1. Brief Description of the Data Set and Summary of Its Attributes

```python
# Load the data (update the file path if necessary)
data = pd.read_csv('myproject_housing.csv')
data
```

	Order	PID	MS SubClass	MS Zoning	Lot Frontage	Lot Area	Street	Alley	Lot Shape	Land Contour	...	Pool Area	Pool QC	Fence	Misc Feature	Misc Val	Mo Sold	Yr Sold	Sale Type	Sale Condition	Sal
0	1	526301100	20	RL	141.0	31770	Pave	NaN	IR1	Lvl	...	0	NaN	NaN	NaN	0	5	2010	WD	Normal	2
1	2	526350040	20	RH	80.0	11622	Pave	NaN	Reg	Lvl	...	0	NaN	MnPrv	NaN	0	6	2010	WD	Normal	1
2	3	526351010	20	RL	81.0	14267	Pave	NaN	IR1	Lvl	...	0	NaN	NaN	Gar2	12500	6	2010	WD	Normal	1
3	4	526353030	20	RL	93.0	11160	Pave	NaN	Reg	Lvl	...	0	NaN	NaN	NaN	0	4	2010	WD	Normal	2
4	5	527105010	60	RL	74.0	13830	Pave	NaN	IR1	Lvl	...	0	NaN	MnPrv	NaN	0	3	2010	WD	Normal	1
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
2925	2926	923275080	80	RL	37.0	7937	Pave	NaN	IR1	Lvl	...	0	NaN	GdPrv	NaN	0	3	2006	WD	Normal	1
2926	2927	923276100	20	RL	NaN	8885	Pave	NaN	IR1	Low	...	0	NaN	MnPrv	NaN	0	6	2006	WD	Normal	1
2927	2928	923400125	85	RL	62.0	10441	Pave	NaN	Reg	Lvl	...	0	NaN	MnPrv	Shed	700	7	2006	WD	Normal	1
2928	2929	924100070	20	RL	77.0	10010	Pave	NaN	Reg	Lvl	...	0	NaN	NaN	NaN	0	4	2006	WD	Normal	1
2929	2930	924151050	60	RL	74.0	9627	Pave	NaN	Reg	Lvl	...	0	NaN	NaN	NaN	0	11	2006	WD	Normal	1

