In [275]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

In [276]:
```
data=pd.read_csv(r"C:\Users\Admin\Desktop\Course DS\Data\House Price\train (2).csv")

pd.set_option('display.max_columns',None)
data.head()
```

Out[276]:

	Id	MSSubClass	MSZoning	LotFrontage	LotArea	Street	Alley	LotShape	LandContour	Utilities	LotConfig	LandSlope	Neighborhood	Condition1	Condition2	BldgType	HouseStyle	OverallQual	OverallCo
0	1	60	RL	65.0	8450	Pave	NaN	Reg	Lvl	AllPub	Inside	Gtl	CollgCr	Norm	Norm	1Fam	2Story	7	
1	2	20	RL	80.0	9600	Pave	NaN	Reg	Lvl	AllPub	FR2	Gtl	Veenker	Feedr	Norm	1Fam	1Story	6	
2	3	60	RL	68.0	11250	Pave	NaN	IR1	Lvl	AllPub	Inside	Gtl	CollgCr	Norm	Norm	1Fam	2Story	7	
3	4	70	RL	60.0	9550	Pave	NaN	IR1	Lvl	AllPub	Corner	Gtl	Crawfor	Norm	Norm	1Fam	2Story	7	
4	5	60	RL	84.0	14260	Pave	NaN	IR1	Lvl	AllPub	FR2	Gtl	NoRidge	Norm	Norm	1Fam	2Story	8	

In [358]:
```
test=pd.read_csv(r"C:\Users\Admin\Desktop\Course DS\Data\House Price\test (2).csv")
test.head()
```

Out[358]:

	Id	MSSubClass	MSZoning	LotFrontage	LotArea	Street	Alley	LotShape	LandContour	Utilities	LotConfig	LandSlope	Neighborhood	Condition1	Condition2	BldgType	HouseStyle	OverallQual	Overall
0	1461	20	RH	80.0	11622	Pave	NaN	Reg	Lvl	AllPub	Inside	Gtl	NAmes	Feedr	Norm	1Fam	1Story	5	
1	1462	20	RL	81.0	14267	Pave	NaN	IR1	Lvl	AllPub	Corner	Gtl	NAmes	Norm	Norm	1Fam	1Story	6	
2	1463	60	RL	74.0	13830	Pave	NaN	IR1	Lvl	AllPub	Inside	Gtl	Gilbert	Norm	Norm	1Fam	2Story	5	
3	1464	60	RL	78.0	9978	Pave	NaN	IR1	Lvl	AllPub	Inside	Gtl	Gilbert	Norm	Norm	1Fam	2Story	6	
4	1465	120	RL	43.0	5005	Pave	NaN	IR1	HLS	AllPub	Inside	Gtl	StoneBr	Norm	Norm	TwnhsE	1Story	8	

In [ ]:

In [278]:
```
data.shape
```

Out[278]:
```
(1460, 81)
```

In [279]:

```
data.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  KitchenAbvGr   1460 non-null   int64  
 53  KitchenQual    1460 non-null   object 
 54  TotRmsAbvGrd   1460 non-null   int64  
 55  Functional     1460 non-null   object 
 56  Fireplaces     1460 non-null   int64  
 57  FireplaceQu    770 non-null    object 
 58  GarageType     1379 non-null   object 
 59  GarageYrBlt    1379 non-null   float64
 60  GarageFinish   1379 non-null   object 
 61  GarageCars     1460 non-null   int64  
 62  GarageArea     1460 non-null   int64  
 63  GarageQual     1379 non-null   object 
 64  GarageCond     1379 non-null   object 
 65  PavedDrive     1460 non-null   object 
 66  WoodDeckSF     1460 non-null   int64  
 67  OpenPorchSF    1460 non-null   int64  
 68  EnclosedPorch  1460 non-null   int64  
 69  3SsnPorch      1460 non-null   int64  
 70  ScreenPorch    1460 non-null   int64  
 71  PoolArea       1460 non-null   int64  
 72  PoolQC         7 non-null      object 
 73  Fence          281 non-null    object 
 74  MiscFeature    54 non-null     object 
 75  MiscVal        1460 non-null   int64  
 76  MoSold         1460 non-null   int64  
 77  YrSold         1460 non-null   int64  
 78  SaleType       1460 non-null   object 
 79  SaleCondition  1460 non-null   object 
 80  SalePrice      1460 non-null   int64  
dtypes: float64(3), int64(35), object(43)
memory usage: 924.0+ KB
```

In [280]:

```
#data.isna().sum()
```

In [281]:

```
for i in list(data.columns):
    if data[i].isnull().sum() >= 0:
        print(i, ': ', data[i].isnull().sum())
```

```
Id :  0
MSSubClass :  0
MSZoning :  0
LotFrontage :  259
LotArea :  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
OverallQual :  0
OverallCond :  0
YearBuilt :  0
YearRemodAdd :  0
RoofStyle :  0
RoofMatl :  0
Exterior1st :  0
Exterior2nd :  0
MasVnrType :  8
MasVnrArea :  8
ExterQual :  0
ExterCond :  0
Foundation :  0
BsmtQual :  37
BsmtCond :  37
BsmtExposure :  38
BsmtFinType1 :  37
BsmtFinSF1 :  0
BsmtFinType2 :  38
BsmtFinSF2 :  0
BsmtUnfSF :  0
TotalBsmtSF :  0
Heating :  0
HeatingQC :  0
CentralAir :  0
Electrical :  1
1stFlrSF :  0
2ndFlrSF :  0
LowQualFinSF :  0
GrLivArea :  0
BsmtFullBath :  0
BsmtHalfBath :  0
FullBath :  0
HalfBath :  0
BedroomAbvGr :  0
KitchenAbvGr :  0
KitchenQual :  0
TotRmsAbvGrd :  0
Functional :  0
```

```
Fireplaces :  0
FireplaceQu :  690
GarageType :  81
GarageYrBlt :  81
GarageFinish :  81
GarageCars :  0
GarageArea :  0
GarageQual :  81
GarageCond :  81
PavedDrive :  0
WoodDeckSF :  0
OpenPorchSF :  0
EnclosedPorch :  0
3SsnPorch :  0
ScreenPorch :  0
PoolArea :  0
PoolQC :  1453
Fence :  1179
MiscFeature :  1406
MiscVal :  0
MoSold :  0
YrSold :  0
SaleType :  0
SaleCondition :  0
SalePrice :  0
```

In [282]:
```
print(data.nunique())
```

```
Id               1460
MSSubClass         15
MSZoning            5
LotFrontage       110
LotArea          1073
                 ... 
MoSold             12
YrSold              5
SaleType            9
SaleCondition       6
SalePrice         663
Length: 81, dtype: int64
```

In [283]:
```
for i in list(data.columns):
    if data[i].isnull().sum() > 0:
        print(i, ': ', data[i].isnull().sum())
```

```
LotFrontage :  259
Alley :  1369
MasVnrType :  8
MasVnrArea :  8
BsmtQual :  37
BsmtCond :  37
BsmtExposure :  38
BsmtFinType1 :  37
BsmtFinType2 :  38
Electrical :  1
FireplaceQu :  690
GarageType :  81
GarageYrBlt :  81
GarageFinish :  81
GarageQual :  81
GarageCond :  81
PoolQC :  1453
Fence :  1179
MiscFeature :  1406
```

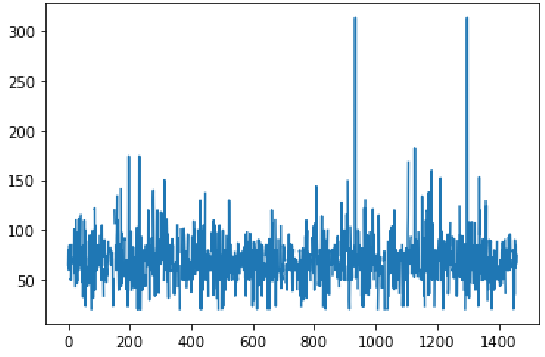
In [284]:
```
data1=data.drop(["Id","MSZoning","Street","Alley","MasVnrType","LowQualFinSF","BsmtExposure","BsmtFinType1","BsmtFinType2","EnclosedPorch","3SsnPorch","ScreenPorch","Poo
"HeatingQC","FireplaceQu","Functional","FireplaceQu","GarageType","GarageFinish","GarageQual","GarageCond","PavedDrive","PoolQC","Fence","MiscFeature","SaleType","BsmtFi
,"2ndFlrSF","GarageYrBlt",  "MiscVal","OpenPorchSF","WoodDeckSF"],axis=1)
data1
```

Out[284]:

	MSSubClass	LotFrontage	LotArea	LotShape	LandContour	Utilities	LotConfig	LandSlope	Neighborhood	Condition1	Condition2	BldgType	HouseStyle	OverallQual	OverallCond	YearBuilt	YearRemod
0	60	65.0	8450	Reg	Lvl	AllPub	Inside	Gtl	CollgCr	Norm	Norm	1Fam	2Story	7	5	2003	2
1	20	80.0	9600	Reg	Lvl	AllPub	FR2	Gtl	Veenker	Feedr	Norm	1Fam	1Story	6	8	1976	1
2	60	68.0	11250	IR1	Lvl	AllPub	Inside	Gtl	CollgCr	Norm	Norm	1Fam	2Story	7	5	2001	2
3	70	60.0	9550	IR1	Lvl	AllPub	Corner	Gtl	Crawfor	Norm	Norm	1Fam	2Story	7	5	1915	1
4	60	84.0	14260	IR1	Lvl	AllPub	FR2	Gtl	NoRidge	Norm	Norm	1Fam	2Story	8	5	2000	2
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
1455	60	62.0	7917	Reg	Lvl	AllPub	Inside	Gtl	Gilbert	Norm	Norm	1Fam	2Story	6	5	1999	2
1456	20	85.0	13175	Reg	Lvl	AllPub	Inside	Gtl	NWAmes	Norm	Norm	1Fam	1Story	6	6	1978	1
1457	70	66.0	9042	Reg	Lvl	AllPub	Inside	Gtl	Crawfor	Norm	Norm	1Fam	2Story	7	9	1941	2
1458	20	68.0	9717	Reg	Lvl	AllPub	Inside	Gtl	NAmes	Norm	Norm	1Fam	1Story	5	6	1950	1
1459	20	75.0	9937	Reg	Lvl	AllPub	Inside	Gtl	Edwards	Norm	Norm	1Fam	1Story	5	6	1965	1

1460 rows × 50 columns

```
In [285]: data1["BldgType"].unique()

Out[285]: array(['1Fam', '2fmCon', 'Duplex', 'TwnhsE', 'Twnhs'], dtype=object)

In [ ]:

In [286]: for i in list(data1.columns):
              if data1[i].isnull().sum() > 0:
                  print(i, ': ', data1[i].isnull().sum())

          LotFrontage :  259
          MasVnrArea :  8
          BsmtQual :  37
          BsmtCond :  37
          Electrical :  1

In [287]: data1["LotFrontage"].unique()

Out[287]: array([ 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.])

In [288]: plt.plot(data1["LotFrontage"])

Out[288]: [<matplotlib.lines.Line2D at 0x1ea23803640>]
```



In [289]:

```
data1["LotFrontage"]=data1["LotFrontage"].replace(np.nan,data1["LotFrontage"].mean())
data1["LotFrontage"]
```

Out[289]:

```
0       65.0
1       80.0
2       68.0
3       60.0
4       84.0
        ...
1455    62.0
1456    85.0
1457    66.0
1458    68.0
1459    75.0
Name: LotFrontage, Length: 1460, dtype: float64
```

In [290]:

```
#data1["MasVnrArea"].unique()
```

In [291]:

```
##Replacing Nan values
data1["MasVnrArea"]=data1["MasVnrArea"].replace(np.nan,data1["MasVnrArea"].mean())
data1["MasVnrArea"]
```

Out[291]:

```
0       196.0
1         0.0
2       162.0
3         0.0
4       350.0
        ...
1455      0.0
1456    119.0
1457      0.0
1458      0.0
1459      0.0
Name: MasVnrArea, Length: 1460, dtype: float64
```

