Feature_Engineering_Code

August 31, 2022

1 Import Libraries/modules

[1]:
```
import pandas as pd
import numpy as np

import warnings
warnings.simplefilter("ignore")

import matplotlib.pyplot as plt
import seaborn as sns
from matplotlib import style
%matplotlib inline

import scipy.stats as stats
from scipy.stats import chi2_contingency
```

2 load the dataset in panda dataframe

[2]:
```
#import the PEP1 dataset CSV into the panda dataframe#

df = pd.read_csv('PEP1.csv', low_memory=False)
```

3 Task 1. Understand the dataset

3.0.1 a. Identify the shape of the dataset

[3]:
```
df.shape
```

[3]:
```
(1460, 81)
```

In the given dataset there are '1460' rows and '81' columns.

[4]:
```
#Printing the name of the columns
df.columns
```

[4]:
```
Index(['Id', 'MSSubClass', 'MSZoning', 'LotFrontage', 'LotArea', 'Street',
       'Alley', 'LotShape', 'LandContour', 'Utilities', 'LotConfig',
```

```
       'LandSlope', 'Neighborhood', 'Condition1', 'Condition2', 'BldgType',
       'HouseStyle', 'OverallQual', 'OverallCond', 'YearBuilt', 'YearRemodAdd',
       'RoofStyle', 'RoofMatl', 'Exterior1st', 'Exterior2nd', 'MasVnrType',
       'MasVnrArea', 'ExterQual', 'ExterCond', 'Foundation', 'BsmtQual',
       'BsmtCond', 'BsmtExposure', 'BsmtFinType1', 'BsmtFinSF1',
       'BsmtFinType2', 'BsmtFinSF2', 'BsmtUnfSF', 'TotalBsmtSF', 'Heating',
       'HeatingQC', 'CentralAir', 'Electrical', '1stFlrSF', '2ndFlrSF',
       'LowQualFinSF', 'GrLivArea', 'BsmtFullBath', 'BsmtHalfBath', 'FullBath',
       'HalfBath', 'BedroomAbvGr', 'KitchenAbvGr', 'KitchenQual', 'TotRmsAbvGrd',
       'Functiol', 'Fireplaces', 'FireplaceQu', 'GarageType', 'GarageYrBlt',
       'GarageFinish', 'GarageCars', 'GarageArea', 'GarageQual', 'GarageCond',
       'PavedDrive', 'WoodDeckSF', 'OpenPorchSF', 'EnclosedPorch', '3SsnPorch',
       'ScreenPorch', 'PoolArea', 'PoolQC', 'Fence', 'MiscFeature', 'MiscVal',
       'MoSold', 'YrSold', 'SaleType', 'SaleCondition', 'SalePrice'],
      dtype='object')
```

[5]:
```
#check indexes
df.index
```

[5]:
```
RangeIndex(start=0, stop=1460, step=1)
```

[6]:
```
#Undrstand data set information
df.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     1452 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  KitchebvGr     1460 non-null   int64
53  KitchenQual    1460 non-null   object
54  TotRmsAbvGrd   1460 non-null   int64
55  Functiol       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
```

[7]:
```
# understand sample data
df.head()
```

[7]:
```
   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]
```

3.0.2 b. Identify variables with null values

[8]:
```
# method 1 - solution

''' isnull function along with sum function can Find columns with Null values␣
↪and their respective count
here in output non 0 value denotes the no of null values a column is having'''

df.isnull().sum()
```

[8]:
```
Id                 0
MSSubClass         0
MSZoning           0
LotFrontage      259
LotArea            0
                ...
MoSold             0
YrSold             0
SaleType           0
SaleCondition      0
SalePrice          0
Length: 81, dtype: int64
```

[9]:
```
# method 2
#Below code can also be used to find only those columns columns which have null␣
↪values,

print("Below are the columns having null data : \n", df.columns[df.isnull().
↪any()].tolist())
print("\n total no of columns : ", len(df.columns))
print("total no of columns having null data : ", len(df.columns[df.isnull().
↪any()]))
print("total no of not null data columns :", len(df.columns[df.notnull().
↪all()]))
```

```
Below are the columns having null data :
 ['LotFrontage', 'Alley', 'MasVnrType', 'MasVnrArea', 'BsmtQual', 'BsmtCond',
'BsmtExposure', 'BsmtFinType1', 'BsmtFinType2', 'Electrical', 'FireplaceQu',
'GarageType', 'GarageYrBlt', 'GarageFinish', 'GarageQual', 'GarageCond',
'PoolQC', 'Fence', 'MiscFeature']

 total no of columns :  81
total no of columns having null data :  19
total no of not null data columns : 62
```

- From above we can see that, there are total of 81 columns in the dataset
- out of which 19 has atleast 1 null record and 18 columns have no null records

3.0.3 c. Identify variables with unique values

[10]:
```
for i in df.columns:
    print (i , ":", df[i].unique())
    print (" _ "*40)
    print (" _ "*40)
```

```
Id : [   1    2    3 … 1458 1459 1460]
 _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _
_  _  _  _  _  _  _  _  _  _  _  _  _
 _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _
_  _  _  _  _  _  _  _  _  _  _  _  _
MSSubClass : [ 60  20  70  50 190  45  90 120  30  85  80 160  75 180  40]
 _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _
_  _  _  _  _  _  _  _  _  _  _  _  _
 _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _
_  _  _  _  _  _  _  _  _  _  _  _  _
MSZoning : ['RL' 'RM' 'C (all)' 'FV' 'RH']
 _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _
_  _  _  _  _  _  _  _  _  _  _  _  _
 _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _
_  _  _  _  _  _  _  _  _  _  _  _  _
LotFrontage : [ 65.  80.  68.  60.  84.  85.  75.  nan  51.  50.  70.  91.  72.
66.
 101.  57.  44. 110.  98.  47. 108. 112.  74. 115.  61.  48.  33.  52.
 100.  24.  89.  63.  76.  81.  95.  69.  21.  32.  78. 121. 122.  40.
 105.  73.  77.  64.  94.  34.  90.  55.  88.  82.  71. 120. 107.  92.
 134.  62.  86. 141.  97.  54.  41.  79. 174.  99.  67.  83.  43. 103.
  93.  30. 129. 140.  35.  37. 118.  87. 116. 150. 111.  49.  96.  59.
  36.  56. 102.  58.  38. 109. 130.  53. 137.  45. 106. 104.  42.  39.
 144. 114. 128. 149. 313. 168. 182. 138. 160. 152. 124. 153.  46.]
 _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _
_  _  _  _  _  _  _  _  _  _  _  _  _
 _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _
_  _  _  _  _  _  _  _  _  _  _  _  _
LotArea : [ 8450  9600 11250 … 17217 13175  9717]
 _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _
_  _  _  _  _  _  _  _  _  _  _  _  _
 _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _
_  _  _  _  _  _  _  _  _  _  _  _  _
Street : ['Pave' 'Grvl']
 _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _
_  _  _  _  _  _  _  _  _  _  _  _  _
 _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _
_  _  _  _  _  _  _  _  _  _  _  _  _
Alley : [nan 'Grvl' 'Pave']
 _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _  _
_  _  _  _  _  _  _  _  _  _  _  _  _
```

```
 - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - -
LotShape : ['Reg' 'IR1' 'IR2' 'IR3']
 - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - -
 - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - -
LandContour : ['Lvl' 'Bnk' 'Low' 'HLS']
 - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - -
 - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - -
Utilities : ['AllPub' 'NoSeWa']
 - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - -
 - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - -
LotConfig : ['Inside' 'FR2' 'Corner' 'CulDSac' 'FR3']
 - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - -
 - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - -
LandSlope : ['Gtl' 'Mod' 'Sev']
 - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - -
 - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - -
Neighborhood : ['CollgCr' 'Veenker' 'Crawfor' 'NoRidge' 'Mitchel' 'Somerst'
'NWAmes'
 'OldTown' 'BrkSide' 'Sawyer' 'NridgHt' 'mes' 'SawyerW' 'IDOTRR' 'MeadowV'
 'Edwards' 'Timber' 'Gilbert' 'StoneBr' 'ClearCr' 'NPkVill' 'Blmngtn'
 'BrDale' 'SWISU' 'Blueste']
 - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - -
 - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - -
Condition1 : ['Norm' 'Feedr' 'PosN' 'Artery' 'RRAe' 'RRNn' 'RRAn' 'PosA' 'RRNe']
 - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - -
 - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - -
Condition2 : ['Norm' 'Artery' 'RRNn' 'Feedr' 'PosN' 'PosA' 'RRAn' 'RRAe']
 - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - -
 - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - -
BldgType : ['1Fam' '2fmCon' 'Duplex' 'TwnhsE' 'Twnhs']
 - - - - - - - - - - - - - - - - - - - - - - - - - - -
```

```
- - - - - - - - - - - - -
 - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - -
HouseStyle : ['2Story' '1Story' '1.5Fin' '1.5Unf' 'SFoyer' 'SLvl' '2.5Unf'
'2.5Fin']
 - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - -
 - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - -
OverallQual : [ 7  6  8  5  9  4 10  3  1  2]
 - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - -
 - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - -
OverallCond : [5 8 6 7 4 2 3 9 1]
 - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - -
 - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - -
YearBuilt : [2003 1976 2001 1915 2000 1993 2004 1973 1931 1939 1965 2005 1962
2006
 1960 1929 1970 1967 1958 1930 2002 1968 2007 1951 1957 1927 1920 1966
 1959 1994 1954 1953 1955 1983 1975 1997 1934 1963 1981 1964 1999 1972
 1921 1945 1982 1998 1956 1948 1910 1995 1991 2009 1950 1961 1977 1985
 1979 1885 1919 1990 1969 1935 1988 1971 1952 1936 1923 1924 1984 1926
 1940 1941 1987 1986 2008 1908 1892 1916 1932 1918 1912 1947 1925 1900
 1980 1989 1992 1949 1880 1928 1978 1922 1996 2010 1946 1913 1937 1942
 1938 1974 1893 1914 1906 1890 1898 1904 1882 1875 1911 1917 1872 1905]
 - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - -
 - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - -
YearRemodAdd : [2003 1976 2002 1970 2000 1995 2005 1973 1950 1965 2006 1962 2007
1960
 2001 1967 2004 2008 1997 1959 1990 1955 1983 1980 1966 1963 1987 1964
 1972 1996 1998 1989 1953 1956 1968 1981 1992 2009 1982 1961 1993 1999
 1985 1979 1977 1969 1958 1991 1971 1952 1975 2010 1984 1986 1994 1988
 1954 1957 1951 1978 1974]
 - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - -
 - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - -
RoofStyle : ['Gable' 'Hip' 'Gambrel' 'Mansard' 'Flat' 'Shed']
 - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - -
 - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - -
RoofMatl : ['CompShg' 'WdShngl' 'Metal' 'WdShake' 'Membran' 'Tar&Grv' 'Roll'
```

```
 'ClyTile']
 - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - -
 - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - -
Exterior1st : ['VinylSd' 'MetalSd' 'Wd Sdng' 'HdBoard' 'BrkFace' 'WdShing'
'CemntBd'
 'Plywood' 'AsbShng' 'Stucco' 'BrkComm' 'AsphShn' 'Stone' 'ImStucc'
 'CBlock']
 - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - -
 - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - -
Exterior2nd : ['VinylSd' 'MetalSd' 'Wd Shng' 'HdBoard' 'Plywood' 'Wd Sdng'
'CmentBd'
 'BrkFace' 'Stucco' 'AsbShng' 'Brk Cmn' 'ImStucc' 'AsphShn' 'Stone'
 'Other' 'CBlock']
 - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - -
 - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - -
MasVnrType : ['BrkFace' 'None' 'Stone' 'BrkCmn' nan]
 - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - -
 - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - -
MasVnrArea : [1.960e+02 0.000e+00 1.620e+02 3.500e+02 1.860e+02 2.400e+02
2.860e+02
 3.060e+02 2.120e+02 1.800e+02 3.800e+02 2.810e+02 6.400e+02 2.000e+02
 2.460e+02 1.320e+02 6.500e+02 1.010e+02 4.120e+02 2.720e+02 4.560e+02
 1.031e+03 1.780e+02 5.730e+02 3.440e+02 2.870e+02 1.670e+02 1.115e+03
 4.000e+01 1.040e+02 5.760e+02 4.430e+02 4.680e+02 6.600e+01 2.200e+01
 2.840e+02 7.600e+01 2.030e+02 6.800e+01 1.830e+02 4.800e+01 2.800e+01
 3.360e+02 6.000e+02 7.680e+02 4.800e+02 2.200e+02 1.840e+02 1.129e+03
 1.160e+02 1.350e+02 2.660e+02 8.500e+01 3.090e+02 1.360e+02 2.880e+02
 7.000e+01 3.200e+02 5.000e+01 1.200e+02 4.360e+02 2.520e+02 8.400e+01
 6.640e+02 2.260e+02 3.000e+02 6.530e+02 1.120e+02 4.910e+02 2.680e+02
 7.480e+02 9.800e+01 2.750e+02 1.380e+02 2.050e+02 2.620e+02 1.280e+02
 2.600e+02 1.530e+02 6.400e+01 3.120e+02 1.600e+01 9.220e+02 1.420e+02
 2.900e+02 1.270e+02 5.060e+02 2.970e+02       nan 6.040e+02 2.540e+02
 3.600e+01 1.020e+02 4.720e+02 4.810e+02 1.080e+02 3.020e+02 1.720e+02
 3.990e+02 2.700e+02 4.600e+01 2.100e+02 1.740e+02 3.480e+02 3.150e+02
 2.990e+02 3.400e+02 1.660e+02 7.200e+01 3.100e+01 3.400e+01 2.380e+02
 1.600e+03 3.650e+02 5.600e+01 1.500e+02 2.780e+02 2.560e+02 2.250e+02
 3.700e+02 3.880e+02 1.750e+02 2.960e+02 1.460e+02 1.130e+02 1.760e+02
 6.160e+02 3.000e+01 1.060e+02 8.700e+02 3.620e+02 5.300e+02 5.000e+02
 5.100e+02 2.470e+02 3.050e+02 2.550e+02 1.250e+02 1.000e+02 4.320e+02
 1.260e+02 4.730e+02 7.400e+01 1.450e+02 2.320e+02 3.760e+02 4.200e+01
```

```
 1.610e+02 1.100e+02 1.800e+01 2.240e+02 2.480e+02 8.000e+01 3.040e+02
 2.150e+02 7.720e+02 4.350e+02 3.780e+02 5.620e+02 1.680e+02 8.900e+01
 2.850e+02 3.600e+02 9.400e+01 3.330e+02 9.210e+02 7.620e+02 5.940e+02
 2.190e+02 1.880e+02 4.790e+02 5.840e+02 1.820e+02 2.500e+02 2.920e+02
 2.450e+02 2.070e+02 8.200e+01 9.700e+01 3.350e+02 2.080e+02 4.200e+02
 1.700e+02 4.590e+02 2.800e+02 9.900e+01 1.920e+02 2.040e+02 2.330e+02
 1.560e+02 4.520e+02 5.130e+02 2.610e+02 1.640e+02 2.590e+02 2.090e+02
 2.630e+02 2.160e+02 3.510e+02 6.600e+02 3.810e+02 5.400e+01 5.280e+02
 2.580e+02 4.640e+02 5.700e+01 1.470e+02 1.170e+03 2.930e+02 6.300e+02
 4.660e+02 1.090e+02 4.100e+01 1.600e+02 2.890e+02 6.510e+02 1.690e+02
 9.500e+01 4.420e+02 2.020e+02 3.380e+02 8.940e+02 3.280e+02 6.730e+02
 6.030e+02 1.000e+00 3.750e+02 9.000e+01 3.800e+01 1.570e+02 1.100e+01
 1.400e+02 1.300e+02 1.480e+02 8.600e+02 4.240e+02 1.047e+03 2.430e+02
 8.160e+02 3.870e+02 2.230e+02 1.580e+02 1.370e+02 1.150e+02 1.890e+02
 2.740e+02 1.170e+02 6.000e+01 1.220e+02 9.200e+01 4.150e+02 7.600e+02
 2.700e+01 7.500e+01 3.610e+02 1.050e+02 3.420e+02 2.980e+02 5.410e+02
 2.360e+02 1.440e+02 4.230e+02 4.400e+01 1.510e+02 9.750e+02 4.500e+02
 2.300e+02 5.710e+02 2.400e+01 5.300e+01 2.060e+02 1.400e+01 3.240e+02
 2.950e+02 3.960e+02 6.700e+01 1.540e+02 4.250e+02 4.500e+01 1.378e+03
 3.370e+02 1.490e+02 1.430e+02 5.100e+01 1.710e+02 2.340e+02 6.300e+01
 7.660e+02 3.200e+01 8.100e+01 1.630e+02 5.540e+02 2.180e+02 6.320e+02
 1.140e+02 5.670e+02 3.590e+02 4.510e+02 6.210e+02 7.880e+02 8.600e+01
 7.960e+02 3.910e+02 2.280e+02 8.800e+01 1.650e+02 4.280e+02 4.100e+02
 5.640e+02 3.680e+02 3.180e+02 5.790e+02 6.500e+01 7.050e+02 4.080e+02
 2.440e+02 1.230e+02 3.660e+02 7.310e+02 4.480e+02 2.940e+02 3.100e+02
 2.370e+02 4.260e+02 9.600e+01 4.380e+02 1.940e+02 1.190e+02]
 - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - -
 - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - -
ExterQual : ['Gd' 'TA' 'Ex' 'Fa']
 - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - -
 - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - -
ExterCond : ['TA' 'Gd' 'Fa' 'Po' 'Ex']
 - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - -
 - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - -
Foundation : ['PConc' 'CBlock' 'BrkTil' 'Wood' 'Slab' 'Stone']
 - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - -
 - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - -
BsmtQual : ['Gd' 'TA' 'Ex' nan 'Fa']
 - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - -
```

```
 - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - -
BsmtCond : ['TA' 'Gd' nan 'Fa' 'Po']
 - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - -
 - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - -
BsmtExposure : ['No' 'Gd' 'Mn' 'Av' nan]
 - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - -
 - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - -
BsmtFinType1 : ['GLQ' 'ALQ' 'Unf' 'Rec' 'BLQ' nan 'LwQ']
 - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - -
 - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - -
BsmtFinSF1 : [ 706  978  486  216  655  732 1369  859    0  851  906  998  737
733
  578  646  504  840  188  234 1218 1277 1018 1153 1213  731  643  967
  747  280  179  456 1351   24  763  182  104 1810  384  490  649  632
  941  739  912 1013  603 1880  565  320  462  228  336  448 1201   33
  588  600  713 1046  648  310 1162  520  108  569 1200  224  705  444
  250  984   35  774  419  170 1470  938  570  300  120  116  512  567
  445  695  405 1005  668  821  432 1300  507  679 1332  209  680  716
 1400  416  429  222   57  660 1016  370  351  379 1288  360  639  495
  288 1398  477  831 1904  436  352  611 1086  297  626  560  390  566
 1126 1036 1088  641  617  662  312 1065  787  468   36  822  378  946
  341   16  550  524   56  321  842  689  625  358  402   94 1078  329
  929  697 1573  270  922  503 1334  361  672  506  714  403  751  226
  620  546  392  421  905  904  430  614  450  210  292  795 1285  819
  420  841  281  894 1464  700  262 1274  518 1236  425  692  987  970
   28  256 1619   40  846 1124  720  828 1249  810  213  585  129  498
 1270  573 1410 1082  236  388  334  874  956  773  399  162  712  609
  371  540   72  623  428  350  298 1445  218  985  631 1280  241  690
  266  777  812  786 1116  789 1056   50 1128  775 1309 1246  986  616
 1518  664  387  471  385  365 1767  133  642  247  331  742 1606  916
  185  544  553  326  778  386  426  368  459 1350 1196  630  994  168
 1261 1567  299  897  607  836  515  374 1231  111  356  400  698 1247
  257  380   27  141  991  650  521 1436 2260  719  377 1330  348 1219
  783  969  673 1358 1260  144  584  554 1002  619  180  559  308  866
  895  637  604 1302 1071  290  728    2 1441  943  231  414  349  442
  328  594  816 1460 1324 1338  685 1422 1283   81  454  903  605  990
  206  150  457   48  871   41  674  624  480 1154  738  493 1121  282
  500  131 1696  806 1361  920 1721  187 1138  988  193  551  767 1186
  892  311  827  543 1003 1059  239  945   20 1455  965  980  863  533
 1084 1173  523 1148  191 1234  375  808  724  152 1180  252  832  575
  919  439  381  438  549  612 1163  437  394 1416  422  762  975 1097
```

```
  251  686  656  568  539  862  197  516  663  608 1636  784  249 1040
  483  196  572  338  330  156 1390  513  460  659  364  564  306  505
  932  750   64  633 1170  899  902 1238  528 1024 1064  285 2188  465
  322  860  599  354   63  223  301  443  489  284  294  814  165  552
  833  464  936  772 1440  748  982  398  562  484  417  699  696  896
  556 1106  651  867  854 1646 1074  536 1172  915  595 1237  273  684
  324 1165  138 1513  317 1012 1022  509  900 1085 1104  240  383  644
  397  740  837  220  586  535  410   75  824  592 1039  510  423  661
  248  704  412 1032  219  708  415 1004  353  702  369  622  212  645
  852 1150 1258  275  176  296  538 1157  492 1198 1387  522  658 1216
 1480 2096 1159  440 1456  883  547  788  485  340 1220  427  344  756
 1540  666  803 1000  885 1386  319  534  125 1314  602  192  593  804
 1053  532 1158 1014  194  167  776 5644  694 1572  746 1406  925  482
  189  765   80 1443  259  735  734 1447  548  315 1282  408  309  203
  865  204  790 1320  769 1070  264  759 1373  976  781   25 1110  404
  580  678  958 1336 1079   49  830]
 - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - -
 - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - -
BsmtFinType2 : ['Unf' 'BLQ' nan 'ALQ' 'Rec' 'LwQ' 'GLQ']
 - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - -
 - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - -
BsmtFinSF2 : [   0   32  668  486   93  491  506  712  362   41  169  869  150
670
   28 1080  181  768  215  374  208  441  184  279  306  180  580  690
  692  228  125 1063  620  175  820 1474  264  479  147  232  380  544
  294  258  121  391  531  344  539  713  210  311 1120  165  532   96
  495  174 1127  139  202  645  123  551  219  606  612  480  182  132
  336  468  287   35  499  723  119   40  117  239   80  472   64 1057
  127  630  128  377  764  345 1085  435  823  500  290  324  634  411
  841 1061  466  396  354  149  193  273  465  400  682  557  230  106
  791  240  547  469  177  108  600  492  211  168 1031  438  375  144
   81  906  608  276  661   68  173  972  105  420  546  334  352  872
  110  627  163 1029]
 - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - -
 - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - -
BsmtUnfSF : [ 150  284  434  540  490   64  317  216  952  140  134  177  175
1494
  520  832  426    0  468  525 1158  637 1777  200  204 1566  180  486
  207  649 1228 1234  380  408 1117 1097   84  326  445  383  167  465
 1296   83 1632  736  192  612  816   32  935  321  860 1410  148  217
  530 1346  576  318 1143 1035  440  747  701  343  280  404  840  724
  295 1768  448   36 1530 1065  384 1288  684 1013  402  635  163  168
```