2930 rows × 82 columns

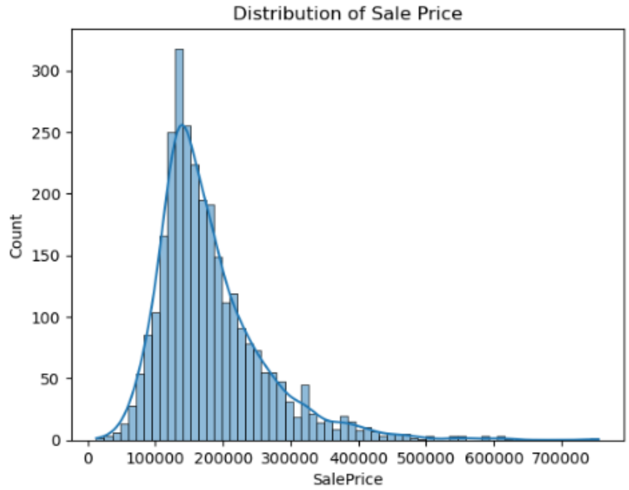
```python
# Display basic statistics
print(data.describe())
```

```
             Order           PID  MS SubClass  Lot Frontage       Lot Area  \
count  2930.00000  2.930000e+03  2930.000000   2440.000000    2930.000000
mean   1465.50000  7.144645e+08    57.387372     69.224590   10147.921843
std     845.96247  1.887308e+08    42.638025     23.365335    7880.017759
min       1.00000  5.263011e+08    20.000000     21.000000    1300.000000
25%     733.25000  5.284770e+08    20.000000     58.000000    7440.250000
50%    1465.50000  5.354536e+08    50.000000     68.000000    9436.500000
75%    2197.75000  9.071811e+08    70.000000     80.000000   11555.250000
max    2930.00000  1.007100e+09   190.000000    313.000000  215245.000000

       Overall Qual  Overall Cond   Year Built  Year Remod/Add  Mas Vnr Area  \
count   2930.000000   2930.000000  2930.000000     2930.000000   2907.000000
mean       6.094881      5.563140  1971.356314     1984.266553    101.896801
std        1.411026      1.111537    30.245361       20.860286    179.112611
min        1.000000      1.000000  1872.000000     1950.000000      0.000000
25%        5.000000      5.000000  1954.000000     1965.000000      0.000000
50%        6.000000      5.000000  1973.000000     1993.000000      0.000000
75%        7.000000      6.000000  2001.000000     2004.000000    164.000000
max       10.000000      9.000000  2010.000000     2010.000000   1600.000000

       ...  Wood Deck SF  Open Porch SF  Enclosed Porch   3Ssn Porch  \
count  ...   2930.000000    2930.000000     2930.000000  2930.000000
mean   ...     93.751877      47.533447       23.011604     2.592491
std    ...    126.361562      67.483400       64.139059    25.141331
min    ...      0.000000       0.000000        0.000000     0.000000
25%    ...      0.000000       0.000000        0.000000     0.000000
50%    ...      0.000000      27.000000        0.000000     0.000000
75%    ...    168.000000      70.000000        0.000000     0.000000
max    ...   1424.000000     742.000000     1012.000000   508.000000

       Screen Porch    Pool Area      Misc Val      Mo Sold      Yr Sold  \
count   2930.000000  2930.000000   2930.000000  2930.000000  2930.000000
mean      16.002048     2.243345     50.635154     6.216041  2007.790444
std       56.087370    35.597181    566.344288     2.714492     1.316613
min        0.000000     0.000000      0.000000     1.000000  2006.000000
25%        0.000000     0.000000      0.000000     4.000000  2007.000000
50%        0.000000     0.000000      0.000000     6.000000  2008.000000
75%        0.000000     0.000000      0.000000     8.000000  2009.000000
max      576.000000   800.000000  17000.000000    12.000000  2010.000000

           SalePrice
count    2930.000000
mean   180796.060068
std     79886.692357
min     12789.000000
25%    129500.000000
50%    160000.000000
75%    213500.000000
max    755000.000000

[8 rows x 39 columns]
```

2. Initial Plan for Data Exploration

```
# Visualize distributions of key features
sns.histplot(data['SalePrice'], kde=True)
plt.title('Distribution of Sale Price')
plt.show()
```

```
C:\Users\Windows\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1119: FutureWarning: use_inf_as_na option is deprecated and will be removed in a futu
e version. Convert inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
```