In [292]:
```
for x in data1.columns:
    if data1[x].all()==0:
        print(x)
#     BsmtFinSF2
#     TotalBsmtSF   GarageArea replace
#     2ndFlrSF
#     GarageYrBlt
# MiscVal
# OpenPorchSF
# WoodDeckSF
```

```
MasVnrArea
BsmtFinSF1
BsmtUnfSF
TotalBsmtSF
BsmtFullBath
BsmtHalfBath
FullBath
HalfBath
BedroomAbvGr
KitchenAbvGr
Fireplaces
GarageCars
GarageArea
```

In [ ]:

In [293]:
```
data1["GarageCars"].value_counts()
```

Out[293]:
```
2    824
1    369
3    181
0     81
4      5
Name: GarageCars, dtype: int64
```

In [294]:
```
#data1["2ndFlrSF"].value_counts()
# data1["OpenPorchSF"].value_counts()
# data1["WoodDeckSF"].value_counts()
```

In [295]:

```
for col in data1:
    print ('\nFrequency of Categories for varible %s'%col)
    print (data1[col].value_counts())
```

```

Frequency of Categories for varible MSSubClass
20     536
60     299
50     144
120     87
30      69
160     63
70      60
80      58
90      52
190     30
85      20
75      16
45      12
180     10
40       4
Name: MSSubClass, dtype: int64

Frequency of Categories for varible LotFrontage
```

In [296]:

```
## For 0 values present in the data
data1["MasVnrArea"]=data1["MasVnrArea"].replace(0,data1["MasVnrArea"].mean())
#data1["2ndFlrSF"]=data1["2ndFlrSF"].replace(0,data1["2ndFlrSF"].mean())
data1["BsmtFinSF1"]=data1["BsmtFinSF1"].replace(0,data1["BsmtFinSF1"].mean())
data1["BsmtUnfSF"]=data1["BsmtUnfSF"].replace(0,data1["BsmtUnfSF"].mean())
data1["TotalBsmtSF"]=data1["TotalBsmtSF"].replace(0,data1["TotalBsmtSF"].mean())
data1["BsmtUnfSF"]=data1["BsmtUnfSF"].replace(0,data1["BsmtUnfSF"].mean())
data1["TotalBsmtSF"]=data1["TotalBsmtSF"].replace(0,data1["TotalBsmtSF"].mean())
data1["GarageArea"]=data1["GarageArea"].replace(0,data1["GarageArea"].mean())
```

```
In [297]: data1
```

Out[297]:

	MSSubClass	LotFrontage	LotArea	LotShape	LandContour	Utilities	LotConfig	LandSlope	Neighborhood	Condition1	Condition2	BldgType	HouseStyle	OverallQual	OverallCond	YearBuilt	YearRemod
0	60	65.0	8450	Reg	Lvl	AllPub	Inside	Gtl	CollgCr	Norm	Norm	1Fam	2Story	7	5	2003	2
1	20	80.0	9600	Reg	Lvl	AllPub	FR2	Gtl	Veenker	Feedr	Norm	1Fam	1Story	6	8	1976	1
2	60	68.0	11250	IR1	Lvl	AllPub	Inside	Gtl	CollgCr	Norm	Norm	1Fam	2Story	7	5	2001	2
3	70	60.0	9550	IR1	Lvl	AllPub	Corner	Gtl	Crawfor	Norm	Norm	1Fam	2Story	7	5	1915	1
4	60	84.0	14260	IR1	Lvl	AllPub	FR2	Gtl	NoRidge	Norm	Norm	1Fam	2Story	8	5	2000	2
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
1455	60	62.0	7917	Reg	Lvl	AllPub	Inside	Gtl	Gilbert	Norm	Norm	1Fam	2Story	6	5	1999	2
1456	20	85.0	13175	Reg	Lvl	AllPub	Inside	Gtl	NWAmes	Norm	Norm	1Fam	1Story	6	6	1978	1
1457	70	66.0	9042	Reg	Lvl	AllPub	Inside	Gtl	Crawfor	Norm	Norm	1Fam	2Story	7	9	1941	2
1458	20	68.0	9717	Reg	Lvl	AllPub	Inside	Gtl	NAmes	Norm	Norm	1Fam	1Story	5	6	1950	1
1459	20	75.0	9937	Reg	Lvl	AllPub	Inside	Gtl	Edwards	Norm	Norm	1Fam	1Story	5	6	1965	1

1460 rows × 50 columns

In [298]:

```
# Label Enconding Techniques ::
label_encoders={}
from sklearn.preprocessing import LabelEncoder
le=LabelEncoder()
for i in data1.columns:
    if data1[i].dtype=="object":
        label_encoders[i] = LabelEncoder()
        data1[i] = label_encoders[i].fit_transform(data1[i])

    print(i)
```

```
MSSubClass
LotFrontage
LotArea
LotShape
LandContour
Utilities
LotConfig
LandSlope
Neighborhood
Condition1
Condition2
BldgType
HouseStyle
OverallQual
OverallCond
YearBuilt
YearRemodAdd
RoofStyle
RoofMatl
Exterior1st
Exterior2nd
MasVnrArea
ExterQual
ExterCond
Foundation
BsmtQual
BsmtCond
BsmtFinSF1
BsmtUnfSF
TotalBsmtSF
Heating
CentralAir
Electrical
1stFlrSF
GrLivArea
BsmtFullBath
BsmtHalfBath
FullBath
HalfBath
BedroomAbvGr
KitchenAbvGr
KitchenQual
TotRmsAbvGrd
Fireplaces
GarageCars
GarageArea
MoSold
YrSold
SaleCondition
SalePrice
```

In [299]:
```
data1
```

Out[299]:

	MSSubClass	LotFrontage	LotArea	LotShape	LandContour	Utilities	LotConfig	LandSlope	Neighborhood	Condition1	Condition2	BldgType	HouseStyle	OverallQual	OverallCond	YearBuilt	YearRemod
0	60	65.0	8450	3	3	0	4	0	5	2	2	0	5	7	5	2003	2
1	20	80.0	9600	3	3	0	2	0	24	1	2	0	2	6	8	1976	1
2	60	68.0	11250	0	3	0	4	0	5	2	2	0	5	7	5	2001	2
3	70	60.0	9550	0	3	0	0	0	6	2	2	0	5	7	5	1915	1
4	60	84.0	14260	0	3	0	2	0	15	2	2	0	5	8	5	2000	2
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
1455	60	62.0	7917	3	3	0	4	0	8	2	2	0	5	6	5	1999	2
1456	20	85.0	13175	3	3	0	4	0	14	2	2	0	2	6	6	1978	1
1457	70	66.0	9042	3	3	0	4	0	6	2	2	0	5	7	9	1941	2
1458	20	68.0	9717	3	3	0	4	0	12	2	2	0	2	5	6	1950	1
1459	20	75.0	9937	3	3	0	4	0	7	2	2	0	2	5	6	1965	1

1460 rows × 50 columns

In [300]:
```
# from sklearn.preprocessing import LabelEncoder
# for i in data1.columns:
#     if i=="object":
#         le=LabelEncoder()
#         i=le.fit_transform(i)
#     print(i)
```

In [301]:
```
for i in list(data1.columns):
    if data1[i].isnull().sum() > 0:
        print(i, ': ', data1[i].isnull().sum())
else:
        print("No Missing Valuesin remaining columns")
```

```
No Missing Valuesin remaining columns
```

In [302]:
```
# data1["MasVnrArea"].interpolate(method='polynomial', order=7)
# data1["MasVnrArea"]
```

In [303]:
```
data1["YrSold"].unique()
```

Out[303]:
```
array([2008, 2007, 2006, 2009, 2010], dtype=int64)
```

In [304]:
```
# ##Outlier detection using Inter Quantile Range
# Q1=data1.quantile(0.25)
# Q3=data1.quantile(0.75)
# IQR=Q3-Q1
# IQR
```

In [305]:
```
# print("outlier Counter of the all features")
# ((data1 < (Q1 - 1.5 * IQR)) | (data1 > (Q3 + 1.5 * IQR))).sum()
```

In [306]:
```
# ##Outliers replace  using Mean
# for col in data1.columns:
#     if data1[col].dtypes != 'object':
#         q1 , q3 =data1[col].quantile(0.25),data1[col].quantile(0.75)
#         iqr = q3 - q1
#         ll = q1-1.5*iqr
#         ul = q3 + 1.5*iqr
#         data1[col] = np.where(data1[col]>ul,data1[col].mean(),np.where(data1[col]<ll,data1[col].mean(),data1[col]))
```

In [307]:
```
# Utilities  #LandSlope ## BldgType _- remove  ,# YearBuilt
```

In [308]:
```
data1.corr()
```

Out[308]:

	MSSubClass	LotFrontage	LotArea	LotShape	LandContour	Utilities	LotConfig	LandSlope	Neighborhood	Condition1	Condition2	BldgType	HouseStyle	OverallQual	OverallCond
MSSubClass	1.000000	-3.570559e-01	-0.139781	0.119289	-0.002940	-2.284384e-02	0.075910	-0.025672	-0.005985	-0.024762	-0.042395	0.746063	0.397161	0.032628	-0.059316
LotFrontage	-0.357056	1.000000e+00	0.306795	-0.144931	-0.075647	5.171411e-18	-0.181253	0.067608	0.084545	-0.008483	0.003214	-0.408564	0.031907	0.234196	-0.052820
LotArea	-0.139781	3.067946e-01	1.000000	-0.165315	-0.149083	1.012318e-02	-0.121161	0.436868	0.044569	0.023846	0.022164	-0.205721	-0.033190	0.105806	-0.005636
LotShape	0.119289	-1.449309e-01	-0.165315	1.000000	0.085434	-3.610068e-02	0.221102	-0.099951	-0.038894	-0.115003	-0.043768	0.116262	-0.104026	-0.190497	0.013693
LandContour	-0.002940	-7.564653e-02	-0.149083	0.085434	1.000000	8.238030e-03	-0.025527	-0.374267	0.019116	0.024801	-0.016185	0.051143	0.075234	0.028907	-0.045271
Utilities	-0.022844	5.171411e-18	0.010123	-0.036101	0.008238	1.000000e+00	-0.032589	-0.005909	0.046809	-0.000950	-0.000831	-0.010778	0.054283	-0.001881	0.009994
LotConfig	0.075910	-1.812535e-01	-0.121161	0.221102	-0.025527	-3.258930e-02	1.000000	-0.007256	-0.036597	0.021457	0.033868	0.107229	-0.032945	-0.031086	-0.030788
LandSlope	-0.025672	6.760810e-02	0.436868	-0.099951	-0.374267	-5.909285e-03	-0.007256	1.000000	-0.080405	-0.016762	-0.026322	-0.053582	-0.031793	-0.066450	0.010355

In [309]:
```
def find_zeros(data1):
    '''creates a list of the indexes of the zeros in the data'''
    zeroidx=np.where(np.any(data1==0, axis=1))
    print(zeroidx)
    return len(zeroidx)
```

In [310]:

```
find_zeros(data1)
```

```
(array([   0,    1,    2, ..., 1457, 1458, 1459], dtype=int64),)
```

Out[310]:

```
1
```

In [311]:

```
for x in data1.columns:
    if data1[x].all()==0:
        print(x)
```

```
LotShape
LandContour
Utilities
LotConfig
LandSlope
Neighborhood
Condition1
Condition2
BldgType
HouseStyle
RoofStyle
RoofMatl
Exterior1st
Exterior2nd
ExterQual
ExterCond
Foundation
BsmtQual
BsmtCond
Heating
CentralAir
Electrical
BsmtFullBath
BsmtHalfBath
FullBath
HalfBath
BedroomAbvGr
KitchenAbvGr
KitchenQual
Fireplaces
GarageCars
SaleCondition
```