176	370	350	381	410	741	1226	1053	641	516	793	1139	550	905
104	310	252	1125	203	728	732	510	899	1362	30	958	556	413
479	297	658	262	891	1304	519	1907	336	107	432	403	811	396
970	506	884	400	896	253	409	93	1200	572	774	769	1335	340
882	779	112	470	294	1686	360	441	354	700	725	320	554	312
968	504	1107	577	660	99	871	474	289	600	755	625	1121	276
186	1424	1140	375	92	305	1176	78	274	311	710	686	457	1232
1498	1010	160	2336	630	638	162	70	1357	1194	773	483	235	125
1390	594	1694	488	357	626	916	1020	1367	798	452	392	975	361
270	602	1482	680	606	88	342	212	1095	96	628	1560	744	2121
768	386	1468	1145	244	698	1079	570	476	131	184	143	1092	324
1541	1470	536	319	599	622	179	292	286	80	712	291	153	1088
1249	166	906	604	100	818	844	596	210	1603	115	103	673	726
995	967	721	1656	972	460	208	191	438	1869	371	624	552	322
598	268	130	484	785	733	953	847	333	1580	411	982	808	1293
939	784	595	229	114	522	735	405	117	961	1286	672	1141	806
165	1064	1063	245	1276	892	1008	499	1316	463	242	444	281	35
356	988	580	651	619	544	387	901	926	135	648	75	788	1307
1078	1258	273	1436	557	930	780	813	878	122	248	588	524	288
389	424	1375	1626	406	298	2153	417	739	225	611	237	290	264
238	363	190	1969	697	414	316	466	420	254	960	397	1191	548
50	178	1368	169	748	689	1264	467	605	1257	551	678	707	880
378	223	578	969	379	765	149	912	620	1709	132	993	197	1374
90	195	706	1163	367	1122	1515	55	1497	450	846	23	390	861
285	1050	331	2042	1237	113	742	924	512	119	314	308	293	537
126	427	309	914	173	1774	823	485	1116	978	636	564	108	1184
796	366	300	542	645	664	756	247	776	849	1392	38	1406	111
545	121	2046	161	261	567	1195	874	1342	151	989	1073	927	219
224	526	1164	761	461	876	859	171	718	138	941	464	250	72
508	1584	415	82	948	893	864	1349	76	487	652	1240	801	279
1030	348	234	1198	740	89	586	323	1836	480	456	1935	338	1594
102	374	1413	491	1129	255	1496	650	1926	154	999	1734	124	1417
15	834	1649	936	778	1489	442	1434	352	458	1221	1099	416	1800
227	907	528	189	1273	563	372	702	1090	435	198	1372	174	1638
894	299	105	676	1120	431	218	110	795	1098	1043	481	666	142
447	783	1670	277	412	794	239	662	1072	717	546	430	422	188
266	1181	1753	964	1450	1905	1480	772	1032	220	187	29	495	640
193	196	720	918	1428	77	1266	1128	692	770	750	1442	1007	501
691	1550	1680	1330	1710	746	814	515	571	359	355	301	668	920
1055	1420	1752	304	1302	833	133	549	705	722	799	462	429	810
155	170	230	1459	1082	758	1290	1074	251	172	868	797	365	418
730	533	671	1012	1528	1005	1373	500	762	752	399	1042	40	26
932	278	459	568	1502	543	574	977	449	983	731	120	538	831
994	341	879	815	1212	866	1630	328	141	364	1380	81	303	940
764	1048	334	1689	690	792	585	473	246	1045	1405	201	14	841
1104	241	925	2002	74	661	708	1152	256	804	812	1085	344	425
1616	976	496	349	971	1393	1622	1352	1795	1017	1588	428	803	693
858	1284	1203	1652	39	539	1217	257	715	616	240	315	1351	1026

```
 1571  156   61   95  482 1094   60  862  221  791  398  777  503  734
  709 1252  656 1319 1422  560 1573  589  877  136]
```

- -

- -

```
TotalBsmtSF : [ 856 1262  920  756 1145  796 1686 1107  952  991 1040 1175  912
1494
 1253  832 1004    0 1114 1029 1158  637 1777 1060 1566  900 1704 1484
  520  649 1228 1234 1398 1561 1117 1097 1297 1057 1088 1350  840  938
 1150 1752 1434 1656  736  955  794  816 1842  384 1425  970  860 1410
  780  530 1370  576 1143 1947 1453  747 1304 2223  845 1086  462  672
 1768  440  896 1237 1563 1065 1288  684  612 1013  990 1235  876 1214
  824  680 1588  960  458  950 1610  741 1226 1053  641  789  793 1844
  994 1264 1809 1028  729 1092 1125 1673  728  732 1080 1199 1362 1078
  660 1008  924  992 1063 1267 1461 1907  928  864 1734  910 1490 1728
  715  884  969 1710  825 1602 1200  572  774 1392 1232 1572 1541  882
 1149  644 1617 1582  720 1064 1606 1202 1151 1052 2216  968  504 1188
 1593  853  725 1431  855 1726 1360  755 1713 1121 1196  617  848 1424
 1140 1100 1157 1212  689 1070 1436  686  798 1248 1498 1010  713 2392
  630 1203  483 1373 1194 1462  894 1414  996 1694  735  540  626  948
 1845 1020 1367 1444 1573 1302 1314  975 1604  963 1482  506  926 1422
  802  740 1095 1385 1152 1240 1560 2121 1160  807 1468 1575  625  858
  698 1079  768  795 1416 1003  702 1165 1470 2000  700  319  861 1896
  697  972 2136  716 1347 1372 1249 1136 1502 1162  710 1719 1383  844
  596 1056 3206 1358  943 1499 1922 1536 1208 1215  967  721 1684  536
  958 1478  764 1848 1869  616  624  940 1142 1062  888  883 1394 1099
 1268  953  744  608  847  683  870 1580 1856  982 1026 1293  939  784
 1256  658 1041 1682  804  788 1144  961 1260 1310 1141  806 1281 1034
 1276 1340 1344  988  651 1518  907  901  765  799  648 3094 1440 1258
  915 1517  930  813 1533  872 1242 1364  588  709  560 1375 1277 1626
 1488  808  547 1976 2153 1705 1833 1792 1216  999 1113 1073  954  264
 1269  190 3200  866 1501  777 1218 1368 1084 2006 1244 3138 1379 1257
 1452  528 2035  611  707  880 1051 1581 1838 1650  723  654 1204 1069
 1709  998  993 1374 1389 1163 1122 1496  846  372 1164 1050 2042 1868
 1437  742  770 1722 1814 1430 1058  908  600  965 1032 1299 1120  936
  783 1822 1522  980 1116  978 1156  636 1554 1386  811 1520 1952 1766
  981 1094 2109  525  776 1486 1629 1138 2077 1406 1021 1408  738 1477
 2046  923 1291 1195 1190  874  551 1419 2444 1210  927 1112 1391 1800
  360 1473 1643 1324  270  859  718 1176 1311  971 1742  941 1698 1584
 1595  868 1153  893 1349 1337 1720 1479 1030 1318 1252  983 1860  836
 1935 1614  761 1413  956  712  650  773 1926  731 1417 1024  849 1442
 1649 1568  778 1489 2078 1454 1516 1067 1559 1127 1390 1273  918 1763
 1090 1054 1039 1148 1002 1638  105  676 1184 1109  892 2217 1505 1059
  951 2330 1670 1623 1017 1105 1001  546  480 1134 1104 1272 1316 1126
 1181 1753  964 1466  925 1905 1500  585 1632  819 1616 1161  828  945
  979  561  696 1330  817 1098 1428  673 1241  944 1225 1266 1128  485
 1930 1396  916  822  750 1700 1007 1187  691 1574 1680 1346  985 1657
```

```
  602 1022 1082  810 1504 1220 1132 1565 1338 1654 1620 1055  800 1306
 1475 2524 1992 1193  973  854  662 1103 1154  942 1048  727  690 1096
 1459 1251 1247 1074 1271  290  655 1463 1836  803  833  408  533 1012
 1552 1005 1530  974 1567 1006 1042 1298  704  932 1219 1296 1198  959
 1261 1598 1683  818 1600 2396 1624  831 1224  663  879  815 1630 2158
  931 1660  559 1300 1702 1075 1361 1106 1476 1689 2076  792 2110 1405
 1192  746 1986  841 2002 1332  935 1019  661 1309 1328 1085 6110 1246
  771  976 1652 1278 1902 1274 1393 1622 1352  420 1795  544 1510  911
  693 1284 1732 2033  570 1980  814  873  757 1108 2633 1571  984 1205
  714 1746 1525  482 1356  862  839 1286 1485 1594  622  791  708 1223
  913  656 1319 1932  539 1221 1542]
 - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - -
 - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - -
Heating : ['GasA' 'GasW' 'Grav' 'Wall' 'OthW' 'Floor']
 - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - -
 - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - -
HeatingQC : ['Ex' 'Gd' 'TA' 'Fa' 'Po']
 - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - -
 - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - -
CentralAir : ['Y' 'N']
 - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - -
 - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - -
Electrical : ['SBrkr' 'FuseF' 'FuseA' 'FuseP' 'Mix' nan]
 - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - -
 - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - -
1stFlrSF : [ 856 1262  920  961 1145  796 1694 1107 1022 1077 1040 1182  912
1494
 1253  854 1004 1296 1114 1339 1158 1108 1795 1060 1600  900 1704  520
  649 1228 1234 1700 1561 1132 1097 1297 1057 1152 1324 1328  884  938
 1150 1752 1518 1656  736  955  794  816 1842 1360 1425  983  860 1426
  780  581 1370  902 1143 2207 1479  747 1304 2223  845  885 1086  840
  526  952 1072 1768  682 1337 1563 1065  804 1301  684  612 1013  990
 1235  964 1260  905  680 1588  960  835 1225 1610  977 1535 1226 1053
 1047  789  997 1844 1216  774 1282 2259 1436  729 1092 1125 1699  728
  988  772 1080 1199 1586  958  660 1327 1721 1682 1214 1959  928  864
 1734  910 1501 1728  970  875  896  969 1710 1252 1200  572  991 1392
 1232 1572 1541  882 1149  808 1867 1707 1064 1362 1651 2158 1164 2234
  968  769  901 1340  936 1217 1224 1593 1549  725 1431  855 1726  929
```

```
 1713 1121 1279  865  848  720 1442 1696 1100 1180 1212  932  689 1236
  810 1137 1248 1498 1010  811 2392  630  483 1555 1194 1490  894 1414
 1014  798 1566  866  889  626 1222 1872  908 1375 1444 1306 1625 1302
 1314 1005 1604  963 1382 1482  926  764 1422  802 1052  778 1113 1095
 1363 1632 1560 2121 1156 1175 1468 1575  625 1085  858  698 1079 1148
 1644 1003  975 1041 1336 1210 1675 2000 1122 1035  861 1944  697  972
  793 2036  832  716 1153 1088 1372 1472 1249 1136 1553 1163 1898  803
 1719 1383 1445  596 1056 1629 1358  943 1619 1922 1536 1621 1215  993
  841 1684  536 1478 1848 1869 1453  616 1192 1167 1142 1352  495  790
  672 1394 1268 1287  953 1120  752 1319  847  904  914 1580 1856 1007
 1026  939  784 1269  658 1742  788  735 1144  876 1112 1288 1310 1165
  806 1620 1166 1071 1050 1276 1028  756 1344 1602 1470 1196  707  907
 1208 1412  765  827  734  694 2402 1440 1128 1258  933 1689 1888  956
  679  813 1533  888  786 1242  624 1663  833  979  575  849 1277 1634
 1502 1161 1976 1652 1493 2069 1718 1131 1850 1792  916  999 1073 1484
 1766  886 3228 1133  899 1801 1218 1368 2020 1378 1244 3138 1266 1476
  605 2515 1509  751  334  820  880 1159 1601 1838 1680  767  664 1377
  915  768  825 1069 1717 1126 1006 1048  897 1557 1389  996 1134 1496
  846  576  877 1320  703 1429 2042 1521  989 2028  838 1473  779  770
  924 1826 1402 1647 1058  927  600 1186 1940 1029 1032 1299 1054  807
 1828 1548  980 1012 1116 1520 1350 1089 1554 1411  800 1567  981 1094
 1051  822  755  909 2113  525  851 1486 1686 1181 2097 1454 1465 1679
 1437  738 1839  792 2046  923 1291 1668 1195 1190  874  551 1419 2444
 1238 1067 1391 1800 1264  372 1824  859 1576 1178 1325  971 1698 1776
 1616 1146  948 1349 1464 1720 1038  742  757 1506 1836 1690 1220 1117
 1973 1204 1614 1430 1110 1342  966  976 1062 1127 1285  773 1966 1428
 1075 1309 1044  686 1661 1008  944 1489 2084 1434 1160  941 1516 1559
 1099 1701 1307 1456  918 1779  702 1512 1039 1002 1646 1547 1036  676
 1184 1462 1155 1090 1187  954  892 1709 1712  872 2217 1505 1068  951
 2364 1670 1063 1636 1020 1105 1015 1001  546  480 1229 1272 1316 1617
 1098 1788 1466  925 1905 1500 1207 1188 1381  965 1168  561  696 1542
  824  783  673  869 1241 1118 1407  750  691 1574 1504  985 1657 1664
 1082 2898 1687 1654 1055 1803 1532 2524 1733 1992 1771  930 1526 1091
 1523 1364 1130 1096 1338 1103 1154  799  893  829 1240 1459 1251 1247
 1390  438  950  887 1021 1552  812 1530  974  986 1042 1298 1811 1265
 1640 1432  959 1831 1261 1170 2129  818 1124 2411  949 1624  831 1622
  842  663  879  815 1630 1074 2196 1283 1660 1318 1211 2136 1138 1702
 1507 1361 1024 1141 1173 2076 1140 1034 2110 1405  760 1987 1104  713
 2018 1968 1332  935 1357  661 1724 1573 1582 1659 4692 1246  753 1203
 1294 1902 1274 1787 1061  708 1584 1334  693 1284 1172 2156 2053  992
 1078 1980 1281  814 2633 1571  984  754 2117  998 1416 1746 1525 1221
  741 1569 1223  962 1537 1932 1423  913 1578 2073 1256]
 - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - -
 - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - -
2ndFlrSF : [ 854    0  866  756 1053  566  983  752 1142 1218  668 1320  631
716
```

```
  676  860 1519  530  808  977 1330  833  765  462  213  548  960  670
 1116  876  612 1031  881  790  755  592  939  520  639  656 1414  884
  729 1523  728  351  688  941 1032  848  836  475  739 1151  448  896
  524 1194  956 1070 1096  467  547  551  880  703  901  720  316 1518
  704 1178  754  601 1360  929  445  564  882  920  518  817 1257  741
  672 1306  504 1304 1100  730  689  591  888 1020  828  700  842 1286
  864  829 1092  709  844 1106  596  807  625  649  698  840  780  568
  795  648  975  702 1242 1818 1121  371  804  325  809 1200  871 1274
 1347 1332 1177 1080  695  167  915  576  605  862  495  403  838  517
 1427  784  711  468 1081  886  793  665  858  874  526  590  406 1157
  299  936  438 1098  766 1101 1028 1017 1254  378 1160  682  110  600
  678  834  384  512  930  868  224 1103  560  811  878  574  910  620
  687  546  902 1000  846 1067  914  660 1538 1015 1237  611  707  527
 1288  832  806 1182 1040  439  717  511 1129 1370  636  533  745  584
  812  684  595  988  800  677  573 1066  778  661 1440  872  788  843
  713  567  651  762  482  738  586  679  644  900  887 1872 1281  472
 1312  319  978 1093  473  664 1540 1276  441  348 1060  714  744 1203
  783 1097  734  767 1589  742  686 1128 1111 1174  787 1072 1088 1063
  545  966  623  432  581  540  769 1051  761  779  514  455 1426  785
  521  252  813 1120 1037 1169 1001 1215  928 1140 1243  571 1196 1038
  561  979  701  332  368  883 1336 1141  634  912  798  985  826  831
  750  456  602  855  336  408  980  998 1168 1208  797  850  898 1054
  895  954  772 1230  727  454  370  628  304  582 1122 1134  885  640
  580 1112  653  220  240 1362  534  539  650  918  933  712 1796  971
 1175  743  523 1216 2065  272  685  776  630  984  875  913  464 1039
 1259  940  892  725  924  764  925 1479  192  589  992  903  430  748
  587  994  950 1323  732 1357  557 1296  390 1185  873 1611  457  796
  908  550  989  932  358 1392  349  691 1349  768  208  622  857  556
 1044  708  626  904  510 1104  830  981  870  694 1152]
 - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - -
 - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - -
LowQualFinSF : [  0 360 513 234 528 572 144 392 371 390 420 473 156 515  80  53
232 481
 120 514 397 479 205 384]
 - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - -
 - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - -
GrLivArea : [1710 1262 1786 1717 2198 1362 1694 2090 1774 1077 1040 2324  912
1494
 1253  854 1004 1296 1114 1339 2376 1108 1795 1060 1600  900 1704  520
 1317 1228 1234 1700 1561 2452 1097 1297 1057 1152 1324 1328  884  938
 1150 1752 2149 1656 1452  955 1470 1176  816 1842 1360 1425 1739 1720
 2945  780 1158 1111 1370 2034 2473 2207 1479  747 2287 2223  845 1718
 1086 1605  988  952 1285 1768 1230 2142 1337 1563 1065 1474 2417 1560
 1224 1526  990 1235  964 2291 1588  960  835 1225 1610 1732 1535 1226
```

