Zillow House Price Prediction Model

import necessary libraries

In [68]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score
from sklearn.ensemble import RandomForestRegressor
from sklearn.model_selection import GridSearchCV
```

Load the Ames Housing dataset

In [71]:
```
data = r"C:\Users\ssing\OneDrive\Desktop\AmesHousing.csv"
data = pd.read_csv(data)
```

Explore the dataset

In [74]:
```
# Explore the dataset
print("Dataset Information:")
data.info()
print("\nMissing Values:")
print(data.isnull().sum())
```

```
Dataset Information:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2930 entries, 0 to 2929
Data columns (total 82 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   Order            2930 non-null   int64
 1   PID              2930 non-null   int64
 2   MS SubClass      2930 non-null   int64
 3   MS Zoning        2930 non-null   object
 4   Lot Frontage     2440 non-null   float64
 5   Lot Area         2930 non-null   int64
 6   Street           2930 non-null   object
 7   Alley            198 non-null    object
 8   Lot Shape        2930 non-null   object
 9   Land Contour     2930 non-null   object
 10  Utilities        2930 non-null   object
 11  Lot Config       2930 non-null   object
 12  Land Slope       2930 non-null   object
 13  Neighborhood     2930 non-null   object
 14  Condition 1      2930 non-null   object
 15  Condition 2      2930 non-null   object
 16  Bldg Type        2930 non-null   object
 17  House Style      2930 non-null   object
 18  Overall Qual     2930 non-null   int64
 19  Overall Cond     2930 non-null   int64
 20  Year Built       2930 non-null   int64
 21  Year Remod/Add   2930 non-null   int64
 22  Roof Style       2930 non-null   object
 23  Roof Matl        2930 non-null   object
 24  Exterior 1st     2930 non-null   object
 25  Exterior 2nd     2930 non-null   object
 26  Mas Vnr Type     1155 non-null   object
 27  Mas Vnr Area     2907 non-null   float64
 28  Exter Qual       2930 non-null   object
 29  Exter Cond       2930 non-null   object
 30  Foundation       2930 non-null   object
 31  Bsmt Qual        2850 non-null   object
 32  Bsmt Cond        2850 non-null   object
 33  Bsmt Exposure    2847 non-null   object
 34  BsmtFin Type 1   2850 non-null   object
 35  BsmtFin SF 1     2929 non-null   float64
```

```
 36  BsmtFin Type 2   2849 non-null   object 
 37  BsmtFin SF 2     2929 non-null   float64
 38  Bsmt Unf SF      2929 non-null   float64
 39  Total Bsmt SF    2929 non-null   float64
 40  Heating          2930 non-null   object 
 41  Heating QC       2930 non-null   object 
 42  Central Air      2930 non-null   object 
 43  Electrical       2929 non-null   object 
 44  1st Flr SF       2930 non-null   int64  
 45  2nd Flr SF       2930 non-null   int64  
 46  Low Qual Fin SF  2930 non-null   int64  
 47  Gr Liv Area      2930 non-null   int64  
 48  Bsmt Full Bath   2928 non-null   float64
 49  Bsmt Half Bath   2928 non-null   float64
 50  Full Bath        2930 non-null   int64  
 51  Half Bath        2930 non-null   int64  
 52  Bedroom AbvGr    2930 non-null   int64  
 53  Kitchen AbvGr    2930 non-null   int64  
 54  Kitchen Qual     2930 non-null   object 
 55  TotRms AbvGrd    2930 non-null   int64  
 56  Functional       2930 non-null   object 
 57  Fireplaces       2930 non-null   int64  
 58  Fireplace Qu     1508 non-null   object 
 59  Garage Type      2773 non-null   object 
 60  Garage Yr Blt    2771 non-null   float64
 61  Garage Finish    2771 non-null   object 
 62  Garage Cars      2929 non-null   float64
 63  Garage Area      2929 non-null   float64
 64  Garage Qual      2771 non-null   object 
 65  Garage Cond      2771 non-null   object 
 66  Paved Drive      2930 non-null   object 
 67  Wood Deck SF     2930 non-null   int64  
 68  Open Porch SF    2930 non-null   int64  
 69  Enclosed Porch   2930 non-null   int64  
 70  3Ssn Porch       2930 non-null   int64  
 71  Screen Porch     2930 non-null   int64  
 72  Pool Area        2930 non-null   int64  
 73  Pool QC          13 non-null     object 
 74  Fence            572 non-null    object 
 75  Misc Feature     106 non-null    object 
 76  Misc Val         2930 non-null   int64  
 77  Mo Sold          2930 non-null   int64  
```