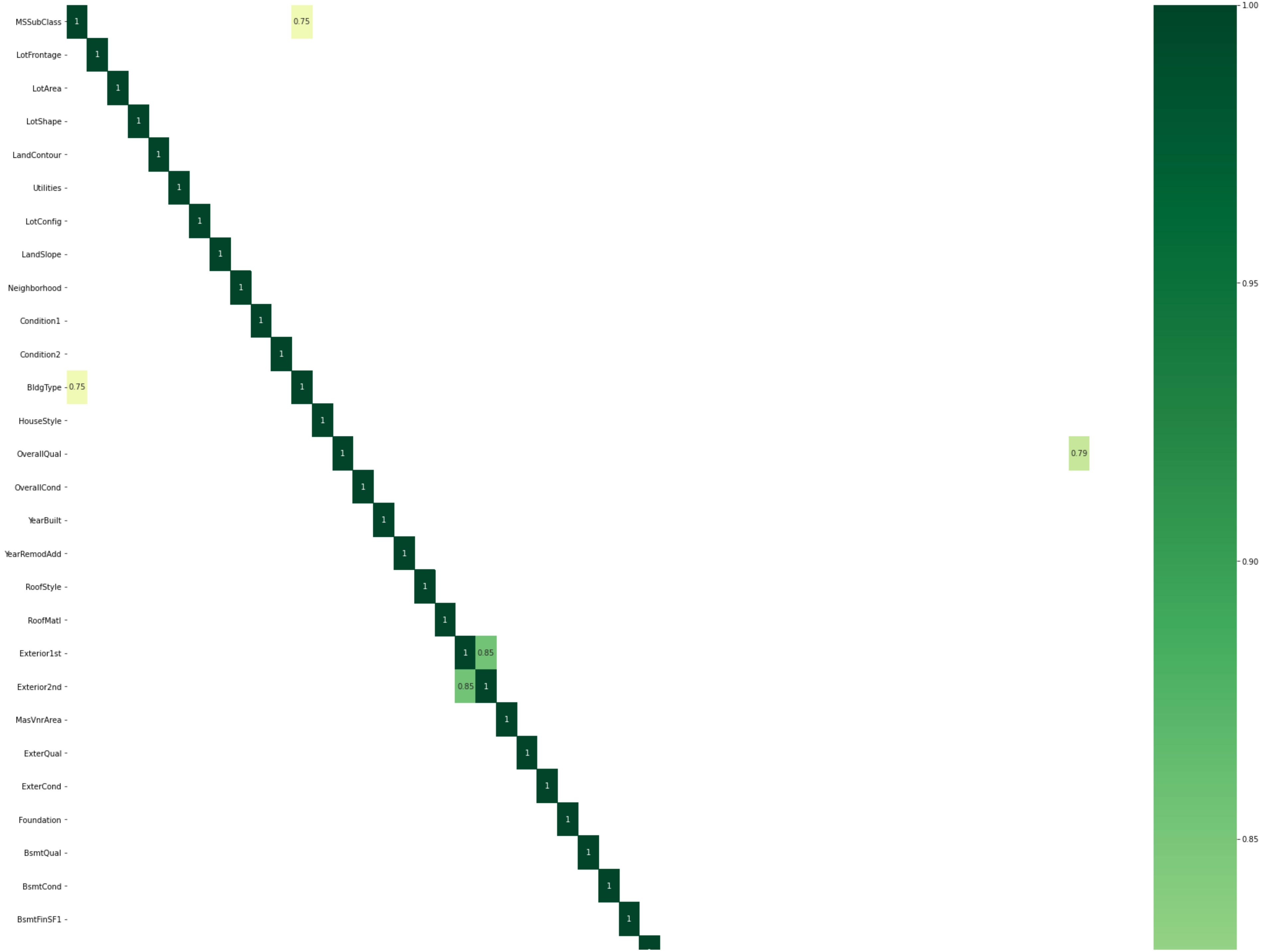
In [312]:
```
data1
```

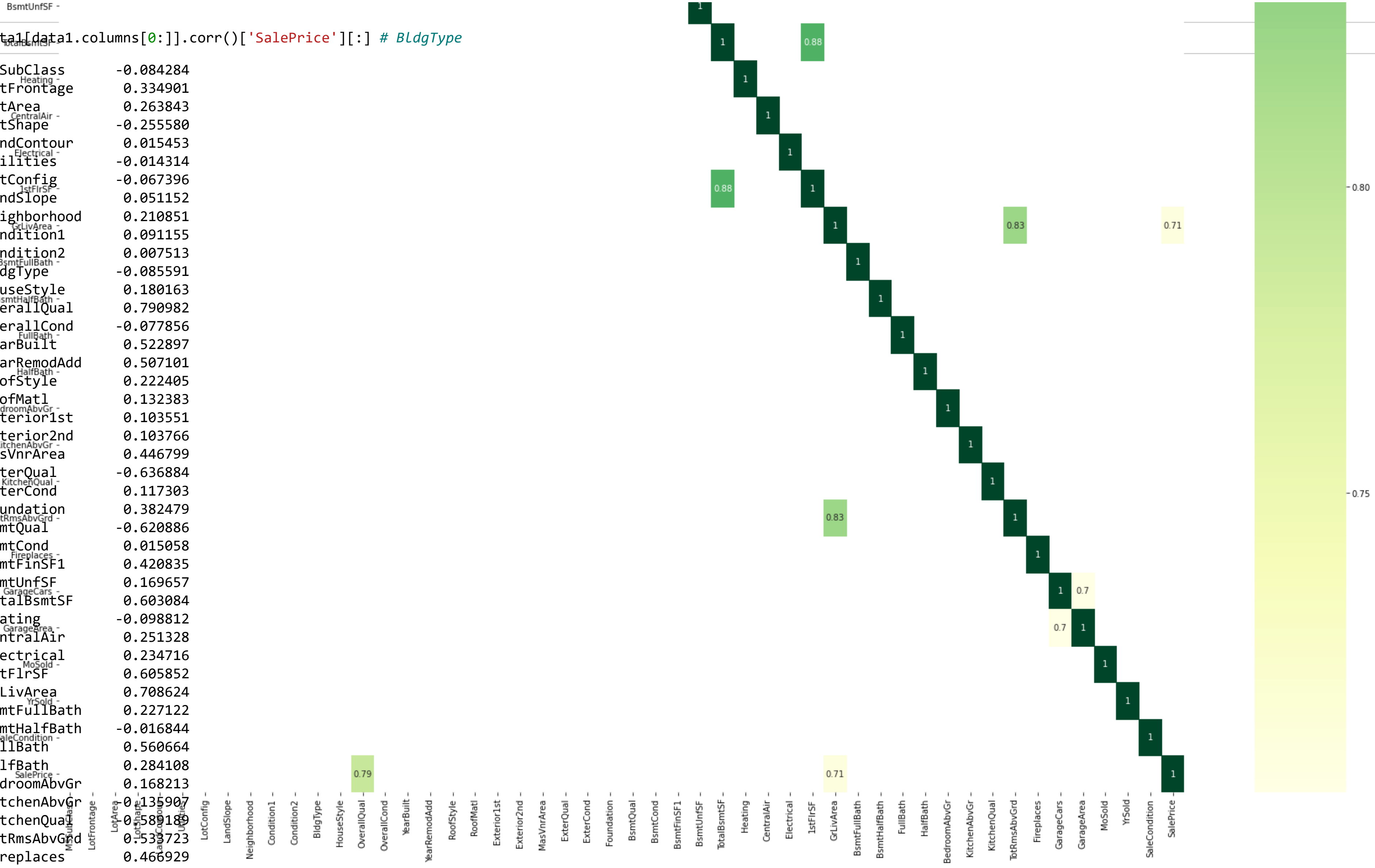
Out[312]:

	MSSubClass	LotFrontage	LotArea	LotShape	LandContour	Utilities	LotConfig	LandSlope	Neighborhood	Condition1	Condition2	BldgType	HouseStyle	OverallQual	OverallCond	YearBuilt	YearRemod
0	60	65.0	8450	3	3	0	4	0	5	2	2	0	5	7	5	2003	2
1	20	80.0	9600	3	3	0	2	0	24	1	2	0	2	6	8	1976	1
2	60	68.0	11250	0	3	0	4	0	5	2	2	0	5	7	5	2001	2
3	70	60.0	9550	0	3	0	0	0	6	2	2	0	5	7	5	1915	1
4	60	84.0	14260	0	3	0	2	0	15	2	2	0	5	8	5	2000	2
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
1455	60	62.0	7917	3	3	0	4	0	8	2	2	0	5	6	5	1999	2
1456	20	85.0	13175	3	3	0	4	0	14	2	2	0	2	6	6	1978	1
1457	70	66.0	9042	3	3	0	4	0	6	2	2	0	5	7	9	1941	2
1458	20	68.0	9717	3	3	0	4	0	12	2	2	0	2	5	6	1950	1
1459	20	75.0	9937	3	3	0	4	0	7	2	2	0	2	5	6	1965	1

1460 rows × 50 columns

In [313]:
```
corr_df = data1.corr()
corr_df = corr_df[(abs(corr_df) >= 0.7)]
plt.figure(figsize=(30,40))
sns.heatmap(corr_df, annot=True, cmap='YlGn')
plt.show()
```


MSSubClass
LotFrontage
LotArea
LotShape
LandContour
Utilities
LotConfig
LandSlope
Neighborhood
Condition1
Condition2
BldgType
HouseStyle
OverallQual
OverallCond
YearBuilt
YearRemodAdd
RoofStyle
RoofMatl
Exterior1st
Exterior2nd
MasVnrArea
ExterQual
ExterCond
Foundation
BsmtQual
BsmtCond
BsmtFinSF1
1
0.75
0.79
0.85
1.00
0.95
0.90
0.85



In [314]:
```
data1[data1.columns[0:]].corr()['SalePrice'][:] # BldgType
```

Out[314]:
```
MSSubClass      -0.084284
LotFrontage      0.334901
LotArea          0.263843
LotShape        -0.255580
LandContour      0.015453
Utilities       -0.014314
LotConfig       -0.067396
LandSlope        0.051152
Neighborhood     0.210851
Condition1       0.091155
Condition2       0.007513
BldgType        -0.085591
HouseStyle       0.180163
OverallQual      0.790982
OverallCond     -0.077856
YearBuilt        0.522897
YearRemodAdd     0.507101
RoofStyle        0.222405
RoofMatl         0.132383
Exterior1st      0.103551
Exterior2nd      0.103766
MasVnrArea       0.446799
ExterQual       -0.636884
ExterCond        0.117303
Foundation       0.382479
BsmtQual        -0.620886
BsmtCond         0.015058
BsmtFinSF1       0.420835
BsmtUnfSF        0.169657
TotalBsmtSF      0.603084
Heating         -0.098812
CentralAir       0.251328
Electrical       0.234716
1stFlrSF         0.605852
GrLivArea        0.708624
BsmtFullBath     0.227122
BsmtHalfBath    -0.016844
FullBath         0.560664
HalfBath         0.284108
BedroomAbvGr     0.168213
KitchenAbvGr    -0.135907
KitchenQual     -0.589189
TotRmsAbvGrd     0.533723
Fireplaces       0.466929
GarageCars       0.640409
GarageArea       0.596139
MoSold           0.046432
YrSold          -0.028923
SaleCondition    0.213092
SalePrice        1.000000
Name: SalePrice, dtype: float64
```

In [315]:

```
sorted_mat = corr_df.unstack().sort_values().dropna()
sorted_mat
```

Out[315]:

```
GarageArea   GarageCars     0.701053
GarageCars   GarageArea     0.701053
SalePrice    GrLivArea      0.708624
GrLivArea    SalePrice      0.708624
MSSubClass   BldgType       0.746063
                              ...
Foundation   Foundation     1.000000
BsmtQual     BsmtQual       1.000000
BsmtFinSF1   BsmtFinSF1     1.000000
OverallCond  OverallCond    1.000000
SalePrice    SalePrice      1.000000
Length: 64, dtype: float64
```

In [ ]:

In [455]:

```
data2=data1.drop(["GarageCars","GrLivArea","BldgType","OverallQual","Exterior2nd","1stFlrSF"],axis=1)
data2
```

Out[455]:

	MSSubClass	LotFrontage	LotArea	LotShape	LandContour	Utilities	LotConfig	LandSlope	Neighborhood	Condition1	Condition2	HouseStyle	OverallCond	YearBuilt	YearRemodAdd	RoofStyle	RoofMa
0	60	65.0	8450	3	3	0	4	0	5	2	2	5	5	2003	2003	1	
1	20	80.0	9600	3	3	0	2	0	24	1	2	2	8	1976	1976	1	
2	60	68.0	11250	0	3	0	4	0	5	2	2	5	5	2001	2002	1	
3	70	60.0	9550	0	3	0	0	0	6	2	2	5	5	1915	1970	1	
4	60	84.0	14260	0	3	0	2	0	15	2	2	5	5	2000	2000	1	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	.
1455	60	62.0	7917	3	3	0	4	0	8	2	2	5	5	1999	2000	1	
1456	20	85.0	13175	3	3	0	4	0	14	2	2	2	6	1978	1988	1	
1457	70	66.0	9042	3	3	0	4	0	6	2	2	5	9	1941	2006	1	
1458	20	68.0	9717	3	3	0	4	0	12	2	2	2	6	1950	1996	3	
1459	20	75.0	9937	3	3	0	4	0	7	2	2	2	6	1965	1965	1	

1460 rows × 44 columns

In [317]:

```
data2["Utilities"].unique()
```

Out[317]:

```
array([0, 1])
```

In [456]:

```
x=data2.iloc[:,:-1].values
y=data2.iloc[:,-1].values
y
```

Out[456]:

```
array([208500, 181500, 223500, ..., 266500, 142125, 147500], dtype=int64)
```

In [457]:

```
from sklearn.preprocessing import StandardScaler
s1=StandardScaler()
x_scaled=s1.fit_transform(x)
#y_scaled=s1.fit_transform(y.reshape(-1,1))
```

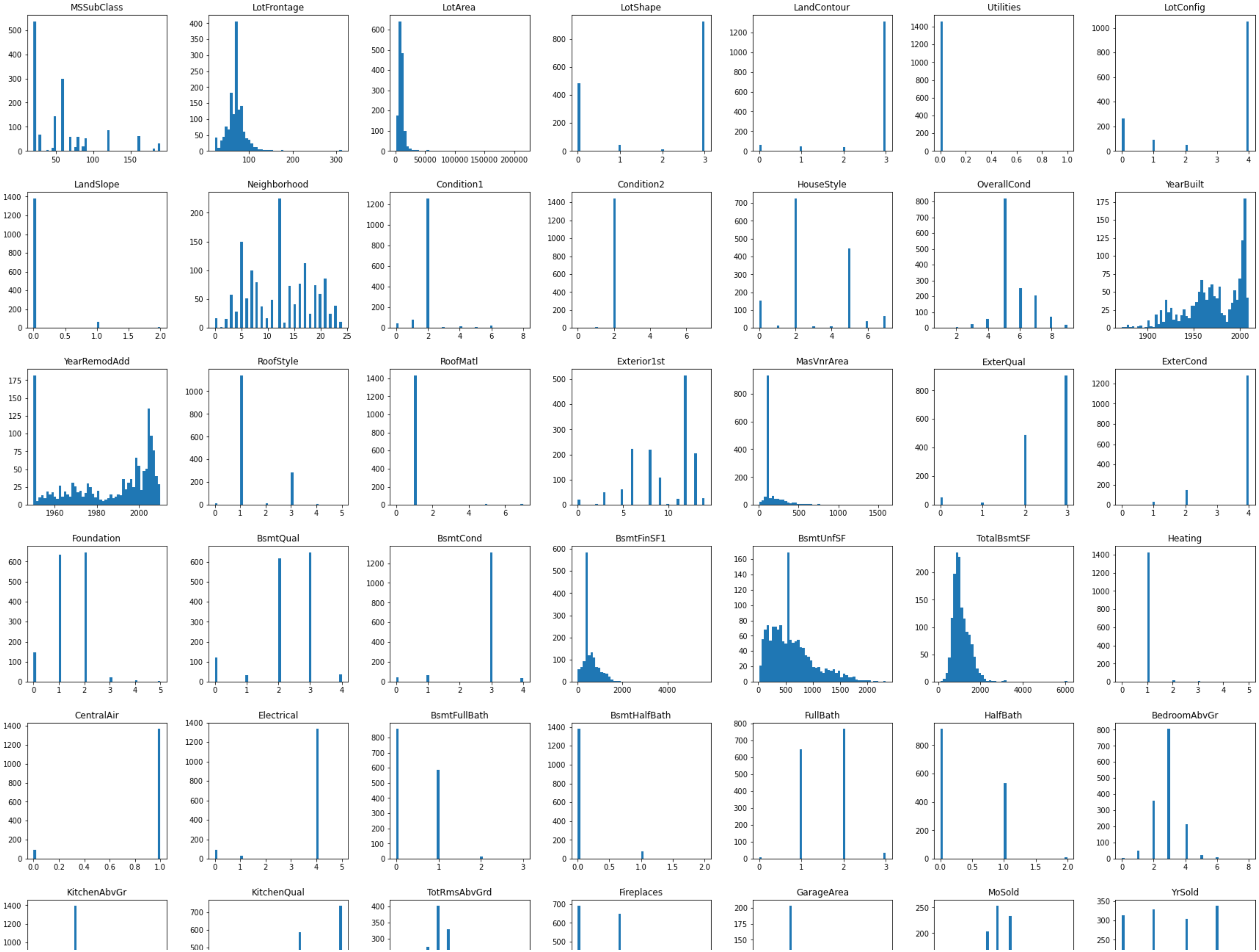
In [458]:

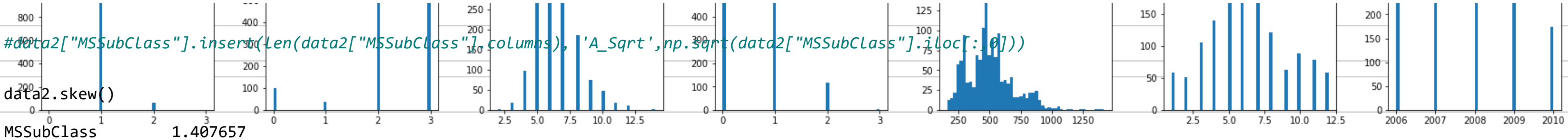
```
print("x_scaled",x_scaled)
```

```
x_scaled [[ 0.07337496 -0.22937175 -0.20714171 ... -1.5991111   0.13877749
   0.2085023 ]
 [-0.87256276  0.4519361  -0.09188637 ... -0.48911005 -0.61443862
   0.2085023 ]
 [ 0.07337496 -0.09311018  0.07347998 ...  0.99089135  0.13877749
   0.2085023 ]
 ...
 [ 0.30985939 -0.18395123 -0.14781027 ... -0.48911005  1.64520971
   0.2085023 ]
 [-0.87256276 -0.09311018 -0.08016039 ... -0.8591104   1.64520971
   0.2085023 ]
 [-0.87256276  0.22483348 -0.05811155 ... -0.1191097   0.13877749
   0.2085023 ]]
```

```
In [321]: # sk=pd.DataFrame(x_scaled)
          # sk.hist(grid=False,figsize=(30,20),bins=50)
          data2.hist(grid=False,figsize=(30,30),bins=50)
```

```
Out[321]: array([[<AxesSubplot:title={'center':'MSSubClass'}>,
                  <AxesSubplot:title={'center':'LotFrontage'}>,
                  <AxesSubplot:title={'center':'LotArea'}>,
                  <AxesSubplot:title={'center':'LotShape'}>,
                  <AxesSubplot:title={'center':'LandContour'}>,
                  <AxesSubplot:title={'center':'Utilities'}>,
                  <AxesSubplot:title={'center':'LotConfig'}>],
                 [<AxesSubplot:title={'center':'LandSlope'}>,
                  <AxesSubplot:title={'center':'Neighborhood'}>,
                  <AxesSubplot:title={'center':'Condition1'}>,
                  <AxesSubplot:title={'center':'Condition2'}>,
                  <AxesSubplot:title={'center':'HouseStyle'}>,
                  <AxesSubplot:title={'center':'OverallCond'}>,
                  <AxesSubplot:title={'center':'YearBuilt'}>],
                 [<AxesSubplot:title={'center':'YearRemodAdd'}>,
                  <AxesSubplot:title={'center':'RoofStyle'}>,
                  <AxesSubplot:title={'center':'RoofMatl'}>,
                  <AxesSubplot:title={'center':'Exterior1st'}>,
                  <AxesSubplot:title={'center':'MasVnrArea'}>,
                  <AxesSubplot:title={'center':'ExterQual'}>,
                  <AxesSubplot:title={'center':'ExterCond'}>],
                 [<AxesSubplot:title={'center':'Foundation'}>,
                  <AxesSubplot:title={'center':'BsmtQual'}>,
                  <AxesSubplot:title={'center':'BsmtCond'}>,
                  <AxesSubplot:title={'center':'BsmtFinSF1'}>,
                  <AxesSubplot:title={'center':'BsmtUnfSF'}>,
                  <AxesSubplot:title={'center':'TotalBsmtSF'}>,
                  <AxesSubplot:title={'center':'Heating'}>],
                 [<AxesSubplot:title={'center':'CentralAir'}>,
                  <AxesSubplot:title={'center':'Electrical'}>,
                  <AxesSubplot:title={'center':'BsmtFullBath'}>,
                  <AxesSubplot:title={'center':'BsmtHalfBath'}>,
                  <AxesSubplot:title={'center':'FullBath'}>,
                  <AxesSubplot:title={'center':'HalfBath'}>,
                  <AxesSubplot:title={'center':'BedroomAbvGr'}>],
                 [<AxesSubplot:title={'center':'KitchenAbvGr'}>,
                  <AxesSubplot:title={'center':'KitchenQual'}>,
                  <AxesSubplot:title={'center':'TotRmsAbvGrd'}>,
                  <AxesSubplot:title={'center':'Fireplaces'}>,
                  <AxesSubplot:title={'center':'GarageArea'}>,
                  <AxesSubplot:title={'center':'MoSold'}>,
                  <AxesSubplot:title={'center':'YrSold'}>],
                 [<AxesSubplot:title={'center':'SaleCondition'}>,
                  <AxesSubplot:title={'center':'SalePrice'}>, <AxesSubplot:>,
                  <AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>]],
                dtype=object)
```

