CUỘC THI DỰ ĐOÁN GIÁ NHÀ - Kaggle

https://www.kaggle.com/competitions/house-prices-advanced-regression-techniques/overview

Yêu cầu người mua nhà mô tả ngôi nhà mơ ước của họ, và họ có thể sẽ không bắt đầu bằng chiều cao của trần tầng hầm hoặc vị trí gần đường sắt đông-tây. Nhưng tập dữ liệu của cuộc thi sân chơi này chứng minh rằng ảnh hưởng đến việc đàm phán giá cả nhiều hơn số lượng phòng ngủ hoặc hàng rào cọc trắng.

Với 79 biến giải thích mô tả (gần như) mọi khía cạnh của nhà ở ở Ames, Iowa, cuộc thi này thách thức bạn dự đoán giá cuối cùng của mỗi căn nhà.

Luyện tập kỹ năng Kỹ thuật tính năng sáng tạo Các kỹ thuật hồi quy nâng cao như rừng ngẫu nhiên và tăng cường độ dốc Sự nhìn nhận Bộ dữ liệu Ames Housing được Dean De Cock biên soạn để sử dụng trong giáo dục khoa học dữ liệu. Đây là một giải pháp thay thế tuyệt vời dành cho các nhà khoa học dữ liệu đang tìm kiếm phiên bản mở rộng và hiện đại hóa của bộ dữ liệu Nhà ở Boston thường được trích dẫn.

- ## Dữ liệu các trường

Dưới đây là phiên bản tóm tắt về những gì bạn sẽ tìm thấy trong tệp mô tả dữ liệu.

SalePrice - giá bán của bất động sản trong đô la. Đây là biến mục tiêu mà bạn đang cố gắng dự đoán.

MSSubClass : Lớp xây dựng

MSZoning : Phân loại chung

LotFrontage : Độ dài đường phố kết nối với bất động sản

LotArea : Diện tích lô trong feet vuông

Street : Loại quyền truy cập đường

Alley : Loại quyền truy cập ngõ hẻm

LotShape : Hình dạng tổng quan của bất động sản

LandContour : Độ bằng phẳng của bất động sản

Utilities : Loại tiện ích có sẵn

LotConfig : Cấu hình lô đất

LandSlope : Độ dốc của bất động sản

Neighborhood : Vị trí vật lý trong giới hạn thành phố Ames

Condition1 : Gần đường chính hoặc đường sắt

Condition2 : Gần đường chính hoặc đường sắt (nếu có một thứ hai)

BldgType : Loại nhà ở

HouseStyle : Kiểu nhà ở

OverallQual : Chất lượng vật liệu và hoàn thiện tổng thể

OverallCond : Xếp loại điều kiện tổng thể

YearBuilt : Ngày xây dựng ban đầu

YearRemodAdd : Ngày cải tạo

RoofStyle : Loại mái nhà

RoofMatl : Vật liệu mái nhà

Exterior1st : Phủ bề mặt bên ngoài trên nhà

Exterior2nd : Phủ bề mặt bên ngoài trên nhà (nếu có nhiều hơn một loại)

MasVnrType : Loại vênе nhân tạo

MasVnrArea : Diện tích vênе nhân tạo trong feet vuông

ExterQual : Chất lượng vật liệu bề mặt bên ngoài

ExterCond : Tình trạng hiện tại của vật liệu bề mặt bên ngoài

Foundation : Loại nền móng

BsmtQual : Chiều cao của tầng hầm

BsmtCond : Tình trạng chung của tầng hầm

BsmtExposure : Tầng hầm có tường ra vào hoặc tầng hầm ở mức đất

BsmtFinType1 : Chất lượng khu vực đã hoàn thành của tầng hầm

BsmtFinSF1 : Diện tích hoàn thành loại 1 trong feet vuông

BsmtFinType2 : Chất lượng khu vực đã hoàn thành thứ hai (nếu có)