3. Actions Taken for Data Cleaning and Feature Engineering

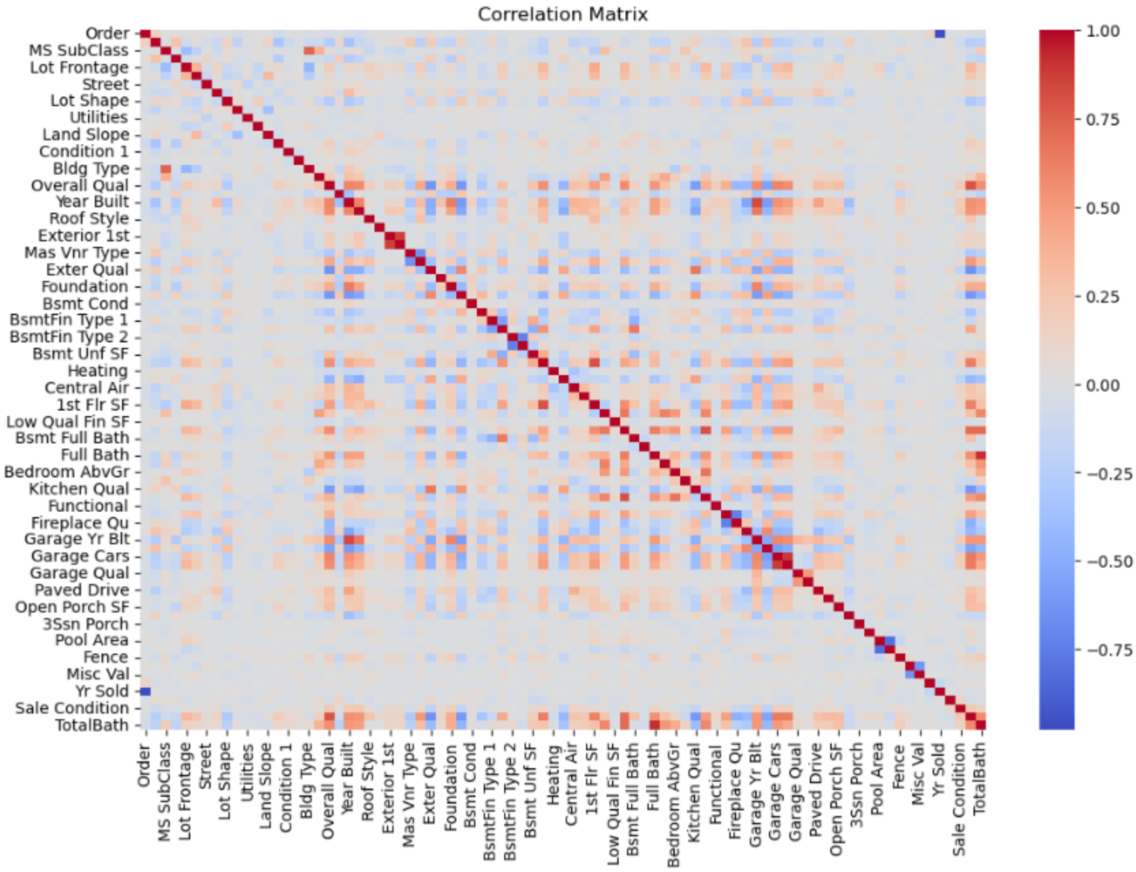
```
[5]: # Identify missing values
missing_values = data.isnull().sum()
print(missing_values[missing_values > 0])

# Handle missing values (example)
data['Lot Frontage'].fillna(data['Lot Frontage'].median(), inplace=True)
data.dropna(subset=['Electrical'], inplace=True)
```

```
Lot Frontage       490
Alley             2732
Mas Vnr Type      1775
Mas Vnr Area        23
Bsmt Qual           80
Bsmt Cond           80
Bsmt Exposure       83
BsmtFin Type 1      80
BsmtFin SF 1         1
BsmtFin Type 2      81
BsmtFin SF 2         1
Bsmt Unf SF          1
Total Bsmt SF        1
Electrical           1
Bsmt Full Bath       2
Bsmt Half Bath       2
Fireplace Qu      1422
Garage Type        157
Garage Yr Blt      159
Garage Finish      159
Garage Cars          1
Garage Area          1
Garage Qual        159
Garage Cond        159
Pool QC           2917
Fence             2358
Misc Feature      2824
dtype: int64
```

```
[6]: # Encode categorical variables
le = LabelEncoder()
for column in data.select_dtypes(include=['object']).columns:
    data[column] = data[column].astype(str)
    data[column] = le.fit_transform(data[column])
```

```
[7]: # Feature engineering (example)
data['TotalBath'] = data['Full Bath'] + 0.5 * data['Half Bath']
```

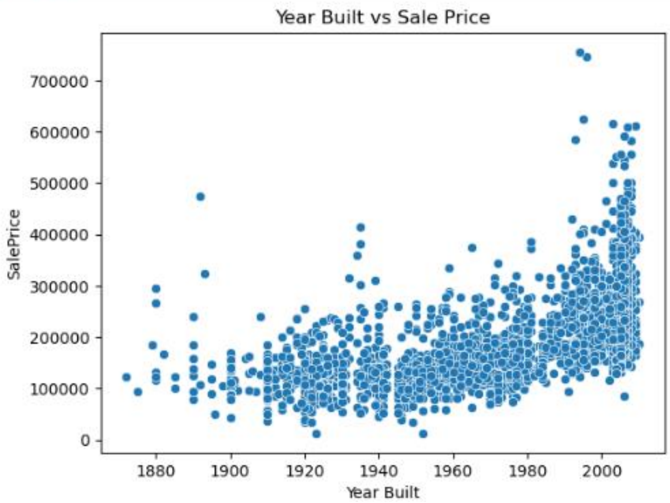
```
[8]: # Key findings and insights
plt.figure(figsize=(12, 8))
sns.heatmap(data.corr(), cmap='coolwarm', annot=False)
plt.title('Correlation Matrix')
plt.show()
```