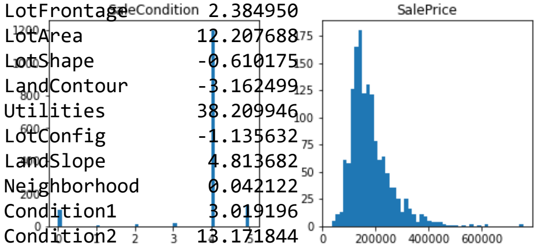

MSSubClass
0 100 200 300 400 500
50 100 150
LotFrontage
0 50 100 150 200 250 300 350 400
100 200 300
LotArea
0 100 200 300 400 500 600
0 50000 100000 150000 200000
LotShape
0 200 400 600 800
0 1 2 3
LandContour
0 200 400 600 800 1000 1200
0 1 2 3
Utilities
0 200 400 600 800 1000 1200 1400
0.0 0.2 0.4 0.6 0.8 1.0
LotConfig
0 200 400 600 800 1000
0 1 2 3 4
LandSlope
0 200 400 600 800 1000 1200 1400
0.0 0.5 1.0 1.5 2.0
Neighborhood
0 50 100 150 200
0 5 10 15 20 25
Condition1
0 200 400 600 800 1000 1200
0 2 4 6 8
Condition2
0 200 400 600 800 1000 1200 1400
0 2 4 6
HouseStyle
0 100 200 300 400 500 600 700
0 2 4 6
OverallCond
0 100 200 300 400 500 600 700 800
2 4 6 8
YearBuilt
0 25 50 75 100 125 150 175
1900 1950 2000
YearRemodAdd
0 25 50 75 100 125 150 175
1960 1980 2000
RoofStyle
0 200 400 600 800 1000
0 1 2 3 4 5
RoofMatl
0 200 400 600 800 1000 1200 1400
0 2 4 6
Exterior1st
0 100 200 300 400 500
0 5 10
MasVnrArea
0 200 400 600 800
0 500 1000 1500
ExterQual
0 200 400 600 800
0 1 2 3
ExterCond
0 200 400 600 800 1000 1200
0 1 2 3 4
Foundation
0 100 200 300 400 500 600
0 1 2 3 4 5
BsmtQual
0 100 200 300 400 500 600
0 1 2 3 4
BsmtCond
0 200 400 600 800 1000 1200
0 1 2 3 4
BsmtFinSF1
0 100 200 300 400 500 600
0 2000 4000
BsmtUnfSF
0 20 40 60 80 100 120 140 160
0 500 1000 1500 2000
TotalBsmtSF
0 50 100 150 200
0 2000 4000 6000
Heating
0 200 400 600 800 1000 1200 1400
0 1 2 3 4 5
CentralAir
0 200 400 600 800 1000 1200 1400
0.0 0.2 0.4 0.6 0.8 1.0
Electrical
0 200 400 600 800 1000 1200 1400
0 1 2 3 4 5
BsmtFullBath
0 100 200 300 400 500 600 700 800
0 1 2 3
BsmtHalfBath
0 200 400 600 800 1000 1200 1400
0.0 0.5 1.0 1.5 2.0
FullBath
0 100 200 300 400 500 600 700 800
0 1 2 3
HalfBath
0 200 400 600 800
0.0 0.5 1.0 1.5 2.0
BedroomAbvGr
0 100 200 300 400 500 600 700 800
0 2 4 6 8
KitchenAbvGr
1000 1200 1400
KitchenQual
500 600 700
TotRmsAbvGrd
300 350 400
Fireplaces
500 600 700
GarageArea
150 175 200
MoSold
200 250
YrSold
250 300 350



In [322]:
```
#data2["MSSubClass"].insert(len(data2["MSSubClass"].columns), 'A_Sqrt',np.sqrt(data2["MSSubClass"].iloc[:,0]))
```

In [323]:
```
data2.skew()
```

Out[323]:



```
MSSubClass        1.407657
LotFrontage       2.384950
LotArea          12.207688
LotShape         -0.610175
LandContour      -3.162499
Utilities        38.209946
LotConfig        -1.135632
LandSlope         4.813682
Neighborhood      0.042122
Condition1        3.019196
Condition2       13.171844
HouseStyle        0.306755
OverallCond       0.693067
YearBuilt        -0.613461
YearRemodAdd     -0.503562
RoofStyle         1.473796
RoofMatl          8.109402
Exterior1st      -0.726314
MasVnrArea        3.565835
ExterQual        -1.830265
ExterCond        -2.565305
Foundation        0.091217
BsmtQual         -1.114573
BsmtCond         -3.056021
BsmtFinSF1        2.986882
BsmtUnfSF         1.040159
TotalBsmtSF       2.203900
Heating           9.855100
CentralAir       -3.530386
Electrical       -3.059107
BsmtFullBath      0.596067
BsmtHalfBath      4.103403
FullBath          0.036562
HalfBath          0.675897
BedroomAbvGr      0.211790
KitchenAbvGr      4.488397
KitchenQual      -1.422808
TotRmsAbvGrd      0.676341
Fireplaces        0.649565
GarageArea        0.858354
MoSold            0.212053
YrSold            0.096269
SaleCondition    -2.741167
SalePrice         1.882876
dtype: float64
```

In [324]:
```
#data2["MSSubClass"]=data2.insert(len(data2.columns), 'B_Sqrt',np.sqrt(max(data2.iloc[:,0]+1) - data2.iloc[:, 0]))
```

In [325]:

```
data2["LandContour"]=np.sqrt(max(data2["LandContour"]+1) - (data2["LandContour"]))
data2["LotConfig"]=np.sqrt(max(data2["LotConfig"]+1) - (data2["LotConfig"]))
data2["ExterQual"]=np.sqrt(max(data2["ExterQual"]+1) - (data2["ExterQual"]))
data2["ExterCond"]=np.sqrt(max(data2["ExterCond"]+1) - (data2["ExterCond"]))
data2["BsmtQual"]=np.sqrt(max(data2["BsmtQual"]+1) - (data2["BsmtQual"]))
data2["BsmtCond"]=np.sqrt(max(data2["BsmtCond"]+1) - (data2["BsmtCond"]))
data2["CentralAir"]=np.sqrt(max(data2["CentralAir"]+1) - (data2["CentralAir"]))
data2["Electrical"]=np.sqrt(max(data2["Electrical"]+1) - (data2["Electrical"]))
data2["KitchenQual"]=np.sqrt(max(data2["KitchenQual"]+1) - (data2["KitchenQual"]))
data2["SaleCondition"]=np.sqrt(max(data2["SaleCondition"]+1) - (data2["SaleCondition"]))
```

In [326]:

```
#data2["LotArea"]=np.cbrt(data2["LotArea"])
```

In [443]:

```
data2["MSSubClass"]=np.sqrt(data2["MSSubClass"])
data2["LotArea"]=np.sqrt(data2["LotArea"])
data2["LotFrontage"]=np.sqrt(data2["LotFrontage"])
data2["Utilities"]=np.sqrt(data2["Utilities"])
data2["LandSlope"]=np.sqrt(data2["LandSlope"])
data2["Condition1"]=np.sqrt(data2["Condition1"])
data2["Condition2"]=np.sqrt(data2["Condition2"])
data2["RoofStyle"]=np.sqrt(data2["RoofStyle"])
data2["RoofMatl"]=np.sqrt(data2["RoofMatl"])
data2["MasVnrArea"]=np.sqrt(data2["MasVnrArea"])
data2["BsmtFinSF1"]=np.sqrt(data2["BsmtFinSF1"])
data2["BsmtUnfSF"]=np.sqrt(data2["BsmtUnfSF"])
data2["TotalBsmtSF"]=np.sqrt(data2["TotalBsmtSF"])
data2["BsmtHalfBath"]=np.sqrt(data2["BsmtHalfBath"])
data2["KitchenAbvGr"]=np.sqrt(data2["KitchenAbvGr"])
#data2["SalePrice"]=np.sqrt(data2["SalePrice"])
data2["Heating"]=np.sqrt(data2["Heating"])
```

In [ ]:

```
data2["Heating"]=np.sqrt(data2["Heating"])
```

In [328]:
```
data2.skew()
```

Out[328]:
```
MSSubClass        0.762448
LotFrontage       0.368212
LotArea           4.139714
LotShape         -0.610175
LandContour       3.017192
Utilities        38.209946
LotConfig         1.084958
LandSlope         4.141956
Neighborhood      0.042122
Condition1       -1.276478
Condition2       -0.287721
HouseStyle        0.306755
OverallCond       0.693067
YearBuilt        -0.613461
YearRemodAdd     -0.503562
RoofStyle         0.902857
RoofMatl          7.485299
Exterior1st      -0.726314
MasVnrArea        2.140237
ExterQual         1.302275
ExterCond         2.446621
Foundation        0.091217
BsmtQual          0.626380
BsmtCond          2.535087
BsmtFinSF1        0.362329
BsmtUnfSF         0.214098
TotalBsmtSF       0.705949
Heating           7.353695
CentralAir        3.530386
Electrical        3.026503
BsmtFullBath      0.596067
BsmtHalfBath      3.893994
FullBath          0.036562
HalfBath          0.675897
BedroomAbvGr      0.211790
KitchenAbvGr      3.105498
KitchenQual       0.962551
TotRmsAbvGrd      0.676341
Fireplaces        0.649565
GarageArea        0.858354
MoSold            0.212053
YrSold            0.096269
SaleCondition     2.164717
SalePrice         0.943153
dtype: float64
```

In [452]:
```
#data2.insert(len(data2.columns), 'A_Sqrt',np.sqrt(data2.iloc[:,0]))
```

In [453]:
```
data2.skew()
```

Out[453]:
```
MSSubClass        0.269695
LotFrontage      -0.836117
LotArea          -0.028086
LotShape         -0.706421
LandContour       2.914990
Utilities        38.209946
LotConfig         1.051414
LandSlope         3.976628
Neighborhood     -3.983562
Condition1       -5.211649
Condition2      -26.657355
HouseStyle       -2.272040
OverallCond      -0.309048
YearBuilt        -0.636494
YearRemodAdd     -0.509767
RoofStyle        -9.840968
RoofMatl        -31.504028
Exterior1st      -5.893393
MasVnrArea        0.435773
ExterQual         0.994446
ExterCond         2.389176
Foundation       -2.486902
BsmtQual          0.140004
BsmtCond          1.821019
BsmtFinSF1       -1.845415
BsmtUnfSF        -0.746153
TotalBsmtSF      -0.200324
Heating         -35.109413
CentralAir        3.530386
Electrical        2.994762
BsmtFullBath      0.353221
BsmtHalfBath      3.859546
FullBath         -6.348436
HalfBath          0.520150
BedroomAbvGr     -7.776898
KitchenAbvGr    -35.083794
KitchenQual       0.649975
TotRmsAbvGrd     -0.039016
Fireplaces       -0.096208
GarageArea       -0.089795
MoSold           -0.995308
YrSold            0.095607
SaleCondition     1.375443
SalePrice         0.259360
A_Sqrt            0.269695
dtype: float64
```

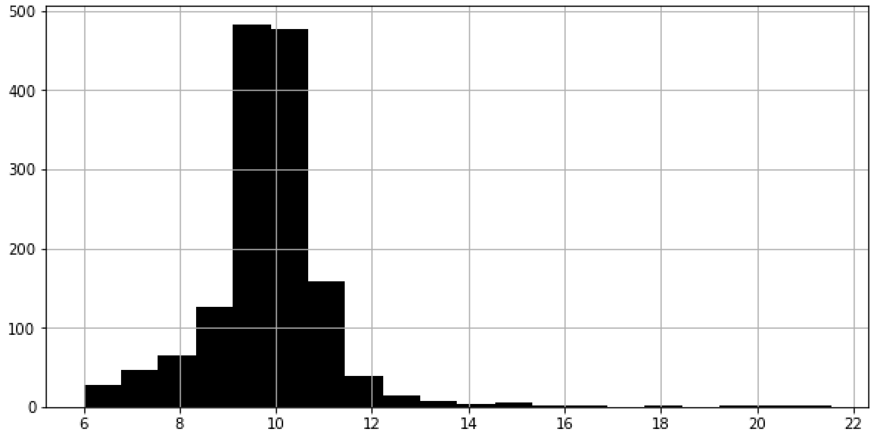
In [ ]:
```

```

```
In [331]: data2.skew()
```

```
Out[331]: MSSubClass        0.762448
          LotFrontage       0.368212
          LotArea           4.139714
          LotShape         -0.610175
          LandContour       3.017192
          Utilities        38.209946
          LotConfig         1.084958
          LandSlope         4.141956
          Neighborhood      0.042122
          Condition1       -1.276478
          Condition2       -0.287721
          HouseStyle        0.306755
          OverallCond       0.693067
          YearBuilt        -0.613461
          YearRemodAdd     -0.503562
          RoofStyle         0.902857
          RoofMatl          7.485299
          Exterior1st      -0.726314
          MasVnrArea        2.140237
          ExterQual         1.302275
          ExterCond         2.446621
          Foundation        0.091217
          BsmtQual          0.626380
          BsmtCond          2.535087
          BsmtFinSF1        0.362329
          BsmtUnfSF         0.214098
          TotalBsmtSF       0.705949
          Heating           7.353695
          CentralAir        3.530386
          Electrical        3.026503
          BsmtFullBath      0.596067
          BsmtHalfBath      3.893994
          FullBath          0.036562
          HalfBath          0.675897
          BedroomAbvGr      0.211790
          KitchenAbvGr      3.105498
          KitchenQual       0.962551
          TotRmsAbvGrd      0.676341
          Fireplaces        0.649565
          GarageArea        0.858354
          MoSold            0.212053
          YrSold            0.096269
          SaleCondition     2.164717
          SalePrice         0.943153
          A_Sqrt            0.475676
          dtype: float64
```