BsmtFinSF2 : Diện tích hoàn thành loại 2 trong feet vuông

BsmtUnfSF : Diện tích chưa hoàn thành của tầng hầm trong feet vuông

TotalBsmtSF : Tổng diện tích của tầng hầm trong feet vuông

Heating : Loại hệ thống sưởi

HeatingQC : Chất lượng và điều kiện sưởi

CentralAir : Hệ thống điều hòa không khí trung tâm

Electrical : Hệ thống điện

1stFlrSF : Diện tích tầng trệt trong feet vuông

2ndFlrSF : Diện tích tầng hai trong feet vuông

LowQualFinSF : Diện tích hoàn thành thấp chất lượng (tất cả các tầng)

GrLivArea : Diện tích sống trên mức độ (mặt đất)

BsmtFullBath : Phòng tắm đầy đủ tầng hầm

BsmtHalfBath : Phòng tắm bán tầng hầm

FullBath : Phòng tắm đầy đủ trên mức độ

HalfBath : Nửa phòng tắm trên mức độ

Bedroom : Số phòng ngủ trên mức độ tầng hầm

Kitchen : Số phòng bếp

KitchenQual : Chất lượng bếp

TotRmsAbvGrd : Tổng số phòng trên mức độ (không bao gồm phòng tắm)

Functional : Xếp hạng chức năng nhà

Fireplaces : Số lượng lò sưởi

FireplaceQu : Chất lượng lò sưởi

GarageType : Vị trí gara

GarageYrBlt : Năm xây dựng garage

GarageFinish : Hoàn thiện nội thất garage

GarageCars : Kích thước garage theo sức chứa xe hơi

GarageArea : Kích thước garage trong feet vuông

GarageQual : Chất lượng garage

GarageCond : Tình trạng garage

PavedDrive : Đường lái nhựa

WoodDeckSF : Diện tích sàn nhựa gỗ trong feet vuông

OpenPorchSF : Diện tích hiên mở trong feet vuông

EnclosedPorch : Diện tích hiên đóng trong feet vuông

3SsnPorch : Diện tích hiên 3 mùa trong feet vuông

ScreenPorch : Diện tích hiên chắn trong feet vuông

PoolArea : Diện tích hồ bơi trong feet vuông

PoolQC : Chất lượng hồ bơi

Fence : Chất lượng hàng rào

MiscFeature : Tính năng đặc biệt

HỒI QUY TUYẾN TÍNH (Linear Regression)

1. Hãy load dữ liệu từ github sau (lưu vào biến df): https://github.com/baoanth/DataAna_DA20TT/blob/main/house_price_train.csv
2. Hãy tạo cột thuộc tính mới (HouseAge)
3. Hãy tính giá trị tương quan giữa tuổi nhà và giá nhà (SalePrice).
4. Hãy tính lập mô hình hồi quy tuyến tính đơn biến dự đoán giá nhà (SalePrice) dựa vào tuổi nhà (HouseAge).
5. Hãy viết ra công thức hồi quy, tính hệ số R2, RMSE và kết luận cho câu 4.
6. Hãy tính lập công thức hồi quy đa biến dự đoán giá nhà (SalePrice) dựa vào tuổi nhà (HouseAge), diện tích (LotArea) và điểm chất lượng tổng quan (OverallQual)
7. Hãy viết ra công thức hồi quy, tính hệ số R2, RMSE và kết luận cho câu 6.
8. Hãy vẽ heatmap thể hiện hệ số tương quan cho tất cả các cặp biến định tính.
9. Dựa vào heatmap ở câu 8, hãy chọn 5 thuộc tính độc lập để dự đoán giá nhà.
10. Hãy viết ra công thức hồi quy, tính hệ số R2, RMSE và kết luận cho câu 9.

```
Bắt đầu lập trình hoặc tạo mã bằng trí tuệ nhân tạo (AI).
```

```
import pandas as pd
import numpy as np
import matplotlib as mpl
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns

df = pd.read_csv('https://raw.githubusercontent.com/baoanth/DataAna_DA20TT/main/house_price_train.csv')
```

```
#2. Hãy tạo cột thuộc tính mới (HouseAge)
import datetime
import pandas as pd

# df['HouseAge'] = df['YrSold'] - df['YearBuilt']  age từ thời điểm ngôi nhà được bán:

# Có thể dùng năm cố định 2026
df['HouseAge'] = 2026 - df['YearBuilt']
df
```

