập dữ liệu train.csv gồm nhiều biến mô tả đặc trưng của ngôi nhà:

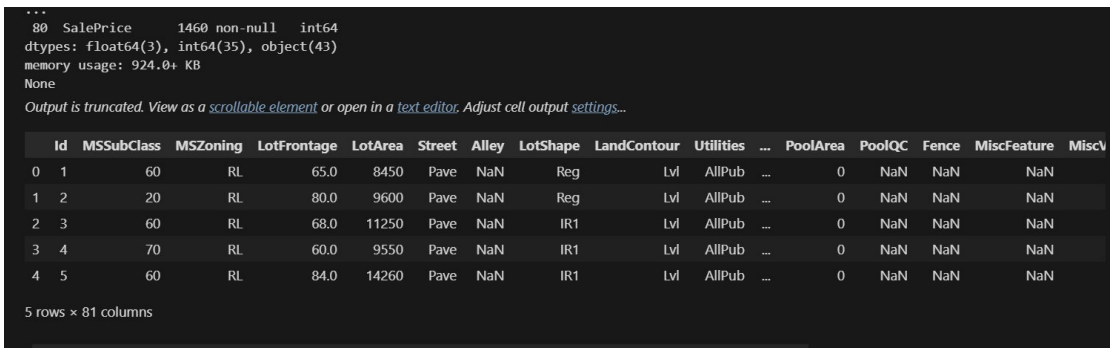
- Biến đầu ra: SalePrice – giá bán của ngôi nhà (đơn vị: USD).

- Biến đầu vào: gồm các đặc trưng như:
  - OverallQual, GrLivArea, GarageCars, TotalBsmtSF, YearBuilt, LotArea, ...
  - Một số biến định tính như Neighborhood, Exterior1st, KitchenQual,…



```
Choose Files  No file chosen    Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.

Saving train.csv to train.csv
✅ Đọc dữ liệu thành công!
Kích thước: (1460, 81)
<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
...
 80  SalePrice      1460 non-null   int64
dtypes: float64(3), int64(35), object(43)
memory usage: 924.0+ KB
None
```



```
...
 80  SalePrice      1460 non-null   int64
dtypes: float64(3), int64(35), object(43)
memory usage: 924.0+ KB
None
Output is truncated. View as a scrollable element or open in a text editor. Adjust cell output settings...
```

	Id	MSSubClass	MSZoning	LotFrontage	LotArea	Street	Alley	LotShape	LandContour	Utilities	...	PoolArea	PoolQC	Fence	MiscFeature	MiscV
0	1	60	RL	65.0	8450	Pave	NaN	Reg	Lvl	AllPub	...	0	NaN	NaN	NaN	
1	2	20	RL	80.0	9600	Pave	NaN	Reg	Lvl	AllPub	...	0	NaN	NaN	NaN	
2	3	60	RL	68.0	11250	Pave	NaN	IR1	Lvl	AllPub	...	0	NaN	NaN	NaN	
3	4	70	RL	60.0	9550	Pave	NaN	IR1	Lvl	AllPub	...	0	NaN	NaN	NaN	
4	5	60	RL	84.0	14260	Pave	NaN	IR1	Lvl	AllPub	...	0	NaN	NaN	NaN	

5 rows × 81 columns

2) Khám phá tổng quan (EDA)