In [332]:
```
data2['LotArea'] = np.sqrt(data2['LotArea'])
data2['LotArea'].hist(bins = 20, figsize = (10,5), color = 'black')
```

Out[332]:
```
<AxesSubplot:>
```

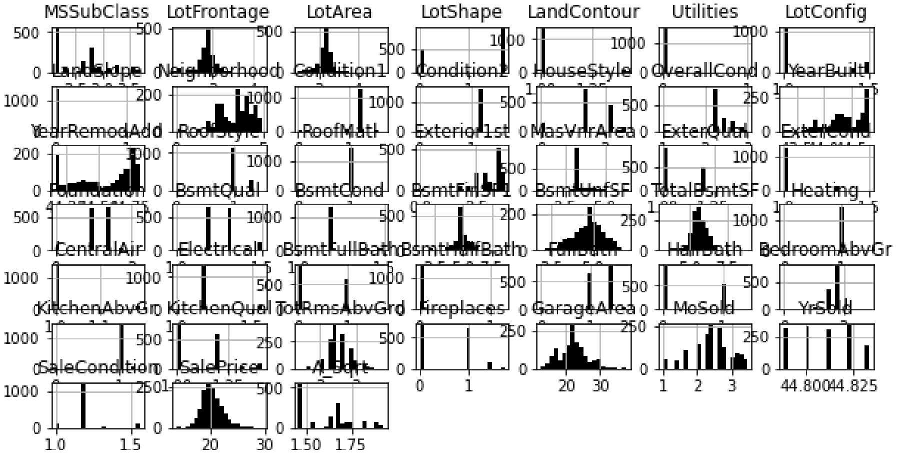


In [333]:

```
data2 = np.sqrt(data2)
data2.hist(bins = 20, figsize = (10,5), color = 'black')
```

Out[333]:

```
array([[<AxesSubplot:title={'center':'MSSubClass'}>,
        <AxesSubplot:title={'center':'LotFrontage'}>,
        <AxesSubplot:title={'center':'LotArea'}>,
        <AxesSubplot:title={'center':'LotShape'}>,
        <AxesSubplot:title={'center':'LandContour'}>,
        <AxesSubplot:title={'center':'Utilities'}>,
        <AxesSubplot:title={'center':'LotConfig'}>],
       [<AxesSubplot:title={'center':'LandSlope'}>,
        <AxesSubplot:title={'center':'Neighborhood'}>,
        <AxesSubplot:title={'center':'Condition1'}>,
        <AxesSubplot:title={'center':'Condition2'}>,
        <AxesSubplot:title={'center':'HouseStyle'}>,
        <AxesSubplot:title={'center':'OverallCond'}>,
        <AxesSubplot:title={'center':'YearBuilt'}>],
       [<AxesSubplot:title={'center':'YearRemodAdd'}>,
        <AxesSubplot:title={'center':'RoofStyle'}>,
        <AxesSubplot:title={'center':'RoofMatl'}>,
        <AxesSubplot:title={'center':'Exterior1st'}>,
        <AxesSubplot:title={'center':'MasVnrArea'}>,
        <AxesSubplot:title={'center':'ExterQual'}>,
        <AxesSubplot:title={'center':'ExterCond'}>],
       [<AxesSubplot:title={'center':'Foundation'}>,
        <AxesSubplot:title={'center':'BsmtQual'}>,
        <AxesSubplot:title={'center':'BsmtCond'}>,
        <AxesSubplot:title={'center':'BsmtFinSF1'}>,
        <AxesSubplot:title={'center':'BsmtUnfSF'}>,
        <AxesSubplot:title={'center':'TotalBsmtSF'}>,
        <AxesSubplot:title={'center':'Heating'}>],
       [<AxesSubplot:title={'center':'CentralAir'}>,
        <AxesSubplot:title={'center':'Electrical'}>,
        <AxesSubplot:title={'center':'BsmtFullBath'}>,
        <AxesSubplot:title={'center':'BsmtHalfBath'}>,
        <AxesSubplot:title={'center':'FullBath'}>,
        <AxesSubplot:title={'center':'HalfBath'}>,
        <AxesSubplot:title={'center':'BedroomAbvGr'}>],
       [<AxesSubplot:title={'center':'KitchenAbvGr'}>,
        <AxesSubplot:title={'center':'KitchenQual'}>,
        <AxesSubplot:title={'center':'TotRmsAbvGrd'}>,
        <AxesSubplot:title={'center':'Fireplaces'}>,
        <AxesSubplot:title={'center':'GarageArea'}>,
        <AxesSubplot:title={'center':'MoSold'}>,
        <AxesSubplot:title={'center':'YrSold'}>],
       [<AxesSubplot:title={'center':'SaleCondition'}>,
        <AxesSubplot:title={'center':'SalePrice'}>,
        <AxesSubplot:title={'center':'A_Sqrt'}>, <AxesSubplot:>,
        <AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>]], dtype=object)
```


MSSubClass
LotFrontage
LotArea
LotShape
LandContour
Utilities
LotConfig
LandSlope
Neighborhood
Condition1
Condition2
HouseStyle
OverallCond
YearBuilt
YearRemodAdd
RoofStyle
RoofMatl
Exterior1st
MasVnrArea
ExterQual
ExterCond
Foundation
BsmtQual
BsmtCond
BsmtFinSF1
BsmtUnfSF
TotalBsmtSF
Heating
CentralAir
Electrical
BsmtFullBath
BsmtHalfBath
FullBath
HalfBath
BedroomAbvGr
KitchenAbvGr
KitchenQual
TotRmsAbvGrd
Fireplaces
GarageArea
MoSold
YrSold
SaleCondition
SalePrice
All_Sqft
500
0
1000
200
250
1.0
1.5
20
30
1.50
1.75
0
1
2
3
44.800
44.825

In [459]:

```
data2.skew()
```

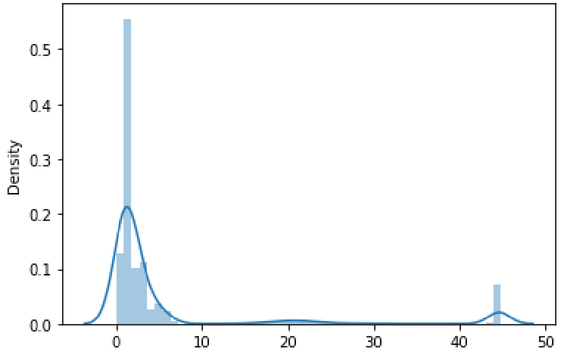
Out[459]:

```
MSSubClass        1.407657
LotFrontage       2.384950
LotArea          12.207688
LotShape         -0.610175
LandContour      -3.162499
Utilities        38.209946
LotConfig        -1.135632
LandSlope         4.813682
Neighborhood      0.042122
Condition1        3.019196
Condition2       13.171844
HouseStyle        0.306755
OverallCond       0.693067
YearBuilt        -0.613461
YearRemodAdd     -0.503562
RoofStyle         1.473796
RoofMatl          8.109402
Exterior1st      -0.726314
MasVnrArea        3.565835
ExterQual        -1.830265
ExterCond        -2.565305
Foundation        0.091217
BsmtQual         -1.114573
BsmtCond         -3.056021
BsmtFinSF1        2.986882
BsmtUnfSF         1.040159
TotalBsmtSF       2.203900
Heating           9.855100
CentralAir       -3.530386
Electrical       -3.059107
BsmtFullBath      0.596067
BsmtHalfBath      4.103403
FullBath          0.036562
HalfBath          0.675897
BedroomAbvGr      0.211790
KitchenAbvGr      4.488397
KitchenQual      -1.422808
TotRmsAbvGrd      0.676341
Fireplaces        0.649565
GarageArea        0.858354
MoSold            0.212053
YrSold            0.096269
SaleCondition    -2.741167
SalePrice         1.882876
dtype: float64
```

```
In [335]: sns.distplot(data2)
```

```
F:\Rahul\lib\site-packages\seaborn\distributions.py:2551: FutureWarning: `distplot` is a deprecated function and will be removed in a future version. Please adapt your
code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

```
Out[335]: <AxesSubplot:ylabel='Density'>
```

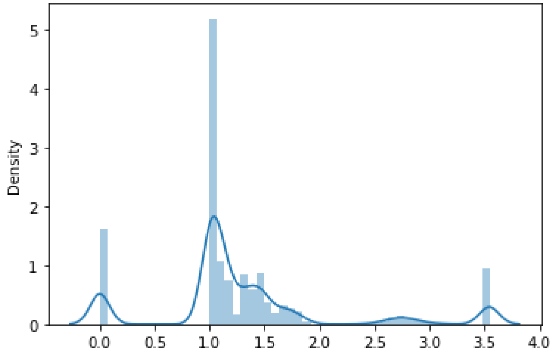


```
In [336]: data2=np.cbrt(data2)
```

```
In [337]: sns.distplot(data2)
```

```
F:\Rahul\lib\site-packages\seaborn\distributions.py:2551: FutureWarning: `distplot` is a deprecated function and will be removed in a future version. Please adapt your
code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

```
Out[337]: <AxesSubplot:ylabel='Density'>
```



In [338]:
```
data2
```

Out[338]:

	MSSubClass	LotFrontage	LotArea	LotShape	LandContour	Utilities	LotConfig	LandSlope	Neighborhood	Condition1	Condition2	HouseStyle	OverallCond	YearBuilt	YearRemodAdd	RoofStyle	RoofMa
0	1.406628	1.416042	1.457535	1.200937	1.0	0.0	1.000000	0.0	1.307660	1.059463	1.059463	1.307660	1.307660	3.550423	3.550423	1.000000	1
1	1.283569	1.440757	1.465305	1.200937	1.0	0.0	1.095873	0.0	1.698381	1.000000	1.059463	1.122462	1.414214	3.542402	3.542402	1.000000	1
2	1.406628	1.421376	1.475020	0.000000	1.0	0.0	1.000000	0.0	1.307660	1.059463	1.059463	1.307660	1.307660	3.549832	3.550128	1.000000	1
3	1.424814	1.406628	1.464986	0.000000	1.0	0.0	1.143530	0.0	1.348006	1.059463	1.059463	1.307660	1.307660	3.523937	3.540607	1.000000	1
4	1.406628	1.446627	1.489664	0.000000	1.0	0.0	1.095873	0.0	1.570418	1.059463	1.059463	1.307660	1.307660	3.549537	3.549537	1.000000	1
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
1455	1.406628	1.410477	1.453584	1.200937	1.0	0.0	1.000000	0.0	1.414214	1.059463	1.059463	1.307660	1.307660	3.549241	3.549537	1.000000	1
1456	1.283569	1.448055	1.484760	1.200937	1.0	0.0	1.000000	0.0	1.552463	1.059463	1.059463	1.122462	1.348006	3.542999	3.545978	1.000000	1
1457	1.424814	1.417845	1.461653	1.200937	1.0	0.0	1.000000	0.0	1.348006	1.059463	1.059463	1.307660	1.442250	3.531866	3.551309	1.000000	1
1458	1.283569	1.421376	1.466045	1.200937	1.0	0.0	1.000000	0.0	1.513086	1.059463	1.059463	1.122462	1.348006	3.534590	3.548352	1.095873	1
1459	1.283569	1.433029	1.467413	1.200937	1.0	0.0	1.000000	0.0	1.383088	1.059463	1.059463	1.122462	1.348006	3.539108	3.539108	1.000000	1

1460 rows × 45 columns

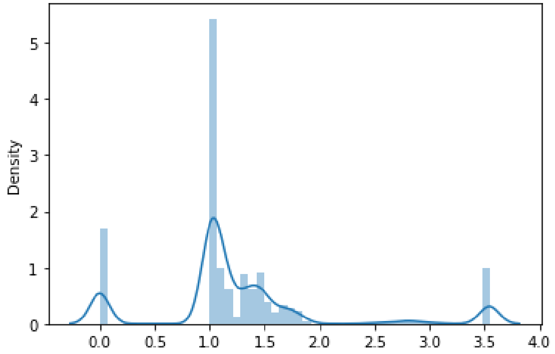
In [349]:
```
x=data2.iloc[:,:-2]
y=data2.iloc[:,-2]
y
```