1818	1992	1047	789	1517	1844	1855	1430	2696	2259	2320	1458	1092	1125
3222	1456	1123	1080	1199	1586	754	958	840	1348	1053	2157	2054	1327
1721	1682	1214	1959	1852	1764	864	1734	1385	1501	1728	1709	875	2035
1344	969	1993	1252	1200	1096	1968	1947	2462	1232	2668	1541	882	1616
1355	1867	2161	1707	1382	1767	1651	2158	2060	1920	2234	968	1525	1802
1340	2082	3608	1217	1593	2727	1431	1726	3112	2229	1713	1121	1279	1310
848	1284	1442	1696	1100	2062	1212	1392	1236	1436	1954	1248	1498	2267
1552	2392	1302	2520	987	1555	1194	2794	894	1960	1414	1744	1487	1566
866	1440	2110	1872	1928	1375	1668	2144	1306	1625	1640	1314	1604	1792
2574	1316	764	1422	1511	2192	778	1113	1939	1363	2270	1632	1548	2121
2022	1982	1468	1575	1250	858	1396	1919	1716	2263	1644	1003	1558	1950
1743	1336	3493	2000	2243	1406	861	1944	972	1118	2036	1641	1432	2353
2646	1472	2596	2468	2730	1163	2978	803	1719	1383	2134	1192	1056	1629
1358	1638	1922	1536	1621	1215	1908	841	1684	1112	1577	1478	1626	2728
1869	1453	720	1595	1167	1142	1352	1924	1505	1574	1394	1268	1287	1664
752	1319	904	914	2466	1856	1800	1691	1301	1797	784	1953	1269	1184
2332	1367	1961	788	1034	1144	1812	1550	1288	672	1572	1620	1639	1680
2172	2078	1276	1028	2097	1400	2624	1134	1602	2630	1196	1389	907	1208
1412	1198	1365	630	1661	694	2402	1573	1258	1689	1888	1886	1376	1183
813	1533	1756	1590	1242	1663	1666	1203	1935	1135	1660	1277	1634	1502
1969	1072	1976	1652	970	1493	2643	1131	1850	1826	1216	999	1073	1484
2414	1304	1578	886	3228	1820	899	1218	1801	1322	1911	1378	1041	1368
2020	2119	2344	1796	2080	1294	1244	4676	2398	1266	928	2713	605	2515
1509	827	334	1347	1724	1159	1601	1838	2285	767	1496	2183	1635	768
825	2094	1069	1126	2046	1048	1446	1557	996	1674	2295	1647	2504	2132
943	1692	1109	1477	1320	1429	2042	2775	2028	838	860	1473	935	1582
2296	924	1402	1556	1904	1915	1986	2008	3194	1029	2153	1032	1120	1054
832	1828	2262	2614	980	1512	1790	1116	1520	1350	1750	1554	1411	3395
800	1387	796	1567	1518	1929	2704	1766	981	1094	1839	1665	1510	1469
2113	1486	2448	1181	1936	2380	1679	1437	1180	1476	1369	1136	1441	792
923	1291	1761	1102	1419	4316	2519	1539	1137	616	1148	1391	1164	2576
1824	729	1178	2554	2418	971	1742	1698	1776	1146	2031	948	1349	1464
2715	2256	2640	1529	1140	2098	1026	1471	1386	2531	1547	2365	1506	1714
1836	3279	1220	1117	1973	1204	1614	1603	1110	1342	2084	901	2087	1145
1062	2013	1895	1564	773	3140	1688	2822	1128	1428	1576	2138	1309	1044
1008	1052	936	1733	1489	1434	2126	1223	1829	1516	1067	1559	1099	1482
1165	1416	1701	1775	2358	1646	1445	1779	1481	2654	1426	1039	1372	1002
1949	910	2610	2224	1155	1090	2230	892	1712	1393	2217	1683	1068	951
2240	2364	1670	902	1063	1636	2057	2274	1015	2002	480	1229	2127	2200
1617	1686	2374	1978	1788	2236	1466	925	1905	1500	2069	1971	1962	2403
1381	965	1958	2872	1894	1308	1098	1095	918	2019	869	1241	2612	2290
1940	2030	1851	1050	944	691	1504	985	1657	1522	1271	1022	1082	1132
2898	1264	3082	1654	954	1803	2329	2524	2868	1771	930	1977	1989	1523
1364	2184	1991	1338	2337	1103	1154	2260	1571	1611	2521	893	1240	1740
1459	1251	1247	1088	438	950	2622	2021	1690	1658	1964	833	1012	698
1005	1530	1981	974	2210	986	1020	1868	2828	1006	1298	932	1811	1265
1580	1876	1671	2108	3627	1261	3086	2345	1343	1124	2514	4476	1130	1221
1699	1624	1804	1622	1863	1630	1074	2196	1283	1845	1902	1211	1846	2136

```
 1490 1138 1933 1702 1507 2620 1190 1188 1784 1948 1141 1173 2076 1553
 2058 1405  874 2167 1987 1166 1675 1889 2018 3447 1524 1357 1395 2447
 1659 1970 2372 5642 1246 1983 2526 1708 1122 1274 2810 2599 2112 1787
 1923  708  774 2792 1334  693 1861  872 2169 1913 2156 2634 3238 1865
 1078 1980 2601 1738 1475 1374 2633  790 2117 1762 2784 1746 1584 1912
 2482 1687 1513 1608 2093 1840 1848 1569 2450 2201  804 1537 1932 1725
 2555 2007  913 1346 2073 2340 1256]
 - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - -
 - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - -
BsmtFullBath : [1 0 2 3]
 - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - -
 - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - -
BsmtHalfBath : [0 1 2]
 - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - -
 - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - -
FullBath : [2 1 3 0]
 - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - -
 - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - -
HalfBath : [1 0 2]
 - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - -
 - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - -
BedroomAbvGr : [3 4 1 2 0 5 6 8]
 - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - -
 - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - -
KitchebvGr : [1 2 3 0]
 - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - -
 - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - -
KitchenQual : ['Gd' 'TA' 'Ex' 'Fa']
 - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - -
 - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - -
TotRmsAbvGrd : [ 8  6  7  9  5 11  4 10 12  3  2 14]
 - - - - - - - - - - - - - - - - - - - - - - - - - - -
```

```
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
Functiol : ['Typ' 'Min1' 'Maj1' 'Min2' 'Mod' 'Maj2' 'Sev']
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
Fireplaces : [0 1 2 3]
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
FireplaceQu : [nan 'TA' 'Gd' 'Fa' 'Ex' 'Po']
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
GarageType : ['Attchd' 'Detchd' 'BuiltIn' 'CarPort' nan 'Basment' '2Types']
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
GarageYrBlt : [2003. 1976. 2001. 1998. 2000. 1993. 2004. 1973. 1931. 1939. 1965.
2005.
 1962. 2006. 1960. 1991. 1970. 1967. 1958. 1930. 2002. 1968. 2007. 2008.
 1957. 1920. 1966. 1959. 1995. 1954. 1953.   nan 1983. 1977. 1997. 1985.
 1963. 1981. 1964. 1999. 1935. 1990. 1945. 1987. 1989. 1915. 1956. 1948.
 1974. 2009. 1950. 1961. 1921. 1900. 1979. 1951. 1969. 1936. 1975. 1971.
 1923. 1984. 1926. 1955. 1986. 1988. 1916. 1932. 1972. 1918. 1980. 1924.
 1996. 1940. 1949. 1994. 1910. 1978. 1982. 1992. 1925. 1941. 2010. 1927.
 1947. 1937. 1942. 1938. 1952. 1928. 1922. 1934. 1906. 1914. 1946. 1908.
 1929. 1933.]
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
GarageFinish : ['RFn' 'Unf' 'Fin' nan]
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
GarageCars : [2 3 1 0 4]
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
GarageArea : [ 548  460  608  642  836  480  636  484  468  205  384  736  352
```

```
840
  576  516  294  853  280  534  572  270  890  772  319  240  250  271
  447  556  691  672  498  246    0  440  308  504  300  670  826  386
  388  528  894  565  641  288  645  852  558  220  667  360  427  490
  379  297  283  509  405  758  461  400  462  420  432  506  684  472
  366  476  410  740  648  273  546  325  792  450  180  430  594  390
  540  264  530  435  453  750  487  624  471  318  766  660  470  720
  577  380  434  866  495  564  312  625  680  678  726  532  216  303
  789  511  616  521  451 1166  252  497  682  666  786  795  856  473
  398  500  349  454  644  299  210  431  438  675  968  721  336  810
  494  457  818  463  604  389  538  520  309  429  673  884  868  492
  413  924 1053  439  671  338  573  732  505  575  626  898  529  685
  281  539  418  588  282  375  683  843  552  870  888  746  708  513
 1025  656  872  292  441  189  880  676  301  474  706  617  445  200
  592  566  514  296  244  610  834  639  501  846  560  596  600  373
  947  350  396  864  304  784  696  569  628  550  493  578  198  422
  228  526  525  908  499  508  694  874  164  402  515  286  603  900
  583  889  858  502  392  403  527  765  367  426  615  871  570  406
  590  612  650 1390  275  452  842  816  621  544  486  230  261  531
  393  774  749  364  627  260  256  478  442  562  512  839  330  711
 1134  416  779  702  567  832  326  551  606  739  408  475  704  983
  768  632  541  320  800  831  554  878  752  614  481  496  423  841
  895  412  865  630  605  602  618  444  397  455  409  820 1020  598
  857  595  433  776 1220  458  613  456  436  812  686  611  425  343
  479  619  902  574  523  414  738  354  483  327  756  690  284  833
  601  533  522  788  555  689  796  808  510  255  424  305  368  824
  328  160  437  665  290  912  905  542  716  586  467  582 1248 1043
  254  712  719  862  928  782  466  714 1052  225  234  324  306  830
  807  358  186  693  482  813  995  757 1356  459  701  322  315  668
  404  543  954  850  477  276  518 1014  753 1418  213  844  860  748
  248  287  825  647  342  770  663  377  804  936  722  208  662  754
  622  620  370 1069  372  923  192]
 - - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - -
 - - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - -
GarageQual : ['TA' 'Fa' 'Gd' nan 'Ex' 'Po']
 - - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - -
 - - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - -
GarageCond : ['TA' 'Fa' nan 'Gd' 'Po' 'Ex']
 - - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - -
 - - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - -
PavedDrive : ['Y' 'N' 'P']
 - - - - - - - - - - - - - - - - - - - - - - - - - - - -
```

```
- - - - - - - - - - - - -
 - - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - -
WoodDeckSF : [  0 298 192  40 255 235  90 147 140 160  48 240 171 100 406 222
288  49
 203 113 392 145 196 168 112 106 857 115 120  12 576 301 144 300  74 127
 232 158 352 182 180 166 224  80 367  53 188 105  24  98 276 200 409 239
 400 476 178 574 237 210 441 116 280 104  87 132 238 149 355  60 139 108
 351 209 216 248 143 365 370  58 197 263 123 138 333 250 292  95 262  81
 289 124 172 110 208 468 256 302 190 340 233 184 201 142 122 155 670 135
 495 536 306  64 364 353  66 159 146 296 125  44 215 264  88  89  96 414
 519 206 141 260 324 156 220  38 261 126  85 466 270  78 169 320 268  72
 349  42  35 326 382 161 179 103 253 148 335 176 390 328 312 185 269 195
  57 236 517 304 198 426  28 316 322 307 257 219 416 344 380  68 114 327
 165 187 181  92 228 245 503 315 241 303 133 403  36  52 265 207 150 290
 486 278  70 418 234  26 342  97 272 121 243 511 154 164 173 384 202  56
 321  86 194 421 305 117 550 509 153 394 371  63 252 136 186 170 474 214
 199 728 436  55 431 448 361 362 162 229 439 379 356  84 635 325  33 212
 314 242 294  30 128  45 177 227 218 309 404 500 668 402 283 183 175 586
 295  32 366 736]
 - - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - -
 - - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - -
OpenPorchSF : [ 61   0  42  35  84  30  57 204   4  21  33 213 112 102 154 159
110  90
  56  32  50 258  54  65  38  47  64  52 138 104  82  43 146  75  72  70
  49  11  36 151  29  94 101 199  99 234 162  63  68  46  45 122 184 120
  20  24 130 205 108  80  66  48  25  96 111 106  40 114   8 136 132  62
 228  60 238 260  27  74  16 198  26  83  34  55  22  98 172 119 208 105
 140 168  28  39 148  12  51 150 117 250  10  81  44 144 175 195 128  76
  17  59 214 121  53 231 134 192 123  78 187  85 133 176 113 137 125 523
 100 285  88 406 155  73 182 502 274 158 142 243 235 312 124 267 265  87
 288  23 152 341 116 160 174 247 291  18 170 156 166 129 418 240  77 364
 188 207  67  69 131 191  41 118 252 189 282 135  95 224 169 319  58  93
 244 185 200  92 180 263 304 229 103 211 287 292 241 547  91  86 262 210
 141  15 126 236]
 - - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - -
 - - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - - -
EnclosedPorch : [  0 272 228 205 176  87 172 102  37 144  64 114 202 128 156  44
77 192
 140 180 183  39 184  40 552  30 126  96  60 150 120 112 252  52 224 234
 244 268 137  24 108 294 177 218 242  91 160 130 169 105  34 248 236  32
  80 115 291 116 158 210  36 200  84 148 136 240  54 100 189 293 164 216
 239  67  90  56 129  98 143  70 386 154 185 134 196 264 275 230 254  68
 194 318  48  94 138 226 174  19 170 220 214 280 190 330 208 145 259  81
```

```
  42 123 162 286 168  20 301 198 221 212  50  99]
 - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - -
 - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - -
3SsnPorch : [  0 320 407 130 180 168 140 508 238 245 196 144 182 162  23 216  96
153
 290 304]
 - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - -
 - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - -
ScreenPorch : [  0 176 198 291 252  99 184 168 130 142 192 410 224 266 170 154
153 144
 128 259 160 271 234 374 185 182  90 396 140 276 180 161 145 200 122  95
 120  60 126 189 260 147 385 287 156 100 216 210 197 204 225 152 175 312
 222 265 322 190 233  63  53 143 273 288 263  80 163 116 480 178 440 155
 220 119 165  40]
 - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - -
 - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - -
PoolArea : [  0 512 648 576 555 480 519 738]
 - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - -
 - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - -
PoolQC : [nan 'Ex' 'Fa' 'Gd']
 - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - -
 - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - -
Fence : [nan 'MnPrv' 'GdWo' 'GdPrv' 'MnWw']
 - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - -
 - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - -
MiscFeature : [nan 'Shed' 'Gar2' 'Othr' 'TenC']
 - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - -
 - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - -
MiscVal : [    0   700   350   500   400   480   450 15500  1200   800  2000
600
  3500  1300    54   620   560  1400  8300  1150  2500]
 - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - -
 - - - - - - - - - - - - - - - - - - - - - - - - - - -
```

```
- - - - - - - - - - - - -
MoSold : [ 2  5  9 12 10  8 11  4  1  7  3  6]
 - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - -
 - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - -
YrSold : [2008 2007 2006 2009 2010]
 - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - -
 - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - -
SaleType : ['WD' 'New' 'COD' 'ConLD' 'ConLI' 'CWD' 'ConLw' 'Con' 'Oth']
 - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - -
 - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - -
SaleCondition : ['Normal' 'Abnorml' 'Partial' 'AdjLand' 'Alloca' 'Family']
 - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - -
 - - - - - - - - - - - - - - - - - - - - - - - - - - -
- - - - - - - - - - - - -
SalePrice : [208500 181500 223500 140000 250000 143000 307000 200000 129900
118000
 129500 345000 144000 279500 157000 132000 149000  90000 159000 139000
 325300 139400 230000 154000 256300 134800 306000 207500  68500  40000
 149350 179900 165500 277500 309000 145000 153000 109000  82000 160000
 170000 130250 141000 319900 239686 249700 113000 127000 177000 114500
 110000 385000 130000 180500 172500 196500 438780 124900 158000 101000
 202500 219500 317000 180000 226000  80000 225000 244000 185000 144900
 107400  91000 135750 136500 193500 153500 245000 126500 168500 260000
 174000 164500  85000 123600 109900  98600 163500 133900 204750 214000
  94750  83000 128950 205000 178000 118964 198900 169500 100000 115000
 190000 136900 383970 217000 259500 176000 155000 320000 163990 136000
 153900 181000  84500 128000  87000 150000 150750 220000 171000 231500
 166000 204000 125000 105000 222500 122000 372402 235000  79000 109500
 269500 254900 162500 412500 103200 152000 127500 325624 183500 228000
 128500 215000 239000 163000 184000 243000 211000 501837 200100 120000
 475000 173000 135000 153337 286000 315000 192000 148500 311872 104000
 274900 171500 112000 143900 277000  98000 186000 252678 156000 161750
 134450 210000 107000 311500 167240 204900  97000 386250 290000 106000
 192500 148000 403000  94500 128200 216500  89500 185500 194500 318000
 262500 110500 241500 137000  76500 276000 151000  73000 175500 179500
 120500 266000 124500 201000 415298 228500 244600 179200 164700  88000
 153575 233230 135900 131000 167000 142500 175000 158500 267000 149900
 295000 305900  82500 360000 165600 119900 375000 188500 270000 187500
 342643 354000 301000 126175 242000 324000 145250 214500  78000 119000
 284000 207000 228950 377426 202900  87500 140200 151500 157500 437154
 318061  95000 105900 177500 134000 280000 198500 147000 165000 162000
```

```
172400 134432 123000  61000 340000 394432 179000 187750 213500  76000
240000  81000 191000 426000 106500 129000  67000 241000 245500 164990
108000 258000 168000 339750  60000 222000 181134 149500 126000 142000
206300 275000 109008 195400  85400  79900 122500 212000 116000  90350
555000 162900 199900 119500 188000 256000 161000 263435  62383 188700
124000 178740 146500 187000 440000 251000 132500 208900 380000 297000
 89471 326000 374000 164000  86000 133000 172785  91300  34900 430000
226700 289000 208300 164900 202665  96500 402861 265000 234000 106250
184750 315750 446261 200624 107500  39300 111250 272000 248000 213250
179665 229000 263000 112500 255500 121500 268000 325000 316600 135960
142600 224500 118500 146000 131500 181900 253293 369900  79500 185900
451950 138000 319000 114504 194201 217500 221000 359100 313000 261500
 75500 137500 183200 105500 314813 305000 165150 139900 209500  93000
264561 274000 370878 143250  98300 205950 350000 145500  97500 197900
402000 423000 230500 173500 103600 257500 372500 159434 285000 227875
148800 392000 194700 755000 335000 108480 141500  89000 123500 138500
196000 312500 361919 213000  55000 302000 254000 179540  52000 102776
189000 130500 159500 341000 103000 236500 131400  93500 239900 299800
236000 265979 260400 275500 158900 179400 215200 337000 264132 216837
538000 134900 102000 395000 221500 175900 187100 161500 233000 107900
160200 146800 269790 143500 485000 582933 227680 135500 159950 144500
 55993 157900 224900 271000 224000 183000 139500 232600 147400 237000
139950 174900 133500 189950 250580 248900 169000 200500  66500 303477
132250 328900 122900 154500 118858 142953 611657 125500 255000 154300
173733  75000  35311 238000 176500 145900 169990 193000 117500 184900
253000 239799 244400 150900 197500 172000 116500 214900 178900  37900
 99500 182000 167500  85500 178400 336000 159895 255900 117000 395192
195000 197000 348000 173900 337500 121600 206000 232000 136905 119200
227000 203000 213490 194000 287000 293077 310000 119750  84000 315500
262280 278000 139600 556581  84900 176485 200141 185850 328000 167900
151400  91500 138800 155900  83500 252000  92900 176432 274725 134500
184100 133700 118400 212900 163900 259000 239500  94000 424870 174500
116900 201800 218000 235128 108959 233170 245350 625000 171900 154900
392500 745000 186700 104900 262000 219210 116050 271900 229456  80500
137900 367294 101800 138887 265900 248328 465000 186500 169900 171750
294000 165400 301500  99900 128900 183900 378500 381000 185750  68400
150500 281000 333168 206900 295493 111000 156500  72500  52500 155835
108500 283463 410000 156932 144152 216000 274300 466500  58500 237500
377500 246578 281213 137450 193879 282922 257000 223000 274970 182900
192140 143750  64500 394617 149700 149300 121000 179600  92000 287090
266500 142125 147500]
 _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _
 _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ _ _
```

