House Prices - Advanced Regression Techniques

February 5, 2025

1 Case : House Prices - Advanced Regression Techniques

1.0.1 by Hossein Izadi

CRISP-DM methodology

1.1 Business Understanding

Goal: ### Predict sales prices and practice feature engineering, RFs, and gradient boosting

1.2 For more information, see the project metadata: Project Metadata

[2]:
```
#Required libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import statsmodels.api as sm
import seaborn as sns
from scipy.stats import skew
from scipy.stats import zscore
from sklearn.preprocessing import LabelEncoder, OneHotEncoder, RobustScaler,␣
 ↪OrdinalEncoder
from sklearn.linear_model import Ridge, Lasso
from sklearn.metrics import mean_squared_error, f1_score, mean_absolute_error,␣
 ↪r2_score, make_scorer
import scipy.stats as stats
from catboost import CatBoostRegressor
from sklearn.ensemble import RandomForestRegressor, RandomForestClassifier
from sklearn.exceptions import ConvergenceWarning
from sklearn.linear_model import LinearRegression
from sklearn.neighbors import KNeighborsRegressor
from sklearn.svm import SVR
from sklearn.tree import DecisionTreeRegressor
from xgboost import XGBRegressor
from sklearn.model_selection import train_test_split, cross_val_score,␣
 ↪GridSearchCV
from sklearn.preprocessing import StandardScaler
```

2 Data Undrestanding

2.1 Load and preview the dataset

[4]:
```
#Read data from file
data_train = pd.read_csv('train.csv')
data_train.head()
```

[4]:
```
   Id  MSSubClass MSZoning  LotFrontage  LotArea Street Alley LotShape  \
0   1          60       RL         65.0     8450   Pave   NaN      Reg
1   2          20       RL         80.0     9600   Pave   NaN      Reg
2   3          60       RL         68.0    11250   Pave   NaN      IR1
3   4          70       RL         60.0     9550   Pave   NaN      IR1
4   5          60       RL         84.0    14260   Pave   NaN      IR1

  LandContour Utilities  … PoolArea PoolQC Fence MiscFeature MiscVal MoSold  \
0         Lvl    AllPub  …        0    NaN   NaN         NaN       0      2
1         Lvl    AllPub  …        0    NaN   NaN         NaN       0      5
2         Lvl    AllPub  …        0    NaN   NaN         NaN       0      9
3         Lvl    AllPub  …        0    NaN   NaN         NaN       0      2
4         Lvl    AllPub  …        0    NaN   NaN         NaN       0     12

  YrSold  SaleType  SaleCondition  SalePrice
0   2008        WD         Normal     208500
1   2007        WD         Normal     181500
2   2008        WD         Normal     223500
3   2006        WD        Abnorml     140000
4   2008        WD         Normal     250000

[5 rows x 81 columns]
```

[5]:
```
data_train.shape
```

[5]:
```
(1460, 81)
```

[ ]:
```

```

[6]:
```
data_train['SalePrice']
```

[6]:
```
0       208500
1       181500
2       223500
3       140000
4       250000
         …
1455    175000
1456    210000
1457    266500
```

```
1458    142125
1459    147500
Name: SalePrice, Length: 1460, dtype: int64
```

[7]:
```
# Summary statistics
data_train.info()
```

```
<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
 17  OverallQual    1460 non-null   int64
 18  OverallCond    1460 non-null   int64
 19  YearBuilt      1460 non-null   int64
 20  YearRemodAdd   1460 non-null   int64
 21  RoofStyle      1460 non-null   object
 22  RoofMatl       1460 non-null   object
 23  Exterior1st    1460 non-null   object
 24  Exterior2nd    1460 non-null   object
 25  MasVnrType     588 non-null    object
 26  MasVnrArea     1452 non-null   float64
 27  ExterQual      1460 non-null   object
 28  ExterCond      1460 non-null   object
 29  Foundation     1460 non-null   object
 30  BsmtQual       1423 non-null   object
 31  BsmtCond       1423 non-null   object
 32  BsmtExposure   1422 non-null   object
 33  BsmtFinType1   1423 non-null   object
 34  BsmtFinSF1     1460 non-null   int64
 35  BsmtFinType2   1422 non-null   object
```

```
 36  BsmtFinSF2     1460 non-null   int64
 37  BsmtUnfSF      1460 non-null   int64
 38  TotalBsmtSF    1460 non-null   int64
 39  Heating        1460 non-null   object
 40  HeatingQC      1460 non-null   object
 41  CentralAir     1460 non-null   object
 42  Electrical     1459 non-null   object
 43  1stFlrSF       1460 non-null   int64
 44  2ndFlrSF       1460 non-null   int64
 45  LowQualFinSF   1460 non-null   int64
 46  GrLivArea      1460 non-null   int64
 47  BsmtFullBath   1460 non-null   int64
 48  BsmtHalfBath   1460 non-null   int64
 49  FullBath       1460 non-null   int64
 50  HalfBath       1460 non-null   int64
 51  BedroomAbvGr   1460 non-null   int64
 52  KitchenAbvGr   1460 non-null   int64
 53  KitchenQual    1460 non-null   object
 54  TotRmsAbvGrd   1460 non-null   int64
 55  Functional     1460 non-null   object
 56  Fireplaces     1460 non-null   int64
 57  FireplaceQu    770 non-null    object
 58  GarageType     1379 non-null   object
 59  GarageYrBlt    1379 non-null   float64
 60  GarageFinish   1379 non-null   object
 61  GarageCars     1460 non-null   int64
 62  GarageArea     1460 non-null   int64
 63  GarageQual     1379 non-null   object
 64  GarageCond     1379 non-null   object
 65  PavedDrive     1460 non-null   object
 66  WoodDeckSF     1460 non-null   int64
 67  OpenPorchSF    1460 non-null   int64
 68  EnclosedPorch  1460 non-null   int64
 69  3SsnPorch      1460 non-null   int64
 70  ScreenPorch    1460 non-null   int64
 71  PoolArea       1460 non-null   int64
 72  PoolQC         7 non-null      object
 73  Fence          281 non-null    object
 74  MiscFeature    54 non-null     object
 75  MiscVal        1460 non-null   int64
 76  MoSold         1460 non-null   int64
 77  YrSold         1460 non-null   int64
 78  SaleType       1460 non-null   object
 79  SaleCondition  1460 non-null   object
 80  SalePrice      1460 non-null   int64
dtypes: float64(3), int64(35), object(43)
memory usage: 924.0+ KB
```

[8]:
```
data_test = pd.read_csv('test.csv')
data_test.head()
```

[8]:

	Id	MSSubClass	MSZoning	LotFrontage	LotArea	Street	Alley	LotShape	…	ScreenPorch	PoolArea	PoolQC	Fence	MiscFeature	MiscVal	MoSold	YrSold	SaleType	SaleCondition
0	1461	20	RH	80.0	11622	Pave	NaN	Reg	…	120	0	NaN	MnPrv	NaN	0	6	2010	WD	Normal
1	1462	20	RL	81.0	14267	Pave	NaN	IR1	…	0	0	NaN	NaN	Gar2	12500	6	2010	WD	Normal
2	1463	60	RL	74.0	13830	Pave	NaN	IR1	…	0	0	NaN	MnPrv	NaN	0	3	2010	WD	Normal
3	1464	60	RL	78.0	9978	Pave	NaN	IR1	…	0	0	NaN	NaN	NaN	0	6	2010	WD	Normal
4	1465	120	RL	43.0	5005	Pave	NaN	IR1	…	144	0	NaN	NaN	NaN	0	1	2010	WD	Normal

[5 rows x 80 columns]

[9]:
```
data_test.shape
```

[9]:
```
(1459, 80)
```

3 Data Preparation

3.1 Missing Values Analysis

[11]:
```
np.sum(data_train.isnull())
np.sum(data_test.isnull())
```

```
C:\ProgramData\anaconda3\Lib\site-packages\numpy\core\fromnumeric.py:86:
FutureWarning: The behavior of DataFrame.sum with axis=None is deprecated, in a
future version this will reduce over both axes and return a scalar. To retain
the old behavior, pass axis=0 (or do not pass axis)
  return reduction(axis=axis, out=out, **passkwargs)
```

[11]:
```
Id               0
MSSubClass       0
MSZoning         4
LotFrontage    227
LotArea          0
```

```
                ...
MiscVal            0
MoSold             0
YrSold             0
SaleType           1
SaleCondition      0
Length: 80, dtype: int64
```

[12]:
```
msv_summary_train = data_train.isnull().sum().sort_values(ascending = False)
print('msv_summary_train :\n', msv_summary_train[msv_summary_train > 0])

msv_summary_test = data_test.isnull().sum().sort_values(ascending = False)
print('msv_summary_test :\n', msv_summary_test[msv_summary_test > 0])
```

```
msv_summary_train :
 PoolQC          1453
MiscFeature     1406
Alley           1369
Fence           1179
MasVnrType       872
FireplaceQu      690
LotFrontage      259
GarageYrBlt       81
GarageCond        81
GarageType        81
GarageFinish      81
GarageQual        81
BsmtFinType2      38
BsmtExposure      38
BsmtQual          37
BsmtCond          37
BsmtFinType1      37
MasVnrArea         8
Electrical         1
dtype: int64
msv_summary_test :
 PoolQC          1456
MiscFeature     1408
Alley           1352
Fence           1169
MasVnrType       894
FireplaceQu      730
LotFrontage      227
GarageYrBlt       78
GarageQual        78
GarageFinish      78
GarageCond        78
GarageType        76
```

```
BsmtCond          45
BsmtQual          44
BsmtExposure      44
BsmtFinType1      42
BsmtFinType2      42
MasVnrArea        15
MSZoning           4
BsmtHalfBath       2
Utilities          2
Functional         2
BsmtFullBath       2
BsmtFinSF1         1
BsmtFinSF2         1
BsmtUnfSF          1
KitchenQual        1
TotalBsmtSF        1
Exterior2nd        1
GarageCars         1
Exterior1st        1
GarageArea         1
SaleType           1
dtype: int64
```

[13]:
```
msv_summary_train = pd.DataFrame({'freq': np.sum(data_train.isnull())})
msv_summary_train ['pct'] = round(msv_summary_train['freq']/data_train.shape[0]␣
 ↪* 100, 1)
msv_summary_train.sort_values(by = 'pct', ascending = False)

print(msv_summary_train.sort_values(by = 'pct', ascending =␣
 ↪False)[msv_summary_train.sort_values(by = 'pct', ascending = False) > 50])
```

```
               freq   pct
PoolQC       1453.0  99.5
MiscFeature  1406.0  96.3
Alley        1369.0  93.8
Fence        1179.0  80.8
MasVnrType    872.0  59.7
…               …     …
ExterQual       NaN   NaN
Exterior2nd     NaN   NaN
Exterior1st     NaN   NaN
RoofMatl        NaN   NaN
SalePrice       NaN   NaN

[81 rows x 2 columns]

C:\ProgramData\anaconda3\Lib\site-packages\numpy\core\fromnumeric.py:86:
FutureWarning: The behavior of DataFrame.sum with axis=None is deprecated, in a
```

```
future version this will reduce over both axes and return a scalar. To retain
the old behavior, pass axis=0 (or do not pass axis)
  return reduction(axis=axis, out=out, **passkwargs)
```

[14]:
```
msv_summary_test = pd.DataFrame({'freq': np.sum(data_test.isnull())})
msv_summary_test ['pct'] = round(msv_summary_test['freq']/data_test.shape[0] *
  ↪100, 1)
msv_summary_test.sort_values(by = 'pct', ascending = False)

print(msv_summary_test.sort_values(by = 'pct', ascending =
  ↪False)[msv_summary_test.sort_values(by = 'pct', ascending = False) > 50])
```

```
                freq   pct
PoolQC         1456.0  99.8
MiscFeature    1408.0  96.5
Alley          1352.0  92.7
Fence          1169.0  80.1
MasVnrType      894.0  61.3
…                …   …
Electrical        NaN   NaN
1stFlrSF          NaN   NaN
2ndFlrSF          NaN   NaN
LowQualFinSF      NaN   NaN
SaleCondition     NaN   NaN

[80 rows x 2 columns]
C:\ProgramData\anaconda3\Lib\site-packages\numpy\core\fromnumeric.py:86:
FutureWarning: The behavior of DataFrame.sum with axis=None is deprecated, in a
future version this will reduce over both axes and return a scalar. To retain
the old behavior, pass axis=0 (or do not pass axis)
  return reduction(axis=axis, out=out, **passkwargs)
```

[15]:
```
print(data_train.duplicated().sum())
print(data_train.loc[:, data_train.nunique() == 1].columns)

print(data_test.duplicated().sum())
print(data_test.loc[:, data_train.nunique() == 1].columns)
```

```
0
Index([], dtype='object')
0
Index([], dtype='object')
```

[16]:
```
# List of columns to drop
columns_to_drop = ['PoolQC', 'MiscFeature', 'Alley', 'Fence', 'MasVnrType',
  ↪'FireplaceQu']

# Drop the columns
```

```
data_train.drop(columns = columns_to_drop, inplace = True, errors = 'ignore')
data_test.drop(columns = columns_to_drop, inplace = True, errors = 'ignore')

# Check the shape after dropping columns
print("Shape after dropping columns:", data_train.shape)
print("Shape after dropping columns:", data_test.shape)
# Check missing value summary
msv_summary_train = data_train.isnull().sum() / len(data_train) * 100 #or␣
 ↪data_train.shape[0]
msv_summary_train = pd.DataFrame(msv_summary_train, columns=['pct'])
msv_summary_test = data_test.isnull().sum() / len(data_test) * 100 #or␣
 ↪data_train.shape[0]
msv_summary_test = pd.DataFrame(msv_summary_test, columns=['pct'])
print('msv_summary_train_new : \n', msv_summary_train.sort_values(by='pct',␣
 ↪ascending=False))
print('msv_summary_test_new : \n', msv_summary_test.sort_values(by='pct',␣
 ↪ascending=False))
```

```
Shape after dropping columns: (1460, 75)
Shape after dropping columns: (1459, 74)
msv_summary_train_new :
                    pct
LotFrontage   17.739726
GarageYrBlt    5.547945
GarageCond     5.547945
GarageType     5.547945
GarageFinish   5.547945
…                 …
BsmtUnfSF      0.000000
TotalBsmtSF    0.000000
MSSubClass     0.000000
HeatingQC      0.000000
SalePrice      0.000000

[75 rows x 1 columns]
msv_summary_test_new :
                     pct
LotFrontage    15.558602
GarageYrBlt     5.346127
GarageFinish    5.346127
GarageQual      5.346127
GarageCond      5.346127
…                  …
CentralAir      0.000000
Electrical      0.000000
1stFlrSF        0.000000
2ndFlrSF        0.000000
```

```
SaleCondition   0.000000

[74 rows x 1 columns]
```

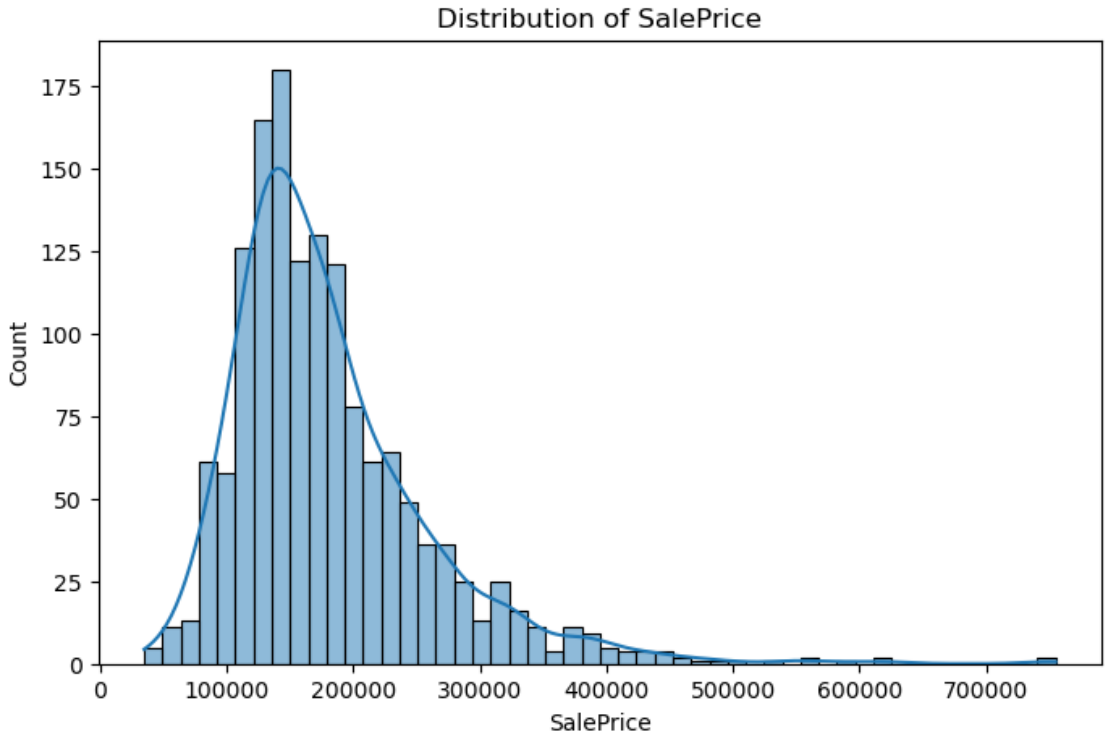
4 Variables Analysis

[18]:
```
data_train.set_index('Id', inplace=True)
data_test.set_index('Id', inplace=True)
```

[19]:
```
print(data_train.describe())

plt.figure(figsize=(8, 5))
sns.histplot(data_train['SalePrice'], kde=True, bins=50)
plt.title("Distribution of SalePrice")
plt.show()
```

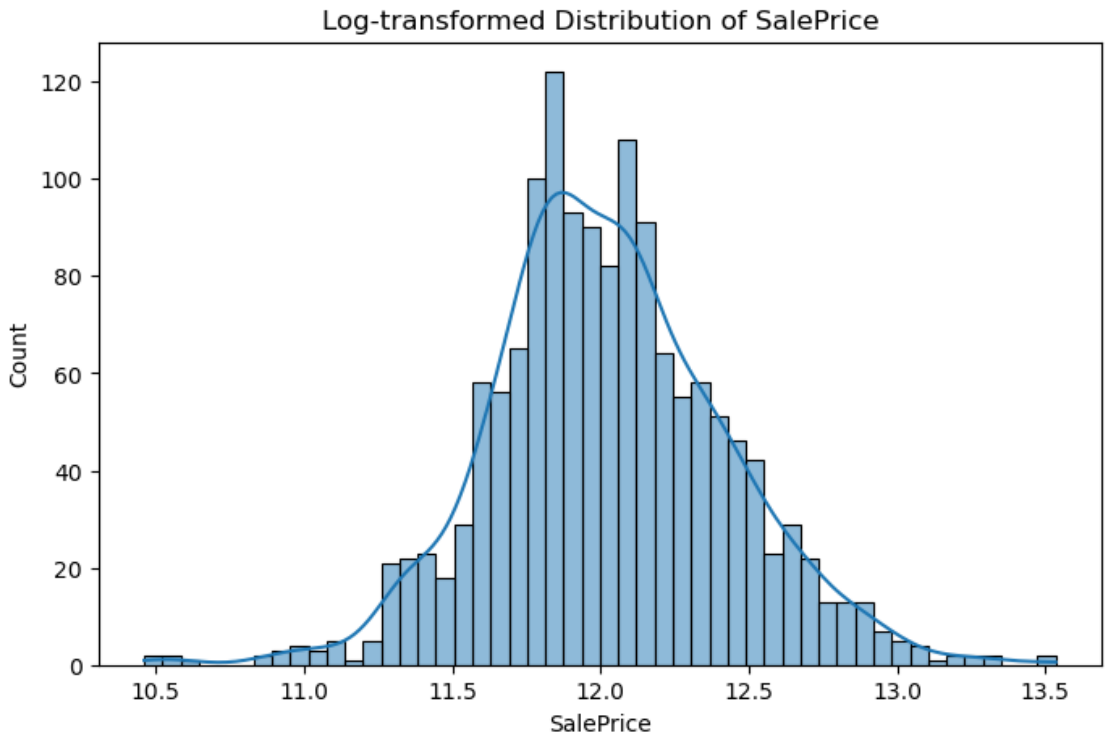
```
        MSSubClass  LotFrontage        LotArea  OverallQual  OverallCond  \
count  1460.000000  1201.000000    1460.000000  1460.000000  1460.000000
mean     56.897260    70.049958   10516.828082     6.099315     5.575342
std      42.300571    24.284752    9981.264932     1.382997     1.112799
min      20.000000    21.000000    1300.000000     1.000000     1.000000
25%      20.000000    59.000000    7553.500000     5.000000     5.000000
50%      50.000000    69.000000    9478.500000     6.000000     5.000000
75%      70.000000    80.000000   11601.500000     7.000000     6.000000
max     190.000000   313.000000  215245.000000    10.000000     9.000000

         YearBuilt  YearRemodAdd   MasVnrArea   BsmtFinSF1   BsmtFinSF2  …  \
count  1460.000000   1460.000000  1452.000000  1460.000000  1460.000000  …
mean   1971.267808   1984.865753   103.685262   443.639726    46.549315  …
std      30.202904     20.645407   181.066207   456.098091   161.319273  …
min    1872.000000   1950.000000     0.000000     0.000000     0.000000  …
25%    1954.000000   1967.000000     0.000000     0.000000     0.000000  …
50%    1973.000000   1994.000000     0.000000   383.500000     0.000000  …
75%    2000.000000   2004.000000   166.000000   712.250000     0.000000  …
max    2010.000000   2010.000000  1600.000000  5644.000000  1474.000000  …

        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
25%       0.000000     0.000000       0.000000     0.000000     0.000000
50%       0.000000    25.000000       0.000000     0.000000     0.000000
75%     168.000000    68.000000       0.000000     0.000000     0.000000
max     857.000000   547.000000     552.000000   508.000000   480.000000

          PoolArea       MiscVal       MoSold       YrSold      SalePrice
```

```
count  1460.000000   1460.000000  1460.000000  1460.000000    1460.000000
mean      2.758904     43.489041     6.321918  2007.815753  180921.195890
std      40.177307    496.123024     2.703626     1.328095   79442.502883
min       0.000000      0.000000     1.000000  2006.000000   34900.000000
25%       0.000000      0.000000     5.000000  2007.000000  129975.000000
50%       0.000000      0.000000     6.000000  2008.000000  163000.000000
75%       0.000000      0.000000     8.000000  2009.000000  214000.000000
max     738.000000  15500.000000    12.000000  2010.000000  755000.000000

[8 rows x 37 columns]
```