Out[349]:
```
0       2.774973
1       2.743088
2       2.791085
3       2.684380
4       2.817269
          ...   
1455    2.734764
1456    2.776632
1457    2.832314
1458    2.687752
1459    2.696079
Name: SalePrice, Length: 1460, dtype: float64
```

In [340]:
```
sns.distplot(x)
```

```
F:\Rahul\lib\site-packages\seaborn\distributions.py:2551: FutureWarning: `distplot` is a deprecated function and will be removed in a future version. Please adapt your
code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

Out[340]:
```
<AxesSubplot:ylabel='Density'>
```

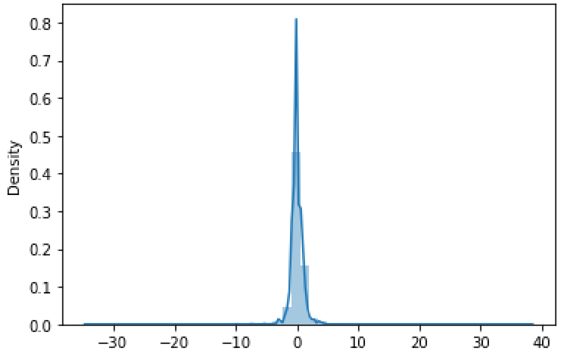


In [342]:
```
from sklearn.preprocessing import StandardScaler
s1=StandardScaler()
x1=s1.fit_transform(x)
```

In [343]:
```
sns.distplot(x1)
```

```
F:\Rahul\lib\site-packages\seaborn\distributions.py:2551: FutureWarning: `distplot` is a deprecated function and will be removed in a future version. Please adapt your
code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

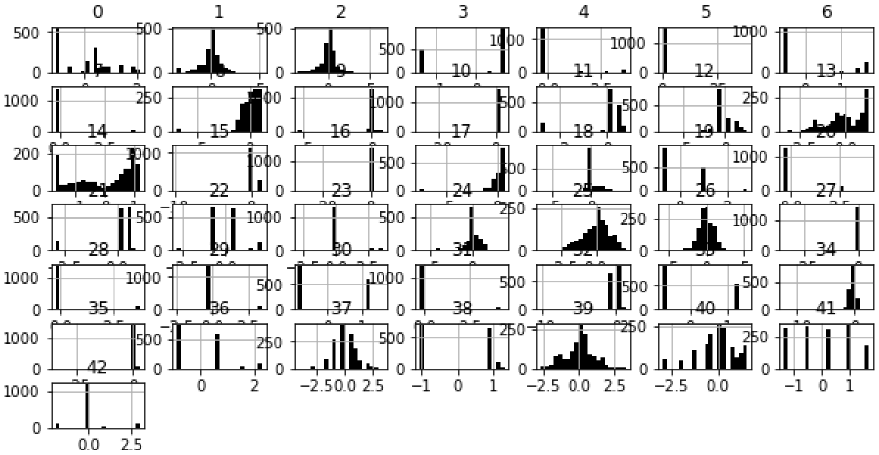
Out[343]:
```
<AxesSubplot:ylabel='Density'>
```



In [345]:
```
x2=pd.DataFrame(x1)
```

```
In [346]: x2.hist(bins = 20, figsize = (10,5), color = 'black')
```

```
Out[346]: array([[<AxesSubplot:title={'center':'0'}>,
                  <AxesSubplot:title={'center':'1'}>,
                  <AxesSubplot:title={'center':'2'}>,
                  <AxesSubplot:title={'center':'3'}>,
                  <AxesSubplot:title={'center':'4'}>,
                  <AxesSubplot:title={'center':'5'}>,
                  <AxesSubplot:title={'center':'6'}>],
                 [<AxesSubplot:title={'center':'7'}>,
                  <AxesSubplot:title={'center':'8'}>,
                  <AxesSubplot:title={'center':'9'}>,
                  <AxesSubplot:title={'center':'10'}>,
                  <AxesSubplot:title={'center':'11'}>,
                  <AxesSubplot:title={'center':'12'}>,
                  <AxesSubplot:title={'center':'13'}>],
                 [<AxesSubplot:title={'center':'14'}>,
                  <AxesSubplot:title={'center':'15'}>,
                  <AxesSubplot:title={'center':'16'}>,
                  <AxesSubplot:title={'center':'17'}>,
                  <AxesSubplot:title={'center':'18'}>,
                  <AxesSubplot:title={'center':'19'}>,
                  <AxesSubplot:title={'center':'20'}>],
                 [<AxesSubplot:title={'center':'21'}>,
                  <AxesSubplot:title={'center':'22'}>,
                  <AxesSubplot:title={'center':'23'}>,
                  <AxesSubplot:title={'center':'24'}>,
                  <AxesSubplot:title={'center':'25'}>,
                  <AxesSubplot:title={'center':'26'}>,
                  <AxesSubplot:title={'center':'27'}>],
                 [<AxesSubplot:title={'center':'28'}>,
                  <AxesSubplot:title={'center':'29'}>,
                  <AxesSubplot:title={'center':'30'}>,
                  <AxesSubplot:title={'center':'31'}>,
                  <AxesSubplot:title={'center':'32'}>,
                  <AxesSubplot:title={'center':'33'}>,
                  <AxesSubplot:title={'center':'34'}>],
                 [<AxesSubplot:title={'center':'35'}>,
                  <AxesSubplot:title={'center':'36'}>,
                  <AxesSubplot:title={'center':'37'}>,
                  <AxesSubplot:title={'center':'38'}>,
                  <AxesSubplot:title={'center':'39'}>,
                  <AxesSubplot:title={'center':'40'}>,
                  <AxesSubplot:title={'center':'41'}>],
                 [<AxesSubplot:title={'center':'42'}>, <AxesSubplot:>,
                  <AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>, <AxesSubplot:>,
                  <AxesSubplot:>]], dtype=object)
```


0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
−2.5 0.0 2.5
−1 0 1
0 2
0.0 2.5

In [347]:

```
x2.skew()
```

Out[347]:

```
0      0.308239
1     -0.739661
2      0.085805
3     -0.706421
4      2.914990
5     38.209946
6      1.051414
7      3.979267
8     -3.983562
9     -5.110249
10   -25.643333
11    -2.272040
12    -0.309048
13    -0.636494
14    -0.509767
15    -8.198637
16   -17.969334
17    -5.893393
18     0.694437
19     0.994446
20     2.389176
21    -2.486902
22     0.140004
23     1.821019
24    -1.603097
25    -0.643592
26    -0.115715
27   -27.147839
28     3.530386
29     2.994762
30     0.353221
31     3.860003
32    -6.348436
33     0.520150
34    -7.776898
35   -27.429523
36     0.649975
37    -0.039016
38    -0.096208
39    -0.089795
40    -0.995308
41     0.095607
42     1.375443
dtype: float64
```

In [450]:

```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x1,y,test_size=0.25,random_state=25)
```

In [451]:

```
print(y_train)
```

```
1276    2.717229
1217    2.797209
1036    2.872433
1320    2.709419
80      2.757762
          ...
1341    2.707246
143     2.769932
474     2.818206
318     2.826492
1156    2.741065
Name: SalePrice, Length: 1095, dtype: float64
```

In [355]:

```
from sklearn.ensemble import RandomForestRegressor
r1=RandomForestRegressor()
model=r1.fit(x_train,y_train)
```

In [356]:

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

```
Out[356]: array([2.89558737, 2.8609608 , 2.72870184, 2.82442986, 2.73308414,
                 2.7666847 , 2.76919563, 2.81413435, 2.73602404, 2.74657472,
                 2.87715106, 2.70928637, 2.87101666, 2.8318737 , 2.85738989,
                 2.74719592, 2.71624017, 2.68496609, 2.71541865, 2.79936644,
                 2.66502192, 2.9379566 , 2.73090894, 2.73611977, 2.65049524,
                 2.62005026, 2.70041685, 2.87345055, 2.86413529, 2.65068703,
                 2.77398078, 2.7363107 , 2.65535875, 2.88233673, 2.73674205,
                 2.77183701, 2.81129063, 2.72878339, 2.6447669 , 2.66023964,
                 2.77360139, 2.75056557, 2.76384462, 2.68434657, 2.73777882,
                 2.75189869, 2.71938884, 2.79461772, 2.64729976, 2.78325777,
                 2.71803749, 2.64382602, 2.63298156, 2.76067396, 2.77606528,
                 2.59530131, 2.76167274, 2.6634013 , 2.74565273, 2.77836283,
                 2.64805256, 2.69763949, 2.5902493 , 2.8160018 , 2.82241963,
                 2.75550319, 2.6892658 , 2.81212089, 2.74280056, 2.62602897,
                 2.68684137, 2.87945594, 2.78503813, 2.81758834, 2.76248889,
                 2.67594863, 2.61153089, 2.68498192, 2.67283058, 2.92511277,
                 2.68917067, 2.60003624, 2.70331042, 2.68622897, 2.74615719,
                 2.67027273, 2.77386046, 2.66409594, 2.9404404 , 2.65190196,
                 2.81727694, 2.68932268, 2.82849466, 2.67823286, 2.78984467,
                 2.78872017, 2.80905895, 2.76435831, 2.6867273 , 2.71175809,
                 2.71811821, 2.74701648, 2.7401438 , 2.72014526, 2.66947468,
                 2.7339536 , 2.64991129, 2.66734659, 2.68500538, 2.75753876,
                 2.74918932, 2.85965451, 2.62832388, 2.65042825, 2.78059553,
                 2.77782851, 2.77551284, 2.60818391, 2.75988043, 2.66848368,
                 2.91940828, 2.72529063, 2.62622346, 2.65056093, 2.76503618,
                 2.70791682, 2.78426368, 2.74021632, 2.71115243, 2.78244742,
                 2.53397713, 2.67988836, 2.59939592, 2.83234732, 2.74221019,
                 2.79458615, 2.65080585, 2.73583523, 2.72761873, 2.72724806,
                 2.7396986 , 2.62696622, 2.85628077, 2.68226056, 2.79353517,
                 2.63435519, 2.64536514, 2.6779073 , 2.80716227, 2.7725951 ,
                 2.83304688, 2.78798465, 2.66644836, 2.70651985, 2.94645305,
                 2.82067766, 2.69241529, 2.88112504, 2.66824969, 2.76354157,
                 2.78867347, 2.81112214, 2.67473305, 2.74485899, 2.66747724,
                 2.81410152, 2.74511204, 2.70580786, 2.76402975, 2.64786385,
                 2.68284493, 2.7234062 , 2.68855194, 2.76331852, 2.6402006 ,
                 2.66808221, 2.69009234, 2.68406889, 2.66316383, 2.70563041,
                 2.76077867, 2.66695286, 2.85910105, 2.61603249, 2.7049709 ,
                 2.66704312, 2.65467445, 2.77118633, 2.76985777, 2.82687068,
                 2.78373741, 2.65183476, 2.65715116, 2.77439104, 2.61360892,
                 2.70219442, 2.7085913 , 2.71238791, 2.85107911, 2.65162653,
                 2.71232554, 2.57695928, 2.5672688 , 2.68980273, 2.64899294,
                 2.72989543, 2.68227558, 2.74408778, 2.72978749, 2.7076687 ,
                 2.77678269, 2.82767811, 2.72440377, 2.68568704, 2.80997178,
                 2.73467706, 2.65581823, 2.63927201, 2.80156893, 2.90889293,
                 2.65252478, 2.77506483, 2.71934315, 2.6754719 , 2.83133628,
                 2.91959062, 2.67705963, 2.71164741, 2.78613425, 2.67805737,
                 2.76628489, 2.70391002, 2.73442028, 2.73984941, 2.80946982,
                 2.69523381, 2.66661998, 2.82082948, 2.73922439, 2.67056256,
                 2.84657825, 2.68349584, 2.6647458 , 2.76165648, 2.69398953,
                 2.73087331, 2.7038098 , 2.91663314, 2.68460532, 2.92304118,
                 2.70064948, 2.74022108, 2.81559073, 2.68625294, 2.59130212,
                 2.75746055, 2.67943602, 2.64744059, 2.6995501 , 2.7159473 ,
                 2.85331793, 2.77101104, 2.66629107, 2.65157196, 2.892263  ,
                 2.68801902, 2.89284523, 2.64338864, 2.73588715, 2.76056868,
                 2.75779715, 2.61290288, 2.7667422 , 2.65785843, 2.7638479 ,
                 2.66377273, 2.85204283, 2.69313495, 2.65580731, 2.7148608 ,
                 2.61648975, 2.61773096, 2.68944724, 2.6554663 , 2.6720377 ,
                 2.65590765, 2.77174379, 2.73353221, 2.8170845 , 2.64335523,
                 2.78408902, 2.60733953, 2.65953406, 2.79982832, 2.78971305,
                 2.67275564, 2.59853054, 2.71706784, 2.83144174, 2.87939113,
                 2.78249633, 2.7626758 , 2.73635276, 2.62934031, 2.67024979,
                 2.68952954, 2.70457137, 2.67121257, 2.66260813, 2.74200995,
                 2.64355543, 2.67720241, 2.65146983, 2.58834245, 2.64374271,
                 2.65201297, 2.71910611, 2.73379737, 2.76032921, 2.7407884 ,
                 2.6853539 , 2.70416558, 2.70952011, 2.65769046, 2.59035084,
                 2.73747571, 2.62422384, 2.67744993, 2.6557038 , 2.70170236,
                 2.75776396, 2.84531939, 2.61726791, 2.79024394, 2.80412949,
                 2.76625742, 2.8073853 , 2.78974502, 2.75303125, 2.71379995,
                 2.68560097, 2.6463054 , 2.81885015, 2.74401954, 2.88800663,
                 2.75498784, 2.76261919, 2.72710806, 2.65690857, 2.764514  ,
                 2.73026291, 2.67019576, 2.65804489, 2.96403921, 2.72785076,
                 2.74143046, 2.77259022, 2.6474107 , 2.69620599, 2.64847915,
                 2.91106894, 2.65207336, 2.72709247, 2.80523646, 2.8580758 ])
```