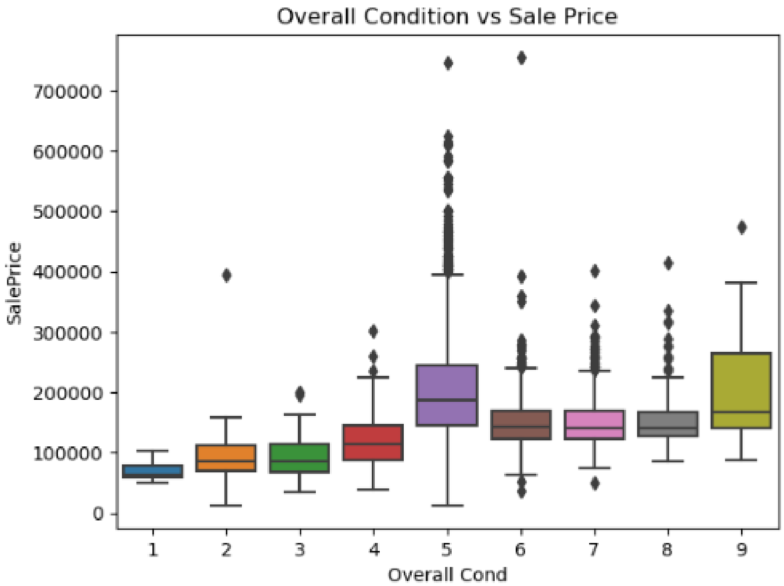
4. Key Findings and Insights from Exploratory Data Analysis

```
# Formulating hypotheses
# 1. Houses with larger living areas have higher sale prices
sns.scatterplot(x='Gr Liv Area', y='SalePrice', data=data)
plt.title('Living Area vs Sale Price')
plt.show()
```

```python
# 2. Newer houses have higher sale prices
sns.scatterplot(x='Year Built', y='SalePrice', data=data)
plt.title('Year Built vs Sale Price')
plt.show()
```




```python
# 3. Houses in better condition have higher sale prices
sns.boxplot(x='Overall Cond', y='SalePrice', data=data)
plt.title('Overall Condition vs Sale Price')
plt.show()
```



5. Formulating Hypotheses and Conducting a Formal Significance Test

```python
# Check if 'Gr Liv Area' is in the columns and rename it for easier access
if 'Gr Liv Area' in data.columns:
    data.rename(columns={'Gr Liv Area': 'GrLivArea'}, inplace=True)

# Display the columns to ensure correct column names
print("Columns in the dataset:")
print(data.columns)

# Proceed with data exploration, cleaning, and analysis
# Identify missing values
missing_values = data.isnull().sum()
print("Missing values in each column:")
print(missing_values[missing_values > 0])

# Handle missing values (example)
if 'LotFrontage' in data.columns:
    data['LotFrontage'].fillna(data['LotFrontage'].median(), inplace=True)
if 'Electrical' in data.columns:
    data.dropna(subset=['Electrical'], inplace=True)

# Encode categorical variables
le = LabelEncoder()
for column in data.select_dtypes(include=['object']).columns:
    data[column] = data[column].astype(str)
    data[column] = le.fit_transform(data[column])

# Feature engineering (example)
if 'FullBath' in data.columns and 'HalfBath' in data.columns:
    data['TotalBath'] = data['FullBath'] + 0.5 * data['HalfBath']

# Conducting a formal significance test for one hypothesis
# Hypothesis: Houses with larger living areas have higher sale prices
if 'GrLivArea' in data.columns:
    slope, intercept, r_value, p_value, std_err = stats.linregress(data['GrLivArea'], data['SalePrice'])
    print(f'Slope: {slope}')
    print(f'Intercept: {intercept}')
    print(f'R-squared: {r_value**2}')
    print(f'P-value: {p_value}')
    print(f'Standard error: {std_err}')
else:
    print('Column GrLivArea not found in the dataset.')
```