[20]:
```python
print(data_train['SalePrice'].skew())

if data_train['SalePrice'].skew() > 1:
    data_train['SalePrice'] = np.log1p(data_train['SalePrice'])

plt.figure(figsize=(8, 5))
sns.histplot(data_train['SalePrice'], kde=True, bins=50)
plt.title("Log-transformed Distribution of SalePrice")
plt.show()
```

```
1.8828757597682129
```



[21]:
```
numeric_features_train = data_train.select_dtypes(include=['int64', 'float64'])

msv_numeric_features_train = [col for col in numeric_features_train.columns if␣
 ↪data_train[col].isnull().sum() > 0]

print(msv_numeric_features_train)

print("------------------------------------------------------------------------")
numeric_features_test = data_test.select_dtypes(include=['int64', 'float64'])

msv_numeric_features_test = [col for col in numeric_features_test.columns if␣
 ↪data_test[col].isnull().sum() > 0]

print(msv_numeric_features_test)
```

```
['LotFrontage', 'MasVnrArea', 'GarageYrBlt']
------------------------------------------------------------------------
['LotFrontage', 'MasVnrArea', 'BsmtFinSF1', 'BsmtFinSF2', 'BsmtUnfSF',
'TotalBsmtSF', 'BsmtFullBath', 'BsmtHalfBath', 'GarageYrBlt', 'GarageCars',
'GarageArea']
```

[22]:
```
# Identify all columns that end with '_missing_flag'
```

```
missing_flag_columns = [col for col in data_train.columns if col.
 ↪endswith('_missing_flag')]

# Drop those columns from the dataframe
data_train.drop(missing_flag_columns, axis=1, inplace=True)

data_train
```

[22]:

Id	MSSubClass	MSZoning	LotFrontage	LotArea	Street	LotShape	LandContour	…
1	60	RL	65.0	8450	Pave	Reg	Lvl	
2	20	RL	80.0	9600	Pave	Reg	Lvl	
3	60	RL	68.0	11250	Pave	IR1	Lvl	
4	70	RL	60.0	9550	Pave	IR1	Lvl	
5	60	RL	84.0	14260	Pave	IR1	Lvl	
…	…	…	…	…	…	…	…	
1456	60	RL	62.0	7917	Pave	Reg	Lvl	
1457	20	RL	85.0	13175	Pave	Reg	Lvl	
1458	70	RL	66.0	9042	Pave	Reg	Lvl	
1459	20	RL	68.0	9717	Pave	Reg	Lvl	
1460	20	RL	75.0	9937	Pave	Reg	Lvl	

Id	Utilities	LotConfig	LandSlope	…	EnclosedPorch	3SsnPorch	ScreenPorch	…
				…				
1	AllPub	Inside	Gtl	…	0	0	0	
2	AllPub	FR2	Gtl	…	0	0	0	
3	AllPub	Inside	Gtl	…	0	0	0	
4	AllPub	Corner	Gtl	…	272	0	0	
5	AllPub	FR2	Gtl	…	0	0	0	
…	…	…	…	…	…	…	…	
1456	AllPub	Inside	Gtl	…	0	0	0	
1457	AllPub	Inside	Gtl	…	0	0	0	
1458	AllPub	Inside	Gtl	…	0	0	0	
1459	AllPub	Inside	Gtl	…	112	0	0	
1460	AllPub	Inside	Gtl	…	0	0	0	

Id	PoolArea	MiscVal	MoSold	YrSold	SaleType	SaleCondition	SalePrice
1	0	0	2	2008	WD	Normal	12.247699
2	0	0	5	2007	WD	Normal	12.109016
3	0	0	9	2008	WD	Normal	12.317171
4	0	0	2	2006	WD	Abnorml	11.849405
5	0	0	12	2008	WD	Normal	12.429220
…	…	…	…	…	…	…	…
1456	0	0	8	2007	WD	Normal	12.072547
1457	0	0	2	2010	WD	Normal	12.254868
1458	0	2500	5	2010	WD	Normal	12.493133

```
1459        0       0       4    2010        WD         Normal  11.864469
1460        0       0       6    2008        WD         Normal  11.901590

[1460 rows x 74 columns]
```

[23]:
```
# Identify all columns that end with '_missing_flag'
missing_flag_columns = [col for col in data_test.columns if col.
 ↪endswith('_missing_flag')]

# Drop those columns from the dataframe
data_test.drop(missing_flag_columns, axis=1, inplace=True)

data_test
```

[23]:
```
      MSSubClass MSZoning  LotFrontage  LotArea Street LotShape LandContour  \
Id
1461          20       RH         80.0    11622   Pave      Reg         Lvl
1462          20       RL         81.0    14267   Pave      IR1         Lvl
1463          60       RL         74.0    13830   Pave      IR1         Lvl
1464          60       RL         78.0     9978   Pave      IR1         Lvl
1465         120       RL         43.0     5005   Pave      IR1         HLS
…            …        …           …        …      …         …           …
2915         160       RM         21.0     1936   Pave      Reg         Lvl
2916         160       RM         21.0     1894   Pave      Reg         Lvl
2917          20       RL        160.0    20000   Pave      Reg         Lvl
2918          85       RL         62.0    10441   Pave      Reg         Lvl
2919          60       RL         74.0     9627   Pave      Reg         Lvl

     Utilities LotConfig LandSlope  … OpenPorchSF EnclosedPorch 3SsnPorch  \
Id                                  …
1461    AllPub    Inside       Gtl  …           0             0         0
1462    AllPub    Corner       Gtl  …          36             0         0
1463    AllPub    Inside       Gtl  …          34             0         0
1464    AllPub    Inside       Gtl  …          36             0         0
1465    AllPub    Inside       Gtl  …          82             0         0
…          …         …         …  …           …             …         …
2915    AllPub    Inside       Gtl  …           0             0         0
2916    AllPub    Inside       Gtl  …          24             0         0
2917    AllPub    Inside       Gtl  …           0             0         0
2918    AllPub    Inside       Gtl  …          32             0         0
2919    AllPub    Inside       Mod  …          48             0         0

     ScreenPorch PoolArea  MiscVal  MoSold  YrSold  SaleType SaleCondition
Id
1461         120        0        0       6    2010        WD        Normal
1462           0        0    12500       6    2010        WD        Normal
1463           0        0        0       3    2010        WD        Normal
```

```
1464          0        0        0       6    2010        WD         Normal
1465        144        0        0       1    2010        WD         Normal
…           …      …        …      …     …       …           …
2915          0        0        0       6    2006        WD         Normal
2916          0        0        0       4    2006        WD        Abnorml
2917          0        0        0       9    2006        WD        Abnorml
2918          0        0      700       7    2006        WD         Normal
2919          0        0        0      11    2006        WD         Normal

[1459 rows x 73 columns]
```

[24]:
```
# Function to handle missing data in the training dataset
def handle_missing_data_train(data_train, numeric_features, missing_threshold=0.
↪1):
    for col in numeric_features:
        if data_train[col].isnull().sum() > 0:  # Check for missing values
            # Calculate skewness
            col_skewness = skew(data_train[col].dropna())  # Skewness of the␣
↪column (drop NaN values)

            # If skewness is low, fill missing values with the mean
            if abs(col_skewness) < 0.5:
                print(f"{col} has a symmetric distribution (skewness:␣
↪{col_skewness:.2f}). Filling with mean.")
                data_train[col].fillna(data_train[col].mean(), inplace=True)
            else:
                print(f"{col} has a skewed distribution (skewness:␣
↪{col_skewness:.2f}). Using regression model.")
                # Get feature columns (exclude the current column)
                feature_cols = [f for f in numeric_features if f != col]
                train_data = data_train[data_train[col].notnull()]
                X_train = train_data[feature_cols]
                y_train = train_data[col]

                # Train the model and predict missing values
                model = RandomForestRegressor(n_estimators=100, random_state=42)
                model.fit(X_train, y_train)
                missing_data = data_train[data_train[col].isnull()]
                X_missing = missing_data[feature_cols]
                predicted_values = model.predict(X_missing)
                data_train.loc[data_train[col].isnull(), col] = predicted_values
                print(f"Missing values in {col} filled using␣
↪RandomForestRegressor.")
    return data_train

# Function to handle missing data in the test dataset
```

```
def handle_missing_data_test(data_test, numeric_features, model,␣
 ↪missing_threshold=0.1):
    for col in numeric_features:
        if data_test[col].isnull().sum() > 0:  # Check for missing values
            # Calculate skewness
            col_skewness = skew(data_test[col].dropna())  # Skewness of the␣
 ↪column (drop NaN values)

            # If skewness is low, fill missing values with the mean
            if abs(col_skewness) < 0.5:
                print(f"{col} has a symmetric distribution (skewness:␣
 ↪{col_skewness:.2f}). Filling with mean.")
                data_test[col].fillna(data_test[col].mean(), inplace=True)
            else:
                print(f"{col} has a skewed distribution (skewness:␣
 ↪{col_skewness:.2f}). Using regression model.")
                # Get feature columns (exclude the current column)
                feature_cols = [f for f in numeric_features if f != col]
                missing_data = data_test[data_test[col].isnull()]
                X_missing = missing_data[feature_cols]
                predicted_values = model.predict(X_missing)
                data_test.loc[data_test[col].isnull(), col] = predicted_values
                print(f"Missing values in {col} filled using␣
 ↪RandomForestRegressor.")
    return data_test
```

[25]:
```
data_test.select_dtypes(include=['int64', 'float64']).isnull().sum().
 ↪sort_values(ascending = False)
```

[25]:
```
LotFrontage      227
GarageYrBlt       78
MasVnrArea        15
BsmtHalfBath       2
BsmtFullBath       2
TotalBsmtSF        1
GarageCars         1
BsmtFinSF1         1
BsmtFinSF2         1
BsmtUnfSF          1
GarageArea         1
OpenPorchSF        0
WoodDeckSF         0
MSSubClass         0
EnclosedPorch      0
TotRmsAbvGrd       0
3SsnPorch          0
ScreenPorch        0
```

```
PoolArea          0
MiscVal           0
MoSold            0
Fireplaces        0
FullBath          0
KitchenAbvGr      0
BedroomAbvGr      0
HalfBath          0
GrLivArea         0
LowQualFinSF      0
2ndFlrSF          0
1stFlrSF          0
YearRemodAdd      0
YearBuilt         0
OverallCond       0
OverallQual       0
LotArea           0
YrSold            0
dtype: int64
```

[26]:
```
columns_to_fill_test = [ 'LotFrontage',  'GarageYrBlt',  'MasVnrArea',␣
↪'BsmtHalfBath', 'BsmtFullBath',
                        'BsmtFinSF2', 'GarageCars', 'GarageArea',␣
↪'TotalBsmtSF', 'BsmtUnfSF', 'BsmtFinSF1' ]

for column in columns_to_fill_test:
    median_value = data_test[column].median()
    data_test[column].fillna(median_value, inplace=True)
```

```
C:\Users\izadi\AppData\Local\Temp\ipykernel_26524\1365818443.py:6:
FutureWarning: A value is trying to be set on a copy of a DataFrame or Series
through chained assignment using an inplace method.
The behavior will change in pandas 3.0. This inplace method will never work
because the intermediate object on which we are setting values always behaves as
a copy.

For example, when doing 'df[col].method(value, inplace=True)', try using
'df.method({col: value}, inplace=True)' or df[col] = df[col].method(value)
instead, to perform the operation inplace on the original object.


  data_test[column].fillna(median_value, inplace=True)
```

[27]:
```
data_test.select_dtypes(include=['int64', 'float64']).isnull().sum().
↪sort_values(ascending = False)
```

[27]:
```
MSSubClass       0
LotFrontage      0
BedroomAbvGr     0
KitchenAbvGr     0
TotRmsAbvGrd     0
Fireplaces       0
GarageYrBlt      0
GarageCars       0
GarageArea       0
WoodDeckSF       0
OpenPorchSF      0
EnclosedPorch    0
3SsnPorch        0
ScreenPorch      0
PoolArea         0
MiscVal          0
MoSold           0
HalfBath         0
FullBath         0
BsmtHalfBath     0
BsmtFinSF1       0
LotArea          0
OverallQual      0
OverallCond      0
YearBuilt        0
YearRemodAdd     0
MasVnrArea       0
BsmtFinSF2       0
BsmtFullBath     0
BsmtUnfSF        0
TotalBsmtSF      0
1stFlrSF         0
2ndFlrSF         0
LowQualFinSF     0
GrLivArea        0
YrSold           0
dtype: int64
```

[28]:
```
data_train.select_dtypes(include=['int64', 'float64']).isnull().sum().
↪sort_values(ascending = False)
```

[28]:
```
LotFrontage      259
GarageYrBlt       81
MasVnrArea         8
OpenPorchSF        0
KitchenAbvGr       0
TotRmsAbvGrd       0
```

```
Fireplaces         0
GarageCars         0
GarageArea         0
WoodDeckSF         0
MSSubClass         0
BedroomAbvGr       0
3SsnPorch          0
ScreenPorch        0
PoolArea           0
MiscVal            0
MoSold             0
YrSold             0
EnclosedPorch      0
FullBath           0
HalfBath           0
BsmtFinSF2         0
LotArea            0
OverallQual        0
OverallCond        0
YearBuilt          0
YearRemodAdd       0
BsmtFinSF1         0
BsmtUnfSF          0
BsmtHalfBath       0
TotalBsmtSF        0
1stFlrSF           0
2ndFlrSF           0
LowQualFinSF       0
GrLivArea          0
BsmtFullBath       0
SalePrice          0
dtype: int64
```

[29]:
```
columns_to_fill_train = [ 'LotFrontage',  'GarageYrBlt',  'MasVnrArea' ]

for column in columns_to_fill_train:
    median_value = data_train[column].median()
    data_train[column].fillna(median_value, inplace=True)
```

```
C:\Users\izadi\AppData\Local\Temp\ipykernel_26524\3752660924.py:5:
FutureWarning: A value is trying to be set on a copy of a DataFrame or Series
through chained assignment using an inplace method.
The behavior will change in pandas 3.0. This inplace method will never work
because the intermediate object on which we are setting values always behaves as
a copy.

For example, when doing 'df[col].method(value, inplace=True)', try using
'df.method({col: value}, inplace=True)' or df[col] = df[col].method(value)
```

```
instead, to perform the operation inplace on the original object.


  data_train[column].fillna(median_value, inplace=True)
```

[30]:
```
np.sum(data_train.select_dtypes(include=['int64', 'float64']).isnull()).
 ↪sort_values(ascending = False)
```

```
C:\ProgramData\anaconda3\Lib\site-packages\numpy\core\fromnumeric.py:86:
FutureWarning: The behavior of DataFrame.sum with axis=None is deprecated, in a
future version this will reduce over both axes and return a scalar. To retain
the old behavior, pass axis=0 (or do not pass axis)
  return reduction(axis=axis, out=out, **passkwargs)
```

[30]:
```
MSSubClass       0
HalfBath         0
KitchenAbvGr     0
TotRmsAbvGrd     0
Fireplaces       0
GarageYrBlt      0
GarageCars       0
GarageArea       0
WoodDeckSF       0
OpenPorchSF      0
EnclosedPorch    0
3SsnPorch        0
ScreenPorch      0
PoolArea         0
MiscVal          0
MoSold           0
YrSold           0
BedroomAbvGr     0
FullBath         0
LotFrontage      0
BsmtHalfBath     0
LotArea          0
OverallQual      0
OverallCond      0
YearBuilt        0
YearRemodAdd     0
MasVnrArea       0
BsmtFinSF1       0
BsmtFinSF2       0
BsmtUnfSF        0
TotalBsmtSF      0
1stFlrSF         0
2ndFlrSF         0
LowQualFinSF     0
```

```
GrLivArea        0
BsmtFullBath     0
SalePrice        0
dtype: int64
```

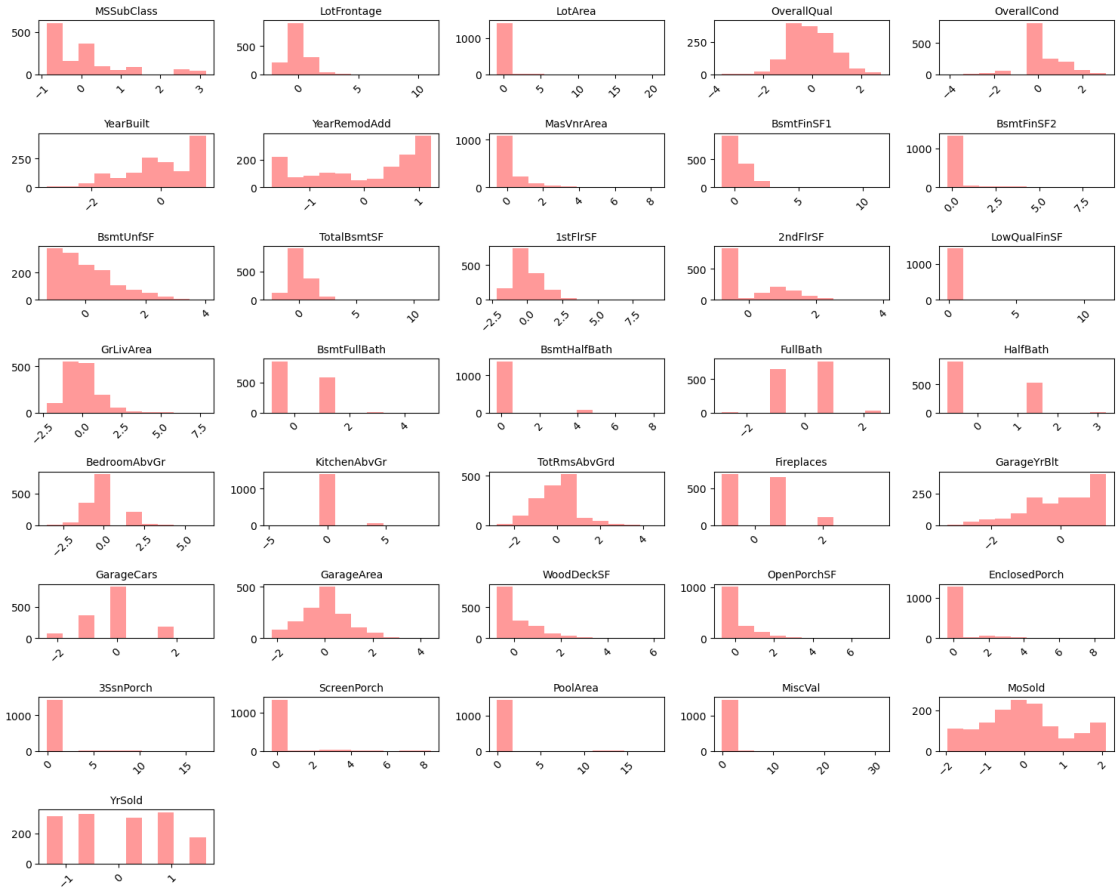
4.1 Feature scaling

[32]:
```
# Define the target column in the training data
target_column = 'SalePrice'

# Select numerical features, excluding the target column in the training data
numeric_features_train = data_train.drop(columns=[target_column]).
 ↪select_dtypes(include=['int64', 'float64']).columns
numeric_features_test = data_test.select_dtypes(include=['int64', 'float64']).
 ↪columns

# Initialize the StandardScaler
scaler = StandardScaler()

# Scale the numerical columns in the training data (fit and transform)
data_train[numeric_features_train] = scaler.
 ↪fit_transform(data_train[numeric_features_train])

# Scale the numerical columns in the test data (transform only, using the same␣
 ↪scaler)
data_test[numeric_features_test] = scaler.
 ↪transform(data_test[numeric_features_test])
```