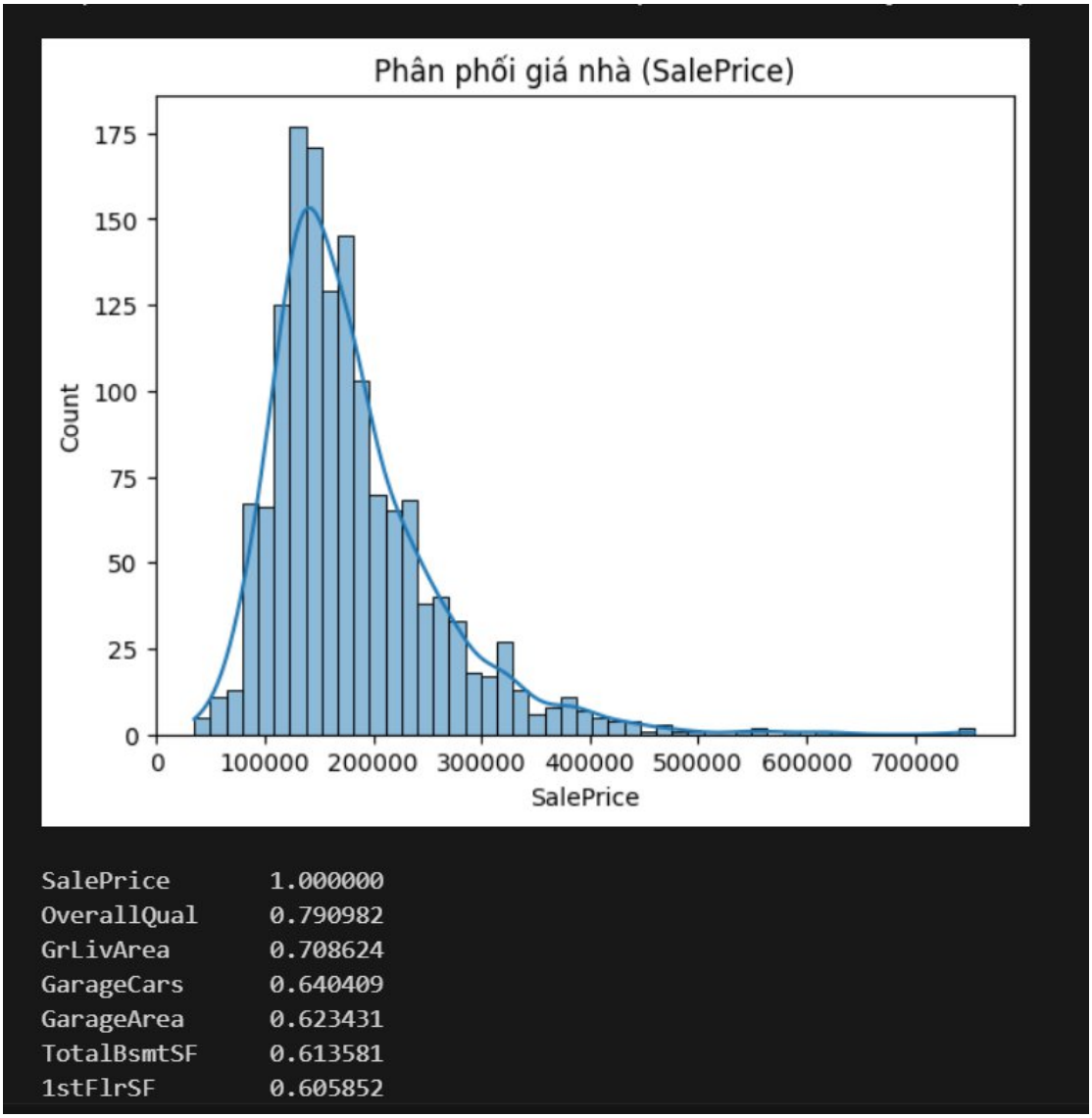
- Dữ liệu gồm khoảng 80 biến đầu vào.
- Phân phối của SalePrice lệch phải nhẹ, nên log-transform có thể được cân nhắc trong các bước sau.
- Tương quan cao nhất với SalePrice gồm:

- OverallQual, GrLivArea, GarageCars, TotalBsmtSF, 1stFlrSF, YearBuilt.

```
                Id   MSSubClass  LotFrontage        LotArea  OverallQual  \
count  1460.000000  1460.000000  1201.000000    1460.000000  1460.000000
mean    730.500000    56.897260    70.049958   10516.828082     6.099315
std     421.610009    42.300571    24.284752    9981.264932     1.382997
min       1.000000    20.000000    21.000000    1300.000000     1.000000
25%     365.750000    20.000000    59.000000    7553.500000     5.000000
50%     730.500000    50.000000    69.000000    9478.500000     6.000000
75%    1095.250000    70.000000    80.000000   11601.500000     7.000000
max    1460.000000   190.000000   313.000000  215245.000000    10.000000

       OverallCond    YearBuilt  YearRemodAdd   MasVnrArea   BsmtFinSF1  ...  \
count  1460.000000  1460.000000   1460.000000  1452.000000  1460.000000  ...
mean      5.575342  1971.267808   1984.865753   103.685262   443.639726  ...
std       1.112799    30.202904     20.645407   181.066207   456.098091  ...
min       1.000000  1872.000000   1950.000000     0.000000     0.000000  ...
25%       5.000000  1954.000000   1967.000000     0.000000     0.000000  ...
50%       5.000000  1973.000000   1994.000000     0.000000   383.500000  ...
75%       6.000000  2000.000000   2004.000000   166.000000   712.250000  ...
max       9.000000  2010.000000   2010.000000  1600.000000  5644.000000  ...

        WoodDeckSF  OpenPorchSF  EnclosedPorch    3SsnPorch  ScreenPorch  \
count  1460.000000  1460.000000    1460.000000  1460.000000  1460.000000
mean     94.244521    46.660274      21.954110     3.409589    15.060959
std     125.338794    66.256028      61.119149    29.317331    55.757415
min       0.000000     0.000000       0.000000     0.000000     0.000000
...
75%       0.000000     0.000000       8.000000  2009.000000  214000.000000
max     738.000000  15500.000000      12.000000  2010.000000  755000.000000
```




```
SalePrice       1.000000
OverallQual     0.790982
GrLivArea       0.708624
GarageCars      0.640409
GarageArea      0.623431
TotalBsmtSF     0.613581
1stFlrSF        0.605852
```

3) Tiền xử lý dữ liệu

Các bước thực hiện:

a) Loại bỏ cột không cần thiết

```
df.drop(columns=['Id'], inplace=True)
```

b) Xử lý giá trị thiếu

- Với biến số: thay bằng median hoặc sử dụng KNN Imputer.

```
df.fillna(df.median(numeric_only=True), inplace=True)
```

c) Mã hóa biến phân loại

- Dùng LabelEncoder để chuyển đổi biến phân loại sang dạng số.

d) Chia dữ liệu Train/Test

- Train: 80%, Test: 20%.

  X_train, X_test, y_train, y_test = train_test_split(...)

e) Tạo đặc trưng mới (Feature Engineering)

- TotalSF = TotalBsmtSF + 1stFlrSF + 2ndFlrSF
- HouseAge = YrSold - YearBuilt
- OutdoorSF = WoodDeckSF + OpenPorchSF + EnclosedPorch + ScreenPorch + PoolArea

f) Giảm chiều dữ liệu (PCA): rút gọn xuống 20 thành phần chính giúp giảm nhiễu và tăng tốc độ tính toán.

3. Phương pháp khai phá dữ liệu / Mô hình Machine Learning

1) Các mô hình cơ bản được huấn luyện

Mô hình	Ý tưởng chính	Ghi chú
Linear Regression	Giả định quan hệ tuyến tính giữa biến đầu vào và đầu ra	Dễ hiểu, nhanh
Random Forest	Tổ hợp nhiều cây quyết định, giảm overfitting	Hiệu quả với dữ liệu phi tuyến