```
 78  Yr Sold          2930 non-null   int64
 79  Sale Type        2930 non-null   object
 80  Sale Condition   2930 non-null   object
 81  SalePrice        2930 non-null   int64
dtypes: float64(11), int64(28), object(43)
memory usage: 1.8+ MB

Missing Values:
Order               0
PID                 0
MS SubClass         0
MS Zoning           0
Lot Frontage      490
                 ...
Mo Sold             0
Yr Sold             0
Sale Type           0
Sale Condition      0
SalePrice           0
Length: 82, dtype: int64
```

Handle Missing Values

Numerical columns: Fill with mean

In [161…

```
#Numerical columns: Fill with mean
num_cols = data.select_dtypes(include=['float64', 'int64']).columns
for col in num_cols:
    if data[col].isnull().sum() > 0:
        data[col].fillna(data[col].mean(), inplace=True)
```

Categorical columns: Fill with mode

In [163…

```
# Categorical columns: Fill with mode
cat_cols = data.select_dtypes(include=['object']).columns
for col in cat_cols:
    if data[col].isnull().sum() > 0:
        data[col].fillna(data[col].mode()[0], inplace=True)
```

Verify no missing values remain

In [84]:
```
# Verify no missing values remain
print("\nMissing Values After Imputation:")
print(data.isnull().sum().sum())
```

```
Missing Values After Imputation:
0
```

Remove duplicates

In [168…
```
data = data.drop_duplicates()
print(f"Number of rows after removing duplicates: {data.shape[0]}")
```

```
Number of rows after removing duplicates: 2930
```

Convert categorical variables to numerical

In [94]:
```
data = pd.get_dummies(data, drop_first=True)
```

Define features and target

In [97]:
```
print(data.columns)
```

```
Index(['Order', 'PID', 'MS SubClass', 'Lot Frontage', 'Lot Area',
       'Overall Qual', 'Overall Cond', 'Year Built', 'Year Remod/Add',
       'Mas Vnr Area',
       ...
       'Sale Type_ConLw', 'Sale Type_New', 'Sale Type_Oth', 'Sale Type_VWD',
       'Sale Type_WD ', 'Sale Condition_AdjLand', 'Sale Condition_Alloca',
       'Sale Condition_Family', 'Sale Condition_Normal',
       'Sale Condition_Partial'],
      dtype='object', length=263)
```

In [170…
```
# Define features and target
X = data[['Lot Area']]  # Example: Selecting a single feature
y = data['SalePrice']
```

Dropping less relevant columns

In [45]:
```
# Dropping less relevant columns (if needed based on domain knowledge)
X = data.drop('SalePrice', axis=1)
```

```
y = data['SalePrice']
```

Split data into training and testing sets

In [172…

```
# Step 3: Split the dataset into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

Build and train the linear regression model

In [174…

```
# Build and train the linear regression model
model = LinearRegression()
model.fit(X_train, y_train)
```

Out[174…

```
▾ LinearRegression
LinearRegression()
```

Make predictions on the set

In [176…

```
predictions = model.predict(X_test)
```

Evaluate The model

In [178…

```
print("\nModel Performance:")
print("MAE:", mean_absolute_error(y_test, predictions))
print("MSE:", mean_squared_error(y_test, predictions))
print("R-squared:", r2_score(y_test, predictions))
```