4.2 Visualization

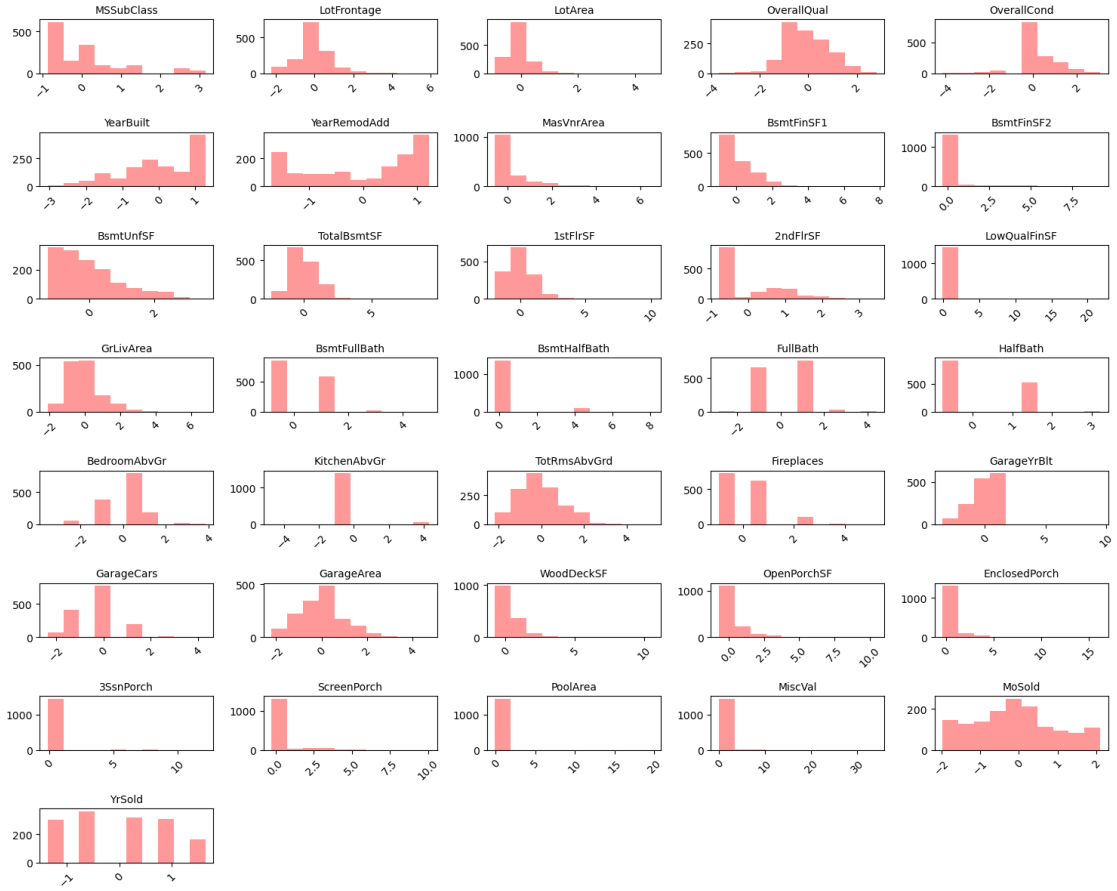
[34]:
```
# Ensure numeric_features_train is a DataFrame and extract column names properly
if isinstance(numeric_features_train, pd.DataFrame):
    numeric_features_train_list = numeric_features_train.columns.tolist()
else:
    numeric_features_train_list = numeric_features_train  # assuming it's␣
 ↪already a list of column names

# Get the indices of numeric columns in data_train using column names
numeric_indices_train = [data_train.columns.get_loc(col) for col in␣
 ↪numeric_features_train_list if col in data_train.columns]

# Plot histograms for numeric variables
plt.figure(figsize=(15, 12))  # Adjust the figure size for better visibility
plt.subplots_adjust(hspace=0.5, wspace=0.5)

# Loop through each index in numeric_indices
for i in range(len(numeric_indices_train)):
```

```
    plt.subplot(8, 5, i + 1)  # Correct subplot indexing (1-based)
    plt.hist(x=data_train.iloc[:, numeric_indices_train[i]], alpha=0.4,␣
 ↪color='red')
    plt.title(data_train.columns[numeric_indices_train[i]], fontsize=10)
    plt.xticks(rotation=45)

plt.tight_layout()
plt.show()
```



[35]:
```
# Ensure numeric_features_test is a DataFrame and extract column names properly
if isinstance(numeric_features_test, pd.DataFrame):
    numeric_features_test_list = numeric_features_test.columns.tolist()
else:
    numeric_features_test_list = numeric_features_test  # assuming it's already␣
 ↪a list of column names

# Get the indices of numeric columns in data_test using column names
numeric_indices_test = [data_test.columns.get_loc(col) for col in␣
 ↪numeric_features_test_list if col in data_test.columns]
```

```
# Plot histograms for numeric variables
plt.figure(figsize=(15, 12))  # Adjust the figure size for better visibility
plt.subplots_adjust(hspace=0.5, wspace=0.5)  # Adjust space between plots

# Loop through each index in numeric_indices_test
for i in range(len(numeric_indices_test)):
    plt.subplot(8, 5, i + 1)  # Correct subplot indexing (1-based)
    plt.hist(x=data_test.iloc[:, numeric_indices_test[i]], alpha=0.4,␣
 ↪color='red')
    plt.title(data_test.columns[numeric_indices_test[i]], fontsize=10)
    plt.xticks(rotation=45)

plt.tight_layout()  # Adjust subplots to fit in the figure area
plt.show()
```



Handle categorical data

[37]:
```
data_train.select_dtypes(include=['object', 'category']).isnull().sum().
 ↪sort_values(ascending = False)
```

[37]:
```
GarageCond       81
GarageQual       81
GarageFinish     81
GarageType       81
BsmtExposure     38
BsmtFinType2     38
BsmtCond         37
BsmtFinType1     37
BsmtQual         37
Electrical        1
HeatingQC         0
Heating           0
MSZoning          0
CentralAir        0
Functional        0
PavedDrive        0
SaleType          0
KitchenQual       0
Foundation        0
Street            0
ExterCond         0
LotShape          0
LandContour       0
Utilities         0
LotConfig         0
LandSlope         0
Neighborhood      0
Condition1        0
Condition2        0
BldgType          0
HouseStyle        0
RoofStyle         0
RoofMatl          0
Exterior1st       0
Exterior2nd       0
ExterQual         0
SaleCondition     0
dtype: int64
```

[38]:
```
#Handle categorical data (Categorical columns in Data train)
categoric_features = data_train.select_dtypes(include=['object', 'category']).
 ↪columns
for col in categoric_features:
    if data_train[col].isnull().sum() > 0:  # If there are missing values
```

```
        mode_value = data_train[col].mode()[0]  # Find the mode (most frequent␣
↪value)
        data_train[col].fillna(mode_value, inplace=True)  # Fill missing values␣
↪with the mode
```

```
C:\Users\izadi\AppData\Local\Temp\ipykernel_26524\4249127474.py:6:
FutureWarning: A value is trying to be set on a copy of a DataFrame or Series
through chained assignment using an inplace method.
The behavior will change in pandas 3.0. This inplace method will never work
because the intermediate object on which we are setting values always behaves as
a copy.

For example, when doing 'df[col].method(value, inplace=True)', try using
'df.method({col: value}, inplace=True)' or df[col] = df[col].method(value)
instead, to perform the operation inplace on the original object.


  data_train[col].fillna(mode_value, inplace=True)  # Fill missing values with
the mode
```

[39]:
```
data_train.select_dtypes(include=['object', 'category']).isnull().sum().
↪sort_values(ascending = False)
```

[39]:
```
MSZoning         0
BsmtQual         0
BsmtExposure     0
BsmtFinType1     0
BsmtFinType2     0
Heating          0
HeatingQC        0
CentralAir       0
Electrical       0
KitchenQual      0
Functional       0
GarageType       0
GarageFinish     0
GarageQual       0
GarageCond       0
PavedDrive       0
SaleType         0
BsmtCond         0
Foundation       0
Street           0
ExterCond        0
LotShape         0
LandContour      0
Utilities        0
```

```
LotConfig        0
LandSlope        0
Neighborhood     0
Condition1       0
Condition2       0
BldgType         0
HouseStyle       0
RoofStyle        0
RoofMatl         0
Exterior1st      0
Exterior2nd      0
ExterQual        0
SaleCondition    0
dtype: int64
```

[40]:
```
#Handle categorical data (Categorical columns in Data test)
categoric_features = data_test.select_dtypes(include=['object', 'category']).
 ↪columns
for col in categoric_features:
    if data_test[col].isnull().sum() > 0:  # If there are missing values
        mode_value = data_test[col].mode()[0]  # Find the mode (most frequent␣
 ↪value)
        data_test[col].fillna(mode_value, inplace=True)  # Fill missing values␣
 ↪with the mode
```

```
C:\Users\izadi\AppData\Local\Temp\ipykernel_26524\2336348955.py:6:
FutureWarning: A value is trying to be set on a copy of a DataFrame or Series
through chained assignment using an inplace method.
The behavior will change in pandas 3.0. This inplace method will never work
because the intermediate object on which we are setting values always behaves as
a copy.

For example, when doing 'df[col].method(value, inplace=True)', try using
'df.method({col: value}, inplace=True)' or df[col] = df[col].method(value)
instead, to perform the operation inplace on the original object.


  data_test[col].fillna(mode_value, inplace=True)  # Fill missing values with
the mode
```

[41]:
```
data_test.select_dtypes(include=['object', 'category']).isnull().sum().
 ↪sort_values(ascending = False)
```

[41]:
```
MSZoning         0
BsmtQual         0
BsmtExposure     0
BsmtFinType1     0
```

```
BsmtFinType2     0
Heating          0
HeatingQC        0
CentralAir       0
Electrical       0
KitchenQual      0
Functional       0
GarageType       0
GarageFinish     0
GarageQual       0
GarageCond       0
PavedDrive       0
SaleType         0
BsmtCond         0
Foundation       0
Street           0
ExterCond        0
LotShape         0
LandContour      0
Utilities        0
LotConfig        0
LandSlope        0
Neighborhood     0
Condition1       0
Condition2       0
BldgType         0
HouseStyle       0
RoofStyle        0
RoofMatl         0
Exterior1st      0
Exterior2nd      0
ExterQual        0
SaleCondition    0
dtype: int64
```

5 Correlation Analysis

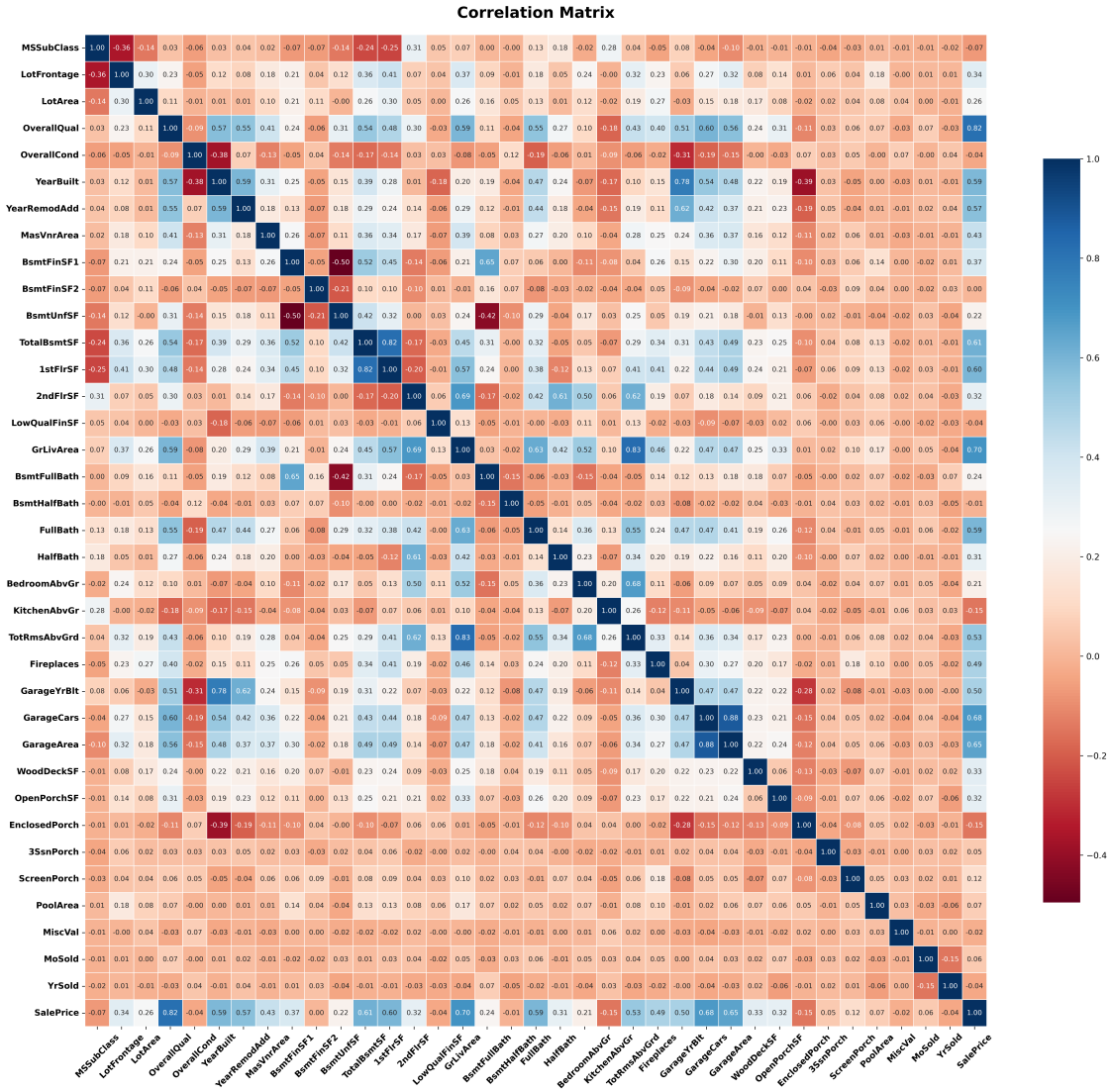
[43]:
```python
def high_correlated_cols(data_train, plot=False, corr_th=0.90):
    # Compute correlation matrix
    corr = data_train.corr(numeric_only=True)

    # Create an absolute correlation matrix
    cor_matrix = corr.abs()

    # Extract upper triangle matrix to identify high correlations
```

```
    upper_triangle_matrix = cor_matrix.where(np.triu(np.ones(cor_matrix.shape),␣
↪k=1).astype(bool))
    drop_list = [col for col in upper_triangle_matrix.columns if␣
↪any(upper_triangle_matrix[col] > corr_th)]

    if plot:
        # Enhance figure size and quality
        plt.figure(figsize=(20, 18), dpi=300)  # Increased size and resolution

        # Improved heatmap with adjusted annotations and line widths
        sns.heatmap(
            corr,
            annot=True,                # Show values
            fmt=".2f",                 # Format for annotation
            cmap="RdBu",               # Diverging color map
            annot_kws={"size": 8},     # Font size for annotations
            linewidths=0.5,            # Line width between cells
            linecolor="white",         # Line color between cells
            cbar_kws={"shrink": 0.75}  # Shrink color bar for better fit
        )

        # Add title and adjust layout
        plt.title("Correlation Matrix", fontsize=18, weight="bold", pad=20)
        plt.xticks(fontsize=10, rotation=45, weight="bold")
        plt.yticks(fontsize=10, weight="bold")
        plt.tight_layout()  # Adjust layout to prevent clipping
        plt.show()

    return drop_list
```

[44]:
```
# Ensure plots are displayed directly in the Jupyter Notebook
%matplotlib inline

# Configure high-resolution output for Retina displays
%config InlineBackend.figure_format = 'retina'

# Set default configurations for matplotlib plots
plt.rcParams['figure.figsize'] = [15, 10]  # Set default figure size (width x␣
↪height in inches)
plt.rcParams['figure.dpi'] = 120           # Set default resolution (dots per␣
↪inch)
plt.rcParams['axes.titlesize'] = 14        # Set default font size for titles
plt.rcParams['axes.labelsize'] = 12        # Set default font size for axis␣
↪labels
plt.rcParams['xtick.labelsize'] = 10       # Set default font size for x-axis␣
↪ticks
```

```
plt.rcParams['ytick.labelsize'] = 10       # Set default font size for y-axis ticks
```

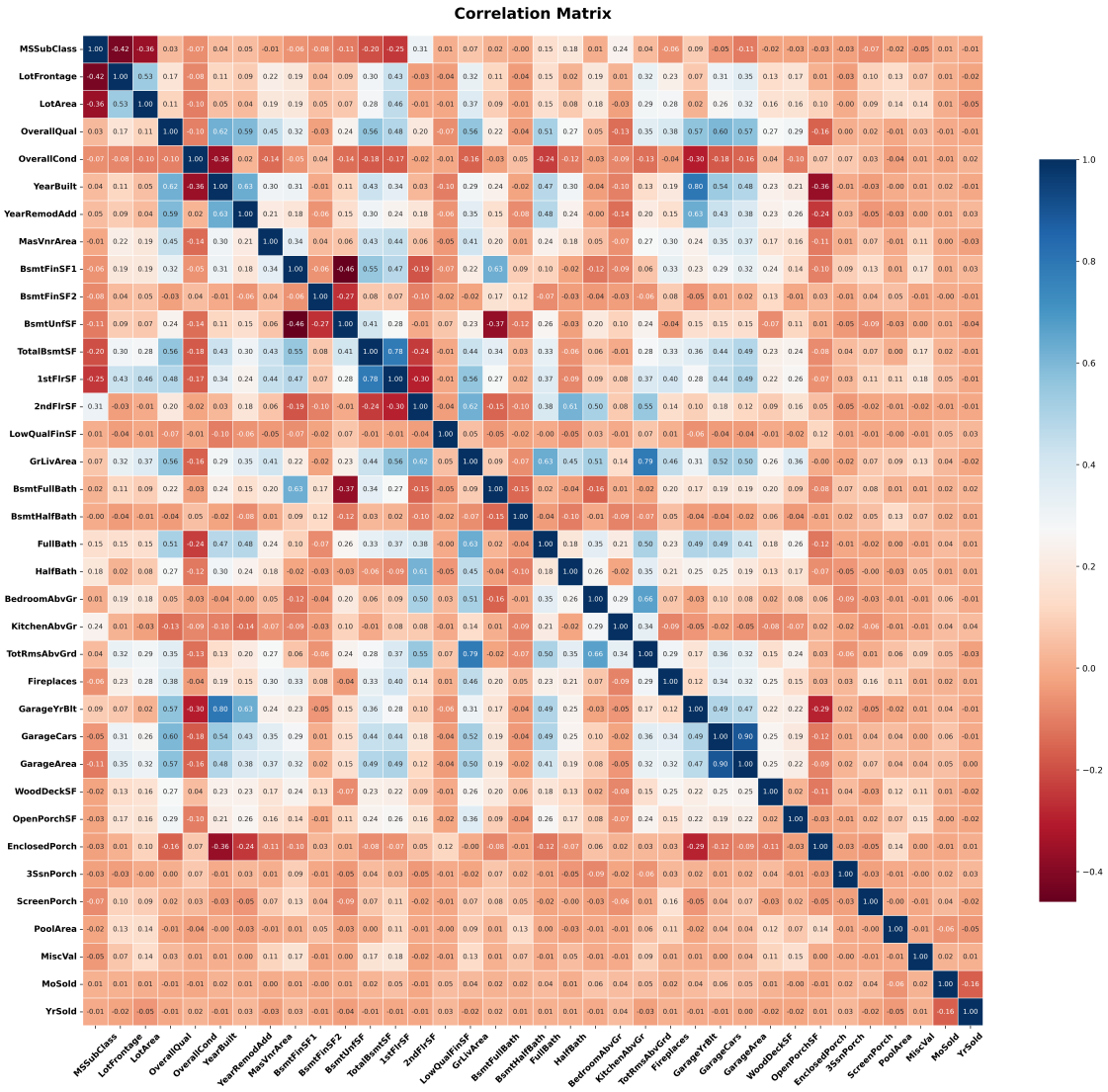
[45]:
```
high_correlated_cols(data_train, plot=True)
```