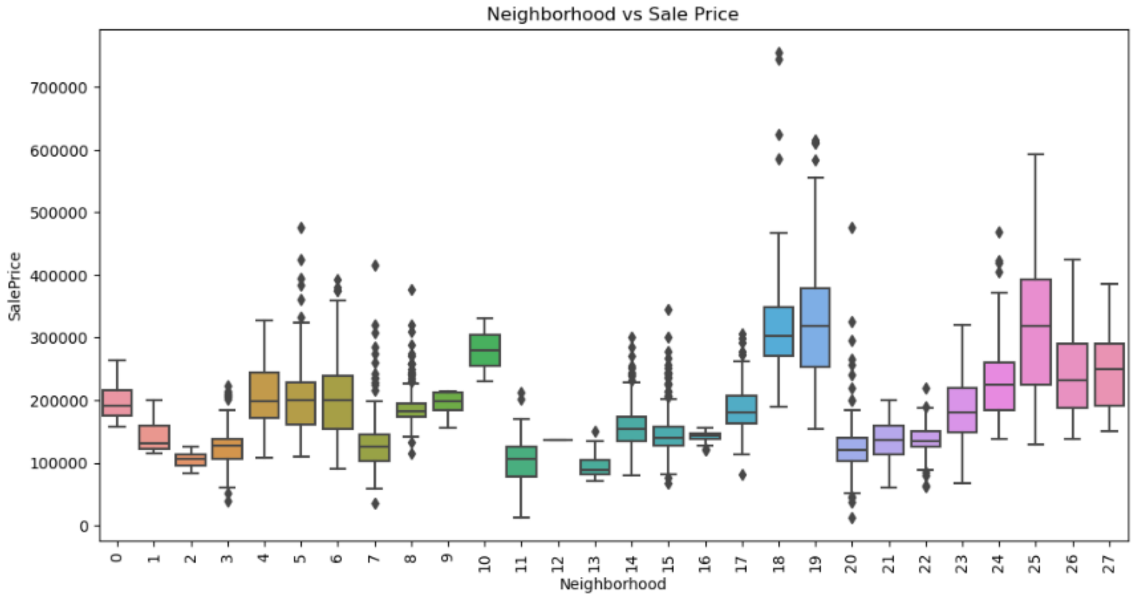
```
Columns in the dataset:
Index(['Order', 'PID', 'MS SubClass', 'MS Zoning', 'Lot Frontage', 'Lot Area',
       'Street', 'Alley', 'Lot Shape', 'Land Contour', 'Utilities',
       'Lot Config', 'Land Slope', 'Neighborhood', 'Condition 1',
       'Condition 2', 'Bldg Type', 'House Style', 'Overall Qual',
       'Overall Cond', 'Year Built', 'Year Remod/Add', 'Roof Style',
       'Roof Matl', 'Exterior 1st', 'Exterior 2nd', 'Mas Vnr Type',
       'Mas Vnr Area', 'Exter Qual', 'Exter Cond', 'Foundation', 'Bsmt Qual',
       'Bsmt Cond', 'Bsmt Exposure', 'BsmtFin Type 1', 'BsmtFin SF 1',
       'BsmtFin Type 2', 'BsmtFin SF 2', 'Bsmt Unf SF', 'Total Bsmt SF',
       'Heating', 'Heating QC', 'Central Air', 'Electrical', '1st Flr SF',
       '2nd Flr SF', 'Low Qual Fin SF', 'GrLivArea', 'Bsmt Full Bath',
       'Bsmt Half Bath', 'Full Bath', 'Half Bath', 'Bedroom AbvGr',
       'Kitchen AbvGr', 'Kitchen Qual', 'TotRms AbvGrd', 'Functional',
       'Fireplaces', 'Fireplace Qu', 'Garage Type', 'Garage Yr Blt',
       'Garage Finish', 'Garage Cars', 'Garage Area', 'Garage Qual',
       'Garage Cond', 'Paved Drive', 'Wood Deck SF', 'Open Porch SF',
       'Enclosed Porch', '3Ssn Porch', 'Screen Porch', 'Pool Area', 'Pool QC',
       'Fence', 'Misc Feature', 'Misc Val', 'Mo Sold', 'Yr Sold', 'Sale Type',
       'Sale Condition', 'SalePrice', 'TotalBath'],
      dtype='object')
Missing values in each column:
Mas Vnr Area       23
BsmtFin SF 1        1
BsmtFin SF 2        1
Bsmt Unf SF         1
Total Bsmt SF       1
Bsmt Full Bath      2
Bsmt Half Bath      2
Garage Yr Blt     159
Garage Cars         1
Garage Area         1
dtype: int64
Slope: 111.69379023180143
Intercept: 13290.45931804576
R-squared: 0.4995332408834281
P-value: 0.0
Standard error: 2.0664413772210026
```