4 Task 2. Generate a separate dataset for numerical and categorical variables

[11]:
```
numerical_df= df.select_dtypes(include=[np.number])
categorical_df=df.select_dtypes(exclude=[np.number])
```

[12]:
```
print ("Numerical columns : \n ",numerical_df.columns)
print ("\n Categorical columns \n :",categorical_df.columns)
```

```
Numerical columns :
  Index(['Id', 'MSSubClass', 'LotFrontage', 'LotArea', 'OverallQual',
       'OverallCond', 'YearBuilt', 'YearRemodAdd', 'MasVnrArea', 'BsmtFinSF1',
       'BsmtFinSF2', 'BsmtUnfSF', 'TotalBsmtSF', '1stFlrSF', '2ndFlrSF',
       'LowQualFinSF', 'GrLivArea', 'BsmtFullBath', 'BsmtHalfBath', 'FullBath',
       'HalfBath', 'BedroomAbvGr', 'KitchebvGr', 'TotRmsAbvGrd', 'Fireplaces',
       'GarageYrBlt', 'GarageCars', 'GarageArea', 'WoodDeckSF', 'OpenPorchSF',
       'EnclosedPorch', '3SsnPorch', 'ScreenPorch', 'PoolArea', 'MiscVal',
       'MoSold', 'YrSold', 'SalePrice'],
      dtype='object')

 Categorical columns
 : Index(['MSZoning', 'Street', 'Alley', 'LotShape', 'LandContour', 'Utilities',
       'LotConfig', 'LandSlope', 'Neighborhood', 'Condition1', 'Condition2',
       'BldgType', 'HouseStyle', 'RoofStyle', 'RoofMatl', 'Exterior1st',
       'Exterior2nd', 'MasVnrType', 'ExterQual', 'ExterCond', 'Foundation',
       'BsmtQual', 'BsmtCond', 'BsmtExposure', 'BsmtFinType1', 'BsmtFinType2',
       'Heating', 'HeatingQC', 'CentralAir', 'Electrical', 'KitchenQual',
       'Functiol', 'FireplaceQu', 'GarageType', 'GarageFinish', 'GarageQual',
       'GarageCond', 'PavedDrive', 'PoolQC', 'Fence', 'MiscFeature',
       'SaleType', 'SaleCondition'],
      dtype='object')
```

5 Task 3. EDA of numerical variables:

Below after identifying the NUMERICAL SIGNIFICANT VARIABLES, there is no missing values in these significant columns hence even though there are 3 columns 'LotFrontage' , 'MasVnrArea', 'GarageYrBlt' where the values are missing but since these columns are not significant hence directly columns are removed.

5.0.1 a. Missing value treatment

[13]:
```
numerical_df.isnull().sum()
```

[13]:
```
Id               0
MSSubClass       0
LotFrontage    259
LotArea          0
```

```
OverallQual       0
OverallCond       0
YearBuilt         0
YearRemodAdd      0
MasVnrArea        8
BsmtFinSF1        0
BsmtFinSF2        0
BsmtUnfSF         0
TotalBsmtSF       0
1stFlrSF          0
2ndFlrSF          0
LowQualFinSF      0
GrLivArea         0
BsmtFullBath      0
BsmtHalfBath      0
FullBath          0
HalfBath          0
BedroomAbvGr      0
KitchebvGr        0
TotRmsAbvGrd      0
Fireplaces        0
GarageYrBlt      81
GarageCars        0
GarageArea        0
WoodDeckSF        0
OpenPorchSF       0
EnclosedPorch     0
3SsnPorch         0
ScreenPorch       0
PoolArea          0
MiscVal           0
MoSold            0
YrSold            0
SalePrice         0
dtype: int64
```

a.1. Find and drop columns having all Null data

[14]:
```
#Find columns with all nul records
print("Below columns does not have any data/ (all rows are null) : \n \n",␣
 ↪numerical_df.columns[numerical_df.isnull().all()])

#remove above columns having all null records
numerical_df.dropna(axis= 1 , how='all', inplace=True)

#print the shape of the dataframe
print("\n After treatment shape of the dataframe is : \n",numerical_df.shape)
```

```
Below columns does not have any data/ (all rows are null) :

 Index([], dtype='object')

 After treatment shape of the dataframe is :
 (1460, 38)
```

* None of the columns having all null records, hence no columns will be removed.

a.2. Find and drop columns having most of the NULL data

[15]:
```
#taken 85% but this value is uaully discussed with business before removing of␣
↪the columns

#Find column having mostly the NULL data
most_Null_data = [i for i in numerical_df.columns if numerical_df[i].isnull().
↪sum() > 0.85*len(numerical_df)]

print("Column having mostly the NULL data :\n \n", most_Null_data)

#drop columns having mostly the NULL data
numerical_df.drop(columns = most_Null_data, inplace=True)


#print the shape of the dataframe
print("\n After treatment shape of the dataframe is : \n",numerical_df.shape)
```

```
Column having mostly the NULL data :

 []

 After treatment shape of the dataframe is :
 (1460, 38)
```

* Above shows that variables having missing values are not mostly Null hence will not remove these columns.

5.0.2 b. Identify the skewness and distribution

Skewness Skewness is a statistical term and it is a way to estimate or measure the shape of a distribution. It is an important statistical methodology that is used to estimate the asymmetrical behavior rather than computing frequency distribution. Skewness can be two types:

Symmetrical: A distribution can be called symmetric if it appears the same from the left and right from the center point. Asymmetrical: A distribution can be called asymmetric if it doesn't appear the same from the left and right from the center point. Distribution on the basis of skewness value:

- Skewness = 0: Then normally distributed.
- Skewness > 0: Then more weight in the left tail of the distribution.
- Skewness < 0: Then more weight in the right tail of the distribution.

Kurtosis: It is also a statistical term and an important characteristic of frequency distribution. It determines whether a distribution is heavy-tailed in respect of the normal distribution. It provides information about the shape of a frequency distribution.

- kurtosis for normal distribution is equal to 3.
- For a distribution having kurtosis < 3: It is called playkurtic.
- For a distribution having kurtosis > 3, It is called leptokurtic and it signifies that it tries to produce more outliers rather than the normal distribution.

[16]:
```python
#For reading numerical dataset
numerical_df = pd.DataFrame(df)

numerical_df.describe()
```

[16]:
```
                Id    MSSubClass  LotFrontage        LotArea  OverallQual  \
count  1460.000000  1460.000000  1201.000000    1460.000000  1460.000000
mean    730.500000    56.897260    70.049958   10516.828082     6.099315
std     421.610009    42.300571    24.284752    9981.264932     1.382997
min       1.000000    20.000000    21.000000    1300.000000     1.000000
25%     365.750000    20.000000    59.000000    7553.500000     5.000000
50%     730.500000    50.000000    69.000000    9478.500000     6.000000
75%    1095.250000    70.000000    80.000000   11601.500000     7.000000
max    1460.000000   190.000000   313.000000  215245.000000    10.000000

       OverallCond    YearBuilt  YearRemodAdd   MasVnrArea   BsmtFinSF1  …  \
count  1460.000000  1460.000000   1460.000000  1452.000000  1460.000000  …
mean      5.575342  1971.267808   1984.865753   103.685262   443.639726  …
std       1.112799    30.202904     20.645407   181.066207   456.098091  …
min       1.000000  1872.000000   1950.000000     0.000000     0.000000  …
25%       5.000000  1954.000000   1967.000000     0.000000     0.000000  …
50%       5.000000  1973.000000   1994.000000     0.000000   383.500000  …
75%       6.000000  2000.000000   2004.000000   166.000000   712.250000  …
max       9.000000  2010.000000   2010.000000  1600.000000  5644.000000  …

        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 38 columns]
```

[17]:
```
#Checking the skewness of entire data
numerical_df.skew(axis = 0, skipna = True)
```

[17]:
```
Id                0.000000
MSSubClass        1.407657
LotFrontage       2.163569
LotArea          12.207688
OverallQual       0.216944
OverallCond       0.693067
YearBuilt        -0.613461
YearRemodAdd     -0.503562
MasVnrArea        2.669084
BsmtFinSF1        1.685503
BsmtFinSF2        4.255261
BsmtUnfSF         0.920268
TotalBsmtSF       1.524255
1stFlrSF          1.376757
2ndFlrSF          0.813030
LowQualFinSF      9.011341
GrLivArea         1.366560
BsmtFullBath      0.596067
BsmtHalfBath      4.103403
FullBath          0.036562
HalfBath          0.675897
BedroomAbvGr      0.211790
KitchebvGr        4.488397
TotRmsAbvGrd      0.676341
Fireplaces        0.649565
GarageYrBlt      -0.649415
GarageCars       -0.342549
GarageArea        0.179981
WoodDeckSF        1.541376
OpenPorchSF       2.364342
EnclosedPorch     3.089872
3SsnPorch        10.304342
ScreenPorch       4.122214
PoolArea         14.828374
MiscVal          24.476794
MoSold            0.212053
```

```
YrSold           0.096269
SalePrice        1.882876
dtype: float64
```

[18]:
```python
#Checking the kurtosis of entire data

numerical_df.kurtosis(axis=0)
```

[18]:
```
Id                -1.200000
MSSubClass         1.580188
LotFrontage       17.452867
LotArea          203.243271
OverallQual        0.096293
OverallCond        1.106413
YearBuilt         -0.439552
YearRemodAdd      -1.272245
MasVnrArea        10.082417
BsmtFinSF1        11.118236
BsmtFinSF2        20.113338
BsmtUnfSF          0.474994
TotalBsmtSF       13.250483
1stFlrSF           5.745841
2ndFlrSF          -0.553464
LowQualFinSF      83.234817
GrLivArea          4.895121
BsmtFullBath      -0.839098
BsmtHalfBath      16.396642
FullBath          -0.857043
HalfBath          -1.076927
BedroomAbvGr       2.230875
KitchebvGr        21.532404
TotRmsAbvGrd       0.880762
Fireplaces        -0.217237
GarageYrBlt       -0.418341
GarageCars         0.220998
GarageArea         0.917067
WoodDeckSF         2.992951
OpenPorchSF        8.490336
EnclosedPorch     10.430766
3SsnPorch        123.662379
ScreenPorch       18.439068
PoolArea         223.268499
MiscVal          701.003342
MoSold            -0.404109
YrSold            -1.190601
SalePrice          6.536282
dtype: float64
```

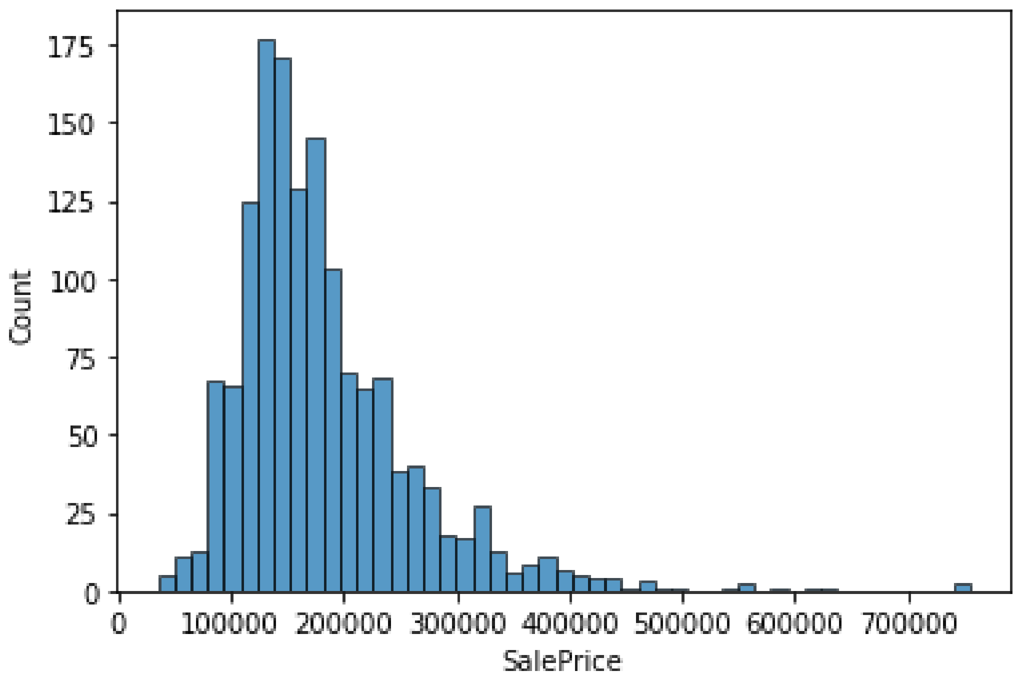
Lets check the skewness and kurtosis of SalePrice, since SalePrice is the columns which we are intrested in

[19]:
```
#Checking skewness and kurtosis of SalePrice
print("Skewness: %f" % numerical_df['SalePrice'].skew())
print("kurtosis: %f" % numerical_df['SalePrice'].kurtosis())
```

```
Skewness: 1.882876
kurtosis: 6.536282
```

[20]:
```
sns.histplot(numerical_df['SalePrice'])
```

[20]: <AxesSubplot:xlabel='SalePrice', ylabel='Count'>



Conclusion: The pair plot, skewness values and kurtosis of the variables and histgram of the Target column 'Salesprice' shows that the dataset is not normally distributed. Therefore, we need to normalize it.