[45]:
```
[]
```

[46]:
```
#data_train = data_train.drop(columns = ["GarageYrBlt", "TotRmsAbvGrd", "GarageCars", "1stFlrSF"])
```

[47]:
```
high_correlated_cols(data_test, plot=True)
```



[47]: []

[48]:
```
#data_test = data_test.drop(columns = ["GarageYrBlt", "TotRmsAbvGrd",␣
↪"GarageCars", "1stFlrSF"])
```

6 Outlier Analysis

[50]:
```
import numpy as np
import pandas as pd
from sklearn.linear_model import LinearRegression
from scipy.stats import zscore
```

```python
# Step 1: Calculate outlier thresholds based on IQR
def outlier_thresholds(dataframe, col_name, q1=0.05, q3=0.95):
    """
    Calculate lower and upper bounds for outliers using IQR.
    """
    quartile1 = dataframe[col_name].quantile(q1)
    quartile3 = dataframe[col_name].quantile(q3)
    iqr = quartile3 - quartile1
    low_limit = quartile1 - 1.5 * iqr
    up_limit = quartile3 + 1.5 * iqr
    return low_limit, up_limit

# Step 2: Identify outliers in a specific column
def check_outliers(dataframe, col_name):
    """
    Identify rows with outliers in a specific column.
    """
    low_limit, up_limit = outlier_thresholds(dataframe, col_name)
    outliers = dataframe[(dataframe[col_name] < low_limit) |␣
 ↪(dataframe[col_name] > up_limit)]
    return outliers

# Step 3: Remove outliers from training data
def remove_outliers(dataframe, numeric_features, target_col):
    """
    Remove outliers from numeric features and target using IQR and linear␣
 ↪regression residuals.
    """
    # Remove feature outliers using IQR
    for col in numeric_features:
        low_limit, up_limit = outlier_thresholds(dataframe, col)
        dataframe = dataframe[(dataframe[col] >= low_limit) & (dataframe[col]␣
 ↪<= up_limit)]

    # Remove outliers based on target residuals
    X = dataframe.drop(columns=[target_col]).select_dtypes(include=[np.number])
    y = dataframe[target_col]

    model = LinearRegression()
    model.fit(X, y)

    residuals = y - model.predict(X)
    outliers = residuals[np.abs(zscore(residuals)) > 3].index

    dataframe = dataframe.drop(index=outliers)
    return dataframe
```

```python
# Step 4: Clip outliers in test data
def clip_outliers(test_dataframe, train_dataframe, numeric_features):
    """
    Adjust test data outliers using thresholds derived from training data.
    """
    for col in numeric_features:
        low_limit, up_limit = outlier_thresholds(train_dataframe, col)
        test_dataframe[col] = test_dataframe[col].clip(lower=low_limit,␣
 ↪upper=up_limit)
    return test_dataframe

# Step 5: Align test data with training data columns
def align_test_data(test_dataframe, train_dataframe, target_col):
    """
    Align test data with training data columns, excluding the target column.
    """
    X_train = train_dataframe.drop(columns=[target_col])
    X_test = test_dataframe.reindex(columns=X_train.columns, fill_value=0)
    return X_train, X_test

# Step 6: Apply the functions to training and test data
numeric_features = data_train.select_dtypes(include=[np.number]).columns.
 ↪tolist()
numeric_features.remove('SalePrice')  # Exclude target column from numeric␣
 ↪features

data_train_cleaned = remove_outliers(data_train, numeric_features,␣
 ↪target_col='SalePrice')
data_test_cleaned = clip_outliers(data_test, data_train_cleaned,␣
 ↪numeric_features)

# Align test data with cleaned training data
X_train, X_test = align_test_data(data_test_cleaned, data_train_cleaned,␣
 ↪target_col='SalePrice')
y_train = data_train_cleaned['SalePrice']

# Display results
print(f"Training data shape before cleaning: {data_train.shape}")
print(f"Training data shape after cleaning: {data_train_cleaned.shape}")
print(f"Testing data shape before cleaning: {data_test.shape}")
print(f"Testing data shape after cleaning: {data_test_cleaned.shape}")
print(f"X_train shape: {X_train.shape}")
print(f"X_test shape: {X_test.shape}")
```

```
Training data shape before cleaning: (1460, 74)
Training data shape after cleaning: (1246, 74)
Testing data shape before cleaning: (1459, 73)
```

```
Testing data shape after cleaning: (1459, 73)
X_train shape: (1246, 73)
X_test shape: (1459, 73)
```