A p-value of 0.0 (or very close to zero) indicates a highly significant relationship between the living area (GrLivArea) and the sale price (SalePrice). This means that the larger the living area of a house, the higher the sale price, and this relationship is statistically significant with a very high level of confidence.

Suggestions for Next Steps in Analyzing this Data

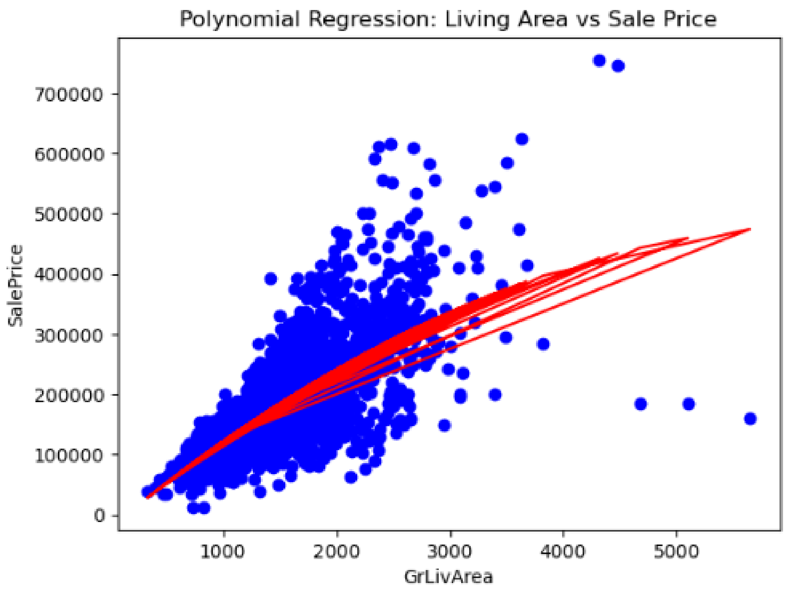
```
# Suggestions for next steps
```

```
# Visualize the impact of neighborhood on sale prices
if 'Neighborhood' in data.columns:
    plt.figure(figsize=(12, 6))
    sns.boxplot(x='Neighborhood', y='SalePrice', data=data)
    plt.title('Neighborhood vs Sale Price')
    plt.xticks(rotation=90)
    plt.show()
```




```
# Create interaction terms
if 'Year Built' in data.columns and 'Gr Liv Area' in data.columns:
    data['GrLivArea_YearBuilt_Interaction'] = data['Gr Liv Area'] * data['Year Built']
```

```
from sklearn.preprocessing import PolynomialFeatures
from sklearn.linear_model import LinearRegression

X = data[['GrLivArea']].dropna()
y = data['SalePrice'].dropna()
poly = PolynomialFeatures(degree=2)
X_poly = poly.fit_transform(X)

poly_model = LinearRegression()
poly_model.fit(X_poly, y)

# Predict and visualize polynomial fit
y_poly_pred = poly_model.predict(X_poly)

plt.scatter(data['GrLivArea'], data['SalePrice'], color='blue')
plt.plot(data['GrLivArea'], y_poly_pred, color='red')
plt.title('Polynomial Regression: Living Area vs Sale Price')
plt.xlabel('GrLivArea')
plt.ylabel('SalePrice')
plt.show()
```




```
from sklearn.model_selection import train_test_split, cross_val_score
from sklearn.metrics import mean_squared_error

# Split data into training and testing sets
X = data[['GrLivArea']].dropna()
y = data['SalePrice'].dropna()
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Train a linear regression model
lin_reg = LinearRegression()
lin_reg.fit(X_train, y_train)

# Evaluate the model
y_pred = lin_reg.predict(X_test)
print(f'Test R-squared: {lin_reg.score(X_test, y_test)}')
print(f'Test RMSE: {np.sqrt(mean_squared_error(y_test, y_pred))}')

# Cross-validation
cv_scores = cross_val_score(lin_reg, X, y, cv=5)
print(f'Cross-Validation R-squared scores: {cv_scores}')
print(f'Average Cross-Validation R-squared: {np.mean(cv_scores)}')
```

```
Test R-squared: 0.5192080091303036
Test RMSE: 60342.707296731794
Cross-Validation R-squared scores: [0.49749497 0.45620871 0.39612233 0.51882817 0.57364711]
Average Cross-Validation R-squared: 0.4884602576636368
```