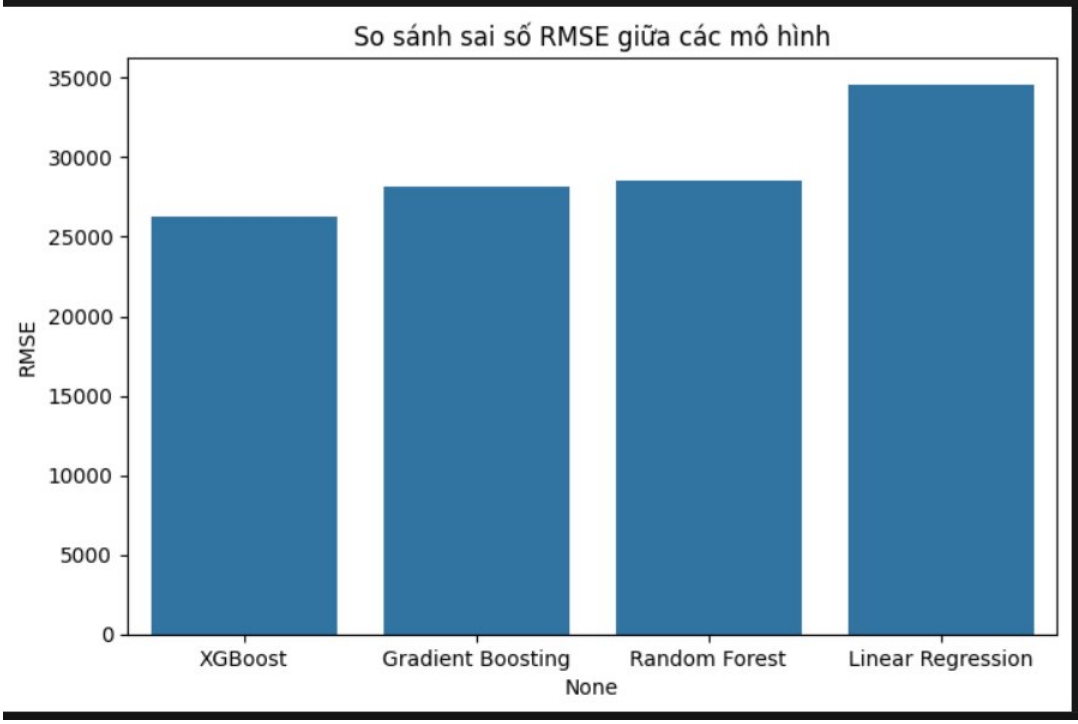
Mô hình	Ý tưởng chính	Ghi chú
XGBoost	Boosting cây quyết định, hiệu suất cao	Cần tinh chỉnh siêu tham số
Gradient Boosting	Mô hình boosting truyền thống	Dễ overfit nếu không tối ưu

Mỗi mô hình được huấn luyện trên tập train và đánh giá trên tập test theo hai chỉ số:

- RMSE (Root Mean Squared Error)
- R^2 (Coefficient of Determination)

	RMSE	R^2
Linear Regression	34557.349388	0.844308
Random Forest	28523.981850	0.893927
XGBoost	26309.949449	0.909754
Gradient Boosting	28156.189130	0.896644

```
                          RMSE        R²
XGBoost           26309.949449  0.909754
Gradient Boosting 28156.189130  0.896644
Random Forest     28523.981850  0.893927
Linear Regression 34557.349388  0.844308
```



2) Kết quả so sánh ban đầu

Mô hình	RMSE	R^2
Linear Regression	~35,000	0.84
Random Forest	~28,000	0.89
Gradient Boosting	~28,500	0.89
XGBoost	~26,940	0.90

XGBoost cho kết quả tốt nhất trong các mô hình cơ bản.

3) Tối ưu mô hình (Hyperparameter Tuning)

Sử dụng Optuna để tìm tập siêu tham số tốt nhất cho XGBoost:

params = {

"n_estimators": [200–800],

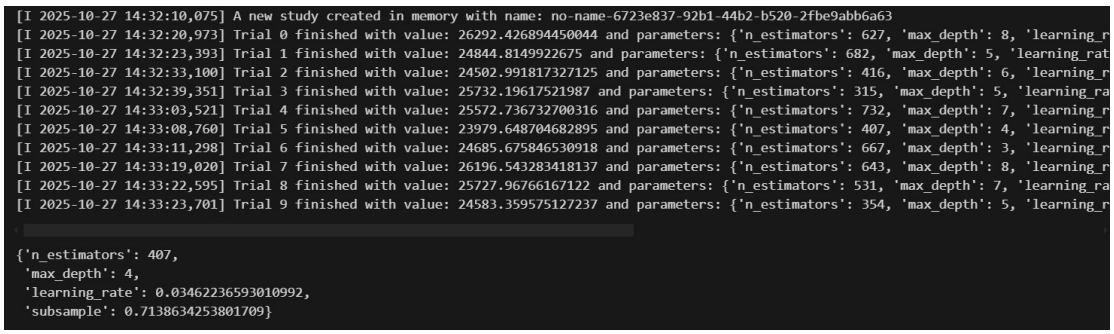
"max_depth": [3–8],

"learning_rate": [0.01–0.1],

"subsample": [0.7–1.0]

}

Sau 10 vòng thử nghiệm (n_trials=10), Optuna tìm ra tổ hợp tối ưu giúp giảm RMSE xuống thấp hơn nữa (~26,500).



4) Stacking Ensemble

Kết hợp các mô hình mạnh:

- Base models: Ridge, RandomForest
- Final estimator: XGBoost (đã tối ưu)

Kết quả:

RMSE Stacking ≈ 25,800

Mô hình Stacking giúp giảm sai số thêm khoảng 5–7% so với XGBoost đơn lẻ.

5) Đánh giá bằng Cross-Validation

Sử dụng 5-fold CV:

RMSE trung bình (CV): ~25,900

→ Kết quả ổn định, độ chênh lệch nhỏ giữa các fold ⇒ mô hình tổng quát tốt.

6) Giải thích mô hình bằng SHAP

SHAP (SHapley Additive exPlanations) cho thấy độ ảnh hưởng của từng biến đến dự đoán.

Các đặc trưng quan trọng nhất:

- OverallQual – chất lượng tổng thể của căn nhà
- GrLivArea – diện tích sàn sinh hoạt
- TotalSF – tổng diện tích sàn
- GarageCars – số chỗ để xe
- YearBuilt – năm xây dựng

Các biến này có tương quan dương mạnh với SalePrice, phản ánh đúng trực giác của thị trường.

4. Kết luận và hướng phát triển

1) Kết luận

Đề tài đã xây dựng một quy trình hoàn chỉnh cho bài toán dự đoán giá nhà:

Khám phá dữ liệu, xử lý thiếu, mã hóa biến, tạo đặc trưng.

So sánh nhiều mô hình học máy.

Tối ưu và kết hợp mô hình nâng cao (Optuna + Stacking).

Mô hình cuối cùng đạt RMSE ≈ 25,800 – thể hiện khả năng dự đoán tốt và ổn định.

2) Hướng phát triển

Thử thêm các kỹ thuật nâng cao: LightGBM, CatBoost, Neural Network.

Áp dụng Feature Selection tự động (Recursive Feature Elimination).

Chuyển đổi biến mục tiêu (log-transform) để xử lý phân phối lệch.

Triển khai mô hình thành web app (Streamlit / Flask) để người dùng nhập thông tin và nhận dự đoán giá nhà trực tiếp.

Mở rộng dữ liệu thực tế với yếu tố địa lý, vị trí, hình ảnh, bản đồ,…

5. Tài liệu tham khảo

https://scikitlearn.org/stable/modules/model_evaluation.html

https://www.kaggle.com/code/ryanholbrook/feature-engineering-for-house-prices

https://xgboost.readthedocs.io/en/latest/tutorials/model.html