[51]:
```python
# Step 1: Define the function for calculating outlier thresholds using IQR
def outlier_thresholds(dataframe, col_name, q1=0.05, q3=0.95):
    """
    Calculate the lower and upper limits for outliers using IQR.
    This is done by calculating the first (Q1) and third quartiles (Q3) and␣
 ↪using the formula:
    lower_limit = Q1 - 1.5 * IQR
    upper_limit = Q3 + 1.5 * IQR
    """
    quartile1 = dataframe[col_name].quantile(q1)
    quartile3 = dataframe[col_name].quantile(q3)
    interquantile_range = quartile3 - quartile1
    up_limit = quartile3 + 1.5 * interquantile_range
    low_limit = quartile1 - 1.5 * interquantile_range
    return low_limit, up_limit

# Step 2: Define the function for checking if a column has outliers
def check_outlier(dataframe, col_name):
    """
    Check if a column contains outliers by comparing each value with the␣
 ↪calculated thresholds.
    """
    # Call the outlier_thresholds function to get the limits
    low_limit, up_limit = outlier_thresholds(dataframe, col_name)

    # Filter the rows where the column values are outside the limits
    outliers = dataframe[(dataframe[col_name] > up_limit) |␣
 ↪(dataframe[col_name] < low_limit)]

    # If there are any outliers, print them
    if outliers.shape[0] > 0:
        print(f"Outliers detected in column: {col_name}")
        print(outliers)
        return True
    else:
        print(f"No outliers detected in column: {col_name}")
        return False

# Example usage with a DataFrame
# Assuming 'data_train' is your DataFrame and contains numeric columns you want␣
 ↪to check for outliers
numeric_features = data_train.select_dtypes(include=['number']).columns.tolist()
```

```
# Checking for outliers in all numeric features
for feature in numeric_features:
    check_outlier(data_train, feature)
```

```
No outliers detected in column: MSSubClass
Outliers detected in column: LotFrontage
      MSSubClass MSZoning  LotFrontage   LotArea Street LotShape LandContour  \
Id
935    -0.872563       RL    11.041546  1.717121   Pave      IR2         HLS
1299    0.073375       RL    11.041546  5.348867   Pave      IR3         Bnk

     Utilities LotConfig LandSlope  … EnclosedPorch 3SsnPorch ScreenPorch  \
Id                                  …
935     AllPub    Inside       Mod  …     -0.359325 -0.116339   -0.270208
1299    AllPub    Corner       Gtl  …     -0.359325 -0.116339   -0.270208

       PoolArea   MiscVal    MoSold    YrSold  SaleType  SaleCondition  \
Id
935   -0.068692 -0.087688  1.730892  0.138777        WD         Normal
1299  11.882444 -0.087688 -1.969111  0.138777       New        Partial

      SalePrice
Id
935   12.396697
1299  11.982935

[2 rows x 74 columns]
Outliers detected in column: LotArea
      MSSubClass MSZoning  LotFrontage    LotArea Street LotShape LandContour  \
Id
54     -0.872563       RL    -0.084636   3.984244   Pave      IR1         Low
250    -0.163109       RL    -0.039223  14.881285   Pave      IR2         Low
272    -0.872563       RL     0.142429   2.865064   Pave      IR1         Low
314    -0.872563       RL     3.639229  20.518273   Pave      IR3         Low
336     3.147673       RL    -0.039223  15.448542   Grvl      IR1         HLS
385     0.073375       RL    -0.039223   4.268474   Pave      IR2         Low
452    -0.872563       RL    -0.357114   6.037793   Pave      IR1         Low
458    -0.872563       RL    -0.039223   4.280500   Pave      IR1         Low
524     0.073375       RL     2.730969   2.964284   Pave      IR1         Bnk
662     0.073375       RL    -0.811244   3.615226   Pave      IR2         Lvl
707    -0.872563       RL    -0.039223  10.486449   Pave      IR2         Low
770     0.073375       RL    -1.038309   4.308262   Pave      IR2         HLS
849    -0.163109       RL     0.233255   3.516107   Pave      IR2         Bnk
1299    0.073375       RL    11.041546   5.348867   Pave      IR3         Bnk
1397   -0.872563       RL    -0.039223   4.678682   Pave      IR1         Bnk

     Utilities LotConfig LandSlope  … EnclosedPorch 3SsnPorch ScreenPorch  \
Id                                  …
```

```
54      AllPub    Inside       Gtl  …     -0.359325 -0.116339   -0.270208
250     AllPub   CulDSac       Sev  …     -0.359325 -0.116339   -0.270208
272     AllPub   CulDSac       Sev  …     -0.359325 -0.116339   -0.270208
314     AllPub    Inside       Sev  …     -0.359325 -0.116339   -0.270208
336     AllPub    Corner       Sev  …     -0.359325 -0.116339   -0.270208
385     AllPub    Corner       Mod  …     -0.359325 -0.116339   -0.270208
452     AllPub    Inside       Mod  …     -0.359325 -0.116339   -0.270208
458     AllPub   CulDSac       Mod  …     -0.359325 -0.116339   -0.270208
524     AllPub    Inside       Gtl  …     -0.359325 -0.116339   -0.270208
662     AllPub   CulDSac       Gtl  …     -0.359325 -0.116339   -0.270208
707     AllPub   CulDSac       Sev  …     -0.359325 -0.116339   -0.270208
770     AllPub   CulDSac       Mod  …     -0.359325 -0.116339    3.497397
849     AllPub    Inside       Gtl  …     -0.359325 -0.116339    2.869463
1299    AllPub    Corner       Gtl  …     -0.359325 -0.116339   -0.270208
1397    AllPub    Inside       Sev  …      0.459030 -0.116339   -0.270208

       PoolArea   MiscVal    MoSold    YrSold  SaleType  SaleCondition  \
Id
54    -0.068692 -0.087688  1.730892 -1.367655        WD         Normal
250   -0.068692  0.920472 -0.119110 -0.614439        WD         Normal
272   -0.068692 -0.087688 -0.859110  0.138777        WD         Normal
314   -0.068692 -0.087688 -0.119110  0.891994        WD         Normal
336   -0.068692  1.323736  0.620891  0.138777        WD         Normal
385   -0.068692 -0.087688 -0.119110 -0.614439        WD         Normal
452   -0.068692 -0.087688  2.100892 -1.367655        WD         Normal
458   -0.068692 -0.087688 -1.229111  0.138777        WD         Normal
524   -0.068692 -0.087688  1.360892 -0.614439       New        Partial
662   -0.068692 -0.087688  0.250891  0.891994        WD         Normal
707   -0.068692 -0.087688 -0.119110 -0.614439        WD         Normal
770   -0.068692 -0.087688 -0.119110  1.645210        WD         Normal
849   -0.068692 -0.087688  0.990891  0.138777        WD         Normal
1299  11.882444 -0.087688 -1.969111  0.138777       New        Partial
1397  -0.068692 -0.087688 -0.119110  1.645210        WD         Normal

      SalePrice
Id
54    12.861001
250   12.531776
272   12.394629
314   12.834684
336   12.341263
385   12.388398
452   12.542548
458   12.452937
524   12.126764
662   12.904210
707   12.618186
770   13.195616
```

```
849   12.388398
1299  11.982935
1397  11.982935

[15 rows x 74 columns]
No outliers detected in column: OverallQual
No outliers detected in column: OverallCond
No outliers detected in column: YearBuilt
No outliers detected in column: YearRemodAdd
Outliers detected in column: MasVnrArea
      MSSubClass MSZoning  LotFrontage   LotArea Street LotShape LandContour  \
Id
298     0.073375       FV    -0.175462 -0.312475   Pave      IR1         Lvl
692     0.073375       RL     1.550231  1.104264   Pave      IR1         Lvl
1170    0.073375       RL     2.186013  2.529922   Pave      IR1         Lvl

     Utilities LotConfig LandSlope  … EnclosedPorch 3SsnPorch ScreenPorch  \
Id                                  …
298     AllPub    Inside       Gtl  …     -0.359325 -0.116339    3.282106
692     AllPub    Corner       Gtl  …     -0.359325 -0.116339   -0.270208
1170    AllPub   CulDSac       Gtl  …     -0.359325 -0.116339   -0.270208

      PoolArea   MiscVal    MoSold    YrSold  SaleType  SaleCondition  \
Id
298  -0.068692 -0.087688 -0.119110 -0.614439        WD         Normal
692  -0.068692 -0.087688 -1.969111 -0.614439        WD         Normal
1170 -0.068692 -0.087688  0.250891 -1.367655        WD         Normal

      SalePrice
Id
298   12.384223
692   13.534474
1170  13.345509

[3 rows x 74 columns]
Outliers detected in column: BsmtFinSF1
      MSSubClass MSZoning  LotFrontage   LotArea Street LotShape LandContour  \
Id
1299    0.073375       RL    11.041546  5.348867   Pave      IR3         Bnk

     Utilities LotConfig LandSlope  … EnclosedPorch 3SsnPorch ScreenPorch  \
Id                                  …
1299    AllPub    Corner       Gtl  …     -0.359325 -0.116339   -0.270208

       PoolArea   MiscVal    MoSold    YrSold  SaleType  SaleCondition  \
Id
1299  11.882444 -0.087688 -1.969111  0.138777       New        Partial
```

```
      SalePrice
Id
1299  11.982935

[1 rows x 74 columns]
Outliers detected in column: BsmtFinSF2
      MSSubClass MSZoning  LotFrontage   LotArea Street LotShape LandContour  \
Id
154    -0.872563       RL    -0.039223  0.298980   Pave      Reg         Lvl
272    -0.872563       RL     0.142429  2.865064   Pave      IR1         Low
323     0.073375       RL     0.732798 -0.013713   Pave      IR1         Lvl
471     1.492282       RL    -0.039223 -0.370504   Pave      IR1         Lvl
543    -0.872563       RL     0.369494 -0.037766   Pave      Reg         Lvl
765     1.492282       RL    -1.810330 -0.096998   Pave      IR1         Lvl
855    -0.872563       RL     1.459405  0.741961   Pave      Reg         Lvl
925    -0.872563       RL     0.414907 -0.027744   Pave      Reg         Lvl
1254    0.073375       RL    -0.039223  0.704077   Pave      IR1         Lvl
1459   -0.872563       RL    -0.084636 -0.080160   Pave      Reg         Lvl

     Utilities LotConfig LandSlope  … EnclosedPorch 3SsnPorch ScreenPorch  \
Id                                  …
154     AllPub    Inside       Gtl  …     -0.359325 -0.116339   -0.270208
272     AllPub   CulDSac       Sev  …     -0.359325 -0.116339   -0.270208
323     AllPub    Inside       Gtl  …     -0.359325 -0.116339   -0.270208
471     AllPub    Corner       Gtl  …     -0.359325 -0.116339    2.241529
543     AllPub    Inside       Gtl  …     -0.359325 -0.116339   -0.270208
765     AllPub   CulDSac       Gtl  …     -0.359325 -0.116339    3.605043
855     AllPub    Inside       Gtl  …     -0.359325 -0.116339    5.327377
925     AllPub    Inside       Gtl  …     -0.359325 -0.116339   -0.270208
1254    AllPub    Inside       Gtl  …     -0.359325 -0.116339   -0.270208
1459    AllPub    Inside       Gtl  …      1.473789 -0.116339   -0.270208

      PoolArea   MiscVal    MoSold    YrSold  SaleType  SaleCondition  \
Id
154  -0.068692 -0.087688 -1.229111  0.138777        WD         Normal
272  -0.068692 -0.087688 -0.859110  0.138777        WD         Normal
323  -0.068692 -0.087688  0.620891 -0.614439        WD         Normal
471  -0.068692 -0.087688 -0.119110  1.645210        WD         Normal
543  -0.068692 -0.087688 -0.119110  0.891994        WD         Normal
765  -0.068692 -0.087688 -0.859110 -1.367655        WD         Normal
855  -0.068692 -0.087688  0.250891 -1.367655        WD        Abnorml
925  -0.068692 -0.087688 -0.489110 -1.367655        WD         Normal
1254 -0.068692 -0.087688  0.250891 -0.614439        WD         Normal
1459 -0.068692 -0.087688 -0.859110  1.645210        WD         Normal

      SalePrice
Id
154   12.367345
```

```
272   12.394629
323   12.614869
471   12.264346
543   12.270225
765   12.506181
855   12.043560
925   12.242891
1254  12.591338
1459  11.864469

[10 rows x 74 columns]
No outliers detected in column: BsmtUnfSF
Outliers detected in column: TotalBsmtSF
      MSSubClass MSZoning  LotFrontage   LotArea Street LotShape LandContour  \
Id
1299    0.073375       RL    11.041546  5.348867   Pave      IR3         Bnk

     Utilities LotConfig LandSlope  … EnclosedPorch 3SsnPorch ScreenPorch  \
Id                                  …
1299    AllPub    Corner       Gtl  …     -0.359325 -0.116339   -0.270208

       PoolArea   MiscVal    MoSold    YrSold  SaleType  SaleCondition  \
Id
1299  11.882444 -0.087688 -1.969111  0.138777       New        Partial

      SalePrice
Id
1299  11.982935

[1 rows x 74 columns]
Outliers detected in column: 1stFlrSF
      MSSubClass MSZoning  LotFrontage   LotArea Street LotShape LandContour  \
Id
1299    0.073375       RL    11.041546  5.348867   Pave      IR3         Bnk

     Utilities LotConfig LandSlope  … EnclosedPorch 3SsnPorch ScreenPorch  \
Id                                  …
1299    AllPub    Corner       Gtl  …     -0.359325 -0.116339   -0.270208

       PoolArea   MiscVal    MoSold    YrSold  SaleType  SaleCondition  \
Id
1299  11.882444 -0.087688 -1.969111  0.138777       New        Partial

      SalePrice
Id
1299  11.982935

[1 rows x 74 columns]
```

```
No outliers detected in column: 2ndFlrSF
Outliers detected in column: LowQualFinSF
      MSSubClass MSZoning  LotFrontage   LotArea Street LotShape LandContour  \
Id
52     -0.163109       RM    -0.811244 -0.428632   Pave      Reg         Lvl
89     -0.163109  C (all)     1.595644 -0.205137   Pave      IR1         Lvl
126     3.147673       RM    -0.447940 -0.374512   Pave      Reg         Lvl
171    -0.163109       RM    -0.039223  0.184526   Pave      IR1         Lvl
186     0.428102       RM     0.914450  1.246078   Pave      IR2         Lvl
188    -0.163109       RL    -0.447940 -0.010707   Pave      Reg         Lvl
198     0.428102       RL     4.729141  1.493526   Pave      Reg         Lvl
199     0.428102       RM     1.005276 -0.500792   Pave      Reg         Lvl
264    -0.163109       RM    -0.902070 -0.502797   Pave      Reg         Lvl
268     0.428102       RL    -0.447940 -0.212153   Pave      Reg         Bnk
407    -0.163109       RL    -0.856657 -0.003691   Pave      Reg         Lvl
590    -0.399594       RM    -0.902070 -0.141997   Pave      Reg         Lvl
636     3.147673       RH    -0.447940  0.038001   Pave      Reg         Bnk
730    -0.636078       RM    -0.811244 -0.428632   Pave      Reg         Lvl
830     2.438219       FV    -2.082808 -0.799053   Pave      Reg         Lvl
832     2.438219       FV    -1.810330 -0.735312   Pave      Reg         Lvl
869     0.073375       RL    -0.039223  0.425460   Pave      IR2         Lvl
874    -0.399594       RL    -0.447940  0.163078   Pave      Reg         Lvl
884     0.428102       RL    -0.447940 -0.432240   Pave      Reg         Bnk
946    -0.163109       RM     1.277754 -0.170060   Pave      Reg         Lvl
1010   -0.163109       RL    -0.447940 -0.452686   Pave      Reg         Lvl
1032    0.428102       RL     1.459405  0.535804   Pave      Reg         Lvl
1174   -0.163109       RL     3.094273  0.752985   Pave      IR1         Bnk
1350    0.309859       RM    -0.902070 -0.527852   Pave      Reg         Lvl
1365    2.438219       FV    -1.810330 -0.735312   Pave      Reg         Lvl
1441    0.309859       RL     0.414907  0.101141   Pave      IR1         Bnk

     Utilities LotConfig LandSlope  … EnclosedPorch 3SsnPorch ScreenPorch  \
Id                                  …
52      AllPub    Inside       Gtl  …     -0.359325 -0.116339   -0.270208
89      AllPub    Corner       Gtl  …      2.193941 -0.116339   -0.270208
126     AllPub    Inside       Gtl  …     -0.359325 -0.116339   -0.270208
171     AllPub    Inside       Gtl  …     -0.359325 -0.116339   -0.270208
186     AllPub    Inside       Gtl  …     -0.359325 -0.116339    7.085593
188     AllPub    Inside       Gtl  …     -0.359325  4.660629   -0.270208
198     AllPub    Corner       Gtl  …      8.675309 -0.116339   -0.270208
199     AllPub    Corner       Gtl  …      0.131688 -0.116339   -0.270208
264     AllPub    Corner       Gtl  …     -0.359325 -0.116339   -0.270208
268     AllPub    Inside       Mod  …     -0.359325 -0.116339   -0.270208
407     AllPub    Inside       Gtl  …     -0.359325 -0.116339   -0.270208
590     AllPub    Inside       Gtl  …     -0.359325 -0.116339   -0.270208
636     AllPub    Inside       Gtl  …     -0.359325 -0.116339   -0.270208
730     AllPub    Inside       Gtl  …      1.473789 -0.116339   -0.270208
830     AllPub    Inside       Gtl  …     -0.359325 -0.116339   -0.270208
```

```
832     AllPub    Inside       Gtl  …     -0.359325 -0.116339   -0.270208
869     AllPub    Corner       Gtl  …     -0.359325 -0.116339   -0.270208
874     AllPub    Inside       Gtl  …     -0.359325 -0.116339   -0.270208
884     AllPub    Inside       Gtl  …     -0.359325 -0.116339   -0.270208
946     AllPub    Corner       Gtl  …      3.634245 -0.116339   -0.270208
1010    AllPub    Inside       Gtl  …      1.899334 -0.116339   -0.270208
1032    AllPub    Corner       Gtl  …     -0.359325 -0.116339   -0.270208
1174    AllPub    Inside       Gtl  …     -0.359325 -0.116339   -0.270208
1350    AllPub    Inside       Gtl  …     -0.031983 -0.116339   -0.270208
1365    AllPub    Inside       Gtl  …     -0.359325 -0.116339   -0.270208
1441    AllPub    Inside       Mod  …     -0.359325 -0.116339   -0.270208

       PoolArea   MiscVal    MoSold    YrSold  SaleType  SaleCondition  \
Id
52    -0.068692  0.718840  0.990891 -1.367655        WD         Normal
89    -0.068692 -0.087688  1.360892  0.891994     ConLD        Abnorml
126   -0.068692 -0.087688 -0.119110 -1.367655        WD         Normal
171   -0.068692 -0.087688 -0.489110 -0.614439        WD         Normal
186   -0.068692 -0.087688 -0.119110 -1.367655        WD         Normal
188   -0.068692 -0.087688  0.620891  0.891994        WD         Normal
198   12.679187 -0.087688 -1.229111 -1.367655        WD        Abnorml
199   -0.068692 -0.087688  0.250891  0.891994        WD        Abnorml
264   -0.068692 -0.087688 -0.859110  1.645210        WD         Normal
268   -0.068692 -0.087688  0.250891  0.138777        WD         Normal
407   -0.068692 -0.087688 -1.229111  0.138777        WD         Normal
590   -0.068692  1.122104  0.620891  0.138777        WD         Normal
636   -0.068692 -0.087688 -1.229111 -0.614439        WD        Abnorml
730   -0.068692 -0.087688 -1.969111  0.891994        WD         Normal
830   -0.068692 -0.087688  0.250891 -1.367655        WD         Normal
832   -0.068692 -0.087688 -0.119110 -1.367655        WD         Normal
869   -0.068692 -0.087688 -0.489110 -1.367655        WD         Normal
874   -0.068692 -0.087688  0.990891  0.891994        WD         Normal
884   -0.068692 -0.087688 -1.229111 -1.367655        WD         Normal
946   -0.068692 -0.087688  0.990891  0.891994        WD         Normal
1010  -0.068692 -0.087688 -0.119110 -1.367655        WD         Normal
1032  -0.068692 -0.087688  0.620891  0.891994        WD         Normal
1174  -0.068692 -0.087688 -1.229111 -0.614439        WD         Normal
1350  -0.068692 -0.087688  2.100892  0.138777        WD         Normal
1365  -0.068692 -0.087688 -0.859110 -1.367655        WD        Abnorml
1441  -0.068692 -0.087688  0.990891  0.138777        WD         Normal

      SalePrice
Id
52    11.648339
89    11.350418
126   11.344519
171   11.763692
186   13.071072
```

```
188   11.813037
198   12.367345
199   11.552156
264   11.775297
268   12.097936
407   11.652696
590   11.283525
636   12.206078
730   11.542494
830   11.900912
832   11.925042
869   12.037660
874   11.798112
884   11.682677
946   11.735277
1010  11.532738
1032  12.190964
1174  12.208575
1350  11.711785
1365  11.878631
1441  12.160034

[26 rows x 74 columns]
Outliers detected in column: GrLivArea
      MSSubClass MSZoning  LotFrontage   LotArea Street LotShape LandContour  \
Id
1299    0.073375       RL    11.041546  5.348867   Pave      IR3         Bnk

     Utilities LotConfig LandSlope  … EnclosedPorch 3SsnPorch ScreenPorch  \
Id                                  …
1299    AllPub    Corner       Gtl  …     -0.359325 -0.116339   -0.270208

       PoolArea   MiscVal    MoSold    YrSold  SaleType  SaleCondition  \
Id
1299  11.882444 -0.087688 -1.969111  0.138777       New        Partial

      SalePrice
Id
1299  11.982935

[1 rows x 74 columns]
Outliers detected in column: BsmtFullBath
     MSSubClass MSZoning  LotFrontage  LotArea Street LotShape LandContour  \
Id
739    0.782828       RL     -0.44794  0.02838   Pave      Reg         Lvl

    Utilities LotConfig LandSlope  … EnclosedPorch 3SsnPorch ScreenPorch  \
Id                                 …
```

```
739    AllPub    Inside       Gtl  …     -0.359325 -0.116339   -0.270208

     PoolArea   MiscVal    MoSold    YrSold  SaleType  SaleCondition  \
Id
739 -0.068692 -0.087688 -1.229111  0.891994        WD         Alloca

     SalePrice
Id
739  12.095147

[1 rows x 74 columns]
No outliers detected in column: BsmtHalfBath
No outliers detected in column: FullBath
No outliers detected in column: HalfBath
Outliers detected in column: BedroomAbvGr
     MSSubClass MSZoning  LotFrontage   LotArea Street LotShape LandContour  \
Id
636    3.147673       RH     -0.44794  0.038001   Pave      Reg         Bnk

    Utilities LotConfig LandSlope  … EnclosedPorch 3SsnPorch ScreenPorch  \
Id                                 …
636    AllPub    Inside       Gtl  …     -0.359325 -0.116339   -0.270208

     PoolArea   MiscVal    MoSold    YrSold  SaleType  SaleCondition  \
Id
636 -0.068692 -0.087688 -1.229111 -0.614439        WD        Abnorml

     SalePrice
Id
636  12.206078

[1 rows x 74 columns]
Outliers detected in column: KitchenAbvGr
      MSSubClass MSZoning  LotFrontage   LotArea Street LotShape LandContour  \
Id
9      -0.163109       RM    -0.856657 -0.440659   Pave      Reg         Lvl
10      3.147673       RL    -0.902070 -0.310370   Pave      Reg         Lvl
18      0.782828       RL     0.097016  0.027478   Pave      Reg         Lvl
40      0.782828       RL    -0.220875 -0.448677   Pave      Reg         Lvl
49      3.147673       RM    -1.674091 -0.607428   Pave      Reg         Lvl
…          …      …           …        …    …      …          …
1392    0.782828       RL    -0.220875 -0.157632   Pave      Reg         Lvl
1394    3.147673       RM    -0.447940  0.028380   Pave      Reg         Lvl
1413    0.782828       RL    -0.447940 -0.332419   Pave      Reg         Lvl
1417    3.147673       RM    -0.447940  0.082500   Pave      Reg         Lvl
1451    0.782828       RL    -0.447940 -0.152020   Pave      Reg         Lvl

     Utilities LotConfig LandSlope  … EnclosedPorch 3SsnPorch ScreenPorch  \
```

```
Id                                  …
9       AllPub    Inside       Gtl  …      2.995929 -0.116339   -0.270208
10      AllPub    Corner       Gtl  …     -0.359325 -0.116339   -0.270208
18      AllPub    Inside       Gtl  …     -0.359325 -0.116339   -0.270208
40      AllPub    Inside       Gtl  …     -0.359325 -0.116339   -0.270208
49      AllPub    Inside       Gtl  …      1.310118 -0.116339   -0.270208
…         …       …        …   …           …        …         …
1392    AllPub    Inside       Gtl  …     -0.359325 -0.116339   -0.270208
1394    AllPub    Inside       Gtl  …      3.110498 -0.116339   -0.270208
1413    AllPub    Inside       Gtl  …     -0.359325 -0.116339   -0.270208
1417    AllPub    Inside       Gtl  …      1.506523 -0.116339   -0.270208
1451    AllPub       FR2       Gtl  …     -0.359325 -0.116339   -0.270208

      PoolArea   MiscVal    MoSold    YrSold  SaleType  SaleCondition  \
Id
9    -0.068692 -0.087688 -0.859110  0.138777        WD        Abnorml
10   -0.068692 -0.087688 -1.969111  0.138777        WD         Normal
18   -0.068692  0.920472  1.360892 -1.367655        WD         Normal
40   -0.068692 -0.087688 -0.119110  0.138777        WD        AdjLand
49   -0.068692 -0.087688 -0.119110  0.891994       New        Partial
…         …       …        …        …        …             …
1392 -0.068692 -0.087688 -0.859110  0.891994        WD         Normal
1394 -0.068692 -0.087688 -0.859110  0.138777        WD         Normal
1413 -0.068692 -0.087688 -0.119110  0.891994        WD         Normal
1417 -0.068692 -0.087688 -0.859110  1.645210        WD         Normal
1451 -0.068692 -0.087688  0.990891  0.891994        WD         Normal

      SalePrice
Id
9     11.774528
10    11.678448
18    11.407576
40    11.314487
49    11.635152
…         …
1392  11.728045
1394  12.001512
1413  11.407576
1417  11.715874
1451  11.820418

[68 rows x 74 columns]
No outliers detected in column: TotRmsAbvGrd
No outliers detected in column: Fireplaces
No outliers detected in column: GarageYrBlt
No outliers detected in column: GarageCars
No outliers detected in column: GarageArea
Outliers detected in column: WoodDeckSF
```

```
    MSSubClass MSZoning  LotFrontage   LotArea Street LotShape LandContour  \
Id
54   -0.872563       RL    -0.084636  3.984244   Pave      IR1         Low

   Utilities LotConfig LandSlope  … EnclosedPorch 3SsnPorch ScreenPorch  \
Id                                …
54    AllPub    Inside       Gtl  …     -0.359325 -0.116339   -0.270208

    PoolArea   MiscVal    MoSold    YrSold  SaleType  SaleCondition  SalePrice
Id
54 -0.068692 -0.087688  1.730892 -1.367655        WD         Normal  12.861001

[1 rows x 74 columns]
Outliers detected in column: OpenPorchSF
      MSSubClass MSZoning  LotFrontage   LotArea Street LotShape LandContour  \
Id
496    -0.636078  C (all)    -0.447940 -0.264368   Pave      Reg         Lvl
584     0.428102       RM     0.233255  0.298980   Pave      Reg         Lvl
1329   -0.163109       RM    -0.447940 -0.007700   Pave      Reg         Lvl

     Utilities LotConfig LandSlope  … EnclosedPorch 3SsnPorch ScreenPorch  \
Id                                 …
496     AllPub    Inside       Gtl  …      1.522891 -0.116339   -0.270208
584     AllPub    Inside       Gtl  …     -0.359325 -0.116339   -0.270208
1329    AllPub    Corner       Gtl  …     -0.359325 -0.116339    8.341462

      PoolArea   MiscVal    MoSold    YrSold  SaleType  SaleCondition  \
Id
496  -0.068692 -0.087688  1.730892  0.891994        WD        Abnorml
584  -0.068692 -0.087688  0.250891  0.138777        WD         Normal
1329 -0.068692  2.231080 -0.119110  0.138777        WD         Normal

      SalePrice
Id
496   10.460271
584   12.691584
1329  12.452937

[3 rows x 74 columns]
Outliers detected in column: EnclosedPorch
     MSSubClass MSZoning  LotFrontage   LotArea Street LotShape LandContour  \
Id
198    0.428102       RL     4.729141  1.493526   Pave      Reg         Lvl

    Utilities LotConfig LandSlope  … EnclosedPorch 3SsnPorch ScreenPorch  \
Id                                 …
198    AllPub    Corner       Gtl  …      8.675309 -0.116339   -0.270208
```

```
      PoolArea   MiscVal    MoSold    YrSold  SaleType  SaleCondition  \
Id
198  12.679187 -0.087688 -1.229111 -1.367655        WD        Abnorml

     SalePrice
Id
198  12.367345

[1 rows x 74 columns]
Outliers detected in column: 3SsnPorch
      MSSubClass MSZoning  LotFrontage   LotArea Street LotShape LandContour  \
Id
6      -0.163109       RL     0.687385  0.360616   Pave      IR1         Lvl
56     -0.872563       RL     1.368580 -0.034259   Pave      IR1         Lvl
121     0.546344       RL    -0.039223  1.096045   Pave      IR1         Low
130    -0.872563       RL    -0.039223 -0.154726   Pave      Reg         Lvl
160     0.073375       RL     2.912621  0.888085   Pave      IR1         HLS
183    -0.872563       RL    -0.447940 -0.146006   Pave      Reg         Lvl
188    -0.163109       RL    -0.447940 -0.010707   Pave      Reg         Lvl
206    -0.872563       RL     1.323167  0.133713   Pave      Reg         Lvl
238     0.073375       RL    -0.039223 -0.106619   Pave      IR1         Lvl
259     0.073375       RL     0.460320  0.192243   Pave      Reg         Lvl
281     0.073375       RL     0.551146  0.077188   Pave      Reg         Lvl
547    -0.163109       RL     0.006190 -0.178378   Pave      IR1         Bnk
705    -0.872563       RL     0.006190 -0.212153   Pave      Reg         Lvl
727    -0.872563       RL    -0.039223  1.120299   Pave      IR1         Lvl
745     1.492282       RL    -1.310787 -0.513320   Pave      IR1         HLS
890    -0.872563       RL     2.640143  0.164682   Pave      Reg         Lvl
923    -0.872563       RL    -0.220875 -0.028045   Pave      Reg         Lvl
942     0.073375       RL    -0.039223 -0.176574   Pave      IR1         Lvl
1081   -0.872563       RL     0.460320  0.052433   Pave      Reg         Lvl
1157    0.546344       RL     0.687385 -0.116942   Pave      Reg         Lvl
1162   -0.872563       RL    -0.039223  0.427063   Pave      IR1         Low
1182    1.492282       RM    -0.266288 -0.494077   Pave      IR1         HLS
1347   -0.872563       RL    -0.039223  1.028696   Pave      IR2         Lvl
1438   -0.872563       RL     1.186928  0.193145   Pave      Reg         Lvl

     Utilities LotConfig LandSlope  … EnclosedPorch  3SsnPorch ScreenPorch  \
Id                                  …
6       AllPub    Inside       Gtl  …     -0.359325  10.802446   -0.270208
56      AllPub    Inside       Gtl  …     -0.359325  13.770991   -0.270208
121     AllPub   CulDSac       Sev  …     -0.359325   4.319417   -0.270208
130     AllPub    Inside       Gtl  …     -0.359325   6.025478   -0.270208
160     AllPub    Corner       Gtl  …     -0.359325   5.616023   -0.270208
183     AllPub    Inside       Gtl  …     -0.359325   6.025478   -0.270208
188     AllPub    Inside       Gtl  …     -0.359325   4.660629   -0.270208
206     AllPub    Corner       Gtl  …     -0.359325  17.217233   -0.270208
238     AllPub   CulDSac       Gtl  …     -0.359325   8.004507   -0.270208
```

```
259     AllPub    Inside       Gtl  …     -0.359325   8.243356   -0.270208
281     AllPub    Inside       Gtl  …     -0.359325   6.571417   -0.270208
547     AllPub    Inside       Gtl  …     -0.359325   4.797114   -0.270208
705     AllPub    Inside       Gtl  …     -0.359325   4.797114   -0.270208
727     AllPub    Corner       Gtl  …     -0.359325   6.093720   -0.270208
745     AllPub    Inside       Gtl  …      0.786371   5.616023   -0.270208
890     AllPub    Inside       Gtl  …     -0.359325   5.411296   -0.270208
923     AllPub    Inside       Gtl  …     -0.359325   0.668448   -0.270208
942     AllPub       FR2       Gtl  …     -0.359325   5.616023   -0.270208
1081    AllPub    Inside       Gtl  …     -0.359325   7.253841   -0.270208
1157    AllPub    Inside       Gtl  …     -0.359325   3.159296   -0.270208
1162    AllPub   CulDSac       Gtl  …     -0.359325   7.253841   -0.270208
1182    AllPub    Inside       Mod  …     -0.359325   5.104205   -0.270208
1347    AllPub   CulDSac       Gtl  …     -0.359325   9.778810   -0.270208
1438    AllPub       FR2       Gtl  …     -0.359325  10.256507   -0.270208

      PoolArea   MiscVal    MoSold    YrSold  SaleType  SaleCondition  \
Id
6    -0.068692  1.323736  1.360892  0.891994        WD         Normal
56   -0.068692 -0.087688  0.250891  0.138777        WD         Normal
121  -0.068692 -0.087688  1.360892 -1.367655        WD         Normal
130  -0.068692 -0.087688  0.250891 -1.367655        WD        Abnorml
160  -0.068692 -0.087688 -1.229111 -1.367655       New        Partial
183  -0.068692 -0.087688 -0.119110 -0.614439        WD         Normal
188  -0.068692 -0.087688  0.620891  0.891994        WD         Normal
206  -0.068692 -0.087688 -0.489110  0.891994        WD         Normal
238  -0.068692 -0.087688 -1.599111  1.645210        WD         Normal
259  -0.068692 -0.087688 -0.489110  0.138777        WD         Normal
281  -0.068692 -0.087688 -1.969111 -0.614439        WD         Normal
547  -0.068692 -0.087688 -0.489110 -0.614439        WD         Normal
705  -0.068692 -0.087688 -0.489110  1.645210        WD         Normal
727  -0.068692 -0.087688  2.100892  0.891994        WD         Normal
745  -0.068692 -0.087688  1.360892  0.138777        WD         Normal
890  -0.068692 -0.087688 -1.599111  0.891994        WD         Normal
923  -0.068692 -0.087688  1.360892 -1.367655       New        Partial
942  -0.068692 -0.087688 -0.119110  0.891994        WD         Normal
1081 -0.068692 -0.087688  1.360892  0.138777       COD        Abnorml
1157 -0.068692 -0.087688  1.360892  0.138777        WD         Normal
1162 -0.068692 -0.087688  1.730892  0.138777        WD         Normal
1182 -0.068692 -0.087688  1.730892  0.138777       New        Partial
1347 -0.068692 -0.087688 -0.119110 -1.367655        WD         Normal
1438 -0.068692 -0.087688  1.730892  0.138777       New        Partial

      SalePrice
Id
6     11.870607
56    12.103492
121   12.100718
```

```
130   11.918397
160   12.676079
183   11.695255
188   11.813037
206   12.103492
238   12.178193
259   12.352339
281   12.339296
547   12.254868
705   12.269052
727   12.310437
745   12.100718
890   11.915058
923   12.043501
942   12.273736
1081  11.884496
1157  12.100162
1162  12.319406
1182  12.880294
1347  12.478010
1438  12.885673

[24 rows x 74 columns]
Outliers detected in column: ScreenPorch
      MSSubClass MSZoning  LotFrontage   LotArea Street LotShape LandContour  \
Id
186     0.428102       RM      0.91445  1.246078   Pave      IR2         Lvl
1329   -0.163109       RM     -0.44794 -0.007700   Pave      Reg         Lvl
1387    0.073375       RL      0.46032  0.618888   Pave      IR1         Lvl

     Utilities LotConfig LandSlope  … EnclosedPorch 3SsnPorch ScreenPorch  \
Id                                  …
186     AllPub    Inside       Gtl  …     -0.359325 -0.116339    7.085593
1329    AllPub    Corner       Gtl  …     -0.359325 -0.116339    8.341462
1387    AllPub    Inside       Gtl  …     -0.359325 -0.116339    7.623823

       PoolArea   MiscVal    MoSold    YrSold  SaleType  SaleCondition  \
Id
186   -0.068692 -0.087688 -0.119110 -1.367655        WD         Normal
1329  -0.068692  2.231080 -0.119110  0.138777        WD         Normal
1387  12.853474  3.944952  0.250891 -1.367655        WD         Normal

      SalePrice
Id
186   13.071072
1329  12.452937
1387  12.429220
```

```
[3 rows x 74 columns]
Outliers detected in column: PoolArea
      MSSubClass MSZoning  LotFrontage   LotArea Street LotShape LandContour  \
Id
198     0.428102       RL     4.729141  1.493526   Pave      Reg         Lvl
811    -0.872563       RL     0.369494 -0.037766   Pave      Reg         Lvl
1171    0.546344       RL     0.278668 -0.063824   Pave      Reg         Lvl
1183    0.073375       RL     4.093359  0.511751   Pave      IR1         Lvl
1299    0.073375       RL    11.041546  5.348867   Pave      IR3         Bnk
1387    0.073375       RL     0.460320  0.618888   Pave      IR1         Lvl
1424    0.546344       RL    -0.039223  0.919354   Pave      IR1         Lvl

     Utilities LotConfig LandSlope  … EnclosedPorch 3SsnPorch ScreenPorch  \
Id                                  …
198     AllPub    Corner       Gtl  …      8.675309 -0.116339   -0.270208
811     AllPub    Inside       Gtl  …     -0.359325 -0.116339   -0.270208
1171    AllPub    Inside       Gtl  …     -0.359325 -0.116339   -0.270208
1183    AllPub    Corner       Gtl  …     -0.359325 -0.116339   -0.270208
1299    AllPub    Corner       Gtl  …     -0.359325 -0.116339   -0.270208
1387    AllPub    Inside       Gtl  …     -0.359325 -0.116339    7.623823
1424    AllPub   CulDSac       Gtl  …     -0.359325 -0.116339   -0.270208

       PoolArea   MiscVal    MoSold    YrSold  SaleType  SaleCondition  \
Id
198   12.679187 -0.087688 -1.229111 -1.367655        WD        Abnorml
811   16.065342 -0.087688 -1.969111 -1.367655        WD         Normal
1171  14.272672 -0.087688  0.250891  0.138777        WD         Normal
1183  13.749810 -0.087688  0.250891 -0.614439        WD        Abnorml
1299  11.882444 -0.087688 -1.969111  0.138777       New        Partial
1387  12.853474  3.944952  0.250891 -1.367655        WD         Normal
1424  18.306180 -0.087688  0.620891 -1.367655        WD         Alloca

      SalePrice
Id
198   12.367345
811   12.106258
1171  12.049425
1183  13.521141
1299  11.982935
1387  12.429220
1424  12.524421

[7 rows x 74 columns]
Outliers detected in column: MiscVal
      MSSubClass MSZoning  LotFrontage    LotArea Street LotShape LandContour  \
Id
6      -0.163109       RL     0.687385   0.360616   Pave      IR1         Lvl
8       0.073375       RL    -0.039223  -0.013513   Pave      IR1         Lvl
```