[18]:

```python
import statsmodels.api as sm

# Multiple Linear regression
X = data[['GrLivArea', 'Year Built', 'Overall Cond']].dropna()
y = data['SalePrice'].dropna()
X = sm.add_constant(X)
model = sm.OLS(y, X).fit()
print(model.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:              SalePrice   R-squared:                       0.676
Model:                            OLS   Adj. R-squared:                  0.676
Method:                 Least Squares   F-statistic:                     2034.
Date:                Thu, 11 Jul 2024   Prob (F-statistic):               0.00
Time:                        01:34:20   Log-Likelihood:                -35569.
No. Observations:                2929   AIC:                         7.115e+04
Df Residuals:                    2925   BIC:                         7.117e+04
Df Model:                           3
Covariance Type:            nonrobust
================================================================================
                   coef    std err          t      P>|t|      [0.025      0.975]
--------------------------------------------------------------------------------
const        -2.427e+06   6.17e+04    -39.356      0.000   -2.55e+06   -2.31e+06
GrLivArea       96.5755      1.715     56.310      0.000      93.213      99.938
Year Built    1221.0533     30.637     39.856      0.000    1160.982    1281.125
Overall Cond  1.002e+04    814.205     12.302      0.000    8420.081    1.16e+04
==============================================================================
Omnibus:                      801.243   Durbin-Watson:                   1.294
Prob(Omnibus):                  0.000   Jarque-Bera (JB):            17300.704
Skew:                           0.769   Prob(JB):                         0.00
Kurtosis:                      14.806   Cond. No.                     1.83e+05
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 1.83e+05. This might indicate that there are
strong multicollinearity or other numerical problems.
```