In [357]:

```
from sklearn.metrics import r2_score
print(r2_score(y_pred,y_test))
```

```
0.7686601026263161
```

In [ ]:

```
#Test Data
```

In [408]:

```
test.shape
```

Out[408]:

```
(1459, 80)
```

In [409]:

```
test.head()
```

Out[409]:

	Id	MSSubClass	MSZoning	LotFrontage	LotArea	Street	Alley	LotShape	LandContour	Utilities	LotConfig	LandSlope	Neighborhood	Condition1	Condition2	BldgType	HouseStyle	OverallQual	Overall
0	1461	20	RH	80.0	11622	Pave	NaN	Reg	Lvl	AllPub	Inside	Gtl	NAmes	Feedr	Norm	1Fam	1Story	5	
1	1462	20	RL	81.0	14267	Pave	NaN	IR1	Lvl	AllPub	Corner	Gtl	NAmes	Norm	Norm	1Fam	1Story	6	
2	1463	60	RL	74.0	13830	Pave	NaN	IR1	Lvl	AllPub	Inside	Gtl	Gilbert	Norm	Norm	1Fam	2Story	5	
3	1464	60	RL	78.0	9978	Pave	NaN	IR1	Lvl	AllPub	Inside	Gtl	Gilbert	Norm	Norm	1Fam	2Story	6	
4	1465	120	RL	43.0	5005	Pave	NaN	IR1	HLS	AllPub	Inside	Gtl	StoneBr	Norm	Norm	TwnhsE	1Story	8	

In [410]:

```
test2=test.drop(["Id","MSZoning","Street","Alley","MasVnrType","LowQualFinSF","BsmtExposure","BsmtFinType1","BsmtFinType2","EnclosedPorch","3SsnPorch","ScreenPorch","Poo
"HeatingQC","FireplaceQu","Functional","FireplaceQu","GarageType","GarageFinish","GarageQual","GarageCond","PavedDrive","PoolQC","Fence","MiscFeature","SaleType","BsmtFi
,"2ndFlrSF","GarageYrBlt",  "MiscVal","OpenPorchSF","WoodDeckSF","GarageCars","GrLivArea","BldgType","OverallQual","Exterior2nd","1stFlrSF"],axis=1)
```

In [444]:

```
test2.shape
```

Out[444]:

```
(1459, 43)
```

In [445]:

```
# Label Enconding Techniques ::
label_encoders={}
from sklearn.preprocessing import LabelEncoder
le=LabelEncoder()
for i in test2.columns:
    if test2[i].dtype=="object":
        label_encoders[i] = LabelEncoder()
        test2[i] = label_encoders[i].fit_transform(test2[i])

    print(i)
```

```
MSSubClass
LotFrontage
LotArea
LotShape
LandContour
Utilities
LotConfig
LandSlope
Neighborhood
Condition1
Condition2
HouseStyle
OverallCond
YearBuilt
YearRemodAdd
RoofStyle
RoofMatl
Exterior1st
MasVnrArea
ExterQual
ExterCond
Foundation
BsmtQual
BsmtCond
BsmtFinSF1
BsmtUnfSF
TotalBsmtSF
Heating
CentralAir
Electrical
BsmtFullBath
BsmtHalfBath
FullBath
HalfBath
BedroomAbvGr
KitchenAbvGr
KitchenQual
TotRmsAbvGrd
Fireplaces
GarageArea
MoSold
YrSold
SaleCondition
```

In [413]:

```
test2.shape
```

Out[413]:

```
(1459, 43)
```

In [414]:

```
test2.isna().sum()
```

Out[414]:

```
MSSubClass         0
LotFrontage      227
LotArea            0
LotShape           0
LandContour        0
Utilities          0
LotConfig          0
LandSlope          0
Neighborhood       0
Condition1         0
Condition2         0
HouseStyle         0
OverallCond        0
YearBuilt          0
YearRemodAdd       0
RoofStyle          0
RoofMatl           0
Exterior1st        0
MasVnrArea        15
ExterQual          0
ExterCond          0
Foundation         0
BsmtQual           0
BsmtCond           0
BsmtFinSF1         1
BsmtUnfSF          1
TotalBsmtSF        1
Heating            0
CentralAir         0
Electrical         0
BsmtFullBath       2
BsmtHalfBath       2
FullBath           0
HalfBath           0
BedroomAbvGr       0
KitchenAbvGr       0
KitchenQual        0
TotRmsAbvGrd       0
Fireplaces         0
GarageArea         1
MoSold             0
YrSold             0
SaleCondition      0
dtype: int64
```

In [415]:

```
##Replacing Nan values
test2["LotFrontage"]=test2["LotFrontage"].replace(np.nan,test2["LotFrontage"].mean())
test2["MasVnrArea"]=test2["MasVnrArea"].replace(np.nan,test2["MasVnrArea"].mean())
test2["BsmtFinSF1"]=test2["BsmtFinSF1"].replace(np.nan,test2["BsmtFinSF1"].mean())
test2["BsmtUnfSF"]=test2["BsmtUnfSF"].replace(np.nan,test2["BsmtUnfSF"].mean())
test2["TotalBsmtSF"]=test2["TotalBsmtSF"].replace(np.nan,test2["TotalBsmtSF"].mean())
test2["BsmtFullBath"]=test2["BsmtFullBath"].replace(np.nan,test2["BsmtFullBath"].mean())
test2["LotFrontage"]=test2["LotFrontage"].replace(np.nan,test2["LotFrontage"].mean())
test2["BsmtHalfBath"]=test2["BsmtHalfBath"].replace(np.nan,test2["BsmtHalfBath"].mean())
test2["GarageArea"]=test2["GarageArea"].replace(np.nan,test2["GarageArea"].mean())
```

In [416]:
```
test2.isna().sum()
```

Out[416]:
```
MSSubClass       0
LotFrontage      0
LotArea          0
LotShape         0
LandContour      0
Utilities        0
LotConfig        0
LandSlope        0
Neighborhood     0
Condition1       0
Condition2       0
HouseStyle       0
OverallCond      0
YearBuilt        0
YearRemodAdd     0
RoofStyle        0
RoofMatl         0
Exterior1st      0
MasVnrArea       0
ExterQual        0
ExterCond        0
Foundation       0
BsmtQual         0
BsmtCond         0
BsmtFinSF1       0
BsmtUnfSF        0
TotalBsmtSF      0
Heating          0
CentralAir       0
Electrical       0
BsmtFullBath     0
BsmtHalfBath     0
FullBath         0
HalfBath         0
BedroomAbvGr     0
KitchenAbvGr     0
KitchenQual      0
TotRmsAbvGrd     0
Fireplaces       0
GarageArea       0
MoSold           0
YrSold           0
SaleCondition    0
dtype: int64
```

In [417]:

```
for col in test2:
    print ('\nFrequency of Categories for varible %s'%col)
    print (test2[col].value_counts())
```

```

Frequency of Categories for varible MSSubClass
20     543
60     276
50     143
120     95
30      70
70      68
160     65
80      60
90      57
190     31
85      28
180      7
75       7
45       6
40       2
150      1
Name: MSSubClass, dtype: int64
```

In [418]:

```
test2["MasVnrArea"]=test2["MasVnrArea"].replace(0,test2["MasVnrArea"].mean())
test2["BsmtFinSF1"]=test2["BsmtFinSF1"].replace(0,test2["BsmtFinSF1"].mean())
test2["BsmtUnfSF"]=test2["BsmtUnfSF"].replace(0,test2["BsmtUnfSF"].mean())
test2["TotalBsmtSF"]=test2["TotalBsmtSF"].replace(0,test2["TotalBsmtSF"].mean())
test2["GarageArea"]=test2["GarageArea"].replace(0,test2["GarageArea"].mean())
```

In [419]:

```
for col in test2:
    print ('\nFrequency of Categories for varible %s'%col)
    print (test2[col].value_counts())
```

```

Frequency of Categories for varible MSSubClass
20     543
60     276
50     143
120     95
30      70
70      68
160     65
80      60
90      57
190     31
85      28
180      7
75       7
45       6
40       2
150      1
Name: MSSubClass, dtype: int64
```

In [420]:
```
for x in test2.columns:
    if test2[x].all()==0:
        print(x)
```

```
LotShape
LandContour
Utilities
LotConfig
LandSlope
Neighborhood
Condition1
Condition2
HouseStyle
RoofStyle
RoofMatl
Exterior1st
ExterQual
ExterCond
Foundation
BsmtQual
BsmtCond
Heating
CentralAir
Electrical
BsmtFullBath
BsmtHalfBath
FullBath
HalfBath
BedroomAbvGr
KitchenAbvGr
KitchenQual
Fireplaces
SaleCondition
```

In [446]:
```
test2.shape
```

Out[446]:
```
(1459, 43)
```

In [447]:
```
test_pred=model.predict(test2)
test_pred
```

```
F:\Rahul\lib\site-packages\sklearn\base.py:443: UserWarning: X has feature names, but RandomForestRegressor was fitted without feature names
  warnings.warn(
```

Out[447]:
```
array([2.9149895 , 2.91651388, 2.91669725, ..., 2.91453306, 2.9149895 ,
       2.91689866])
```

In [448]:
```
submission=pd.read_csv(r"C:\Users\Admin\Desktop\Course DS\Data\House Price\sample_submission.csv")
submission1=submission.iloc[:,-1].values
submission1
```

Out[448]:
```
array([169277.0524984 , 187758.39398877, 183583.68356955, ...,
       219222.42340006, 184924.279659  , 187741.86665748])
```

In [449]:

```
from sklearn.metrics import r2_score
print(r2_score(test_pred,submission1))
```

```
-558517819065024.75
```