Next step is to find the correlation and identifying the factors that affect the SalePrice.

5.0.3 c. Identify significant variables using a correlation matrix

[21]:
```
#correlation
corr = numerical_df.corr()
corr.style.background_gradient(cmap='coolwarm', axis=0 )
```

[21]: <pandas.io.formats.style.Styler at 0x1cd9eaaa3b0>

[22]:
```
#print the column names which have threshold  0.5 or  -0.5 with 'SalePrice':

var= corr['SalePrice'][(corr['SalePrice'] >=0.5) | (corr['SalePrice'] <= -0.5)].
↪index.tolist()
var
```

[22]:
```
['OverallQual',
 'YearBuilt',
 'YearRemodAdd',
 'TotalBsmtSF',
 '1stFlrSF',
 'GrLivArea',
 'FullBath',
 'TotRmsAbvGrd',
 'GarageCars',
 'GarageArea',
 'SalePrice']
```

Conclusion: Now we can identify thevariable which is highly corelated(Postive / Negative) to the column 'SalePrice'.

We can see the below 2 columns have threshold 0.7 or -0.7 with 'SalePrice' :

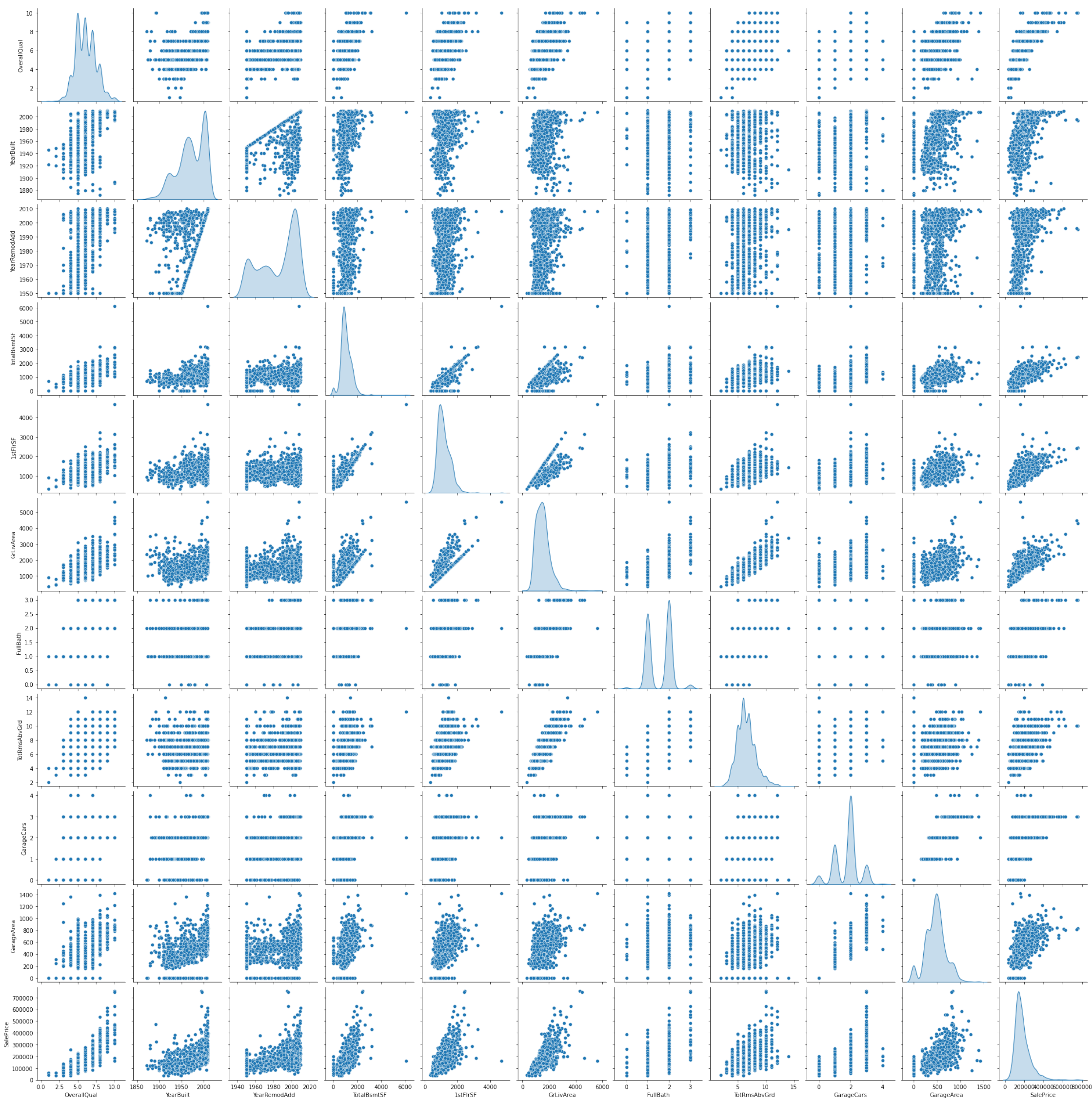
- OverallQual
- GrLivArea

Since number of identified variables are quite less hence , we can see the below columns have threshold 0.5 or -0.5 with 'SalePrice' : List of variables:

- OverallQual
- GrLivArea
- YearBuilt
- YearRemodAdd
- TotalBsmtSF
- 1stFlrSF
- FullBath
- TotRmsAbvGrd
- GarageCars
- GarageArea

5.0.4 d. Pair plot for distribution and density

[23]:
```
PP_DD = sns.pairplot(numerical_df, vars= var ,diag_kind="kde")
```



[24]:
```
corr1 = numerical_df[var].corr()
corr1.style.background_gradient(cmap='coolwarm', axis=0 )
```

[24]: `<pandas.io.formats.style.Styler at 0x1cda0c93910>`

Based on above Pairplot and the Corr matrix, We an Drop independent variables

- Drop YearRemodAdd:

  The relationship of YearRemodAdd with SalePrice has a high resemblance to that of YearBuilt with SalePrice.

- Drop 1stFlrSF:

  The relationship of 1stFlrSF with SalePrice has a high resemblance to that of TotalBsmtSF with SalePrice.

- Drop TotRmsAbvGrd:

  TotRmsAbvGrd is highly correlated to GrLivArea. Therefore, we will drop TotRmsAbvGrd.

- Drop GarageArea:

  GarageArea is highly correlated to GarageCars. Therefore, we will drop GarageArea.

5.0.5 List of significant numerical variables

[25]:
```
# List of significant numerical variables before removing Independent variables:
numerical_df = numerical_df[var]

# Final List of significant numerical variables after removing Independent␣
↪variables:
numerical_df.drop(columns =␣
↪['YearRemodAdd','1stFlrSF','TotRmsAbvGrd','GarageArea'], inplace= True)

numerical_df.columns
```

[25]:
```
Index(['OverallQual', 'YearBuilt', 'TotalBsmtSF', 'GrLivArea', 'FullBath',
       'GarageCars', 'SalePrice'],
      dtype='object')
```

[26]:
```
#Let's see if there is still any missing values in the Numerical Significant␣
↪variables

numerical_df.isnull().sum()
```

[26]:
```
OverallQual    0
YearBuilt      0
TotalBsmtSF    0
GrLivArea      0
FullBath       0
GarageCars     0
SalePrice      0
dtype: int64
```

6 Task 4. EDA of calegorical variables:

6.0.1 a. Missing value treatment

[27]:
```
# Identify columns having missing data in Categorical variables
```

[28]:
```
categorical_df.isnull().sum()
```

[28]:
```
MSZoning            0
Street              0
Alley            1369
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
MasVnrType          8
ExterQual           0
ExterCond           0
Foundation          0
BsmtQual           37
BsmtCond           37
BsmtExposure       38
BsmtFinType1       37
BsmtFinType2       38
Heating             0
HeatingQC           0
CentralAir          0
Electrical          1
KitchenQual         0
Functiol            0
FireplaceQu       690
GarageType         81
GarageFinish       81
GarageQual         81
GarageCond         81
PavedDrive          0
PoolQC           1453
Fence            1179
MiscFeature      1406
SaleType            0
SaleCondition       0
dtype: int64
```

[29]:
```
categorical_df.columns[categorical_df.isnull().any()]
```

```
[29]: Index(['Alley', 'MasVnrType', 'BsmtQual', 'BsmtCond', 'BsmtExposure',
       'BsmtFinType1', 'BsmtFinType2', 'Electrical', 'FireplaceQu',
       'GarageType', 'GarageFinish', 'GarageQual', 'GarageCond', 'PoolQC',
       'Fence', 'MiscFeature'],
      dtype='object')
```

a.1. Find and drop columns having all Null data

```
[30]: #Find columns with all nul records
print("Below columns does not have any data/ (all rows are null) : \n \n",␣
 ↪categorical_df.columns[categorical_df.isnull().all()])

#remove above columns having all null records
categorical_df.dropna(axis= 1 , how='all', inplace=True)

#print the shape of the dataframe
print("\n After treatment shape of the dataframe is : \n",categorical_df.shape)
```

```
Below columns does not have any data/ (all rows are null) :

 Index([], dtype='object')

 After treatment shape of the dataframe is :
 (1460, 43)
```

* None of the columns having all null records, hence no columns will be removed.

a.2. Find and drop columns having most of the NULL data

```
[31]: #taken 85% but this value is uaully discussed with business before removing of␣
 ↪the columns

#Find column having mostly the NULL data
most_Null_data = [i for i in categorical_df.columns if categorical_df[i].
 ↪isnull().sum() > 0.40*len(df)]

print("Column having mostly the NULL data :\n \n", most_Null_data)

#drop columns having mostly the NULL data
categorical_df.drop(columns = most_Null_data, inplace=True)


#print the shape of the dataframe
print("\n After treatment shape of the dataframe is : \n",categorical_df.shape)
```

```
Column having mostly the NULL data :

 ['Alley', 'FireplaceQu', 'PoolQC', 'Fence', 'MiscFeature']
```

```
After treatment shape of the dataframe is :
(1460, 38)
```

- most of the data is null for columns ‘Alley’,‘PoolQC’, ‘Fence’, ‘MiscFeature’, ‘FireplaceQu’
- these columns are dropped

a.3 Drop the missing records

[32]:
```
categorical_df.dropna(inplace=True)

#print the shape of the dataframe
print("\n After treatment shape of the dataframe is : \n",categorical_df.shape)
```

```
After treatment shape of the dataframe is :
(1338, 38)
```

[33]:
```
categorical_df.isnull().sum()
```

[33]:
```
MSZoning        0
Street          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
MasVnrType      0
ExterQual       0
ExterCond       0
Foundation      0
BsmtQual        0
BsmtCond        0
BsmtExposure    0
BsmtFinType1    0
BsmtFinType2    0
Heating         0
HeatingQC       0
CentralAir      0
Electrical      0
```

```
KitchenQual      0
Functiol         0
GarageType       0
GarageFinish     0
GarageQual       0
GarageCond       0
PavedDrive       0
SaleType         0
SaleCondition    0
dtype: int64
```

6.0.2 b. Count plot and box plot for bivariate analysis

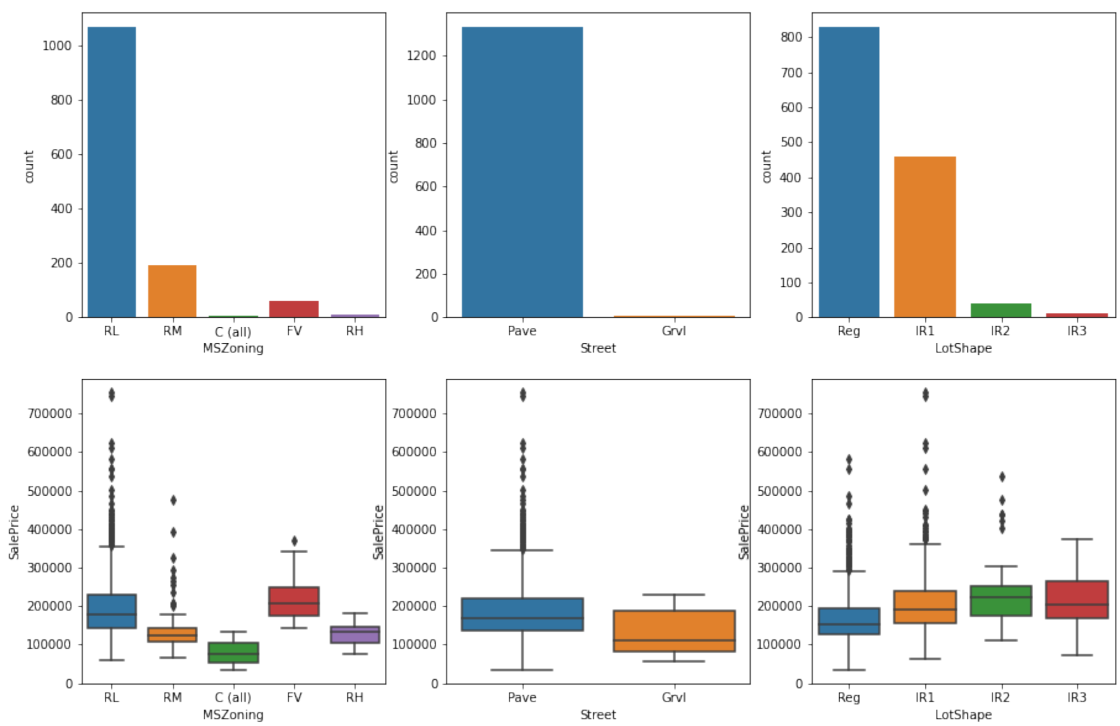
We can analyze the relationship of categorical variables with the dependent variable SalePrice through Count plot and the box plots

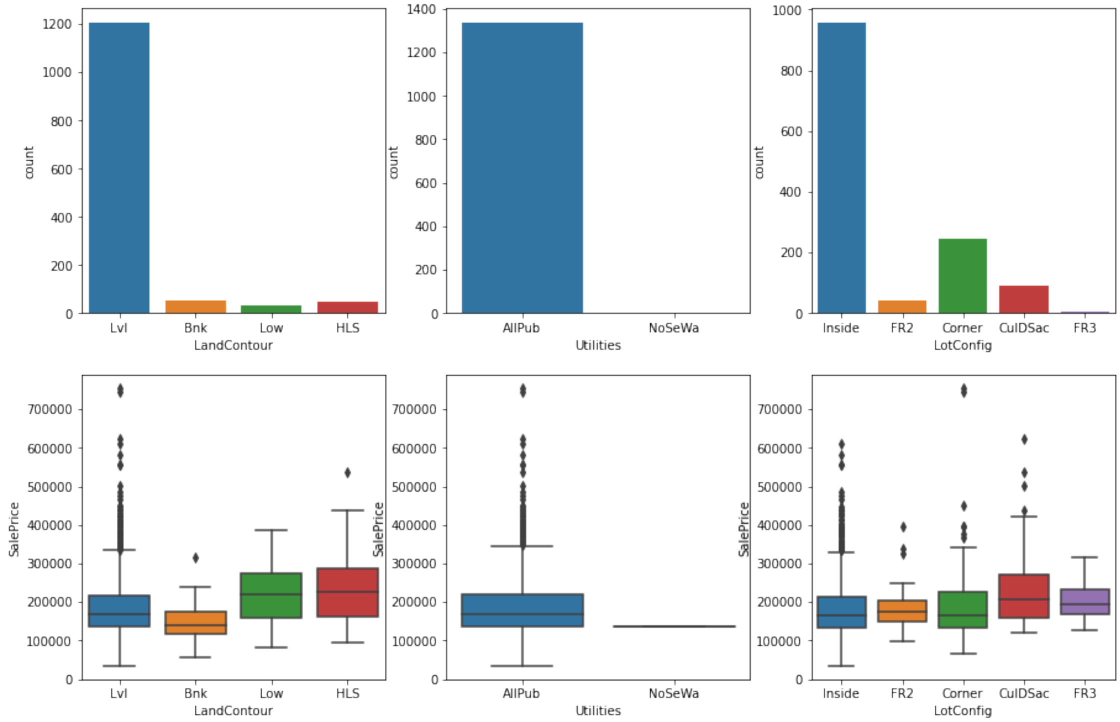
[34]:
```python
#Adding SalePrice to the categorical_df
categorical_df['SalePrice'] = df.loc[categorical_df.index, 'SalePrice'].copy()
categorical_df.head()
```