	Id	MSSubClass	MSZoning	LotFrontage	LotArea	Street	Alley	LotShape	LandContour	Utilities	...	PoolQC	Fence	Misc
0	1	60	RL	65.0	8450	Pave	NaN	Reg	Lvl	AllPub	...	NaN	NaN	
1	2	20	RL	80.0	9600	Pave	NaN	Reg	Lvl	AllPub	...	NaN	NaN	
2	3	60	RL	68.0	11250	Pave	NaN	IR1	Lvl	AllPub	...	NaN	NaN	
3	4	70	RL	60.0	9550	Pave	NaN	IR1	Lvl	AllPub	...	NaN	NaN	
4	5	60	RL	84.0	14260	Pave	NaN	IR1	Lvl	AllPub	...	NaN	NaN	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
1455	1456	60	RL	62.0	7917	Pave	NaN	Reg	Lvl	AllPub	...	NaN	NaN	
1456	1457	20	RL	85.0	13175	Pave	NaN	Reg	Lvl	AllPub	...	NaN	MnPrv	
1457	1458	70	RL	66.0	9042	Pave	NaN	Reg	Lvl	AllPub	...	NaN	GdPrv	
1458	1459	20	RL	68.0	9717	Pave	NaN	Reg	Lvl	AllPub	...	NaN	NaN	
1459	1460	20	RL	75.0	9937	Pave	NaN	Reg	Lvl	AllPub	...	NaN	NaN	

1460 rows × 82 columns

```
Bắt đầu lập trình hoặc tạo mã bằng trí tuệ nhân tạo (AI).
```

```
#3. Hãy tính giá trị tương quan giữa tuổi nhà và giá nhà (SalePrice).
corr = df['HouseAge'].corr(df['SalePrice'])
print("He so tuong quan giua HouseAge va SalePrice", corr)
```

```
He so tuong quan giua HouseAge va SalePrice -0.5228973328794968
```

```
# Import các thư viện cần thiết
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, mean_absolute_error, r2_score
import matplotlib.pyplot as plt
```