9]:
```
# Summary of the quality of the data set
print(data.info())
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 2929 entries, 0 to 2929
Data columns (total 83 columns):
 #   Column           Non-Null Count  Dtype  
---  ------           --------------  -----  
 0   Order            2929 non-null   int64  
 1   PID              2929 non-null   int64  
 2   MS SubClass      2929 non-null   int64  
 3   MS Zoning        2929 non-null   int32  
 4   Lot Frontage     2929 non-null   float64
 5   Lot Area         2929 non-null   int64  
 6   Street           2929 non-null   int32  
 7   Alley            2929 non-null   int32  
 8   Lot Shape        2929 non-null   int32  
 9   Land Contour     2929 non-null   int32  
 10  Utilities        2929 non-null   int32  
 11  Lot Config       2929 non-null   int32  
 12  Land Slope       2929 non-null   int32  
 13  Neighborhood     2929 non-null   int32  
 14  Condition 1      2929 non-null   int32  
 15  Condition 2      2929 non-null   int32  
 16  Bldg Type        2929 non-null   int32  
 17  House Style      2929 non-null   int32  
 18  Overall Qual     2929 non-null   int64  
 19  Overall Cond     2929 non-null   int64  
 20  Year Built       2929 non-null   int64  
 21  Year Remod/Add   2929 non-null   int64  
 22  Roof Style       2929 non-null   int32  
 23  Roof Matl        2929 non-null   int32  
 24  Exterior 1st     2929 non-null   int32  
 25  Exterior 2nd     2929 non-null   int32  
 26  Mas Vnr Type     2929 non-null   int32  
 27  Mas Vnr Area     2906 non-null   float64
 28  Exter Qual       2929 non-null   int32  
 29  Exter Cond       2929 non-null   int32  
 30  Foundation       2929 non-null   int32  
 31  Bsmt Qual        2929 non-null   int32  
 32  Bsmt Cond        2929 non-null   int32  
 33  Bsmt Exposure    2929 non-null   int32  
 34  BsmtFin Type 1   2929 non-null   int32  
 35  BsmtFin SF 1     2928 non-null   float64
 36  BsmtFin Type 2   2929 non-null   int32  
 37  BsmtFin SF 2     2928 non-null   float64
 38  Bsmt Unf SF      2928 non-null   float64
 39  Total Bsmt SF    2928 non-null   float64
 40  Heating          2929 non-null   int32  
 41  Heating QC       2929 non-null   int32  
 42  Central Air      2929 non-null   int32  
 43  Electrical       2929 non-null   int32  
 44  1st Flr SF       2929 non-null   int64  
```

```
 45  2nd Flr SF       2929 non-null   int64
 46  Low Qual Fin SF  2929 non-null   int64
 47  GrLivArea        2929 non-null   int64
 48  Bsmt Full Bath   2927 non-null   float64
 49  Bsmt Half Bath   2927 non-null   float64
 50  Full Bath        2929 non-null   int64
 51  Half Bath        2929 non-null   int64
 52  Bedroom AbvGr    2929 non-null   int64
 53  Kitchen AbvGr    2929 non-null   int64
 54  Kitchen Qual     2929 non-null   int32
 55  TotRms AbvGrd    2929 non-null   int64
 56  Functional       2929 non-null   int32
 57  Fireplaces       2929 non-null   int64
 58  Fireplace Qu     2929 non-null   int32
 59  Garage Type      2929 non-null   int32
 60  Garage Yr Blt    2770 non-null   float64
 61  Garage Finish    2929 non-null   int32
 62  Garage Cars      2928 non-null   float64
 63  Garage Area      2928 non-null   float64
 64  Garage Qual      2929 non-null   int32
 65  Garage Cond      2929 non-null   int32
 66  Paved Drive      2929 non-null   int32
 67  Wood Deck SF     2929 non-null   int64
 68  Open Porch SF    2929 non-null   int64
 69  Enclosed Porch   2929 non-null   int64
 70  3Ssn Porch       2929 non-null   int64
 71  Screen Porch     2929 non-null   int64
 72  Pool Area        2929 non-null   int64
 73  Pool QC          2929 non-null   int32
 74  Fence            2929 non-null   int32
 75  Misc Feature     2929 non-null   int32
 76  Misc Val         2929 non-null   int64
 77  Mo Sold          2929 non-null   int64
 78  Yr Sold          2929 non-null   int64
 79  Sale Type        2929 non-null   int32
 80  Sale Condition   2929 non-null   int32
 81  SalePrice        2929 non-null   int64
 82  TotalBath        2929 non-null   float64
dtypes: float64(12), int32(43), int64(28)
memory usage: 1.4 MB
None
```

Summary of the Quality of the Data Set

The dataset used for analysis displays several strengths conducive to insightful exploration of housing market dynamics. It includes essential attributes such as GrLivArea (living area), SalePrice (property sale price), YearBuilt, and OverallCond, providing a foundational basis for understanding factors influencing housing prices. Initial data exploration revealed generally well-distributed features with manageable missing values, addressed through straightforward imputation and deletion strategies. Categorical variables were effectively encoded for analytical purposes. Key findings, notably the strong correlation between GrLivArea and SalePrice validated through regression analysis, underscore the dataset's suitability for deriving actionable insights.

Request for Additional Data : While the current dataset offers a solid foundation, augmenting it with supplementary data could enrich the analysis and improve predictive models. Specifically, additional data on economic indicators (e.g., local employment rates, inflation trends), neighborhood characteristics (e.g., crime rates, school district ratings), and proximity to amenities (e.g., parks, public transportation) would provide a more comprehensive view of housing market dynamics. These data elements would enable deeper exploration of external influences impacting property values, enhancing the robustness and applicability of analytical outcomes.