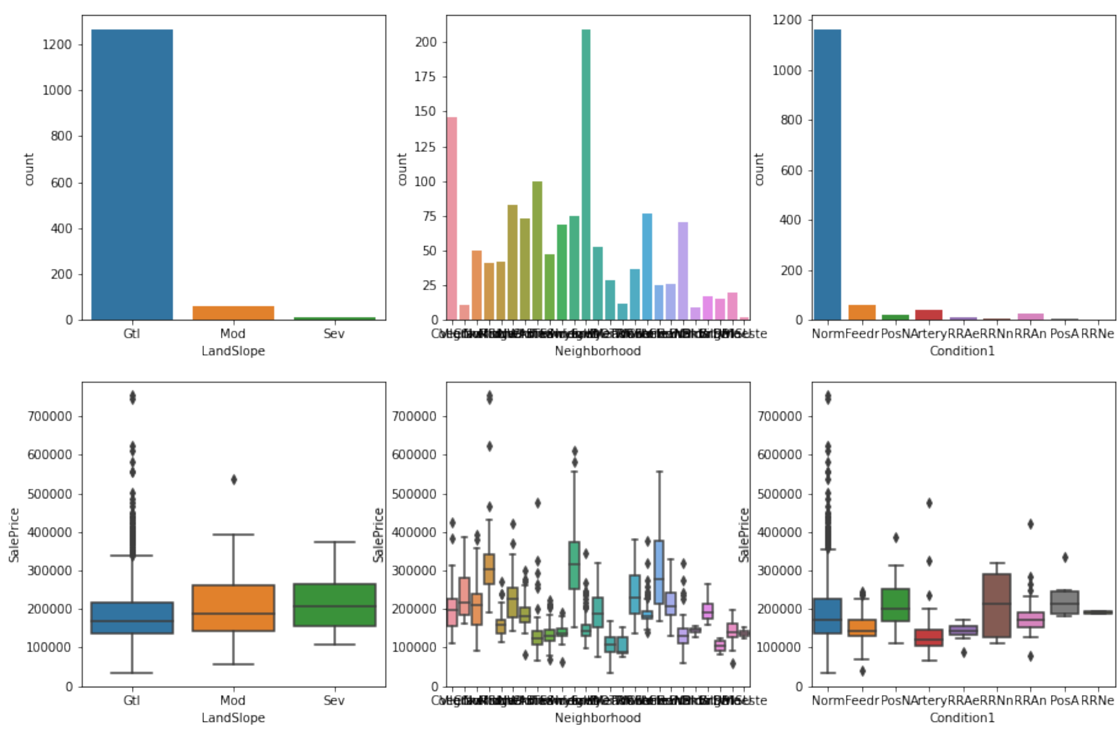
[34]:
```
  MSZoning Street LotShape LandContour Utilities LotConfig LandSlope  \
0       RL   Pave      Reg         Lvl    AllPub    Inside       Gtl
1       RL   Pave      Reg         Lvl    AllPub       FR2       Gtl
2       RL   Pave      IR1         Lvl    AllPub    Inside       Gtl
3       RL   Pave      IR1         Lvl    AllPub    Corner       Gtl
4       RL   Pave      IR1         Lvl    AllPub       FR2       Gtl

  Neighborhood Condition1 Condition2  … KitchenQual Functiol GarageType  \
0      CollgCr       Norm       Norm  …          Gd      Typ     Attchd
1      Veenker      Feedr       Norm  …          TA      Typ     Attchd
2      CollgCr       Norm       Norm  …          Gd      Typ     Attchd
3      Crawfor       Norm       Norm  …          Gd      Typ     Detchd
4      NoRidge       Norm       Norm  …          Gd      Typ     Attchd

  GarageFinish GarageQual GarageCond PavedDrive SaleType SaleCondition  \
0          RFn         TA         TA          Y       WD        Normal
1          RFn         TA         TA          Y       WD        Normal
2          RFn         TA         TA          Y       WD        Normal
3          Unf         TA         TA          Y       WD       Abnorml
4          RFn         TA         TA          Y       WD        Normal

  SalePrice
0    208500
1    181500
2    223500
3    140000
4    250000
```

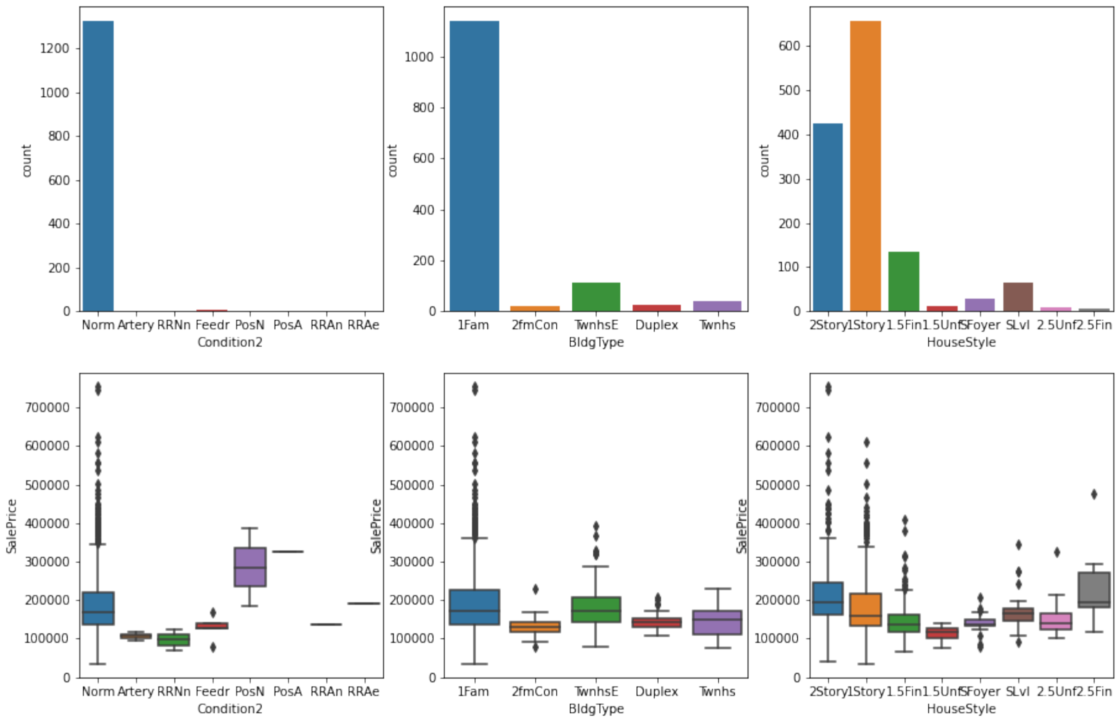
[5 rows x 39 columns]

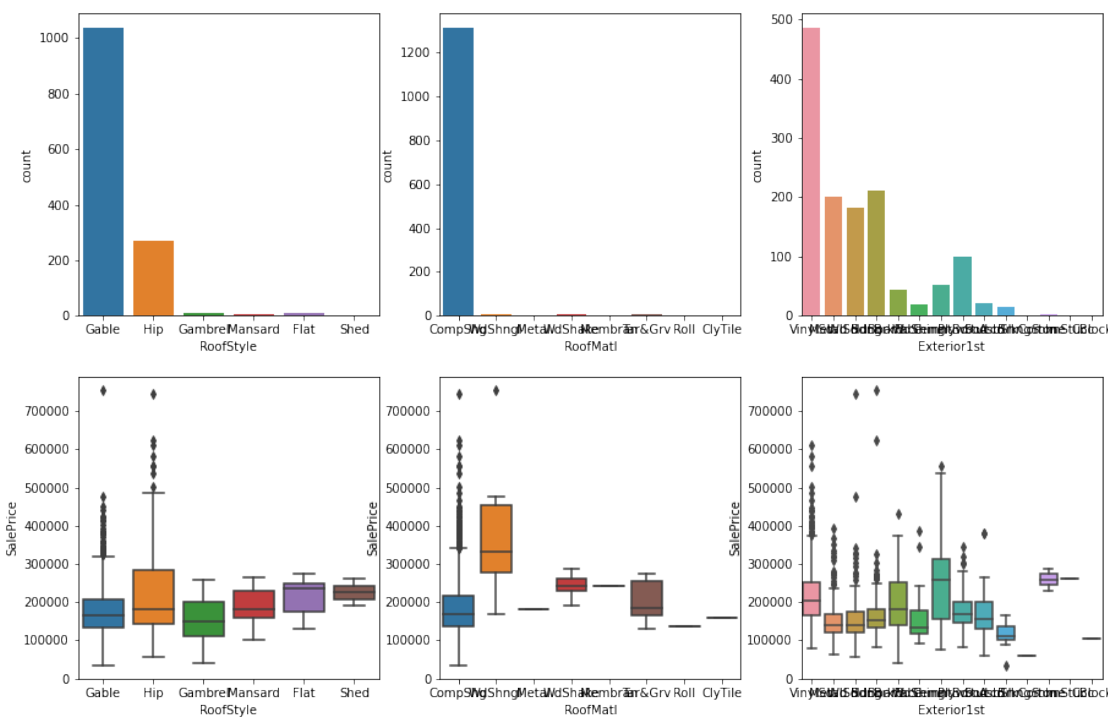
[35]:
```
#Function to box plot all independent categorical variables with SalePrice and␣
↪count plot
ix = 1
fig = plt.figure(figsize = (15,10))
for c in list(categorical_df.columns):
    if ix <= 3:
        if c != 'SalePrice':
            ax1 = fig.add_subplot(2,3,ix)
            sns.countplot(data = categorical_df, x=c, ax = ax1) #For countplot
            ax2 = fig.add_subplot(2,3,ix+3)
            sns.boxplot(data=categorical_df, x=c, y='SalePrice', ax=ax2) #For␣
↪boxplot

    ix = ix +1
    if ix == 4:
        fig = plt.figure(figsize = (15,10))
        ix =1
```

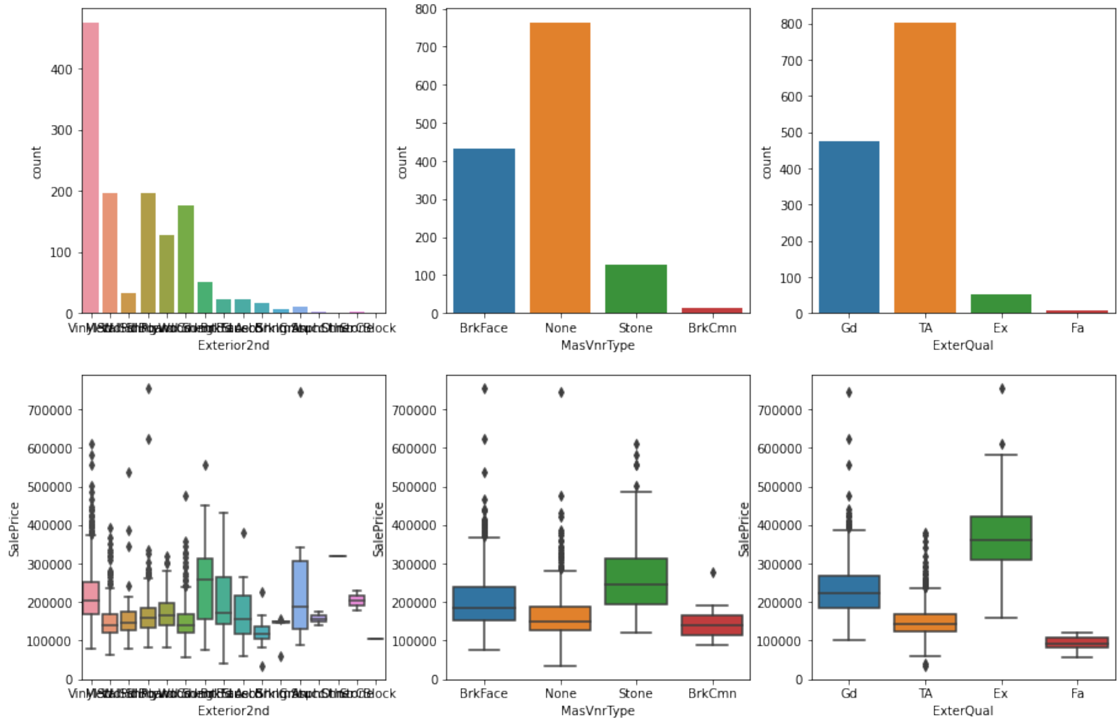


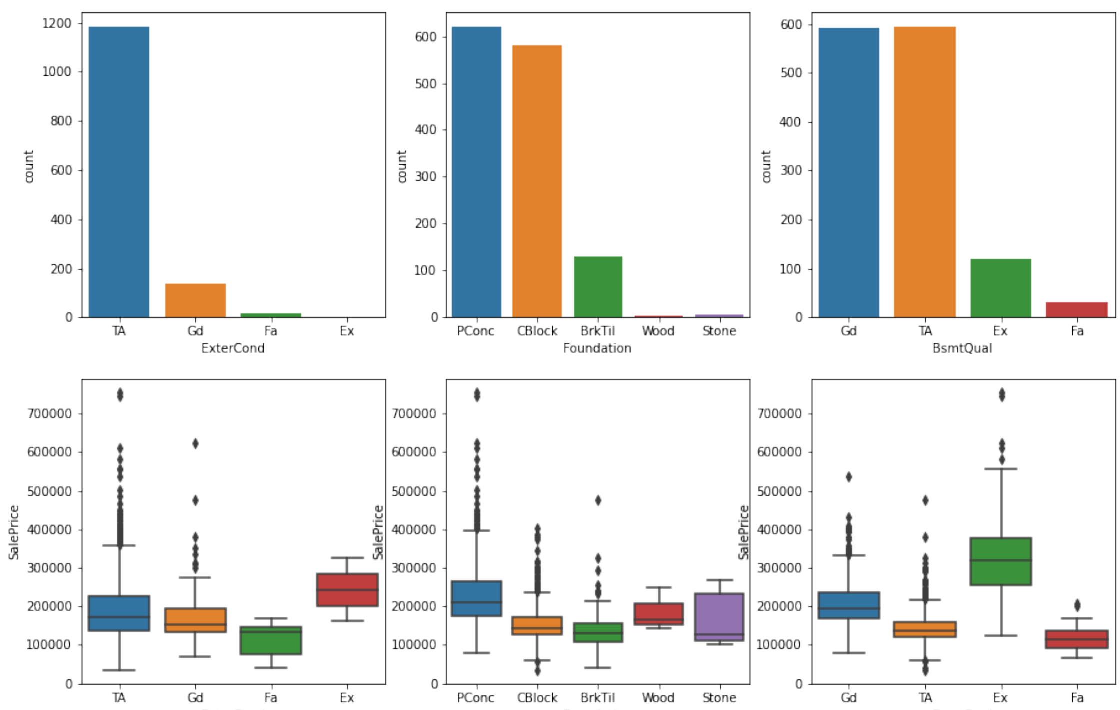

count
Lvl
Bnk
Low
HLS
LandContour
AllPub
NoSeWa
Utilities
Inside
FR2
Corner
CulDSac
FR3
LotConfig
SalePrice

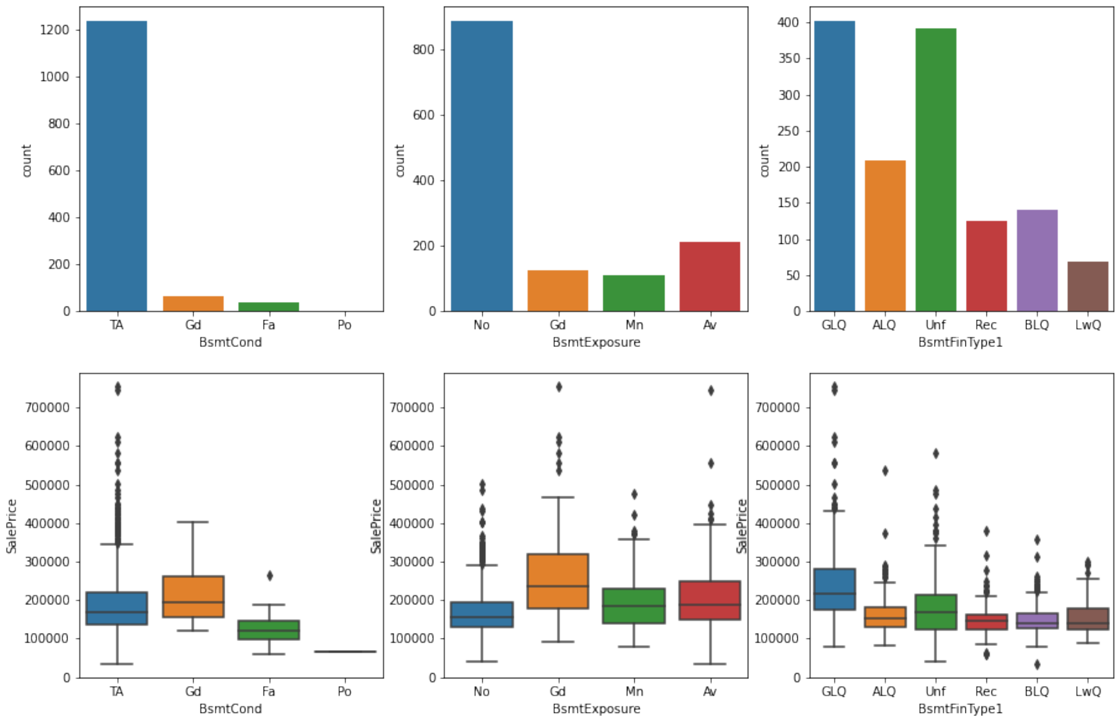

count
Gtl
Mod
Sev
LandSlope
Neighborhood
Norm
Feedr
PosN
Artery
RRAe
RRNn
RRAn
PosA
RRNe
Condition1
SalePrice

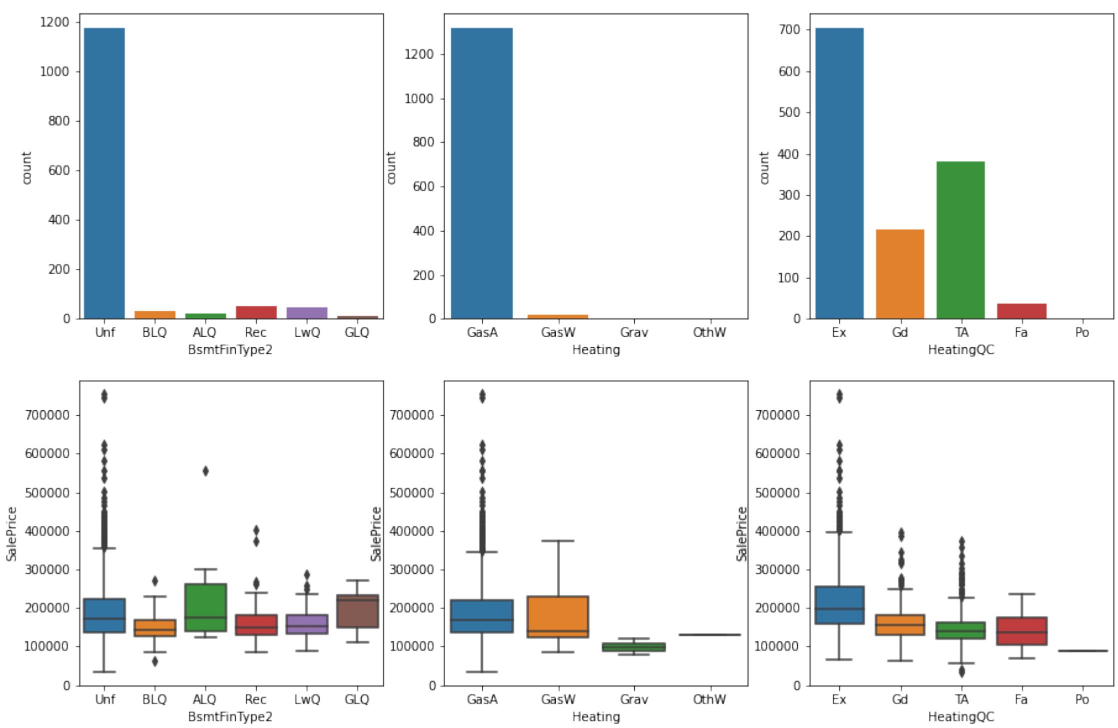

count
Norm Artery RRNn Feedr PosN PosA RRAn RRAe
Condition2
1Fam 2fmCon TwnhsE Duplex Twnhs
BldgType
2Story 1Story 1.5Fin 1.5Unf SFoyer SLvl 2.5Unf 2.5Fin
HouseStyle
SalePrice
Condition2
BldgType
HouseStyle