17	-0.872563	RL	-0.039223	0.072578	Pave	IR1	Lvl
18	0.782828	RL	0.097016	0.027478	Pave	Reg	Lvl
52	-0.163109	RM	-0.811244	-0.428632	Pave	Reg	Lvl
85	0.546344	RL	-0.039223	-0.199124	Pave	IR1	Lvl
96	0.073375	RL	-0.039223	-0.075350	Pave	IR2	Lvl
99	-0.636078	RL	0.687385	0.010841	Pave	Reg	Lvl
100	-0.872563	RL	0.324081	-0.119949	Pave	IR1	Lvl
107	-0.636078	RM	-0.447940	0.028380	Pave	Reg	Lvl
215	0.073375	RL	-0.039223	0.038402	Pave	IR1	Lvl
250	-0.163109	RL	-0.039223	14.881285	Pave	IR2	Low
251	-0.636078	RL	-0.675005	-0.517830	Pave	IR1	Lvl
336	3.147673	RL	-0.039223	15.448542	Grvl	IR1	HLS
339	-0.872563	RL	0.959863	0.363623	Pave	Reg	Lvl
347	-0.872563	RL	-0.039223	0.226018	Pave	IR1	Lvl
393	-0.872563	RL	-0.039223	-0.218266	Pave	IR1	Lvl
440	-0.163109	RL	-0.130049	0.184125	Pave	Reg	Lvl
500	-0.872563	RL	0.006190	-0.298845	Pave	IR1	Lvl
503	-0.872563	RL	0.006190	-0.134982	Pave	Reg	Lvl
511	-0.872563	RL	0.233255	0.405115	Pave	Reg	Lvl
540	-0.872563	RL	-0.039223	0.090818	Pave	Reg	Lvl
590	-0.399594	RM	-0.902070	-0.141997	Pave	Reg	Lvl
612	0.546344	RL	-0.039223	-0.012210	Pave	IR1	Lvl
627	-0.872563	RL	-0.039223	0.182922	Pave	IR1	Lvl
635	0.782828	RL	-0.266288	-0.354568	Pave	Reg	Lvl
706	3.147673	RM	0.006190	-0.492774	Pave	Reg	Lvl
726	-0.872563	RL	-0.447940	-0.356473	Pave	Reg	Lvl
734	-0.872563	RL	0.460320	-0.051798	Pave	Reg	Lvl
761	-0.872563	RL	0.006190	-0.141997	Pave	Reg	Lvl
767	0.073375	RL	0.460320	-0.009604	Pave	Reg	Lvl
768	-0.163109	RL	0.233255	0.199559	Pave	IR1	Lvl
787	-0.163109	RM	-0.447940	0.028380	Pave	Reg	Lvl
795	0.073375	RL	-0.039223	0.031587	Pave	IR1	Lvl
801	0.073375	RL	0.414907	0.228624	Pave	IR1	HLS
813	-0.872563	C (all)	-0.175462	-0.180884	Grvl	Reg	Bnk
814	-0.872563	RL	0.233255	-0.076853	Pave	Reg	Lvl
868	-0.872563	RL	0.687385	-0.355470	Pave	Reg	Lvl
879	0.664586	RL	0.823624	0.126798	Pave	IR1	Lvl
891	-0.163109	RL	-0.447940	-0.245827	Pave	Reg	Lvl
913	-0.636078	RM	-0.856657	-0.440659	Pave	Reg	Lvl
954	0.073375	RL	-0.039223	0.055941	Pave	IR1	Lvl
1062	-0.636078	C (all)	2.276839	0.749979	Grvl	Reg	Low
1077	-0.163109	RL	-0.447940	0.028380	Pave	Reg	Lvl
1084	-0.872563	RL	0.460320	-0.172064	Pave	Reg	Lvl
1172	-0.872563	RL	0.278668	-0.139993	Pave	Reg	Lvl
1211	0.073375	RL	0.006190	0.070273	Pave	Reg	Lvl
1231	0.782828	RL	-0.039223	0.839176	Pave	IR1	Lvl
1253	-0.872563	RL	-0.357114	-0.066029	Pave	Reg	Lvl
1329	-0.163109	RM	-0.447940	-0.007700	Pave	Reg	Lvl

```
1387    0.073375       RL     0.460320   0.618888   Pave      IR1         Lvl
1458    0.309859       RL    -0.175462  -0.147810   Pave      Reg         Lvl

     Utilities LotConfig LandSlope  … EnclosedPorch  3SsnPorch ScreenPorch  \
Id                                  …
6       AllPub    Inside       Gtl  …     -0.359325  10.802446   -0.270208
8       AllPub    Corner       Gtl  …      3.372372  -0.116339   -0.270208
17      AllPub   CulDSac       Gtl  …     -0.359325  -0.116339   -0.270208
18      AllPub    Inside       Gtl  …     -0.359325  -0.116339   -0.270208
52      AllPub    Inside       Gtl  …     -0.359325  -0.116339   -0.270208
85      AllPub    Inside       Gtl  …     -0.359325  -0.116339   -0.270208
96      AllPub    Corner       Gtl  …     -0.359325  -0.116339   -0.270208
99      AllPub    Corner       Gtl  …      0.900941  -0.116339   -0.270208
100     AllPub    Inside       Gtl  …     -0.359325  -0.116339   -0.270208
107     AllPub    Inside       Gtl  …     -0.359325  -0.116339   -0.270208
215     AllPub       FR2       Gtl  …     -0.359325  -0.116339   -0.270208
250     AllPub   CulDSac       Sev  …     -0.359325  -0.116339   -0.270208
251     AllPub    Inside       Gtl  …     -0.359325  -0.116339   -0.270208
336     AllPub    Corner       Sev  …     -0.359325  -0.116339   -0.270208
339     AllPub    Corner       Gtl  …     -0.359325  -0.116339   -0.270208
347     AllPub   CulDSac       Gtl  …     -0.359325  -0.116339   -0.270208
393     AllPub    Inside       Gtl  …     -0.359325  -0.116339   -0.270208
440     AllPub    Corner       Gtl  …     -0.359325  -0.116339   -0.270208
500     AllPub    Inside       Gtl  …     -0.359325  -0.116339   -0.270208
503     AllPub    Corner       Gtl  …      2.652220  -0.116339   -0.270208
511     AllPub    Inside       Gtl  …     -0.359325  -0.116339    2.743876
540     AllPub    Inside       Gtl  …     -0.359325  -0.116339   -0.270208
590     AllPub    Inside       Gtl  …     -0.359325  -0.116339   -0.270208
612     AllPub       FR2       Gtl  …     -0.359325  -0.116339   -0.270208
627     AllPub    Inside       Gtl  …      0.229890  -0.116339   -0.270208
635     AllPub    Inside       Gtl  …     -0.359325  -0.116339   -0.270208
706     AllPub    Inside       Gtl  …     -0.359325  -0.116339   -0.270208
726     AllPub    Inside       Gtl  …     -0.359325  -0.116339   -0.270208
734     AllPub    Corner       Gtl  …     -0.359325  -0.116339   -0.270208
761     AllPub    Inside       Gtl  …     -0.359325  -0.116339   -0.270208
767     AllPub    Inside       Gtl  …      2.193941  -0.116339   -0.270208
768     AllPub    Inside       Gtl  …      2.193941  -0.116339   -0.270208
787     AllPub    Inside       Gtl  …      2.848625  -0.116339   -0.270208
795     AllPub    Corner       Gtl  …     -0.359325  -0.116339   -0.270208
801     AllPub    Inside       Mod  …      2.668587  -0.116339   -0.270208
813     AllPub    Inside       Mod  …     -0.359325  -0.116339   -0.270208
814     AllPub    Inside       Gtl  …      4.141625  -0.116339   -0.270208
868     AllPub    Corner       Gtl  …     -0.359325  -0.116339   -0.270208
879     AllPub    Inside       Gtl  …     -0.359325  -0.116339   -0.270208
891     AllPub    Corner       Gtl  …     -0.359325  -0.116339   -0.270208
913     AllPub    Inside       Gtl  …      1.473789  -0.116339    1.882709
954     AllPub    Inside       Mod  …     -0.359325  -0.116339   -0.270208
1062    AllPub    Inside       Gtl  …     -0.359325  -0.116339   -0.270208
```

1077	AllPub	Inside	Gtl	…	-0.359325	-0.116339	-0.270208
1084	AllPub	Inside	Gtl	…	-0.359325	-0.116339	-0.270208
1172	AllPub	Inside	Gtl	…	-0.359325	-0.116339	4.896794
1211	AllPub	Inside	Gtl	…	-0.359325	-0.116339	-0.270208
1231	AllPub	Inside	Gtl	…	2.013903	-0.116339	-0.270208
1253	AllPub	Inside	Gtl	…	-0.359325	-0.116339	-0.270208
1329	AllPub	Corner	Gtl	…	-0.359325	-0.116339	8.341462
1387	AllPub	Inside	Gtl	…	-0.359325	-0.116339	7.623823
1458	AllPub	Inside	Gtl	…	-0.359325	-0.116339	-0.270208

Id	PoolArea	MiscVal	MoSold	YrSold	SaleType	SaleCondition	\
6	-0.068692	1.323736	1.360892	0.891994	WD	Normal	
8	-0.068692	0.618024	1.730892	0.891994	WD	Normal	
17	-0.068692	1.323736	-1.229111	1.645210	WD	Normal	
18	-0.068692	0.920472	1.360892	-1.367655	WD	Normal	
52	-0.068692	0.718840	0.990891	-1.367655	WD	Normal	
85	-0.068692	1.323736	-0.489110	0.891994	WD	Normal	
96	-0.068692	0.880146	-0.859110	0.891994	WD	Normal	
99	-0.068692	0.718840	-0.489110	1.645210	COD	Abnorml	
100	-0.068692	0.718840	-1.969111	1.645210	WD	Normal	
107	-0.068692	0.819656	0.620891	-0.614439	WD	Normal	
215	-0.068692	0.819656	-1.229111	1.645210	WD	Normal	
250	-0.068692	0.920472	-0.119110	-0.614439	WD	Normal	
251	-0.068692	0.819656	-0.489110	1.645210	WD	Normal	
336	-0.068692	1.323736	0.620891	0.138777	WD	Normal	
339	-0.068692	0.718840	-0.489110	-1.367655	WD	Normal	
347	-0.068692	31.165268	-0.859110	-0.614439	WD	Normal	
393	-0.068692	2.331896	0.250891	-0.614439	WD	Normal	
440	-0.068692	1.525368	0.620891	0.891994	ConLI	Normal	
500	-0.068692	0.880146	-0.119110	-0.614439	WD	Normal	
503	-0.068692	0.718840	-0.859110	-0.614439	WD	Normal	
511	-0.068692	3.944952	-0.119110	0.891994	WD	Normal	
540	-0.068692	3.944952	-0.489110	1.645210	WD	Normal	
590	-0.068692	1.122104	0.620891	0.138777	WD	Normal	
612	-0.068692	0.920472	0.250891	-0.614439	WD	Normal	
627	-0.068692	1.122104	0.620891	-0.614439	WD	Normal	
635	-0.068692	1.122104	-0.119110	1.645210	WD	Normal	
706	-0.068692	6.969431	0.250891	1.645210	WD	Normal	
726	-0.068692	0.920472	1.730892	0.891994	WD	Normal	
734	-0.068692	0.718840	-1.229111	0.891994	WD	Normal	
761	-0.068692	0.819656	1.360892	0.891994	WD	Normal	
767	-0.068692	0.920472	-1.229111	1.645210	WD	Normal	
768	-0.068692	2.533528	0.250891	0.138777	WD	Normal	
787	-0.068692	2.331896	-0.119110	1.645210	WD	Normal	
795	-0.068692	0.920472	1.360892	0.138777	WD	Normal	
801	-0.068692	0.718840	-0.489110	0.138777	WD	Normal	
813	-0.068692	0.021193	-0.119110	1.645210	WD	Alloca	

814	-0.068692	0.920472	-0.859110	-0.614439	COD	Normal
868	-0.068692	0.718840	-0.489110	-0.614439	WD	Normal
879	-0.068692	0.718840	-0.119110	1.645210	WD	Normal
891	-0.068692	3.944952	0.250891	-0.614439	WD	Normal
913	-0.068692	1.162430	0.250891	-1.367655	WD	Abnorml
954	-0.068692	0.718840	0.990891	0.138777	WD	Normal
1062	-0.068692	1.041451	0.620891	0.138777	ConLD	Normal
1077	-0.068692	0.920472	-0.859110	-1.367655	WD	Normal
1084	-0.068692	1.323736	-1.229111	-1.367655	WD	Normal
1172	-0.068692	2.735160	1.730892	0.138777	WD	Normal
1211	-0.068692	0.718840	-0.489110	1.645210	WD	Normal
1231	-0.068692	16.647766	0.620891	-0.614439	WD	Normal
1253	-0.068692	1.122104	1.730892	0.891994	WD	Normal
1329	-0.068692	2.231080	-0.119110	0.138777	WD	Normal
1387	12.853474	3.944952	0.250891	-1.367655	WD	Normal
1458	-0.068692	4.953112	-0.489110	1.645210	WD	Normal

Id	SalePrice
6	11.870607
8	12.206078
17	11.911708
18	11.407576
52	11.648339
85	12.034697
96	12.128117
99	11.326608
100	11.767188
107	11.512935
215	11.993813
250	12.531776
251	11.245059
336	12.341263
339	12.218500
347	11.928348
393	11.575910
440	11.608245
500	11.695255
503	11.849405
511	12.013101
540	12.513561
590	11.283525
612	11.904974
627	11.848690
635	11.877576
706	10.915107
726	11.699413
734	11.786009

```
761    11.755879
767    12.188423
768    11.982935
787    11.842236
795    12.178193
801    12.206078
813    10.933000
814    11.969724
868    11.767575
879    11.904974
891    11.719134
913    11.385103
954    12.055256
1062   11.302217
1077   12.043560
1084   11.982935
1172   12.001512
1211   12.149508
1231   12.154785
1253   11.775297
1329   12.452937
1387   12.429220
1458   12.493133

[52 rows x 74 columns]
No outliers detected in column: MoSold
No outliers detected in column: YrSold
No outliers detected in column: SalePrice
```

[52]:
```
check_outlier(data_train, feature)
```

```
No outliers detected in column: SalePrice
```

[52]:
```
False
```

[53]:
```
# Add new feature: Age of the house
#Add new feature: Total Square Footage
data_train["NEW_HouseAge"] = data_train["YrSold"] - data_train["YearBuilt"]
data_train["NEW_TotalSF"] = data_train["TotalBsmtSF"] + data_train["1stFlrSF"]␣
 ↪+ data_train["2ndFlrSF"]
data_test["NEW_HouseAge"] = data_test["YrSold"] - data_test["YearBuilt"]
data_test["NEW_TotalSF"] = data_test["TotalBsmtSF"] + data_test["1stFlrSF"] +␣
 ↪data_test["2ndFlrSF"]
```