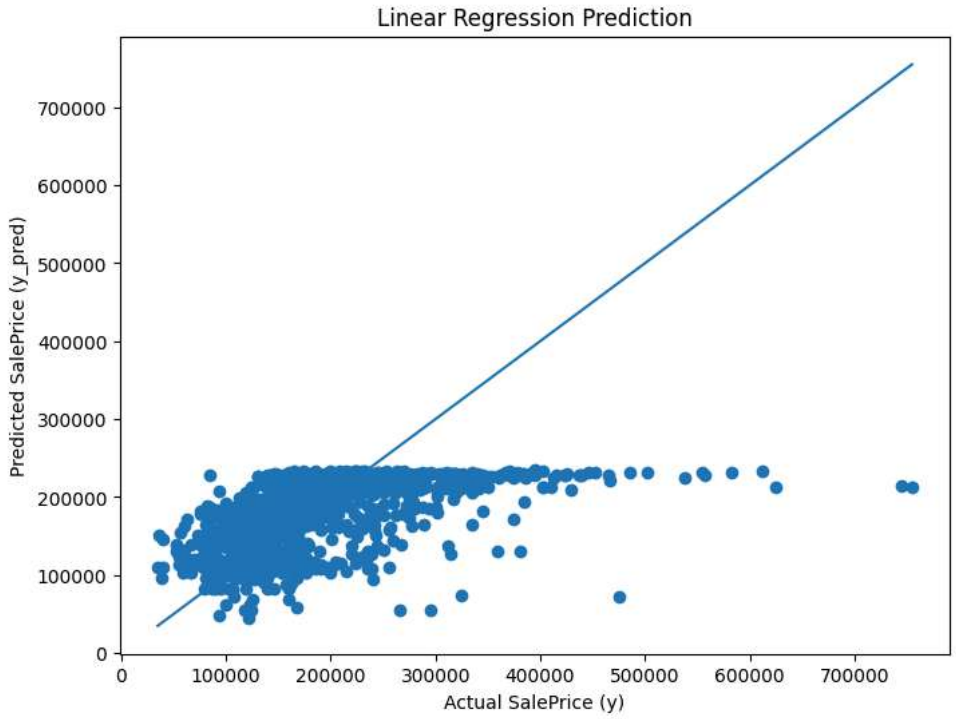
```
#4. Hãy viết ra công thức hồi quy, tính hệ số R2, RMSE và kết luận cho câu 4.
# Import thư viện
import numpy as np
from sklearn.linear_model import LinearRegression
from sklearn.metrics import r2_score, mean_squared_error

# Biến độc lập và biến phụ thuộc
X_simple = df[["HouseAge"]]
y = df["SalePrice"]

# Xây dựng mô hình hồi quy tuyến tính đơn biến
model_simple = LinearRegression()
model_simple.fit(X_simple, y)

# Dự đoán
y_pred_simple = model_simple.predict(X_simple)

# Hệ số hồi quy
beta_0 = model_simple.intercept_
beta_1 = model_simple.coef_[0]

# Đánh giá mô hình
r2_simple = r2_score(y, y_pred_simple)
rmse_simple = np.sqrt(mean_squared_error(y, y_pred_simple))

# In kết quả
print("Phương trình hồi quy:")
print(f"SalePrice = {beta_0:.2f} + ({beta_1:.2f}) * HouseAge")
print("R2:", r2_simple)
print("RMSE:", rmse_simple)
```

```
Phương trình hồi quy:
SalePrice = 256198.40 + (-1375.37) * HouseAge
R2: 0.27342162073249154
RMSE: 67693.25098357971
```

```
#5. Hãy tính lập mô hình hồi quy tuyến tính đơn biến dự đoán giá nhà (SalePrice) dựa vào tuổi nhà (HouseAge).
X_simple = df[["HouseAge"]]
y = df["SalePrice"]

model_simple = LinearRegression()
model_simple.fit(X_simple, y)

y_pred_simple = model_simple.predict(X_simple)

# Hệ số
beta_1 = model_simple.coef_[0]
beta_0 = model_simple.intercept_

# Đánh giá
r2_simple = r2_score(y, y_pred_simple)
rmse_simple = np.sqrt(mean_squared_error(y, y_pred_simple))

print("Intercept:", beta_0)
print("Coefficient (HouseAge):", beta_1)
print("R2:", r2_simple)
print("RMSE:", rmse_simple)


plt.figure(figsize=(8, 6))
```

```
plt.scatter(y, y_pred_simple)

min_val = min(y.min(), y_pred_simple.min())
max_val = max(y.max(), y_pred_simple.max())

plt.plot([min_val, max_val], [min_val, max_val])

plt.xlabel("Actual SalePrice (y)")
plt.ylabel("Predicted SalePrice (y_pred)")
plt.title("Linear Regression Prediction")

plt.show()
```

```
Intercept: 256198.4003077871
Coefficient (HouseAge): -1375.3734679368927
R2: 0.27342162073249154
RMSE: 67693.25098357971
```




```
#6. Hãy tính lập công thức hồi quy đa biến dự đoán giá nhà (SalePrice) dựa vào tuổi nhà (HouseAge), diện tích (LotArea) và điểm
import numpy as np
from sklearn.linear_model import LinearRegression
from sklearn.metrics import r2_score, mean_squared_error, mean_absolute_error

# =========================
# Dữ liệu
# =========================
X = df[["HouseAge", "LotArea", "OverallQual"]]
y = df["SalePrice"]

# =========================
# Fit mô hình (KHÔNG chia train/test)
# =========================
model = LinearRegression()
model.fit(X, y)

y_pred = model.predict(X)

# =========================
# Hệ số
# =========================
beta_0 = model.intercept_
beta_1 = model.coef_
```

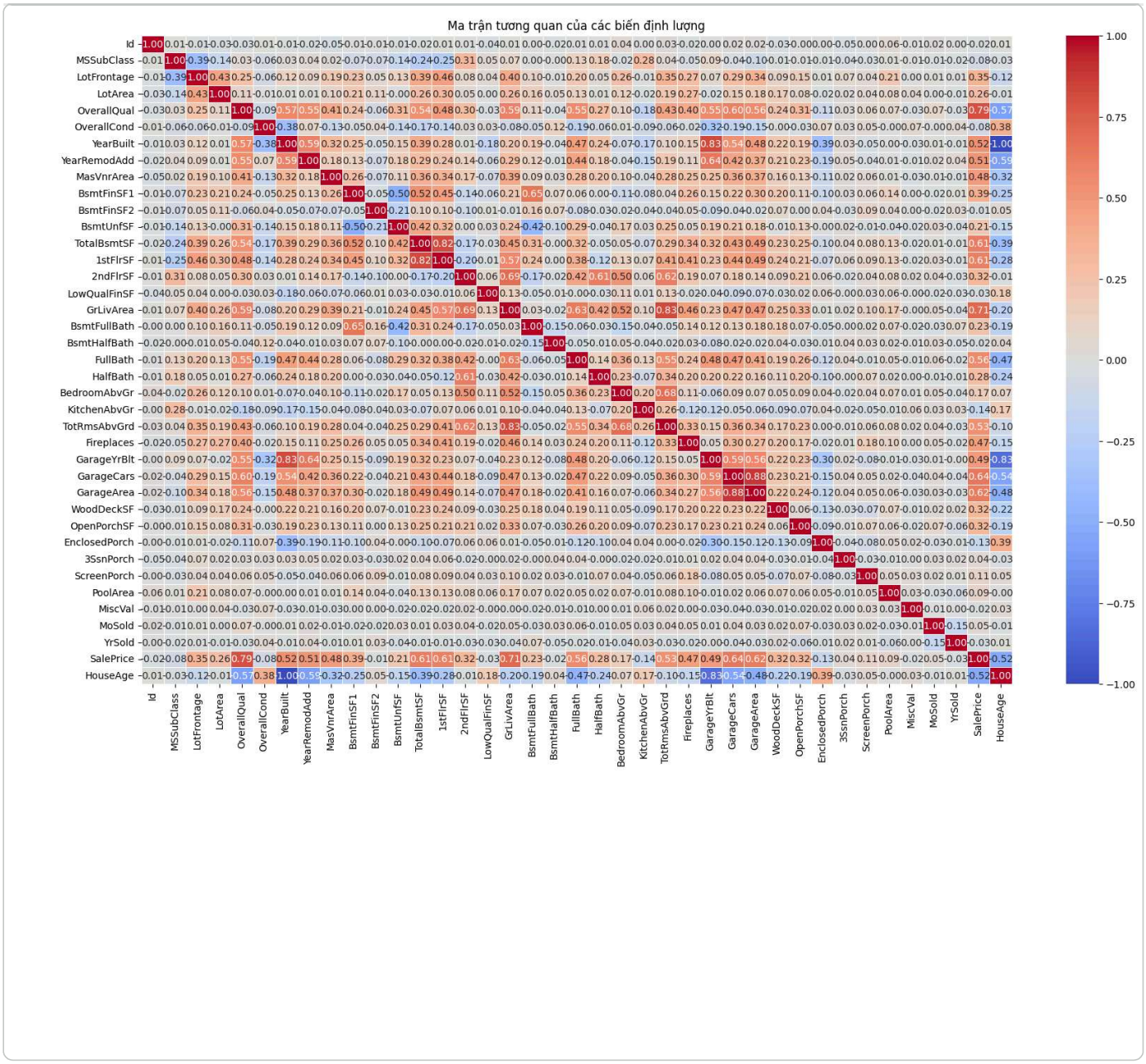
```
# =========================
# Đánh giá
# =========================
r2 = r2_score(y, y_pred)
rmse = np.sqrt(mean_squared_error(y, y_pred))
mae = mean_absolute_error(y, y_pred)
mse = mean_squared_error(y, y_pred)

# =========================
# In kết quả
# =========================
print("Intercept (beta_0):", beta_0)
print("Coefficient (beta_1):", beta_1)
print("R^2:", r2)
print("RMSE:", rmse)
print("MAE:", mae)
print("MSE:", mse)

print(
    f"SalePrice = {beta_1[0]:.2f}*HouseAge + "
    f"{beta_1[1]:.2f}*LotArea + "
    f"{beta_1[2]:.2f}*OverallQual + "
    f"{beta_0:.2f}"
)
```

```
Intercept (beta_0): -64542.43122273148
Coefficient (beta_1): [-3.08602768e+02  1.49385142e+00  4.04379101e+04]
R^2: 0.6677010937757392
RMSE: 45779.24146050506
MAE: 31044.854373556707
MSE: 2095738948.6992254
SalePrice = -308.60*HouseAge + 1.49*LotArea + 40437.91*OverallQual + -64542.43
```

```
#7. Hãy viết ra công thức hồi quy, tính hệ số R2, RMSE và kết luận cho câu 6.

# Import thư viện
import numpy as np
from sklearn.linear_model import LinearRegression
from sklearn.metrics import r2_score, mean_squared_error

# =========================
# CÂU 6: Hồi quy tuyến tính đa biến
# SalePrice ~ HouseAge + LotArea + OverallQual
# =========================

# Biến độc lập và biến phụ thuộc
X_multi = df[["HouseAge", "LotArea", "OverallQual"]]
y = df["SalePrice"]

# Xây dựng mô hình
model_multi = LinearRegression()
model_multi.fit(X_multi, y)

# Dự đoán
y_pred_multi = model_multi.predict(X_multi)

# Hệ số hồi quy
beta_0 = model_multi.intercept_
beta_1, beta_2, beta_3 = model_multi.coef_

# Đánh giá mô hình
r2_multi = r2_score(y, y_pred_multi)
rmse_multi = np.sqrt(mean_squared_error(y, y_pred_multi))

# In kết quả
print("Phương trình hồi quy đa biến:")
print(f"SalePrice = {beta_0:.2f} "
      f"+ ({beta_1:.2f}) * HouseAge "
      f"+ ({beta_2:.2f}) * LotArea "
      f"+ ({beta_3:.2f}) * OverallQual")

print("R2:", r2_multi)
print("RMSE:", rmse_multi)
```

```
Phương trình hồi quy đa biến:
SalePrice = -64542.43 + (-308.60) * HouseAge + (1.49) * LotArea + (40437.91) * OverallQual
R2: 0.6677010937757392
RMSE: 45779.24146050506
```

```
#8. Hãy vẽ heatmap thể hiện hệ số tương quan cho tất cả các cặp biến định tính.
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt

# Chỉ lấy các biến định lượng
numeric_df = df.select_dtypes(include=[np.number])

# Tính ma trận tương quan
correlation_matrix = numeric_df.corr()

# Vẽ heatmap
plt.figure(figsize=(20, 12))
sns.heatmap(
    correlation_matrix,
    annot=True,
    cmap='coolwarm',
    fmt=".2f",
    linewidths=0.5
)

plt.title("Ma trận tương quan của các biến định lượng")
plt.show()
```




```
#9. Dựa vào heatmap ở câu 8, hãy chọn 5 thuộc tính độc lập để dự đoán giá nhà.
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import r2_score, mean_squared_error
import numpy as np

# 5 biến độc lập
X = df[["OverallQual", "GrLivArea",  "GarageCars","GarageArea", "TotalBsmtSF"]]
y = df["SalePrice"]

# Chia train / test
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.3, random_state=42
)

# Huấn luyện mô hình
model = LinearRegression()
model.fit(X_train, y_train)

# Dự đoán
y_pred = model.predict(X_test)

# Đánh giá mô hình
r2 = r2_score(y_test, y_pred)
rmse = np.sqrt(mean_squared_error(y_test, y_pred))
```

```
print("R2:", r2)
print("RMSE:", rmse)

# In công thức hồi quy
print("\nPhương trình hồi quy:")
print(f"SalePrice = {model.intercept_:.2f}", end=" ")

for coef, col in zip(model.coef_, X.columns):
    print(f"+ ({coef:.2f})*{col}", end=" ")
```

```
R2: 0.7936613418610964
RMSE: 37945.36207495814

Phương trình hồi quy:
SalePrice = -93371.59 + (24066.80)*OverallQual + (41.17)*GrLivArea + (18531.03)*GarageCars + (9.39)*GarageArea + (25.20)*TotalBs
```

```
#10. Hãy viết ra công thức hồi quy, tính hệ số R2, RMSE và kết luận cho câu 9.
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
```