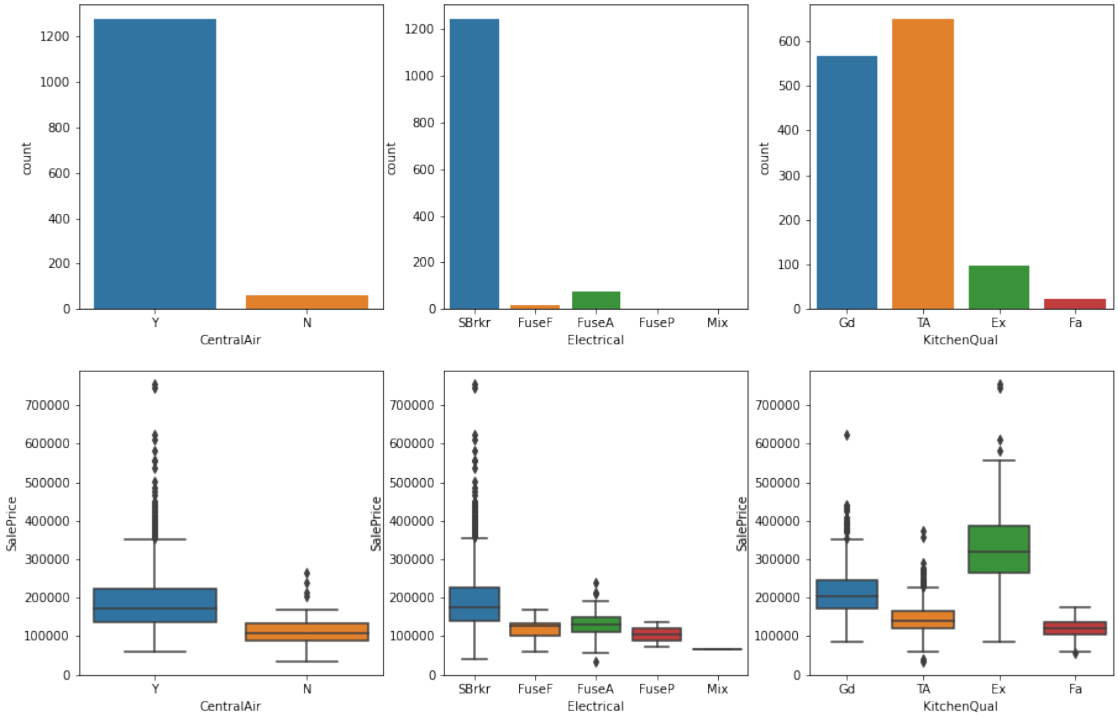

count
Gable Hip Gambrel Mansard Flat Shed
RoofStyle
RoofMatl
Exterior1st
SalePrice
RoofStyle
RoofMatl
Exterior1st

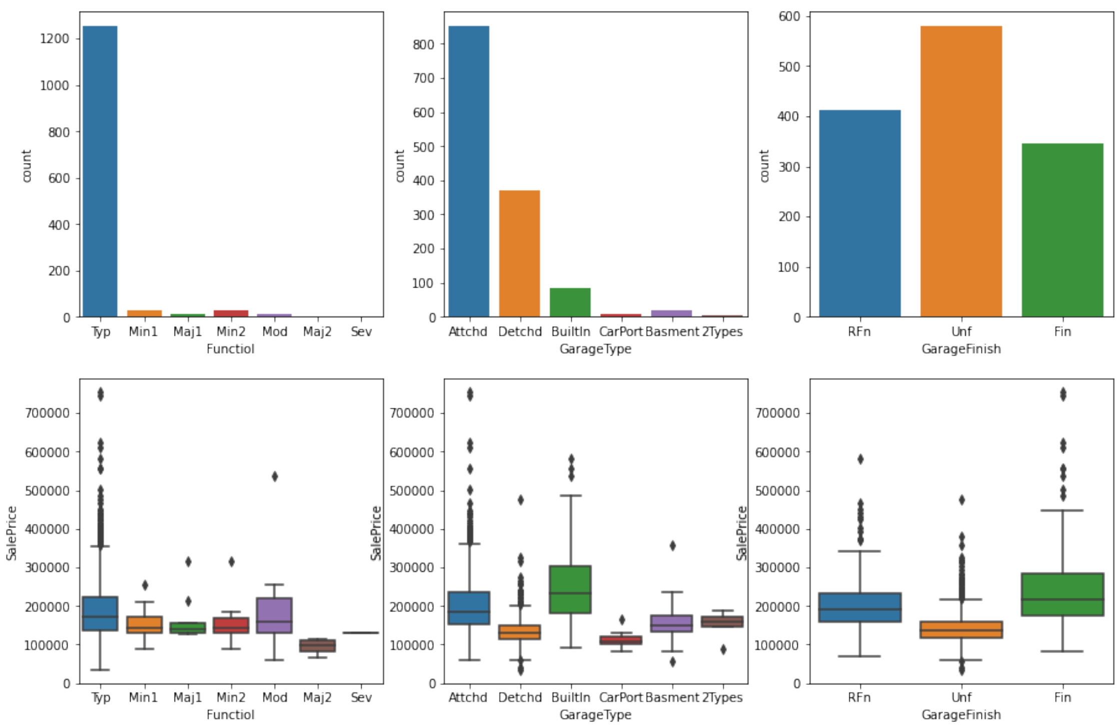

count
Exterior2nd
BrkFace
None
Stone
BrkCmn
MasVnrType
Gd
TA
Ex
Fa
ExterQual
SalePrice

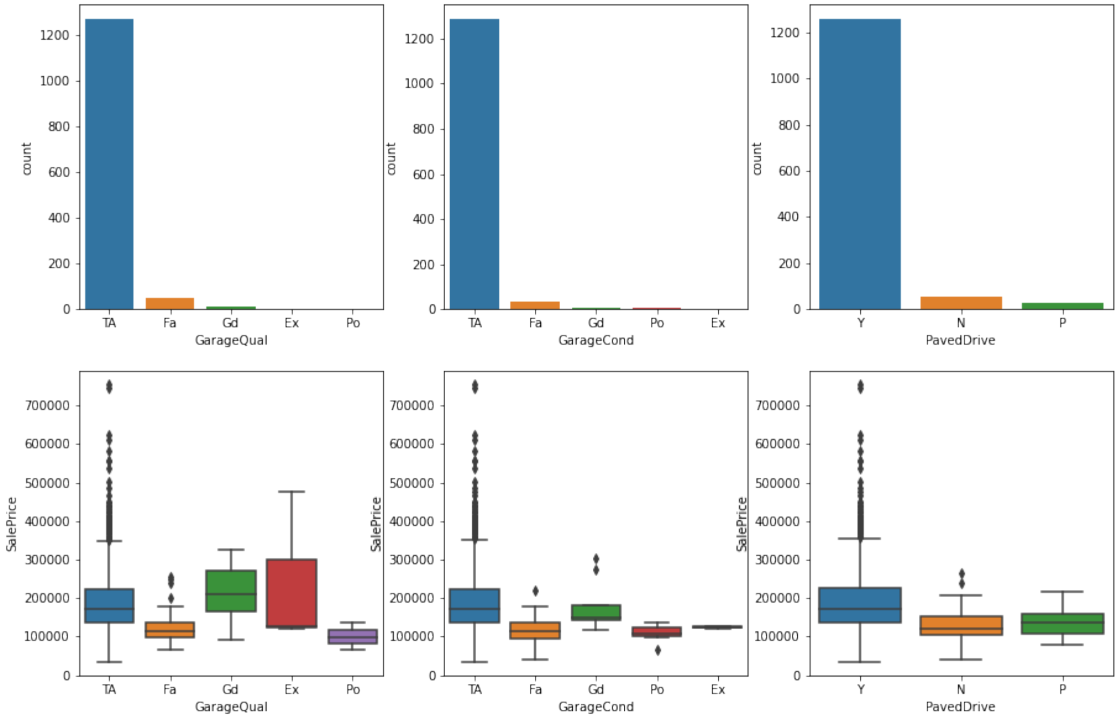

count
TA
Gd
Fa
Ex
ExterCond
PConc
CBlock
BrkTil
Wood
Stone
Foundation
BsmtQual
SalePrice

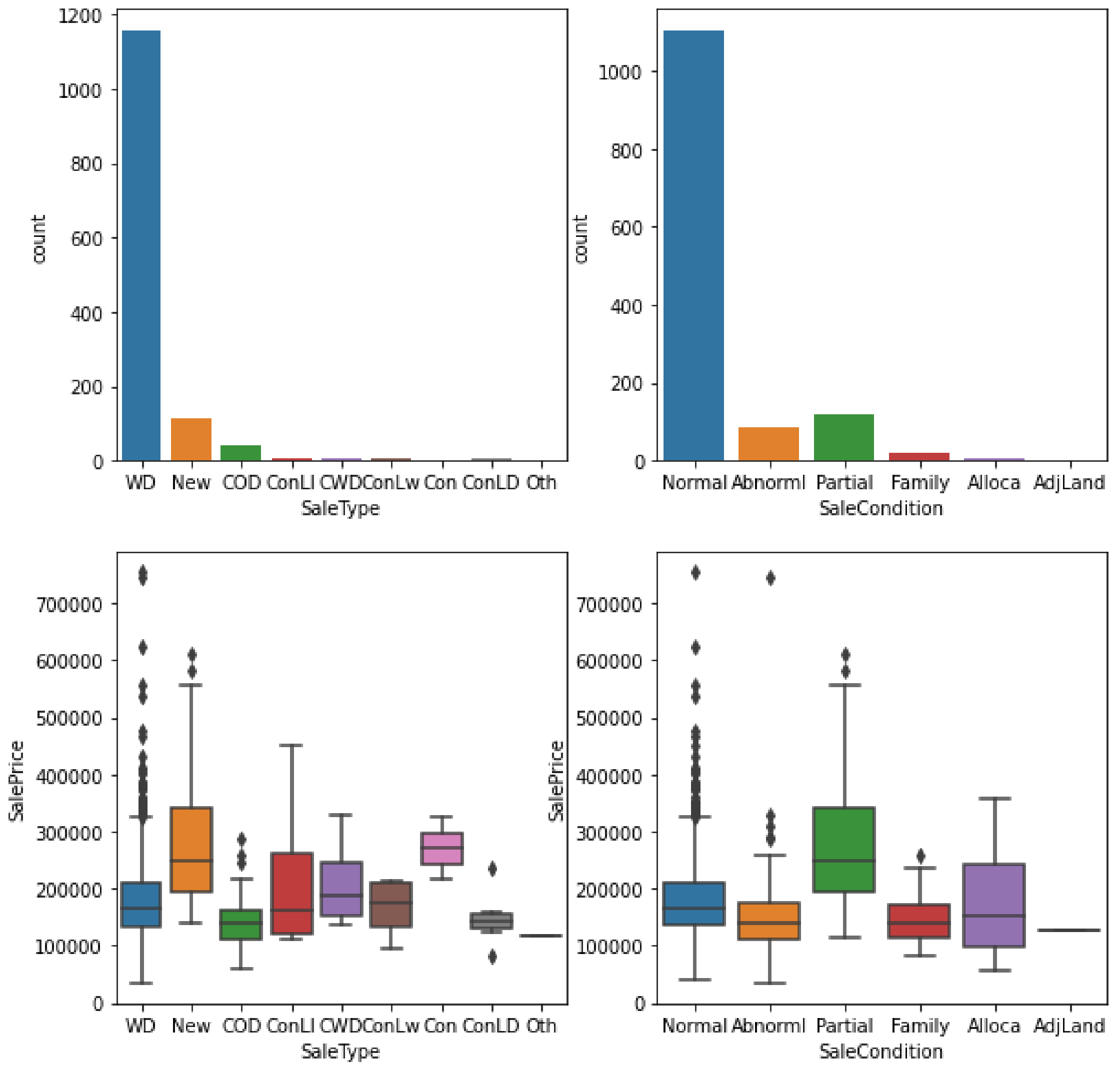

count
BsmtCond
TA
Gd
Fa
Po
BsmtExposure
No
Mn
Av
BsmtFinType1
GLQ
ALQ
Unf
Rec
BLQ
LwQ
SalePrice


count
BsmtFinType2
Unf
BLQ
ALQ
Rec
LwQ
GLQ
Heating
GasA
GasW
Grav
OthW
HeatingQC
Ex
Gd
TA
Fa
Po
SalePrice


count
CentralAir
Y
N
SBrkr
FuseF
FuseA
FuseP
Mix
Electrical
Gd
TA
Ex
Fa
KitchenQual
SalePrice


count
Typ
Min1
Maj1
Min2
Mod
Maj2
Sev
Functiol
Attchd
Detchd
BuiltIn
CarPort
Basment
2Types
GarageType
RFn
Unf
Fin
GarageFinish
SalePrice


count
GarageQual
TA
Fa
Gd
Ex
Po
count
GarageCond
TA
Fa
Gd
Po
Ex
count
PavedDrive
Y
N
P
SalePrice
GarageQual
SalePrice
GarageCond
SalePrice
PavedDrive




```
<Figure size 1080x720 with 0 Axes>
```

Conclusion:

- The above box plot shows that the all the categorical variables have outliers which needs outlier treatment.

- Box plot median shows that the data is heavily skewed either left or right

6.0.3 c. Identify significant variables using p-values and Chi-Square values

Hypothesis Testing

- Null Hypothesis : Identified columns is NOT an important predictor (Here p >= 0.5)
- Alternative Hypothesis : Identified columns is an important predictor (Here p < 0.5)

[36]:

```
class ChiSquare:

#Function to determine p-value and perform chi-square test
    def __init__(self, dataframe):
        self.df = dataframe
        self.p = None #P-Value
        self.chi2 = None #Chi-square Test Statistic
        self.dof = None

        self.dfObserved = None
        self.dfExpected = None
        global significant_variable_list
        significant_variable_list = list()

#Function to print the results of p-value and chi-square test
    def _print_chisquare_result(self, colX, alpha):
        result = ""
        if self.p<alpha:
            result="{0} is IMPORTANT for Prediction".format(colX)
            significant_variable_list.append(colX)
        else:
            result="{0} is NOT an important predictor. (Discard {0} from␣
 ↪model)".format(colX)
        print(result)

#Function to determine chi-square and p-value less than or equal to alpha, here␣
 ↪alpha is considered as 0.05
    def TestIndependence(self,colX,colY, alpha=0.05):
        X = self.df[colX].astype(str)
        Y = self.df[colY].astype(str)

        self.dfObserved = pd.crosstab(Y,X)
        chi2, p, dof, expected = stats.chi2_contingency(self.dfObserved.values)
        self.p = p
        self.chi2 = chi2
        self.dof = dof

        self.dfExpected = pd.DataFrame(expected, columns=self.dfObserved.
 ↪columns, index = self.dfObserved.index)

        self._print_chisquare_result(colX,alpha)


#Initializing ChiSquare Class
cT = ChiSquare(categorical_df)

#Perform Feature Selection
```

```python
testColumns = ['MSZoning', 'Street', 'LotShape', 'LandContour', 'Utilities',␣
↪'LotConfig', 'LandSlope', 'Neighborhood', 'Condition1', 'Condition2',␣
↪'BldgType', 'HouseStyle', 'RoofStyle', 'RoofMatl', 'Exterior1st',␣
↪'Exterior2nd', 'MasVnrType', 'ExterQual', 'ExterCond', 'Foundation',␣
↪'BsmtQual', 'BsmtCond', 'BsmtExposure', 'BsmtFinType1', 'BsmtFinType2',␣
↪'Heating', 'HeatingQC', 'CentralAir', 'Electrical', 'KitchenQual',␣
↪'Functiol', 'GarageType', 'GarageFinish', 'GarageQual', 'GarageCond',␣
↪'PavedDrive', 'SaleType', 'SaleCondition','SalePrice']
for var in testColumns:
    cT.TestIndependence(colX=var,colY="SalePrice" )
```

```
MSZoning is IMPORTANT for Prediction
Street is IMPORTANT for Prediction
LotShape is IMPORTANT for Prediction
LandContour is NOT an important predictor. (Discard LandContour from model)
Utilities is NOT an important predictor. (Discard Utilities from model)
LotConfig is NOT an important predictor. (Discard LotConfig from model)
LandSlope is NOT an important predictor. (Discard LandSlope from model)
Neighborhood is IMPORTANT for Prediction
Condition1 is NOT an important predictor. (Discard Condition1 from model)
Condition2 is IMPORTANT for Prediction
BldgType is NOT an important predictor. (Discard BldgType from model)
HouseStyle is NOT an important predictor. (Discard HouseStyle from model)
RoofStyle is NOT an important predictor. (Discard RoofStyle from model)
RoofMatl is NOT an important predictor. (Discard RoofMatl from model)
Exterior1st is NOT an important predictor. (Discard Exterior1st from model)
Exterior2nd is NOT an important predictor. (Discard Exterior2nd from model)
MasVnrType is IMPORTANT for Prediction
ExterQual is IMPORTANT for Prediction
ExterCond is NOT an important predictor. (Discard ExterCond from model)
Foundation is IMPORTANT for Prediction
BsmtQual is IMPORTANT for Prediction
BsmtCond is IMPORTANT for Prediction
BsmtExposure is IMPORTANT for Prediction
BsmtFinType1 is NOT an important predictor. (Discard BsmtFinType1 from model)
BsmtFinType2 is NOT an important predictor. (Discard BsmtFinType2 from model)
Heating is NOT an important predictor. (Discard Heating from model)
HeatingQC is NOT an important predictor. (Discard HeatingQC from model)
CentralAir is IMPORTANT for Prediction
Electrical is IMPORTANT for Prediction
KitchenQual is IMPORTANT for Prediction
Functiol is NOT an important predictor. (Discard Functiol from model)
GarageType is IMPORTANT for Prediction
GarageFinish is IMPORTANT for Prediction
GarageQual is IMPORTANT for Prediction
GarageCond is NOT an important predictor. (Discard GarageCond from model)
PavedDrive is NOT an important predictor. (Discard PavedDrive from model)
SaleType is IMPORTANT for Prediction
```

```
SaleCondition is IMPORTANT for Prediction
SalePrice is IMPORTANT for Prediction
```