7 Label- Ordinal- One-Hot- Encoding / Scale

[55]:
```
# Define a dictionary for Ordinal Encoding with specified categories
ordinal_columns = {
    'ExterQual': ['Fa', 'TA', 'Gd', 'Ex'],  # Poor (Fa), Average (TA), Good␣
 ↪(Gd), Excellent (Ex)
    'ExterCond': ['Fa', 'TA', 'Gd', 'Ex'],  # Poor (Fa), Average (TA), Good␣
 ↪(Gd), Excellent (Ex)
}

# List to store encoders for potential reuse
encoders = {}

# ------------------------------ Processing Data for Training␣
 ↪------------------------------
# Extract and remove 'SalePrice' from training data
sale_price = data_train['SalePrice']  # Extract 'SalePrice' column
data_train = data_train.drop(columns=['SalePrice'], errors='ignore')  # Drop␣
 ↪'SalePrice' for preprocessing

# Process categorical features
for col in categoric_features:
    if col in ordinal_columns:
        # Apply Ordinal Encoding for specified columns
        try:
            oe = OrdinalEncoder(categories=[ordinal_columns[col]],␣
 ↪handle_unknown='use_encoded_value', unknown_value=-1)
            data_train[col] = oe.fit_transform(data_train[[col]])
            encoders[col] = oe  # Save the encoder for later use
            print(f"Ordinal Encoding applied on {col}.")
        except ValueError as e:
            print(f"Error in Ordinal Encoding for {col}: {e}")
    elif col in data_train.columns and data_train[col].nunique() < 10:  # If␣
 ↪the number of categories is less than 10
        # Apply One-Hot Encoding for features with a small number of categories
        data_train = pd.get_dummies(data_train, columns=[col], drop_first=True)
        print(f"One-Hot Encoding applied on {col}.")
    elif col == 'Neighborhood':  # Example for Target Encoding
        # Apply Target Encoding for 'Neighborhood' feature
        try:
            if 'SalePrice' in data_train.columns:
                target_mean = data_train.groupby(col)['SalePrice'].mean()
                data_train[col] = data_train[col].map(target_mean)
                print(f"Target Encoding applied on {col}.")
            else:
                print(f"SalePrice column not found. Skipping Target Encoding␣
 ↪for {col}.")
```

```
        except Exception as e:
            print(f"Error in Target Encoding for {col}: {e}")
    elif col in data_train.columns:
        # Apply Label Encoding for other categorical features
        try:
            le = LabelEncoder()
            data_train[col] = le.fit_transform(data_train[col].astype(str))
            encoders[col] = le  # Save the encoder for later use
            print(f"Label Encoding applied on {col}.")
        except Exception as e:
            print(f"Error in Label Encoding for {col}: {e}")

# ------------------------------ Processing Data for Testing␣
 ↪------------------------------
# Drop 'SalePrice' column from the test data if it exists
data_test = data_test.drop(columns=['SalePrice'], errors='ignore')  # Drop␣
 ↪'SalePrice' from test data

# Align the test data with the training data columns
data_train, data_test = data_train.align(data_test, join='left', axis=1)

# Fill missing columns in the test data with 0 (or another appropriate value)
data_test = data_test.fillna(0)

# Apply the same encoding transformations to test data
for col in categoric_features:
    if col in ordinal_columns:
        # Apply Ordinal Encoding for test data
        try:
            data_test[col] = encoders[col].transform(data_test[[col]])
            print(f"Ordinal Encoding applied on {col} for test data.")
        except Exception as e:
            print(f"Error in Ordinal Encoding for {col} in test data: {e}")
    elif col in data_test.columns and data_test[col].nunique() < 10:  # If the␣
 ↪number of categories is less than 10
        # Apply One-Hot Encoding for test data
        data_test = pd.get_dummies(data_test, columns=[col], drop_first=True)
        print(f"One-Hot Encoding applied on {col} for test data.")
    elif col in data_test.columns:
        # Apply Label Encoding for test data
        try:
            data_test[col] = encoders[col].transform(data_test[col].astype(str))
            print(f"Label Encoding applied on {col} for test data.")
        except Exception as e:
            print(f"Error in Label Encoding for {col} in test data: {e}")
```

```
# ------------------------------ Adding 'SalePrice' back to Training Data␣
 ↪------------------------------
data_train['SalePrice'] = sale_price  # Re-add 'SalePrice' column to the␣
 ↪training data

# ------------------------------ Converting Boolean Columns to Integer␣
 ↪------------------------------
# Check for boolean columns
bool_columns = data_train.select_dtypes(include=['bool']).columns
if len(bool_columns) > 0:
    # Convert boolean columns to integer
    data_train[bool_columns] = data_train[bool_columns].astype(int)
    data_test[bool_columns] = data_test[bool_columns].astype(int)
    print(f"Converted boolean columns to integer: {bool_columns}")

# ------------------------------ Final Check ------------------------------
# Check the data types of the columns
print(data_train.dtypes)

# Verify the shapes of the training and test data to ensure they match
print(f"Training data shape: {data_train.shape}")
print(f"Test data shape: {data_test.shape}")
```

```
One-Hot Encoding applied on MSZoning.
One-Hot Encoding applied on Street.
One-Hot Encoding applied on LotShape.
One-Hot Encoding applied on LandContour.
One-Hot Encoding applied on Utilities.
One-Hot Encoding applied on LotConfig.
One-Hot Encoding applied on LandSlope.
SalePrice column not found. Skipping Target Encoding for Neighborhood.
One-Hot Encoding applied on Condition1.
One-Hot Encoding applied on Condition2.
One-Hot Encoding applied on BldgType.
One-Hot Encoding applied on HouseStyle.
One-Hot Encoding applied on RoofStyle.
One-Hot Encoding applied on RoofMatl.
Label Encoding applied on Exterior1st.
Label Encoding applied on Exterior2nd.
Ordinal Encoding applied on ExterQual.
Ordinal Encoding applied on ExterCond.
One-Hot Encoding applied on Foundation.
One-Hot Encoding applied on BsmtQual.
One-Hot Encoding applied on BsmtCond.
One-Hot Encoding applied on BsmtExposure.
One-Hot Encoding applied on BsmtFinType1.
One-Hot Encoding applied on BsmtFinType2.
```

```
One-Hot Encoding applied on Heating.
One-Hot Encoding applied on HeatingQC.
One-Hot Encoding applied on CentralAir.
One-Hot Encoding applied on Electrical.
One-Hot Encoding applied on KitchenQual.
One-Hot Encoding applied on Functional.
One-Hot Encoding applied on GarageType.
One-Hot Encoding applied on GarageFinish.
One-Hot Encoding applied on GarageQual.
One-Hot Encoding applied on GarageCond.
One-Hot Encoding applied on PavedDrive.
One-Hot Encoding applied on SaleType.
One-Hot Encoding applied on SaleCondition.
Error in Label Encoding for Neighborhood in test data: 'Neighborhood'
Label Encoding applied on Exterior1st for test data.
Label Encoding applied on Exterior2nd for test data.
Ordinal Encoding applied on ExterQual for test data.
Ordinal Encoding applied on ExterCond for test data.
Converted boolean columns to integer: Index(['MSZoning_FV', 'MSZoning_RH',
'MSZoning_RL', 'MSZoning_RM',
       'Street_Pave', 'LotShape_IR2', 'LotShape_IR3', 'LotShape_Reg',
       'LandContour_HLS', 'LandContour_Low',
       …
       'SaleType_ConLI', 'SaleType_ConLw', 'SaleType_New', 'SaleType_Oth',
       'SaleType_WD', 'SaleCondition_AdjLand', 'SaleCondition_Alloca',
       'SaleCondition_Family', 'SaleCondition_Normal',
       'SaleCondition_Partial'],
      dtype='object', length=133)
MSSubClass               float64
LotFrontage              float64
LotArea                  float64
Neighborhood              object
OverallQual              float64
                          …
SaleCondition_Alloca       int32
SaleCondition_Family       int32
SaleCondition_Normal       int32
SaleCondition_Partial      int32
SalePrice                float64
Length: 177, dtype: object
Training data shape: (1460, 177)
Test data shape: (1459, 176)
```

[56]:
```
le = LabelEncoder()
data_train['Neighborhood'] = le.fit_transform(data_train['Neighborhood'])
data_test['Neighborhood'] = le.transform(data_test['Neighborhood'])
```

```
print(data_train['Neighborhood'].head())
```

```
Id
1     5
2    24
3     5
4     6
5    15
Name: Neighborhood, dtype: int32
```

[57]:
```
data_train.reset_index(inplace=True)
data_test.reset_index(inplace=True)
```

```
C:\Users\izadi\AppData\Local\Temp\ipykernel_26524\3411960026.py:1:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  data_train.reset_index(inplace=True)
C:\Users\izadi\AppData\Local\Temp\ipykernel_26524\3411960026.py:2:
PerformanceWarning: DataFrame is highly fragmented.  This is usually the result
of calling `frame.insert` many times, which has poor performance.  Consider
joining all columns at once using pd.concat(axis=1) instead. To get a de-
fragmented frame, use `newframe = frame.copy()`
  data_test.reset_index(inplace=True)
```

8 Modelling

8.1 Linear Regression

[59]:
```
y = data_train["SalePrice"]
X = data_train.drop(["Id", "SalePrice"], axis=1)
X_train, X_test, y_train, y_test = train_test_split(X,y,test_size=0.3,␣
 ↪random_state=42)
```

[60]:
```
# Linear Regression

reg_model = LinearRegression().fit(X_train,y_train) #built model
reg_pred = reg_model.predict(X_test) #make prediction

print('Linear Regression Results :')
print('MAE:', mean_absolute_error(y_test, reg_pred))
print('MSE:', mean_squared_error(y_test, reg_pred))
print('RMSE:', np.sqrt(mean_squared_error(y_test, reg_pred)))
```

```
Linear Regression Results :
MAE: 0.09907442898562975
MSE: 0.034254639424272294
```

```
RMSE: 0.1850800892161885
```

[61]:
```
liner_r2_score=r2_score(y_test, reg_pred)
liner_r2_score
```

[61]:
```
0.7980792059243104
```

[62]:
```
X_train_ols = sm.add_constant(X_train)
ols_model = sm.OLS(y_train, X_train_ols).fit()
print("OLS Regression Results:")
print(ols_model.summary())
```

```
OLS Regression Results:
                            OLS Regression Results
==============================================================================
Dep. Variable:              SalePrice   R-squared:                       0.937
Model:                            OLS   Adj. R-squared:                  0.924
Method:                 Least Squares   F-statistic:                     75.03
Date:                Wed, 05 Feb 2025   Prob (F-statistic):               0.00
Time:                        11:30:47   Log-Likelihood:                912.14
No. Observations:                1022   AIC:                            -1486.
Df Residuals:                     853   BIC:                            -653.2
Df Model:                         168
Covariance Type:            nonrobust
================================================================================
=========
                               coef    std err          t      P>|t|      [0.025
0.975]
--------------------------------------------------------------------------------
---------
const                        8.6822      0.305     28.513      0.000       8.085
9.280
MSSubClass                  -0.0076      0.024     -0.325      0.746      -0.054
0.039
LotFrontage                  0.0069      0.005      1.312      0.190      -0.003
0.017
LotArea                      0.0165      0.007      2.487      0.013       0.003
0.030
Neighborhood                 0.0010      0.001      1.426      0.154      -0.000
0.002
OverallQual                  0.0747      0.008      9.778      0.000       0.060
0.090
OverallCond                  0.0432      0.005      7.862      0.000       0.032
0.054
YearBuilt                    0.0341      0.008      4.355      0.000       0.019
0.049
YearRemodAdd                 0.0227      0.007      3.476      0.001       0.010
0.035
```

Exterior1st	-0.0084	0.002	-3.580	0.000	-0.013 -0.004
Exterior2nd	0.0053	0.002	2.570	0.010	0.001 0.009
MasVnrArea	0.0041	0.005	0.843	0.400	-0.005 0.014
ExterQual	0.0174	0.011	1.543	0.123	-0.005 0.040
ExterCond	-0.0150	0.012	-1.260	0.208	-0.038 0.008
BsmtFinSF1	0.0135	0.006	2.424	0.016	0.003 0.024
BsmtFinSF2	0.0111	0.007	1.505	0.133	-0.003 0.025
BsmtUnfSF	-0.0151	0.004	-3.374	0.001	-0.024 -0.006
TotalBsmtSF	0.0029	0.006	0.451	0.652	-0.010 0.015
1stFlrSF	0.0147	0.009	1.716	0.086	-0.002 0.032
2ndFlrSF	0.0278	0.008	3.475	0.001	0.012 0.043
LowQualFinSF	0.0084	0.006	1.433	0.152	-0.003 0.020
GrLivArea	0.0347	0.005	6.492	0.000	0.024 0.045
BsmtFullBath	0.0092	0.006	1.591	0.112	-0.002 0.021
BsmtHalfBath	0.0018	0.004	0.437	0.662	-0.006 0.010
FullBath	0.0091	0.007	1.375	0.169	-0.004 0.022
HalfBath	0.0082	0.006	1.377	0.169	-0.003 0.020
BedroomAbvGr	0.0018	0.006	0.281	0.778	-0.011 0.014
KitchenAbvGr	-0.0162	0.008	-2.068	0.039	-0.032 -0.001
TotRmsAbvGrd	0.0109	0.009	1.266	0.206	-0.006 0.028
Fireplaces	0.0238	0.005	5.065	0.000	0.015 0.033
GarageYrBlt	-0.0183	0.008	-2.434	0.015	-0.033 -0.004
GarageCars	0.0217	0.009	2.339	0.020	0.003 0.040
GarageArea	0.0333	0.009	3.607	0.000	0.015 0.051

WoodDeckSF	0.0058	0.004	1.451	0.147	-0.002
0.014					
OpenPorchSF	-9.322e-05	0.004	-0.023	0.981	-0.008
0.008					
EnclosedPorch	0.0102	0.004	2.351	0.019	0.002
0.019					
3SsnPorch	0.0055	0.003	1.658	0.098	-0.001
0.012					
ScreenPorch	0.0118	0.004	2.992	0.003	0.004
0.020					
PoolArea	-0.0021	0.005	-0.425	0.671	-0.012
0.007					
MiscVal	-0.0052	0.009	-0.551	0.582	-0.024
0.013					
MoSold	0.0008	0.004	0.215	0.830	-0.007
0.008					
YrSold	0.0162	0.005	3.441	0.001	0.007
0.025					
NEW_HouseAge	-0.0179	0.004	-4.474	0.000	-0.026
-0.010					
NEW_TotalSF	0.0454	0.003	13.506	0.000	0.039
0.052					
MSZoning_FV	0.4144	0.070	5.887	0.000	0.276
0.553					
MSZoning_RH	0.3937	0.074	5.349	0.000	0.249
0.538					
MSZoning_RL	0.3871	0.068	5.715	0.000	0.254
0.520					
MSZoning_RM	0.3204	0.068	4.730	0.000	0.187
0.453					
Street_Pave	0.0753	0.071	1.058	0.291	-0.064
0.215					
LotShape_IR2	0.0260	0.023	1.125	0.261	-0.019
0.071					
LotShape_IR3	0.0082	0.049	0.168	0.867	-0.088
0.105					
LotShape_Reg	-0.0049	0.009	-0.548	0.584	-0.022
0.013					
LandContour_HLS	0.0809	0.030	2.661	0.008	0.021
0.141					
LandContour_Low	0.0149	0.037	0.400	0.689	-0.058
0.088					
LandContour_Lvl	0.0328	0.023	1.436	0.151	-0.012
0.078					
Utilities_NoSeWa	-0.1578	0.128	-1.232	0.218	-0.409
0.094					
LotConfig_CulDSac	0.0217	0.018	1.231	0.219	-0.013
0.056					

```
LotConfig_FR2          -0.0439      0.022     -1.985      0.047      -0.087
-0.001
LotConfig_FR3          -0.0783      0.072     -1.088      0.277      -0.220
0.063
LotConfig_Inside       -0.0087      0.010     -0.870      0.384      -0.028
0.011
LandSlope_Mod           0.0205      0.022      0.916      0.360      -0.023
0.065
LandSlope_Sev          -0.0505      0.084     -0.598      0.550      -0.216
0.115
Condition1_Feedr        0.0896      0.028      3.253      0.001       0.036
0.144
Condition1_Norm         0.1191      0.023      5.288      0.000       0.075
0.163
Condition1_PosA         0.0179      0.056      0.317      0.751      -0.093
0.128
Condition1_PosN         0.1242      0.042      2.983      0.003       0.042
0.206
Condition1_RRAe         0.0143      0.050      0.287      0.775      -0.083
0.112
Condition1_RRAn         0.0871      0.037      2.336      0.020       0.014
0.160
Condition1_RRNe         0.0933      0.113      0.827      0.408      -0.128
0.315
Condition1_RRNn         0.0983      0.062      1.596      0.111      -0.023
0.219
Condition2_Feedr        0.2104      0.135      1.561      0.119      -0.054
0.475
Condition2_Norm         0.0357      0.102      0.349      0.727      -0.165
0.237
Condition2_PosA       -4.2e-15   8.39e-16     -5.005      0.000   -5.85e-15
-2.55e-15
Condition2_PosN        -0.9111      0.137     -6.661      0.000      -1.180
-0.643
Condition2_RRAe        -0.2683      0.289     -0.929      0.353      -0.835
0.298
Condition2_RRAn        -0.0309      0.155     -0.199      0.842      -0.335
0.273
Condition2_RRNn      2.835e-15   8.05e-16      3.521      0.000    1.25e-15
4.42e-15
BldgType_2fmCon         0.0524      0.077      0.682      0.496      -0.099
0.203
BldgType_Duplex        -0.0278      0.042     -0.659      0.510      -0.111
0.055
BldgType_Twnhs         -0.0603      0.062     -0.975      0.330      -0.182
0.061
BldgType_TwnhsE        -0.0081      0.059     -0.138      0.890      -0.123
0.107
```

HouseStyle_1.5Unf	0.0261	0.043	0.611	0.541	-0.058
0.110					
HouseStyle_1Story	-0.0321	0.026	-1.231	0.219	-0.083
0.019					
HouseStyle_2.5Fin	-0.0771	0.071	-1.087	0.277	-0.216
0.062					
HouseStyle_2.5Unf	0.0147	0.047	0.313	0.754	-0.078
0.107					
HouseStyle_2Story	-0.0479	0.020	-2.373	0.018	-0.088
-0.008					
HouseStyle_SFoyer	-0.0742	0.038	-1.959	0.050	-0.149
0.000					
HouseStyle_SLvl	-0.0487	0.032	-1.513	0.131	-0.112
0.014					
RoofStyle_Gable	-0.1483	0.121	-1.221	0.223	-0.387
0.090					
RoofStyle_Gambrel	-0.0957	0.128	-0.746	0.456	-0.348
0.156					
RoofStyle_Hip	-0.1488	0.122	-1.222	0.222	-0.388
0.090					
RoofStyle_Mansard	-0.0310	0.139	-0.223	0.824	-0.304
0.242					
RoofStyle_Shed	0.1054	0.205	0.513	0.608	-0.298
0.509					
RoofMatl_CompShg	2.6602	0.168	15.840	0.000	2.331
2.990					
RoofMatl_Membran	2.884e-15	5.63e-16	5.121	0.000	1.78e-15
3.99e-15					
RoofMatl_Metal	2.6719	0.256	10.437	0.000	2.169
3.174					
RoofMatl_Roll	2.6726	0.207	12.896	0.000	2.266
3.079					
RoofMatl_Tar&Grv	2.5504	0.212	12.003	0.000	2.133
2.967					
RoofMatl_WdShake	2.6381	0.191	13.819	0.000	2.263
3.013					
RoofMatl_WdShngl	2.6884	0.179	15.057	0.000	2.338
3.039					
Foundation_CBlock	0.0137	0.018	0.752	0.452	-0.022
0.049					
Foundation_PConc	0.0446	0.020	2.259	0.024	0.006
0.083					
Foundation_Slab	-0.0461	0.044	-1.054	0.292	-0.132
0.040					
Foundation_Stone	0.0593	0.064	0.931	0.352	-0.066
0.184					
Foundation_Wood	-0.1621	0.072	-2.256	0.024	-0.303
-0.021					

BsmtQual_Fa	-0.0657	0.034	-1.935	0.053	-0.132
0.001					
BsmtQual_Gd	-0.0566	0.017	-3.263	0.001	-0.091
-0.023					
BsmtQual_TA	-0.0828	0.022	-3.730	0.000	-0.126
-0.039					
BsmtCond_Gd	0.0460	0.030	1.542	0.123	-0.013
0.105					
BsmtCond_Po	0.1318	0.158	0.833	0.405	-0.179
0.442					
BsmtCond_TA	0.0654	0.024	2.770	0.006	0.019
0.112					
BsmtExposure_Gd	0.0318	0.017	1.928	0.054	-0.001
0.064					
BsmtExposure_Mn	-0.0136	0.017	-0.787	0.432	-0.048
0.020					
BsmtExposure_No	-0.0123	0.012	-0.996	0.319	-0.036
0.012					
BsmtFinType1_BLQ	-0.0070	0.015	-0.458	0.647	-0.037
0.023					
BsmtFinType1_GLQ	0.0168	0.014	1.202	0.230	-0.011
0.044					
BsmtFinType1_LwQ	-0.0018	0.021	-0.085	0.932	-0.043
0.039					
BsmtFinType1_Rec	0.0117	0.017	0.686	0.493	-0.022
0.045					
BsmtFinType1_Unf	-0.0070	0.016	-0.429	0.668	-0.039
0.025					
BsmtFinType2_BLQ	-0.0406	0.042	-0.964	0.335	-0.123
0.042					
BsmtFinType2_GLQ	0.0289	0.053	0.549	0.583	-0.074
0.132					
BsmtFinType2_LwQ	-0.0027	0.040	-0.069	0.945	-0.080
0.075					
BsmtFinType2_Rec	0.0237	0.037	0.634	0.526	-0.050
0.097					
BsmtFinType2_Unf	0.0422	0.039	1.070	0.285	-0.035
0.120					
Heating_GasA	0.0404	0.119	0.340	0.734	-0.193
0.274					
Heating_GasW	0.0623	0.123	0.507	0.612	-0.179
0.304					
Heating_Grav	-0.3593	0.136	-2.649	0.008	-0.626
-0.093					
Heating_OthW	-0.0303	0.148	-0.204	0.839	-0.322
0.261					
Heating_Wall	0.1883	0.142	1.327	0.185	-0.090
0.467					

```
HeatingQC_Fa            -0.0067      0.026     -0.263      0.793      -0.057
0.044
HeatingQC_Gd            -0.0249      0.012     -2.137      0.033      -0.048
-0.002
HeatingQC_Po            -0.1151      0.131     -0.877      0.381      -0.373
0.142
HeatingQC_TA            -0.0340      0.012     -2.863      0.004      -0.057
-0.011
CentralAir_Y             0.0535      0.021      2.562      0.011       0.013
0.094
Electrical_FuseF        -0.0408      0.031     -1.324      0.186      -0.101
0.020
Electrical_FuseP        -0.0856      0.099     -0.867      0.386      -0.279
0.108
Electrical_Mix       -8.867e-17   4.35e-16     -0.204      0.839   -9.43e-16
7.65e-16
Electrical_SBrkr        -0.0231      0.018     -1.319      0.188      -0.057
0.011
KitchenQual_Fa          -0.0013      0.035     -0.036      0.971      -0.070
0.067
KitchenQual_Gd          -0.0616      0.019     -3.193      0.001      -0.099
-0.024
KitchenQual_TA          -0.0556      0.023     -2.461      0.014      -0.100
-0.011
Functional_Maj2         -0.1971      0.073     -2.699      0.007      -0.340
-0.054
Functional_Min1          0.0452      0.048      0.935      0.350      -0.050
0.140
Functional_Min2          0.0587      0.049      1.193      0.233      -0.038
0.155
Functional_Mod          -0.0625      0.054     -1.153      0.249      -0.169
0.044
Functional_Sev          -0.4297      0.145     -2.959      0.003      -0.715
-0.145
Functional_Typ           0.0908      0.042      2.154      0.031       0.008
0.174
GarageType_Attchd        0.0711      0.067      1.066      0.287      -0.060
0.202
GarageType_Basment       0.0619      0.075      0.829      0.407      -0.085
0.209
GarageType_BuiltIn       0.0577      0.070      0.829      0.407      -0.079
0.194
GarageType_CarPort       0.1438      0.083      1.739      0.082      -0.018
0.306
GarageType_Detchd        0.0847      0.066      1.283      0.200      -0.045
0.214
GarageFinish_RFn         0.0079      0.011      0.738      0.461      -0.013
0.029
```