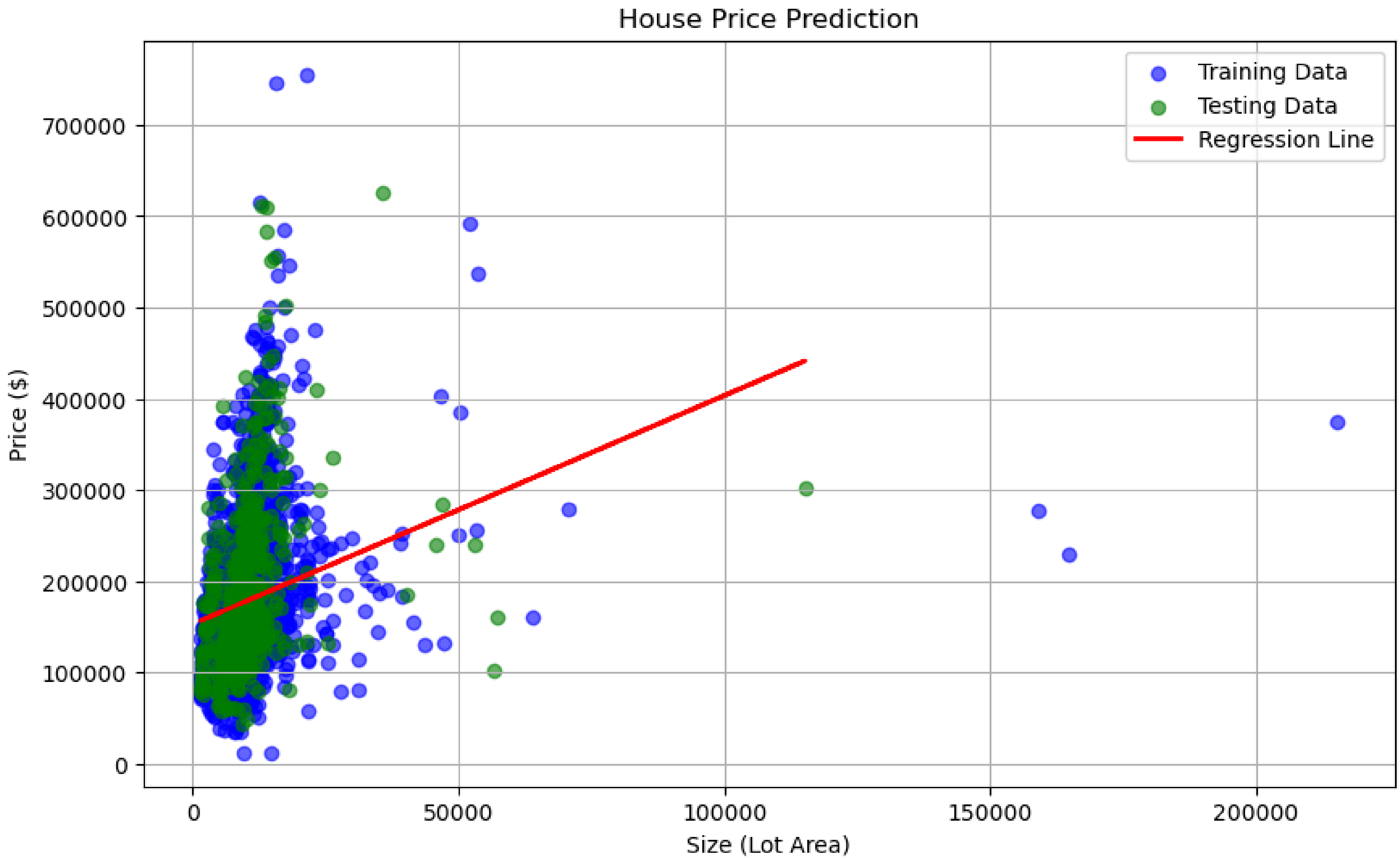
```
Model Performance:
MAE: 62056.86000101161
MSE: 7509189795.222837
R-squared: 0.06340568713349304
```

Visualize the results

In [180…

```
 Visualize the results
plt.figure(figsize=(10, 6))
# Plot training data
```

```
plt.scatter(X_train, y_train, color='blue', label='Training Data', alpha=0.6)
# Plot testing data
plt.scatter(X_test, y_test, color='green', label='Testing Data', alpha=0.6)
# Plot regression line
plt.plot(X_test, predictions, color='red', linewidth=2, label='Regression Line')
plt.title('House Price Prediction')
plt.xlabel('Size (Lot Area)')
plt.ylabel('Price ($)')
plt.legend()
plt.grid()
plt.show()
```


House Price Prediction
Training Data
Testing Data
Regression Line
Price ($)
Size (Lot Area)
0
100000
200000
300000
400000
500000
600000
700000
0
50000
100000
150000
200000

```
In [ ]:
```