List of Identified Significant categorical variables

[37]:
```
# List of significant variable

significant_variable_list
```

[37]:
```
['MSZoning',
 'Street',
 'LotShape',
 'Neighborhood',
 'Condition2',
 'MasVnrType',
 'ExterQual',
 'Foundation',
 'BsmtQual',
 'BsmtCond',
 'BsmtExposure',
 'CentralAir',
 'Electrical',
 'KitchenQual',
 'GarageType',
 'GarageFinish',
 'GarageQual',
 'SaleType',
 'SaleCondition',
 'SalePrice']
```

[38]:
```
# SIgnificant Categorical variable DataFrame

categorical_df =categorical_df[significant_variable_list]
categorical_df.columns
```

[38]:
```
Index(['MSZoning', 'Street', 'LotShape', 'Neighborhood', 'Condition2',
       'MasVnrType', 'ExterQual', 'Foundation', 'BsmtQual', 'BsmtCond',
       'BsmtExposure', 'CentralAir', 'Electrical', 'KitchenQual', 'GarageType',
       'GarageFinish', 'GarageQual', 'SaleType', 'SaleCondition', 'SalePrice'],
      dtype='object')
```

7 Task 5. Combine all the significant categorical and numerical variables

[39]:
```
numerical_df.columns
```

[39]: Index(['OverallQual', 'YearBuilt', 'TotalBsmtSF', 'GrLivArea', 'FullBath',
       'GarageCars', 'SalePrice'],
      dtype='object')

[40]:
```
categorical_df.columns
```

[40]: Index(['MSZoning', 'Street', 'LotShape', 'Neighborhood', 'Condition2',
       'MasVnrType', 'ExterQual', 'Foundation', 'BsmtQual', 'BsmtCond',
       'BsmtExposure', 'CentralAir', 'Electrical', 'KitchenQual', 'GarageType',
       'GarageFinish', 'GarageQual', 'SaleType', 'SaleCondition', 'SalePrice'],
      dtype='object')

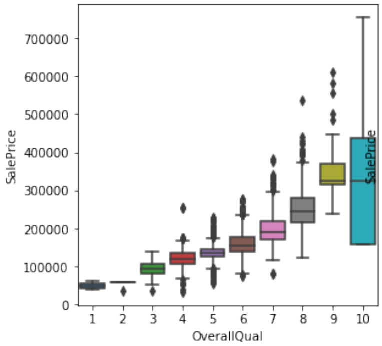
[41]:
```
#Combining the significant categorical and numerical variables datasets
House_Price_Predection_df = pd.merge(numerical_df,categorical_df, how="outer",␣
 ↪on=["SalePrice"])
```

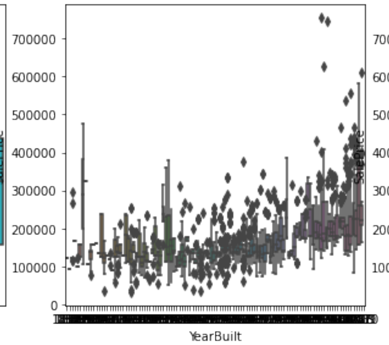
[42]:
```
House_Price_Predection_df.columns
```

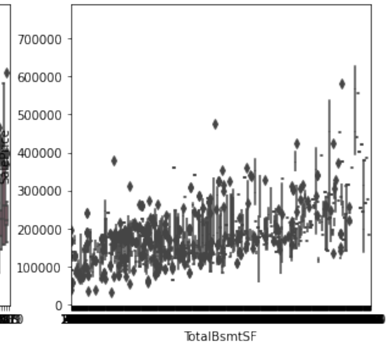
[42]: Index(['OverallQual', 'YearBuilt', 'TotalBsmtSF', 'GrLivArea', 'FullBath',
       'GarageCars', 'SalePrice', 'MSZoning', 'Street', 'LotShape',
       'Neighborhood', 'Condition2', 'MasVnrType', 'ExterQual', 'Foundation',
       'BsmtQual', 'BsmtCond', 'BsmtExposure', 'CentralAir', 'Electrical',
       'KitchenQual', 'GarageType', 'GarageFinish', 'GarageQual', 'SaleType',
       'SaleCondition'],
      dtype='object')

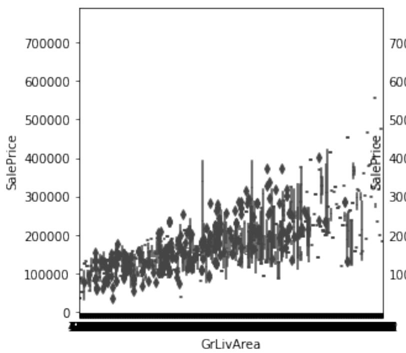
8 Task 6. Plot box plot for the new dataset to find the variables with outliers

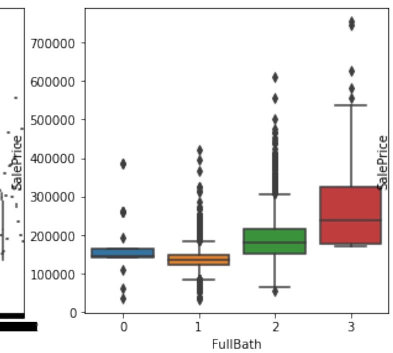
[43]:
```
#Function to plot all independent categorical variables with SalePrice and␣
 ↪count plot
ix = 1
fig = plt.figure(figsize = (15,10))
for c in list(House_Price_Predection_df.columns):
    if ix <= 3:
        if c != 'SalePrice':
            ax2 = fig.add_subplot(2,3,ix)
            sns.boxplot(data=House_Price_Predection_df, x=c, y='SalePrice',␣
 ↪ax=ax2) #for boxplot

    ix = ix +1
    if ix == 4:
        fig = plt.figure(figsize = (15,10))
        ix =1
```

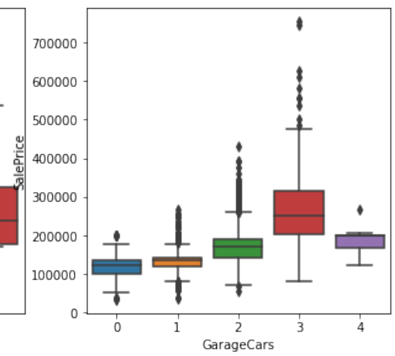

SalePrice
0
100000
200000
300000
400000
500000
600000
700000
1 2 3 4 5 6 7 8 9 10
OverallQual

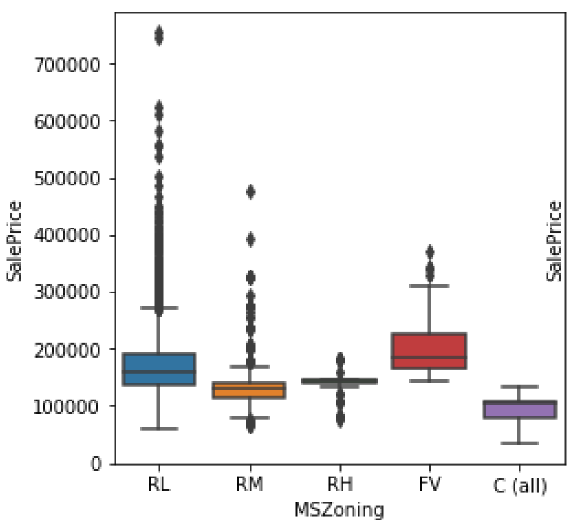

SalePrice
0
100000
200000
300000
400000
500000
600000
700000
YearBuilt

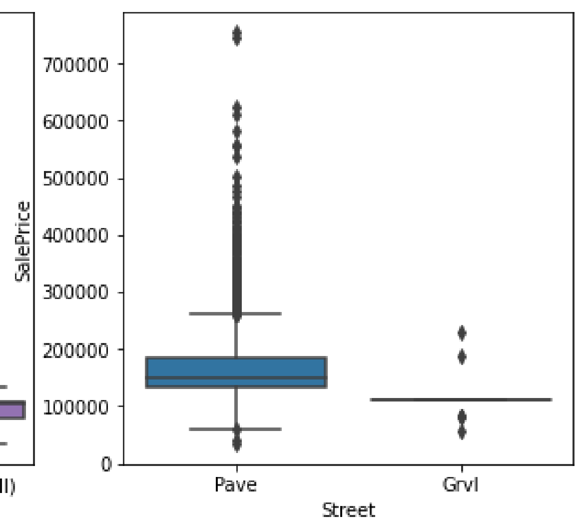

SalePrice
0
100000
200000
300000
400000
500000
600000
700000
TotalBsmtSF

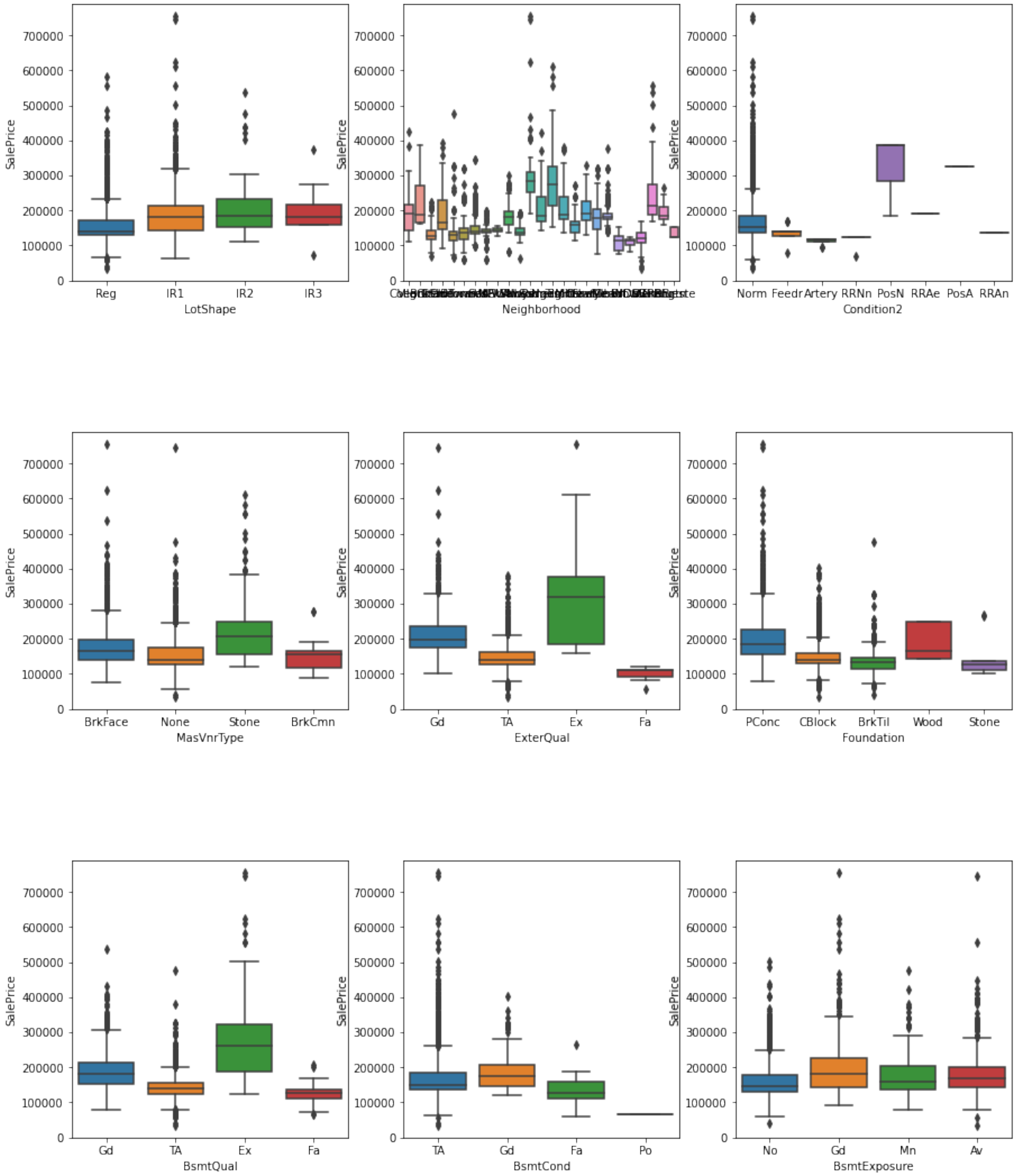

SalePrice
0
100000
200000
300000
400000
500000
600000
700000
GrLivArea

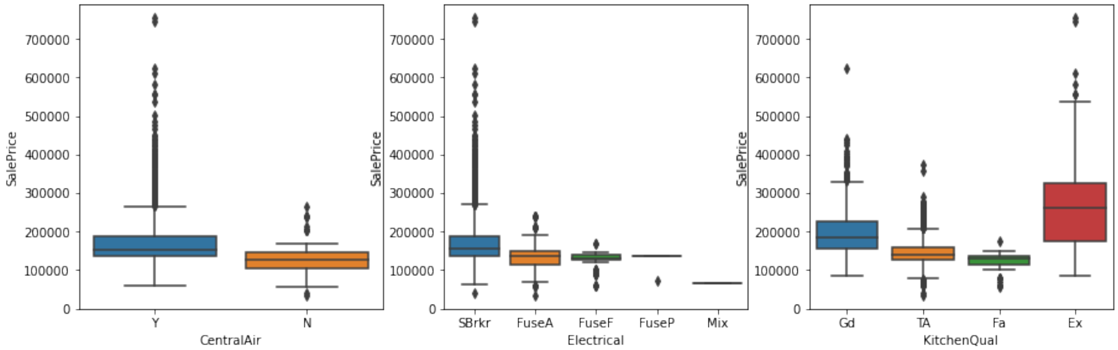

SalePrice
0
100000
200000
300000
400000
500000
600000
700000
0 1 2 3
FullBath

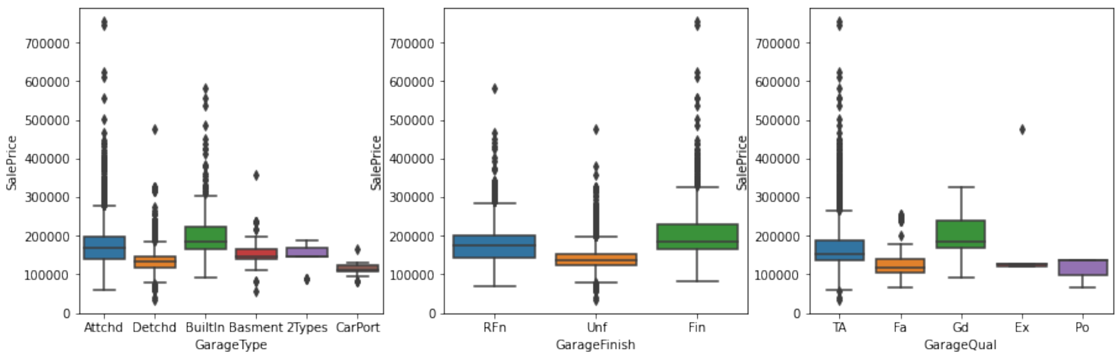

SalePrice
0
100000
200000
300000
400000
500000
600000
700000
0 1 2 3 4
GarageCars

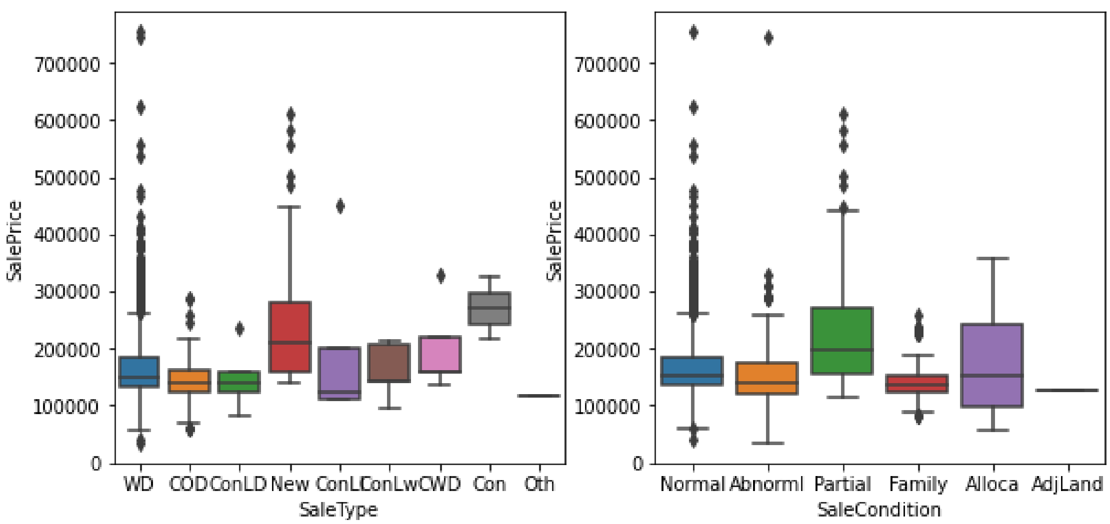

SalePrice
0
100000
200000
300000
400000
500000
600000
700000
RL RM RH FV C (all)
MSZoning


SalePrice
0
100000
200000
300000
400000
500000
600000
700000
Pave Grvl
Street


SalePrice
0
100000
200000
300000
400000
500000
600000
700000
Reg IR1 IR2 IR3
LotShape
Neighborhood
Norm Feedr Artery RRNn PosN RRAe PosA RRAn
Condition2
BrkFace None Stone BrkCmn
MasVnrType
Gd TA Ex Fa
ExterQual
PConc CBlock BrkTil Wood Stone
Foundation
Gd TA Ex Fa
BsmtQual
TA Gd Fa Po
BsmtCond
No Gd Mn Av
BsmtExposure







Conclusion

- In the combined dataset, all the Variables are having outliners

- All the variables are heavily skewed.