GarageFinish_Unf	-0.0010	0.013	-0.075	0.941	-0.027
0.025					
GarageQual_Fa	-0.2319	0.156	-1.485	0.138	-0.538
0.075					
GarageQual_Gd	-0.1967	0.160	-1.229	0.219	-0.511
0.117					
GarageQual_Po	-0.6835	0.249	-2.746	0.006	-1.172
-0.195					
GarageQual_TA	-0.2148	0.154	-1.399	0.162	-0.516
0.087					
GarageCond_Fa	0.1115	0.176	0.635	0.525	-0.233
0.456					
GarageCond_Gd	0.1874	0.182	1.032	0.302	-0.169
0.544					
GarageCond_Po	0.5032	0.201	2.502	0.013	0.108
0.898					
GarageCond_TA	0.1611	0.174	0.927	0.354	-0.180
0.502					
PavedDrive_P	0.0223	0.030	0.732	0.464	-0.037
0.082					
PavedDrive_Y	0.0239	0.019	1.269	0.205	-0.013
0.061					
SaleType_CWD	0.1030	0.064	1.614	0.107	-0.022
0.228					
SaleType_Con	0.1262	0.084	1.508	0.132	-0.038
0.291					
SaleType_ConLD	0.1034	0.057	1.801	0.072	-0.009
0.216					
SaleType_ConLI	-0.0171	0.125	-0.137	0.891	-0.263
0.229					
SaleType_ConLw	-0.0153	0.075	-0.205	0.838	-0.162
0.131					
SaleType_New	0.0567	0.127	0.448	0.654	-0.192
0.305					
SaleType_Oth	0.2649	0.120	2.216	0.027	0.030
0.500					
SaleType_WD	-0.0055	0.025	-0.218	0.827	-0.055
0.044					
SaleCondition_AdjLand	0.0461	0.071	0.651	0.515	-0.093
0.185					
SaleCondition_Alloca	0.2273	0.061	3.721	0.000	0.107
0.347					
SaleCondition_Family	0.0261	0.033	0.802	0.423	-0.038
0.090					
SaleCondition_Normal	0.0686	0.017	4.120	0.000	0.036
0.101					
SaleCondition_Partial	0.0593	0.124	0.477	0.633	-0.184
0.303					

```
==============================================================================
Omnibus:                      261.441   Durbin-Watson:                   1.913
Prob(Omnibus):                  0.000   Jarque-Bera (JB):             4530.677
Skew:                          -0.695   Prob(JB):                         0.00
Kurtosis:                      13.221   Cond. No.                     1.04e+16
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
specified.
[2] The smallest eigenvalue is 3.78e-27. This might indicate that there are
strong multicollinearity problems or that the design matrix is singular.
```

9 Ridge-Lasso Model

[64]:
```python
# 1. Ridge Model
ridge = Ridge(alpha=1.0)  # Set the alpha parameter for Ridge regularization
ridge.fit(X_train, y_train)
ridge_pred = ridge.predict(X_train)

# 2. Lasso Model
lasso = Lasso(alpha=0.0001)
lasso.fit(X_train, y_train)
lasso_pred = lasso.predict(X_train)
# Evaluate the models using MSE and R²
ridge_mse = mean_squared_error(y_train, ridge_pred)
lasso_mse = mean_squared_error(y_train, lasso_pred)
lasso_mae = mean_absolute_error(y_train, lasso_pred)
ridge_mae = mean_absolute_error(y_train, ridge_pred)
ridge_r2 = r2_score(y_train, ridge_pred)
lasso_r2 = r2_score(y_train, lasso_pred)

# Output the results
print(f'Ridge MAE: {ridge_mae}')
print(f'Lasso MAE: {lasso_mae}')
print(f'Ridge MSE: {ridge_mse}, R²: {ridge_r2}')
print(f'Lasso MSE: {lasso_mse}, R²: {lasso_r2}')
```

```
Ridge MAE: 0.07911973928748224
Lasso MAE: 0.07419926954911113
Ridge MSE: 0.012771905353704703, R²: 0.9176025466675709
Lasso MSE: 0.011089882199498056, R²: 0.9284540539810514
```

```
C:\ProgramData\anaconda3\Lib\site-
packages\sklearn\linear_model\_coordinate_descent.py:697: ConvergenceWarning:
Objective did not converge. You might want to increase the number of iterations,
check the scale of the features or consider increasing regularisation. Duality
gap: 1.839e-01, tolerance: 1.584e-02
```

```
model = cd_fast.enet_coordinate_descent(
```

10 Random Forest Regressor

[66]:
```
# Initialize the Random Forest Regressor
rf_model = RandomForestRegressor(n_estimators=100, random_state=42)  # You can␣
↪tune the hyperparameters

# Fit the model on the training data
rf_model.fit(X_train, y_train)

# Make predictions (log scale)
rf_pred_log = rf_model.predict(X_test)

# Calculate metrics in the original scale
rf_mae = mean_absolute_error(y_test, rf_pred_log)
rf_mse = mean_squared_error(y_test, rf_pred_log)
rf_rmse = np.sqrt(rf_mse)
rf_r2 = r2_score(y_test, rf_pred_log)

# Print evaluation metrics
print('Random Forest Results:')
print('MAE:', rf_mae)
print('MSE:', rf_mse)
print('RMSE:', rf_rmse)
print('R²:', rf_r2)
```

```
Random Forest Results:
MAE: 0.09089577051997495
MSE: 0.018691285352520404
RMSE: 0.1367160756916333
R²: 0.8898204960230316
```

11 KNN Regressor

[68]:
```
# Define the KNN model
knn = KNeighborsRegressor()

# Initial settings for GridSearchCV
param_grid = {
    'n_neighbors': [3, 5, 7, 10],  # Number of neighbors
    'weights': ['uniform', 'distance'],  # Type of weighting
    'metric': ['minkowski', 'euclidean', 'manhattan']  # Types of distance␣
↪metrics
}
```

```
# Search for the best hyperparameters
grid_search = GridSearchCV(estimator=knn, param_grid=param_grid, cv=5,␣
 ↪scoring='neg_mean_squared_error', n_jobs=-1)
grid_search.fit(X_train, y_train)

# Best model with optimal settings
best_knn = grid_search.best_estimator_
print(f"Best Parameters for KNN: {grid_search.best_params_}")

# Prediction with the optimized model
knn_pred_log = best_knn.predict(X_test)

# Calculate evaluation metrics
knn_mae = mean_absolute_error(y_test, knn_pred_log)
knn_mse = mean_squared_error(y_test, knn_pred_log)
knn_rmse = np.sqrt(knn_mse)
knn_r2 = r2_score(y_test, knn_pred_log)

print(f"KNN Results:\nMAE: {knn_mae}\nMSE: {knn_mse}\nRMSE: {knn_rmse}\nR²:␣
 ↪{knn_r2}")
```

```
Best Parameters for KNN: {'metric': 'manhattan', 'n_neighbors': 7, 'weights':
'distance'}
KNN Results:
MAE: 0.10820146155474418
MSE: 0.02596909013547263
RMSE: 0.16114927904111961
R²: 0.8469200263173056
```

[69]:
```
# CatBoost Regressor

cb_model = CatBoostRegressor(loss_function='RMSE', logging_level='Silent')
cb_model.fit(X_train, y_train)
preds = cb_model.predict(X_test)

print('CastBoost Regression Results :')
print('MAE:', mean_absolute_error(y_test, preds))
print('MSE:', mean_squared_error(y_test, preds))
print('RMSE:', np.sqrt(mean_squared_error(y_test, preds)))
```

```
CastBoost Regression Results :
MAE: 0.07773776983114379
MSE: 0.014622267870046576
RMSE: 0.12092256972975135
```

[70]:
```
# 1. Splitting features and target from the training data
X_train = data_train.drop(columns=['SalePrice', 'Id'])  # Remove 'SalePrice'␣
 ↪and 'Id'
```

```
y_train = data_train['SalePrice']  # Define the target column

# 2. Splitting features from the test data
X_test = data_test.drop(columns=['Id'])  # Remove 'Id'

# 3. Ensuring column names match between training and test datasets
assert list(X_train.columns) == list(X_test.columns), "Columns in train and␣
 ↪test sets do not match!"

# 4. Defining the CatBoost Regressor model
cb_model = CatBoostRegressor(loss_function='RMSE', logging_level='Silent',␣
 ↪random_state=42)

# 5. Performing Cross-Validation on the training data
scores = cross_val_score(
    cb_model,
    X_train,
    y_train,
    scoring=make_scorer(mean_squared_error, squared=False),  # Use RMSE as the␣
 ↪evaluation metric
    cv=5  # Number of folds in Cross-Validation
)

# Displaying Cross-Validation results
print("Cross-Validation Results (RMSE for each fold):", scores)
print("Mean RMSE:", scores.mean())
print("Standard Deviation of RMSE:", scores.std())

# 6. Training the model on the entire training dataset
cb_model.fit(X_train, y_train)

# 7. Making predictions on the test data
test_predictions = cb_model.predict(X_test)
```

```
C:\ProgramData\anaconda3\Lib\site-packages\sklearn\metrics\_regression.py:492:
FutureWarning: 'squared' is deprecated in version 1.4 and will be removed in
1.6. To calculate the root mean squared error, use the
function'root_mean_squared_error'.
  warnings.warn(
C:\ProgramData\anaconda3\Lib\site-packages\sklearn\metrics\_regression.py:492:
FutureWarning: 'squared' is deprecated in version 1.4 and will be removed in
1.6. To calculate the root mean squared error, use the
function'root_mean_squared_error'.
  warnings.warn(
C:\ProgramData\anaconda3\Lib\site-packages\sklearn\metrics\_regression.py:492:
FutureWarning: 'squared' is deprecated in version 1.4 and will be removed in
1.6. To calculate the root mean squared error, use the
```

```
function'root_mean_squared_error'.
  warnings.warn(
C:\ProgramData\anaconda3\Lib\site-packages\sklearn\metrics\_regression.py:492:
FutureWarning: 'squared' is deprecated in version 1.4 and will be removed in
1.6. To calculate the root mean squared error, use the
function'root_mean_squared_error'.
  warnings.warn(
C:\ProgramData\anaconda3\Lib\site-packages\sklearn\metrics\_regression.py:492:
FutureWarning: 'squared' is deprecated in version 1.4 and will be removed in
1.6. To calculate the root mean squared error, use the
function'root_mean_squared_error'.
  warnings.warn(
```

```
Cross-Validation Results (RMSE for each fold): [0.1118078  0.13564016 0.12962723
0.11169712 0.1281989 ]
Mean RMSE: 0.12339424202339286
Standard Deviation of RMSE: 0.009828180956341859
```

12 Hyperparameter Optimization

[72]:
```
cb = CatBoostRegressor(loss_function='RMSE', logging_level='Silent')
cb_params = {"iterations": [300, 350],
                  "learning_rate": [0.1, 0.2],
                  "depth": [3, 5]}
cb_random = GridSearchCV(estimator=cb,param_grid=cb_params, n_jobs=-1)

cb_random.fit(X_train, y_train)
```

[72]:
```
GridSearchCV(estimator=<catboost.core.CatBoostRegressor object at
0x000002566C2832F0>,
             n_jobs=-1,
             param_grid={'depth': [3, 5], 'iterations': [300, 350],
                         'learning_rate': [0.1, 0.2]})
```

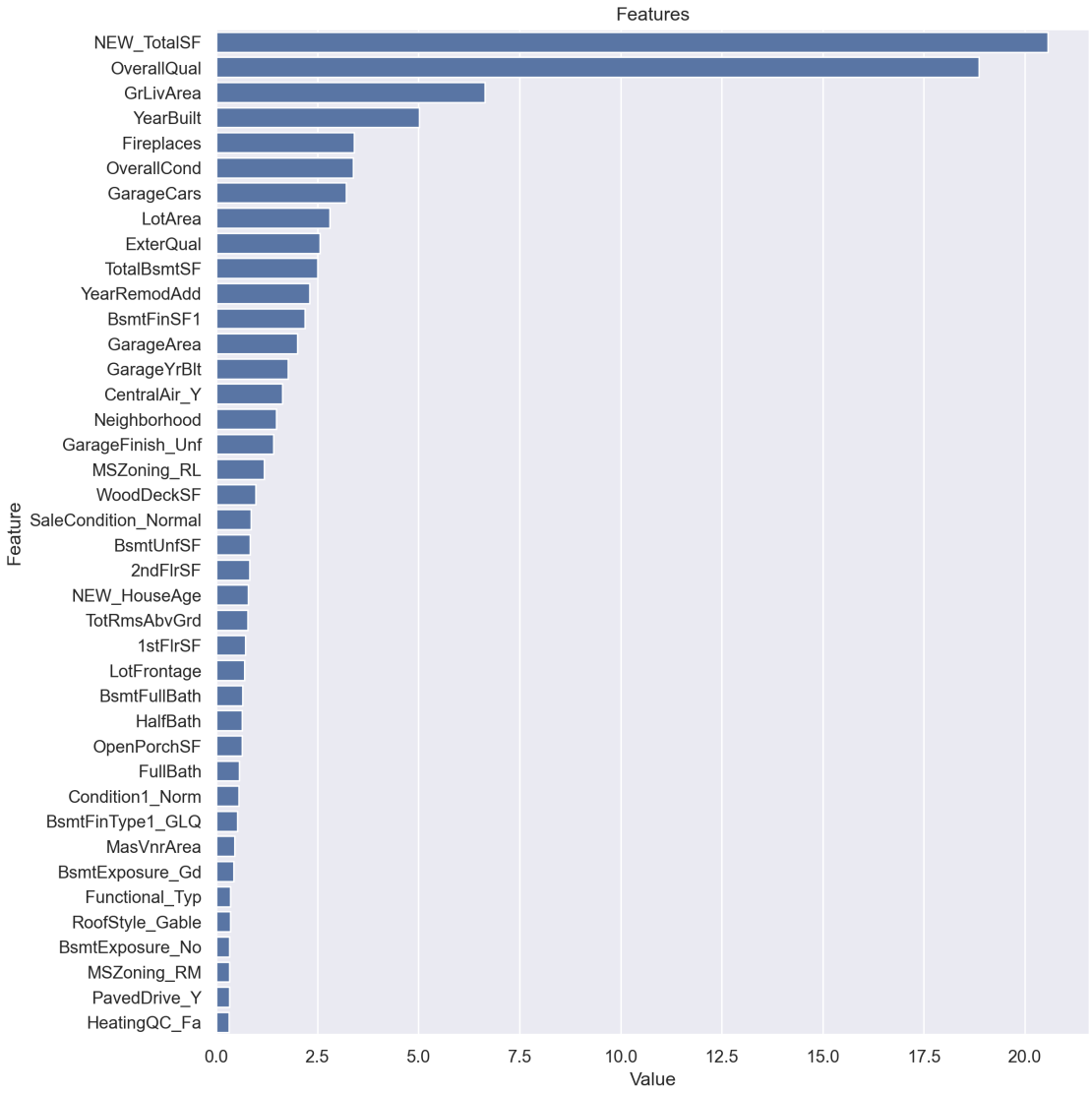
[73]:
```
print(cb_random.best_params_)

final_model = cb.set_params(**cb_random.best_params_, random_state=17).
 ↪fit(X_train, y_train)
```

```
{'depth': 5, 'iterations': 350, 'learning_rate': 0.1}
```

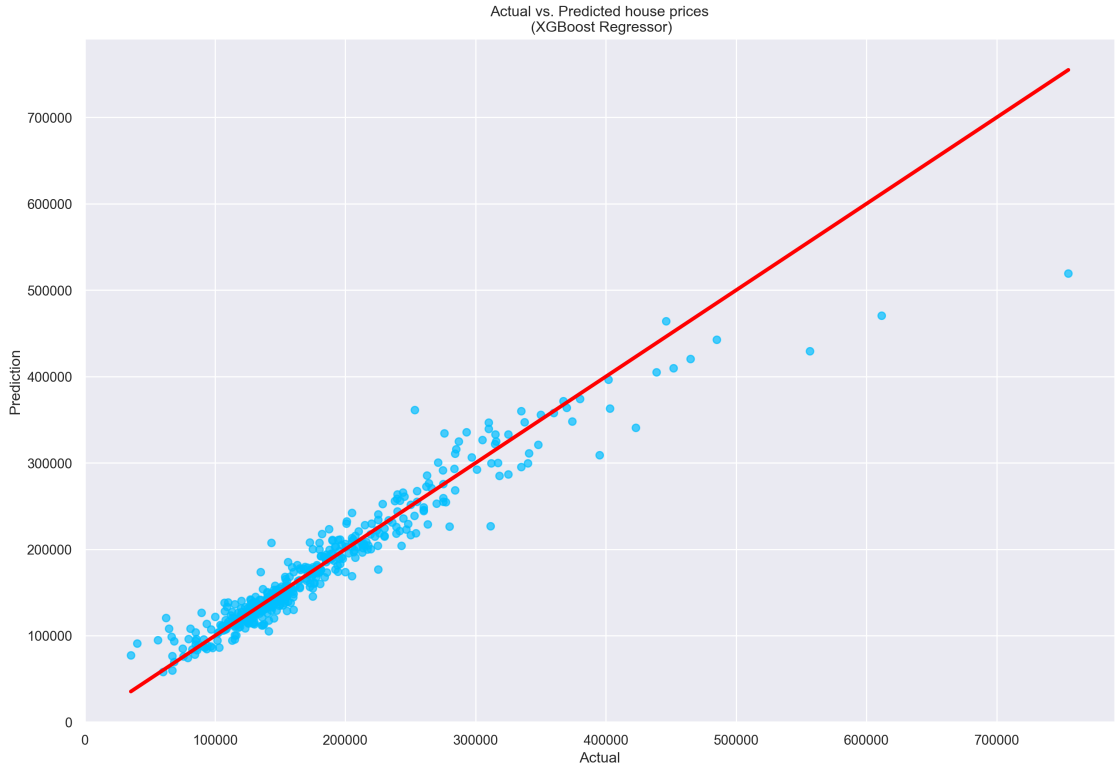
13 Feature Importance

[75]:
```
def plot_importance(model, features, dataframe, save=False):
    num = len(dataframe)
    feature_imp = pd.DataFrame({"Value": model.feature_importances_, "Feature":␣
 ↪features.columns})
```

```
    plt.figure(figsize=(10, 10))
    sns.set_theme(font_scale=1)
    sns.barplot(x="Value", y="Feature", data=feature_imp.
 ↪sort_values(by="Value", ascending=False)[0:40])
    plt.title("Features")
    plt.tight_layout()
    plt.show()
    if save:
        plt.savefig("importances.png")

plot_importance(final_model, X, data_train)
```



[76]:
```
#Actual vs. Prediction
plt.scatter(x = np.exp(y_test), y = np.exp(preds), c = 'deepskyblue', alpha = 0.
↪7)
plt.xlabel('Actual')
plt.ylabel('Prediction')
plt.title('Actual vs. Predicted house prices\n (XGBoost Regressor)')

#Add 45 degree line
xp = np.linspace(np.exp(y_test.min()), np.exp(y_test.max()), 100)
plt.plot(xp, xp, c = 'red', linewidth = 3)
plt.show()
```



[77]:
```
# Predict the prices
prediction_log = final_model.predict(data_test.drop(columns=["Id"]))

# Convert predictions to the original scale
prediction = np.expm1(prediction_log)

# Create the output DataFrame
submission_df = pd.DataFrame({
    "Id": data_test["Id"],
    "SalePrice": prediction
```

```
})
submission_df["Id"] = submission_df["Id"].astype(int)

# Save the output file as CSV
submission_df.to_csv("submission.csv", index=False)

print("Submission file 'submission.csv' has been saved.")
```

```
Submission file 'submission.csv' has been saved.
```

[ ]: