importing the libraries

In [396]:

```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler, PolynomialFeatures
from sklearn.linear_model import LinearRegression, Lasso
from sklearn.metrics import mean_squared_error, r2_score
from sklearn.feature_selection import SelectFromModel
```

Loading the dataset

In [397]:

```
house_df = pd.read_csv('C:/Users/lekshmi/Downloads/houseprice.csv')
```

Data PreProcessing

In [398]:

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

In [399]:

```
# Handle missing values - impute with median for numerical columns and mode for categorical column
```

In [400]:

```
num_cols = house_df.select_dtypes(include=np.number).columns.tolist()
cat_cols = house_df.select_dtypes(include='object').columns.tolist()

for col in num_cols:
    house_df[col].fillna(house_df[col].median(), inplace=True)

for col in cat_cols:
    house_df[col].fillna(house_df[col].mode()[0], inplace=True)
```

In [401]:

```
#### Create new features
```

In [402]:

```
house_df['TotalBathrooms'] = house_df['FullBath'] + house_df['HalfBath']
```

Split data into features (X) and target variable (y)

In [403]:

```
# Select features and target variable
X = house_df[['GrLivArea', 'BedroomAbvGr', 'TotalBathrooms']]
y = house_df['SalePrice']
```

Split data into Training and Test data

In [404]:

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=100)
```

Standardisation of features

In [405]:

```
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)
```

Add polynomial features

In [406]:

```
poly = PolynomialFeatures(degree=2)
X_train_poly = poly.fit_transform(X_train_scaled)
X_test_poly = poly.transform(X_test_scaled)
```

Perform feature selection using Lasso regularization

In [407]:

```
lasso = Lasso(alpha=0.001)  # Adjust alpha as needed
lasso.fit(X_train_poly, y_train)
model = SelectFromModel(lasso, prefit=True)
X_train_selected = model.transform(X_train_poly)
X_test_selected = model.transform(X_test_poly)
```

Initialize and train the Linear Regression model

In [408]:

```
model = LinearRegression()
model.fit(X_train_selected, y_train)
```

Out[408]:

```
▾ LinearRegression
LinearRegression()
```

Make predictions on the test set

In [409]:

```
y_pred = model.predict(X_test_selected)
```

Evaluating the model

In [410]:

```
mse = mean_squared_error(y_test, y_pred)
rmse = mse ** 0.5
r2 = r2_score(y_test, y_pred)

print(f"Mean Squared Error (MSE): {mse}")
print(f"Root Mean Squared Error (RMSE): {rmse}")
print(f"R-squared (R2) Score: {r2}")
```

```
Mean Squared Error (MSE): 2070456205.391273
Root Mean Squared Error (RMSE): 45502.26593688795
R-squared (R2) Score: 0.6779550216322588
```