1. Read the Data

In [1]:
```python
import pandas as pd
import numpy as np
train=pd.read_csv("C:/Users/ASUS/Downloads/training_set.csv")
```

In [2]:
```python
train
```

Out[2]:

	Id	MSSubClass	MSZoning	LotFrontage	LotArea	Street	Alley	LotShape	LandContour	Utilities	...	PoolArea	PoolQC	Fence	Mi
0	1	60	RL	65.0	8450	Pave	NaN	Reg	Lvl	AllPub	...	0	NaN	NaN	
1	2	20	RL	80.0	9600	Pave	NaN	Reg	Lvl	AllPub	...	0	NaN	NaN	
2	3	60	RL	68.0	11250	Pave	NaN	IR1	Lvl	AllPub	...	0	NaN	NaN	
3	4	70	RL	60.0	9550	Pave	NaN	IR1	Lvl	AllPub	...	0	NaN	NaN	
4	5	60	RL	84.0	14260	Pave	NaN	IR1	Lvl	AllPub	...	0	NaN	NaN	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
1455	1456	60	RL	62.0	7917	Pave	NaN	Reg	Lvl	AllPub	...	0	NaN	NaN	
1456	1457	20	RL	85.0	13175	Pave	NaN	Reg	Lvl	AllPub	...	0	NaN	MnPrv	
1457	1458	70	RL	66.0	9042	Pave	NaN	Reg	Lvl	AllPub	...	0	NaN	GdPrv	
1458	1459	20	RL	68.0	9717	Pave	NaN	Reg	Lvl	AllPub	...	0	NaN	NaN	
1459	1460	20	RL	75.0	9937	Pave	NaN	Reg	Lvl	AllPub	...	0	NaN	NaN	

1460 rows × 81 columns

In [3]:
```python
train.head()
```

Out[3]:

	Id	MSSubClass	MSZoning	LotFrontage	LotArea	Street	Alley	LotShape	LandContour	Utilities	...	PoolArea	PoolQC	Fence	MiscFeat
0	1	60	RL	65.0	8450	Pave	NaN	Reg	Lvl	AllPub	...	0	NaN	NaN	N
1	2	20	RL	80.0	9600	Pave	NaN	Reg	Lvl	AllPub	...	0	NaN	NaN	N
2	3	60	RL	68.0	11250	Pave	NaN	IR1	Lvl	AllPub	...	0	NaN	NaN	N
3	4	70	RL	60.0	9550	Pave	NaN	IR1	Lvl	AllPub	...	0	NaN	NaN	N
4	5	60	RL	84.0	14260	Pave	NaN	IR1	Lvl	AllPub	...	0	NaN	NaN	N

5 rows × 81 columns

1. Missing Data treatment

In [4]:
```python
from preprocesser import replacer
replacer(train)
```

In [5]:
```python
train.isna().sum()
```

Out[5]:
```
Id               0
MSSubClass       0
MSZoning         0
LotFrontage      0
LotArea          0
                ..
MoSold           0
YrSold           0
SaleType         0
SaleCondition    0
SalePrice        0
Length: 81, dtype: int64
```

In [6]:
```python
cat=[]
con=[]
for i in train.columns:
    if train[i].dtypes=="object":
        cat.append(i)
    else:
        con.append(i)
```

In [7]:
```python
cat
```

Out[7]:
```
['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',
 'Functional',
 'FireplaceQu',
 'GarageType',
 'GarageFinish',
 'GarageQual',
 'GarageCond',
 'PavedDrive',
 'PoolQC',
 'Fence',
 'MiscFeature',
 'SaleType',
 'SaleCondition']
```

5.1 Standardization of con columns

In [8]:
```
con
```

Out[8]:
```
['Id',
 'MSSubClass',
 'LotFrontage',
 'LotArea',
 'OverallQual',
 'OverallCond',
 'YearBuilt',
 'YearRemodAdd',
 'MasVnrArea',
 'BsmtFinSF1',
 'BsmtFinSF2',
 'BsmtUnfSF',
 'TotalBsmtSF',
 '1stFlrSF',
 '2ndFlrSF',
 'LowQualFinSF',
 'GrLivArea',
 'BsmtFullBath',
 'BsmtHalfBath',
 'FullBath',
 'HalfBath',
 'BedroomAbvGr',
 'KitchenAbvGr',
 'TotRmsAbvGrd',
 'Fireplaces',
 'GarageYrBlt',
 'GarageCars',
 'GarageArea',
 'WoodDeckSF',
 'OpenPorchSF',
 'EnclosedPorch',
 '3SsnPorch',
 'ScreenPorch',
 'PoolArea',
 'MiscVal',
 'MoSold',
 'YrSold',
 'SalePrice']
```

In [9]:
```
Y=train[["SalePrice"]]
```

```
X=train.drop(labels=["SalePrice"],axis=1)
```

1. Drop unnecessary columns(Columns with no statitical importance)

In [10]:
```
Unwanted_cols=X=train.drop(labels=["Id","MSZoning","LotFrontage","Alley","LandSlope","YearBuilt","YearRemodAdd"
```

In [11]:
```
Unwanted_cols
```

Out[11]:

	MSSubClass	LotArea	Street	LotShape	LandContour	Utilities	LotConfig	Neighborhood	Condition1	Condition2	...	OpenPorchSF	P
0	60	8450	Pave	Reg	Lvl	AllPub	Inside	CollgCr	Norm	Norm	...	61	
1	20	9600	Pave	Reg	Lvl	AllPub	FR2	Veenker	Feedr	Norm	...	0	
2	60	11250	Pave	IR1	Lvl	AllPub	Inside	CollgCr	Norm	Norm	...	42	
3	70	9550	Pave	IR1	Lvl	AllPub	Corner	Crawfor	Norm	Norm	...	35	
4	60	14260	Pave	IR1	Lvl	AllPub	FR2	NoRidge	Norm	Norm	...	84	
...	...	...	...	...	...	...	...	...	...	...	...	...	
1455	60	7917	Pave	Reg	Lvl	AllPub	Inside	Gilbert	Norm	Norm	...	40	
1456	20	13175	Pave	Reg	Lvl	AllPub	Inside	NWAmes	Norm	Norm	...	0	
1457	70	9042	Pave	Reg	Lvl	AllPub	Inside	Crawfor	Norm	Norm	...	60	
1458	20	9717	Pave	Reg	Lvl	AllPub	Inside	NAmes	Norm	Norm	...	0	
1459	20	9937	Pave	Reg	Lvl	AllPub	Inside	Edwards	Norm	Norm	...	68	

1460 rows × 50 columns

In [12]:
```
numeric_cols=Unwanted_cols.select_dtypes(include=['int','float']).columns
a=Unwanted_cols[numeric_cols]
```

1. Exploratory Data Analysis

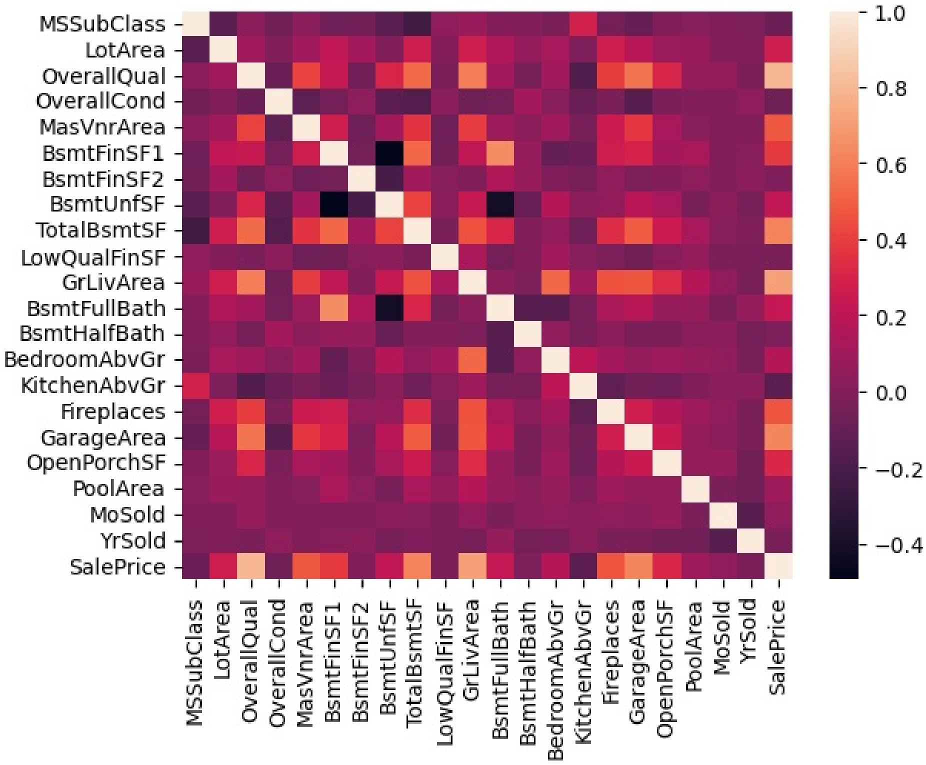
4.1 Correlation

In [13]:
```
corrlation_matrix=a.corr()
```

In [14]:
```
import seaborn as sns
```

In [15]:
```
sns.heatmap(corrlation_matrix)
```

Out[15]: <Axes: >



4.2 Check for Skew in X columns

In [16]:
```
a.skew().sort_values()>0.7
```

Out[16]:
```
YrSold          False
GarageArea      False
BedroomAbvGr    False
MoSold          False
OverallQual     False
BsmtFullBath    False
Fireplaces      False
OverallCond     False
BsmtUnfSF        True
GrLivArea        True
MSSubClass       True
TotalBsmtSF      True
BsmtFinSF1       True
SalePrice        True
OpenPorchSF      True
MasVnrArea       True
BsmtHalfBath     True
BsmtFinSF2       True
KitchenAbvGr     True
LowQualFinSF     True
LotArea          True
PoolArea         True
dtype: bool
```

4.3 Remove skew

In [17]:
```
skew_new=a.drop(labels=["BsmtUnfSF","GrLivArea","MSSubClass","TotalBsmtSF","BsmtFinSF1","SalePrice","OpenPorchS
```

In [18]:
```
from preprocesser import data_prep
Xnew=data_prep(X)
```

In [19]:
```
from sklearn.preprocessing import StandardScaler
ss=StandardScaler()
ss.fit_transform(Xnew)
```

Out[19]:
```
array([[ 0.07337496, -0.20714171,  0.65147924, ..., -0.11785113,
         0.4676514 , -0.30599503],
       [-0.87256276, -0.09188637, -0.07183611, ..., -0.11785113,
         0.4676514 , -0.30599503],
       [ 0.07337496,  0.07347998,  0.65147924, ..., -0.11785113,
         0.4676514 , -0.30599503],
       ...,
       [ 0.30985939, -0.14781027,  0.65147924, ..., -0.11785113,
         0.4676514 , -0.30599503],
       [-0.87256276, -0.08016039, -0.79515147, ..., -0.11785113,
         0.4676514 , -0.30599503],
       [-0.87256276, -0.05811155, -0.79515147, ..., -0.11785113,
         0.4676514 , -0.30599503]])
```

In [20]:
```
import pandas as pd
pd.DataFrame(ss.fit_transform(Xnew))
```

Out[20]:

	0	1	2	3	4	5	6	7	8	9	...	171	172	
0	0.073375	-0.207142	0.651479	-0.517200	0.511418	0.575425	-0.288653	-0.944591	-0.459303	-0.120242	...	-0.058621	-0.301962	-0.(
1	-0.872563	-0.091886	-0.071836	2.179628	-0.574410	1.171992	-0.288653	-0.641228	0.466465	-0.120242	...	-0.058621	-0.301962	-0.(
2	0.073375	0.073480	0.651479	-0.517200	0.323060	0.092907	-0.288653	-0.301643	-0.313369	-0.120242	...	-0.058621	-0.301962	-0.(
3	0.309859	-0.096897	0.651479	-0.517200	-0.574410	-0.499274	-0.288653	-0.061670	-0.687324	-0.120242	...	-0.058621	-0.301962	-0.(
4	0.073375	0.375148	1.374795	-0.517200	1.364570	0.463568	-0.288653	-0.174865	0.199680	-0.120242	...	-0.058621	-0.301962	-0.(
...	...	...	...	...	...	...	...	...	...	...	...	...	...	
1455	0.073375	-0.260560	-0.071836	-0.517200	-0.574410	-0.973018	-0.288653	0.873321	-0.238122	-0.120242	...	-0.058621	-0.301962	-0.(
1456	-0.872563	0.266407	-0.071836	0.381743	0.084843	0.759659	0.722112	0.049262	1.104925	-0.120242	...	-0.058621	-0.301962	-0.(
1457	0.309859	-0.147810	0.651479	3.078570	-0.574410	-0.369871	-0.288653	0.701265	0.215641	-0.120242	...	-0.058621	-0.301962	-0.(
1458	-0.872563	-0.080160	-0.795151	0.381743	-0.574410	-0.865548	6.092188	-1.284176	0.046905	-0.120242	...	-0.058621	-0.301962	-0.(
1459	-0.872563	-0.058112	-0.795151	0.381743	-0.574410	0.847389	1.509640	-0.976285	0.452784	-0.120242	...	-0.058621	-0.301962	-0.(

1460 rows × 181 columns

OHE of categorical columns

In [21]:
```
pd.get_dummies(X,dtype='int')
```

Out[21]:

	MSSubClass	LotArea	OverallQual	OverallCond	MasVnrArea	BsmtFinSF1	BsmtFinSF2	BsmtUnfSF	TotalBsmtSF	LowQualFinSF	...
0	60	8450	7	5	196.0	706	0	150	856	0	...
1	20	9600	6	8	0.0	978	0	284	1262	0	...
2	60	11250	7	5	162.0	486	0	434	920	0	...
3	70	9550	7	5	0.0	216	0	540	756	0	...
4	60	14260	8	5	350.0	655	0	490	1145	0	...
...	...	...	...	...	...	...	...	...	...	...	...
1455	60	7917	6	5	0.0	0	0	953	953	0	...
1456	20	13175	6	6	119.0	790	163	589	1542	0	...
1457	70	9042	7	9	0.0	275	0	877	1152	0	...
1458	20	9717	5	6	0.0	49	1029	0	1078	0	...
1459	20	9937	5	6	0.0	830	290	136	1256	0	...

1460 rows × 181 columns

1. Preprocessing

1. Divide data in training & testing set(Random state: 31)0.8,0.2

In [22]:

```
from sklearn.model_selection import train_test_split
xtrain,xtest,ytrain,ytest=train_test_split(Xnew,Y,test_size=0.2,random_state=21)
```

1. Create a backward elemination OLS model

In [23]:

```
from statsmodels.api import OLS,add_constant
xconst=add_constant(xtrain)
ol=OLS(ytrain,xconst)
model=ol.fit()
model.summary()
```

Out[23]:

OLS Regression Results

Dep. Variable:	SalePrice	**R-squared:**	1.000
Model:	OLS	**Adj. R-squared:**	1.000
Method:	Least Squares	**F-statistic:**	3.430e+29
Date:	Tue, 26 Mar 2024	**Prob (F-statistic):**	0.00
Time:	16:58:05	**Log-Likelihood:**	23774.
No. Observations:	1168	**AIC:**	-4.725e+04
Df Residuals:	1017	**BIC:**	-4.648e+04
Df Model:	150		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	2.398e+04	5.8e-11	4.13e+14	0.000	2.4e+04	2.4e+04
MSSubClass	5.093e-11	7.6e-11	0.671	0.503	-9.81e-11	2e-10
LotArea	-3.638e-12	1.46e-11	-0.250	0.803	-3.22e-11	2.5e-11
OverallQual	4.866e-11	2.41e-11	2.016	0.044	1.3e-12	9.6e-11
OverallCond	-4.729e-11	1.58e-11	-3.001	0.003	-7.82e-11	-1.64e-11
MasVnrArea	-3.82e-11	1.45e-11	-2.627	0.009	-6.67e-11	-9.66e-12
BsmtFinSF1	3.138e-11	1.55e-11	2.026	0.043	9.94e-13	6.18e-11
BsmtFinSF2	-2.547e-11	2.09e-11	-1.219	0.223	-6.65e-11	1.55e-11
BsmtUnfSF	6.048e-11	1.35e-11	4.475	0.000	3.4e-11	8.7e-11
TotalBsmtSF	5.366e-11	1.73e-11	3.104	0.002	1.97e-11	8.76e-11
LowQualFinSF	-5.002e-11	1.56e-11	-3.203	0.001	-8.07e-11	-1.94e-11
GrLivArea	-8.731e-11	3.51e-11	-2.485	0.013	-1.56e-10	-1.84e-11
BsmtFullBath	1.273e-11	1.81e-11	0.703	0.482	-2.28e-11	4.83e-11
BsmtHalfBath	-4.093e-12	1.3e-11	-0.316	0.752	-2.95e-11	2.13e-11
BedroomAbvGr	-2.183e-11	1.73e-11	-1.260	0.208	-5.58e-11	1.22e-11
KitchenAbvGr	-4.366e-11	2.03e-11	-2.153	0.032	-8.34e-11	-3.87e-12
Fireplaces	-8.185e-12	1.53e-11	-0.537	0.592	-3.81e-11	2.18e-11

GarageArea	6.253e-12	1.67e-11	0.374	0.709	-2.66e-11	3.91e-11
OpenPorchSF	-3.797e-11	1.34e-11	-2.827	0.005	-6.43e-11	-1.16e-11
PoolArea	-7.276e-12	1.71e-11	-0.425	0.671	-4.09e-11	2.63e-11
MoSold	2.728e-12	1.19e-11	0.228	0.819	-2.07e-11	2.62e-11
YrSold	1.592e-11	1.22e-11	1.306	0.192	-8.01e-12	3.98e-11
SalePrice	7.942e+04	3.45e-11	2.3e+15	0.000	7.94e+04	7.94e+04
Street_Grvl	1.199e+04	1.11e-10	1.08e+14	0.000	1.2e+04	1.2e+04
Street_Pave	1.199e+04	9.97e-11	1.2e+14	0.000	1.2e+04	1.2e+04
LotShape_IR1	5996.0775	4.56e-11	1.32e+14	0.000	5996.077	5996.077
LotShape_IR2	5996.0775	6.36e-11	9.43e+13	0.000	5996.077	5996.077
LotShape_IR3	5996.0775	1.16e-10	5.18e+13	0.000	5996.077	5996.077
LotShape_Reg	5996.0775	4.76e-11	1.26e+14	0.000	5996.077	5996.077
LandContour_Bnk	5996.0775	5.54e-11	1.08e+14	0.000	5996.077	5996.077
LandContour_HLS	5996.0775	5.8e-11	1.03e+14	0.000	5996.077	5996.077
LandContour_Low	5996.0775	6.89e-11	8.7e+13	0.000	5996.077	5996.077
LandContour_Lvl	5996.0775	3.72e-11	1.61e+14	0.000	5996.077	5996.077
Utilities_AllPub	1.199e+04	2.03e-10	5.91e+13	0.000	1.2e+04	1.2e+04
Utilities_NoSeWa	1.199e+04	2.25e-10	5.33e+13	0.000	1.2e+04	1.2e+04
LotConfig_Corner	4796.8620	5e-11	9.6e+13	0.000	4796.862	4796.862
LotConfig_CulDSac	4796.8620	6.04e-11	7.94e+13	0.000	4796.862	4796.862
LotConfig_FR2	4796.8620	6.87e-11	6.98e+13	0.000	4796.862	4796.862
LotConfig_FR3	4796.8620	1.65e-10	2.91e+13	0.000	4796.862	4796.862
LotConfig_Inside	4796.8620	4.68e-11	1.02e+14	0.000	4796.862	4796.862
Neighborhood_Blmngtn	959.3724	1.12e-10	8.54e+12	0.000	959.372	959.372
Neighborhood_Blueste	959.3724	3.73e-10	2.57e+12	0.000	959.372	959.372
Neighborhood_BrDale	959.3724	1.28e-10	7.49e+12	0.000	959.372	959.372
Neighborhood_BrkSide	959.3724	7.58e-11	1.27e+13	0.000	959.372	959.372
Neighborhood_ClearCr	959.3724	9.15e-11	1.05e+13	0.000	959.372	959.372
Neighborhood_CollgCr	959.3724	4.93e-11	1.95e+13	0.000	959.372	959.372
Neighborhood_Crawfor	959.3724	7.67e-11	1.25e+13	0.000	959.372	959.372
Neighborhood_Edwards	959.3724	5.65e-11	1.7e+13	0.000	959.372	959.372
Neighborhood_Gilbert	959.3724	6.38e-11	1.5e+13	0.000	959.372	959.372
Neighborhood_IDOTRR	959.3724	8.62e-11	1.11e+13	0.000	959.372	959.372
Neighborhood_MeadowV	959.3724	1.32e-10	7.25e+12	0.000	959.372	959.372
Neighborhood_Mitchel	959.3724	6.97e-11	1.38e+13	0.000	959.372	959.372
Neighborhood_NAmes	959.3724	4.71e-11	2.04e+13	0.000	959.372	959.372
Neighborhood_NPkVill	959.3724	1.63e-10	5.89e+12	0.000	959.372	959.372
Neighborhood_NWAmes	959.3724	6.16e-11	1.56e+13	0.000	959.372	959.372
Neighborhood_NoRidge	959.3724	8.28e-11	1.16e+13	0.000	959.372	959.372
Neighborhood_NridgHt	959.3724	6.9e-11	1.39e+13	0.000	959.372	959.372
Neighborhood_OldTown	959.3724	6.33e-11	1.52e+13	0.000	959.372	959.372
Neighborhood_SWISU	959.3724	9.68e-11	9.91e+12	0.000	959.372	959.372
Neighborhood_Sawyer	959.3724	6.06e-11	1.58e+13	0.000	959.372	959.372
Neighborhood_SawyerW	959.3724	6.23e-11	1.54e+13	0.000	959.372	959.372
Neighborhood_Somerst	959.3724	6.02e-11	1.59e+13	0.000	959.372	959.372
Neighborhood_StoneBr	959.3724	1.03e-10	9.3e+12	0.000	959.372	959.372
Neighborhood_Timber	959.3724	7.95e-11	1.21e+13	0.000	959.372	959.372
Neighborhood_Veenker	959.3724	1.22e-10	7.86e+12	0.000	959.372	959.372
Condition1_Artery	2664.9233	8.26e-11	3.22e+13	0.000	2664.923	2664.923
Condition1_Feedr	2664.9233	6.58e-11	4.05e+13	0.000	2664.923	2664.923
Condition1_Norm	2664.9233	5.11e-11	5.21e+13	0.000	2664.923	2664.923
Condition1_PosA	2664.9233	1.52e-10	1.76e+13	0.000	2664.923	2664.923
Condition1_PosN	2664.9233	1.15e-10	2.33e+13	0.000	2664.923	2664.923
Condition1_RRAe	2664.9233	1.45e-10	1.84e+13	0.000	2664.923	2664.923

Condition1_RRAn	2664.9233	1.03e-10	2.59e+13	0.000	2664.923	2664.923
Condition1_RRNe	2664.9233	2.49e-10	1.07e+13	0.000	2664.923	2664.923
Condition1_RRNn	2664.9233	1.73e-10	1.54e+13	0.000	2664.923	2664.923
Condition2_Artery	2998.0387	3.08e-10	9.73e+12	0.000	2998.039	2998.039
Condition2_Feedr	2998.0387	2.19e-10	1.37e+13	0.000	2998.039	2998.039
Condition2_Norm	2998.0387	1.42e-10	2.11e+13	0.000	2998.039	2998.039
Condition2_PosA	2998.0387	5.22e-10	5.74e+12	0.000	2998.039	2998.039
Condition2_PosN	2998.0387	3.79e-10	7.9e+12	0.000	2998.039	2998.039
Condition2_RRAe	2998.0387	5.05e-10	5.94e+12	0.000	2998.039	2998.039
Condition2_RRAn	2998.0387	3.68e-10	8.14e+12	0.000	2998.039	2998.039
Condition2_RRNn	2998.0387	3.76e-10	7.97e+12	0.000	2998.039	2998.039
BldgType_1Fam	4796.8620	1.4e-10	3.43e+13	0.000	4796.862	4796.862
BldgType_2fmCon	4796.8620	1.37e-10	3.5e+13	0.000	4796.862	4796.862
BldgType_Duplex	4796.8620	9.68e-11	4.96e+13	0.000	4796.862	4796.862
BldgType_Twnhs	4796.8620	9.01e-11	5.32e+13	0.000	4796.862	4796.862
BldgType_TwnhsE	4796.8620	7.28e-11	6.59e+13	0.000	4796.862	4796.862
HouseStyle_1.5Fin	2998.0387	5.26e-11	5.7e+13	0.000	2998.039	2998.039
HouseStyle_1.5Unf	2998.0387	1.14e-10	2.63e+13	0.000	2998.039	2998.039
HouseStyle_1Story	2998.0387	7.61e-11	3.94e+13	0.000	2998.039	2998.039
HouseStyle_2.5Fin	2998.0387	1.9e-10	1.58e+13	0.000	2998.039	2998.039
HouseStyle_2.5Unf	2998.0387	1.32e-10	2.26e+13	0.000	2998.039	2998.039
HouseStyle_2Story	2998.0387	4.76e-11	6.29e+13	0.000	2998.039	2998.039
HouseStyle_SFoyer	2998.0387	9.38e-11	3.2e+13	0.000	2998.039	2998.039
HouseStyle_SLvl	2998.0387	8.11e-11	3.7e+13	0.000	2998.039	2998.039
ExterCond_Ex	4796.8620	3.24e-10	1.48e+13	0.000	4796.862	4796.862
ExterCond_Fa	4796.8620	1.33e-10	3.6e+13	0.000	4796.862	4796.862
ExterCond_Gd	4796.8620	1.2e-10	3.99e+13	0.000	4796.862	4796.862
ExterCond_Po	4796.8620	3.5e-10	1.37e+13	0.000	4796.862	4796.862
ExterCond_TA	4796.8620	1.16e-10	4.14e+13	0.000	4796.862	4796.862
Foundation_BrkTil	3997.3850	6.93e-11	5.76e+13	0.000	3997.385	3997.385
Foundation_CBlock	3997.3850	6.5e-11	6.15e+13	0.000	3997.385	3997.385
Foundation_PConc	3997.3850	6.68e-11	5.99e+13	0.000	3997.385	3997.385
Foundation_Slab	3997.3850	1.12e-10	3.57e+13	0.000	3997.385	3997.385
Foundation_Stone	3997.3850	2.23e-10	1.8e+13	0.000	3997.385	3997.385
Foundation_Wood	3997.3850	1.98e-10	2.02e+13	0.000	3997.385	3997.385
BsmtQual_Ex	5996.0775	5.75e-11	1.04e+14	0.000	5996.077	5996.077
BsmtQual_Fa	5996.0775	6.96e-11	8.62e+13	0.000	5996.077	5996.077
BsmtQual_Gd	5996.0775	3.74e-11	1.6e+14	0.000	5996.077	5996.077
BsmtQual_TA	5996.0775	3.87e-11	1.55e+14	0.000	5996.077	5996.077
BsmtCond_Fa	5996.0775	1.18e-10	5.06e+13	0.000	5996.077	5996.077
BsmtCond_Gd	5996.0775	1.22e-10	4.91e+13	0.000	5996.077	5996.077
BsmtCond_Po	5996.0775	3.43e-10	1.75e+13	0.000	5996.077	5996.077
BsmtCond_TA	5996.0775	1.16e-10	5.19e+13	0.000	5996.077	5996.077
BsmtExposure_Av	5996.0775	3.18e-11	1.89e+14	0.000	5996.077	5996.077
BsmtExposure_Gd	5996.0775	4.03e-11	1.49e+14	0.000	5996.077	5996.077
BsmtExposure_Mn	5996.0775	3.88e-11	1.55e+14	0.000	5996.077	5996.077
BsmtExposure_No	5996.0775	2.76e-11	2.17e+14	0.000	5996.077	5996.077
BsmtFinType1_ALQ	3997.3850	3.42e-11	1.17e+14	0.000	3997.385	3997.385
BsmtFinType1_BLQ	3997.3850	3.62e-11	1.1e+14	0.000	3997.385	3997.385
BsmtFinType1_GLQ	3997.3850	3.39e-11	1.18e+14	0.000	3997.385	3997.385
BsmtFinType1_LwQ	3997.3850	5.13e-11	7.79e+13	0.000	3997.385	3997.385
BsmtFinType1_Rec	3997.3850	3.8e-11	1.05e+14	0.000	3997.385	3997.385
BsmtFinType1_Unf	3997.3850	3.55e-11	1.13e+14	0.000	3997.385	3997.385
BsmtFinType2_ALQ	3997.3850	9.5e-11	4.21e+13	0.000	3997.385	3997.385
BsmtFinType2_BLQ	3997.3850	8.14e-11	4.91e+13	0.000	3997.385	3997.385

BsmtFinType2_GLQ	3997.3850	1.19e-10	3.37e+13	0.000	3997.385	3997.385
BsmtFinType2_LwQ	3997.3850	6.79e-11	5.89e+13	0.000	3997.385	3997.385
BsmtFinType2_Rec	3997.3850	6.31e-11	6.34e+13	0.000	3997.385	3997.385
BsmtFinType2_Unf	3997.3850	6.32e-11	6.32e+13	0.000	3997.385	3997.385
CentralAir_N	1.199e+04	4.33e-11	2.77e+14	0.000	1.2e+04	1.2e+04
CentralAir_Y	1.199e+04	4.25e-11	2.82e+14	0.000	1.2e+04	1.2e+04
Electrical_FuseA	5996.0775	8.45e-11	7.1e+13	0.000	5996.077	5996.077
Electrical_FuseF	5996.0775	9.73e-11	6.16e+13	0.000	5996.077	5996.077
Electrical_FuseP	5996.0775	2.18e-10	2.75e+13	0.000	5996.077	5996.077
Electrical_Mix	-1.086e-12	1.5e-25	-7.22e+12	0.000	-1.09e-12	-1.09e-12
Electrical_SBrkr	5996.0775	8.04e-11	7.46e+13	0.000	5996.077	5996.077
KitchenQual_Ex	5996.0775	5.48e-11	1.09e+14	0.000	5996.077	5996.077
KitchenQual_Fa	5996.0775	6.29e-11	9.54e+13	0.000	5996.077	5996.077
KitchenQual_Gd	5996.0775	3.49e-11	1.72e+14	0.000	5996.077	5996.077
KitchenQual_TA	5996.0775	3.42e-11	1.75e+14	0.000	5996.077	5996.077
Functional_Maj1	3426.3300	1.44e-10	2.38e+13	0.000	3426.330	3426.330
Functional_Maj2	3426.3300	2.05e-10	1.67e+13	0.000	3426.330	3426.330
Functional_Min1	3426.3300	9.74e-11	3.52e+13	0.000	3426.330	3426.330
Functional_Min2	3426.3300	1e-10	3.42e+13	0.000	3426.330	3426.330
Functional_Mod	3426.3300	1.27e-10	2.69e+13	0.000	3426.330	3426.330
Functional_Sev	3426.3300	3.48e-10	9.85e+12	0.000	3426.330	3426.330
Functional_Typ	3426.3300	7.56e-11	4.53e+13	0.000	3426.330	3426.330
GarageQual_Ex	4796.8620	2.05e-10	2.34e+13	0.000	4796.862	4796.862
GarageQual_Fa	4796.8620	1.03e-10	4.64e+13	0.000	4796.862	4796.862
GarageQual_Gd	4796.8620	1.38e-10	3.48e+13	0.000	4796.862	4796.862
GarageQual_Po	4796.8620	2.76e-10	1.74e+13	0.000	4796.862	4796.862
GarageQual_TA	4796.8620	8.81e-11	5.45e+13	0.000	4796.862	4796.862
PavedDrive_N	7994.7699	4.67e-11	1.71e+14	0.000	7994.770	7994.770
PavedDrive_P	7994.7699	6.04e-11	1.32e+14	0.000	7994.770	7994.770
PavedDrive_Y	7994.7699	3.96e-11	2.02e+14	0.000	7994.770	7994.770
PoolQC_Ex	8465.0505	2.21e-10	3.83e+13	0.000	8465.051	8465.051
PoolQC_Fa	7054.2088	1.69e-10	4.17e+13	0.000	7054.209	7054.209
PoolQC_Gd	8465.0505	1.88e-10	4.51e+13	0.000	8465.051	8465.051
Fence_GdPrv	5996.0775	6.04e-11	9.93e+13	0.000	5996.077	5996.077
Fence_GdWo	5996.0775	6.02e-11	9.96e+13	0.000	5996.077	5996.077
Fence_MnPrv	5996.0775	4.2e-11	1.43e+14	0.000	5996.077	5996.077
Fence_MnWw	5996.0775	1.03e-10	5.82e+13	0.000	5996.077	5996.077
MiscFeature_Gar2	5643.3670	2.91e-10	1.94e+13	0.000	5643.367	5643.367
MiscFeature_Othr	5643.3670	2.41e-10	2.34e+13	0.000	5643.367	5643.367
MiscFeature_Shed	5643.3670	1.48e-10	3.82e+13	0.000	5643.367	5643.367
MiscFeature_TenC	7054.2088	1.69e-10	4.17e+13	0.000	7054.209	7054.209
SaleType_COD	2664.9233	9.9e-11	2.69e+13	0.000	2664.923	2664.923
SaleType_CWD	2664.9233	2.58e-10	1.03e+13	0.000	2664.923	2664.923
SaleType_Con	2664.9233	2.55e-10	1.04e+13	0.000	2664.923	2664.923
SaleType_ConLD	2664.9233	1.48e-10	1.8e+13	0.000	2664.923	2664.923
SaleType_ConLI	2664.9233	2.14e-10	1.25e+13	0.000	2664.923	2664.923
SaleType_ConLw	2664.9233	1.98e-10	1.35e+13	0.000	2664.923	2664.923
SaleType_New	2664.9233	2.17e-10	1.23e+13	0.000	2664.923	2664.923
SaleType_Oth	2664.9233	2.55e-10	1.04e+13	0.000	2664.923	2664.923
SaleType_WD	2664.9233	7.47e-11	3.57e+13	0.000	2664.923	2664.923
SaleCondition_Abnorml	3997.3850	7.32e-11	5.46e+13	0.000	3997.385	3997.385
SaleCondition_AdjLand	3997.3850	1.93e-10	2.07e+13	0.000	3997.385	3997.385
SaleCondition_Alloca	3997.3850	1.29e-10	3.1e+13	0.000	3997.385	3997.385
SaleCondition_Family	3997.3850	1.02e-10	3.9e+13	0.000	3997.385	3997.385
SaleCondition_Normal	3997.3850	6.18e-11	6.46e+13	0.000	3997.385	3997.385

SaleCondition_Partial	3997.3850	2e-10	2e+13	0.000	3997.385	3997.385

Omnibus:	1171.654	**Durbin-Watson:**	0.794
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	116953.920
Skew:	4.456	**Prob(JB):**	0.00
Kurtosis:	51.205	**Cond. No.**	1.03e+16

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The smallest eigenvalue is 2.3e-28. This might indicate that there are
strong multicollinearity problems or that the design matrix is singular.

1. Remove unncessary columns on the basis of pval

In [24]:
```
D=pd.DataFrame(model.pvalues,columns=["pval"])
column_to_drop=D.sort_values(by="pval",ascending=False)[0:1].index[0]
Xnew=Xnew.drop(labels=[column_to_drop],axis=1)
```

In [25]:
```
Xnew
```

Out[25]:

	MSSubClass	LotArea	OverallQual	OverallCond	MasVnrArea	BsmtFinSF1	BsmtFinSF2	BsmtUnfSF	TotalBsmtSF	LowQualFinSF	...
0	0.073375	-0.207142	0.651479	-0.517200	0.511418	0.575425	-0.288653	-0.944591	-0.459303	-0.120242	...
1	-0.872563	-0.091886	-0.071836	2.179628	-0.574410	1.171992	-0.288653	-0.641228	0.466465	-0.120242	...
2	0.073375	0.073480	0.651479	-0.517200	0.323060	0.092907	-0.288653	-0.301643	-0.313369	-0.120242	...
3	0.309859	-0.096897	0.651479	-0.517200	-0.574410	-0.499274	-0.288653	-0.061670	-0.687324	-0.120242	...
4	0.073375	0.375148	1.374795	-0.517200	1.364570	0.463568	-0.288653	-0.174865	0.199680	-0.120242	...
...	...	...	...	...	...	...	...	...	...	...	...
1455	0.073375	-0.260560	-0.071836	-0.517200	-0.574410	-0.973018	-0.288653	0.873321	-0.238122	-0.120242	...
1456	-0.872563	0.266407	-0.071836	0.381743	0.084843	0.759659	0.722112	0.049262	1.104925	-0.120242	...
1457	0.309859	-0.147810	0.651479	3.078570	-0.574410	-0.369871	-0.288653	0.701265	0.215641	-0.120242	...
1458	-0.872563	-0.080160	-0.795151	0.381743	-0.574410	-0.865548	6.092188	-1.284176	0.046905	-0.120242	...
1459	-0.872563	-0.058112	-0.795151	0.381743	-0.574410	0.847389	1.509640	-0.976285	0.452784	-0.120242	...

1460 rows × 180 columns

1. Create a Linear Regression model on the basis of selected columns.

In [26]:
```
from sklearn.linear_model import LinearRegression
lm=LinearRegression()
model=lm.fit(xtrain,ytrain)
```

In [27]:
```
pred=model.predict(xtest)
```

In [28]:
```
from sklearn.metrics import mean_absolute_error
mean_absolute_error(ytest,pred)
```

Out[28]:
```
0.013268572693399221
```

1. Find training | testing error --> Overfitting or not

In [29]:
```
from sklearn.linear_model import LinearRegression
lm=LinearRegression()
model=lm.fit(xtrain,ytrain)
from sklearn.metrics import mean_squared_error
pred_tr=model.predict(xtrain)
tr_err=round(mean_squared_error(ytrain,pred_tr),3)
pred_ts=model.predict(xtest)
ts_err=round(mean_squared_error(ytest,pred_ts),3)
print("Training Error:",tr_err)
print("\nTesting Error:",ts_err)
if(tr_err<ts_err):
    print("Overfitting")
```

```
Training Error: 0.0

Testing Error: 0.03
Overfitting
```

Ridge, Lasso

In [30]:
```
Q=[]
x=20.5
for i in range(0,100,1):
    x=x+0.01
    x=round(x,4)
    Q.append(x)
```

In [31]:
```
from sklearn.linear_model import Ridge
for i in Q:
    rr=Ridge(alpha=i)
    model=rr.fit(xtrain,ytrain)
    tr_pred=model.predict(xtrain)
    ts_pred=model.predict(xtest)
    from sklearn.metrics import mean_absolute_error
    tr_err=mean_absolute_error(ytrain,tr_pred)
    ts_err=mean_absolute_error(ytest,ts_pred)
    print("alpha",i,"\ttr_Err",round(tr_err,4),"\tts_Err",round(ts_err,4))
```

```
alpha 20.51	tr_Err 1854.3472	ts_Err 2082.9042
alpha 20.52	tr_Err 1855.061	ts_Err 2083.6831
alpha 20.53	tr_Err 1855.7749	ts_Err 2084.4618
alpha 20.54	tr_Err 1856.4887	ts_Err 2085.2403
alpha 20.55	tr_Err 1857.2024	ts_Err 2086.0187
alpha 20.56	tr_Err 1857.916	ts_Err 2086.797
alpha 20.57	tr_Err 1858.6295	ts_Err 2087.5751
alpha 20.58	tr_Err 1859.3428	ts_Err 2088.353
alpha 20.59	tr_Err 1860.056	ts_Err 2089.1308
alpha 20.6	tr_Err 1860.7691	ts_Err 2089.9085
alpha 20.61	tr_Err 1861.4823	ts_Err 2090.686
alpha 20.62	tr_Err 1862.1959	ts_Err 2091.4633
alpha 20.63	tr_Err 1862.9094	ts_Err 2092.2405
alpha 20.64	tr_Err 1863.6228	ts_Err 2093.0176
alpha 20.65	tr_Err 1864.336	ts_Err 2093.7945
alpha 20.66	tr_Err 1865.0492	ts_Err 2094.5712
alpha 20.67	tr_Err 1865.7622	ts_Err 2095.3478
alpha 20.68	tr_Err 1866.4751	ts_Err 2096.1243
alpha 20.69	tr_Err 1867.1879	ts_Err 2096.9006
alpha 20.7	tr_Err 1867.9005	ts_Err 2097.6767
alpha 20.71	tr_Err 1868.613	ts_Err 2098.4527
alpha 20.72	tr_Err 1869.3254	ts_Err 2099.2286
alpha 20.73	tr_Err 1870.0376	ts_Err 2100.0043
alpha 20.74	tr_Err 1870.7498	ts_Err 2100.7798
alpha 20.75	tr_Err 1871.4618	ts_Err 2101.5552
alpha 20.76	tr_Err 1872.1737	ts_Err 2102.3305
alpha 20.77	tr_Err 1872.8854	ts_Err 2103.1056
alpha 20.78	tr_Err 1873.597	ts_Err 2103.8805
alpha 20.79	tr_Err 1874.3085	ts_Err 2104.6553
alpha 20.8	tr_Err 1875.0199	ts_Err 2105.43
alpha 20.81	tr_Err 1875.7311	ts_Err 2106.2045
alpha 20.82	tr_Err 1876.4423	ts_Err 2106.9789
alpha 20.83	tr_Err 1877.1533	ts_Err 2107.7531
alpha 20.84	tr_Err 1877.8641	ts_Err 2108.5271
alpha 20.85	tr_Err 1878.5749	ts_Err 2109.3011
alpha 20.86	tr_Err 1879.2855	ts_Err 2110.0748
alpha 20.87	tr_Err 1879.996	ts_Err 2110.8484
alpha 20.88	tr_Err 1880.7063	ts_Err 2111.6219
alpha 20.89	tr_Err 1881.4166	ts_Err 2112.3952
alpha 20.9	tr_Err 1882.1267	ts_Err 2113.1684
alpha 20.91	tr_Err 1882.8366	ts_Err 2113.9414
alpha 20.92	tr_Err 1883.5466	ts_Err 2114.7143
alpha 20.93	tr_Err 1884.2566	ts_Err 2115.487
alpha 20.94	tr_Err 1884.9665	ts_Err 2116.2596
alpha 20.95	tr_Err 1885.6763	ts_Err 2117.032
alpha 20.96	tr_Err 1886.3859	ts_Err 2117.8043
alpha 20.97	tr_Err 1887.0954	ts_Err 2118.5765
alpha 20.98	tr_Err 1887.8048	ts_Err 2119.3485
alpha 20.99	tr_Err 1888.5141	ts_Err 2120.1203
alpha 21.0	tr_Err 1889.2233	ts_Err 2120.892
alpha 21.01	tr_Err 1889.9323	ts_Err 2121.6635
alpha 21.02	tr_Err 1890.6412	ts_Err 2122.4349
alpha 21.03	tr_Err 1891.35	ts_Err 2123.2062
alpha 21.04	tr_Err 1892.0586	ts_Err 2123.9773
alpha 21.05	tr_Err 1892.7671	ts_Err 2124.7482
alpha 21.06	tr_Err 1893.4755	ts_Err 2125.5191
alpha 21.07	tr_Err 1894.1838	ts_Err 2126.2897
alpha 21.08	tr_Err 1894.8919	ts_Err 2127.0602
alpha 21.09	tr_Err 1895.6	ts_Err 2127.8306
alpha 21.1	tr_Err 1896.3079	ts_Err 2128.6008
alpha 21.11	tr_Err 1897.0156	ts_Err 2129.3709
alpha 21.12	tr_Err 1897.7233	ts_Err 2130.1408
alpha 21.13	tr_Err 1898.4308	ts_Err 2130.9106
alpha 21.14	tr_Err 1899.1382	ts_Err 2131.6802
alpha 21.15	tr_Err 1899.8455	ts_Err 2132.4497
alpha 21.16	tr_Err 1900.5526	ts_Err 2133.2191
alpha 21.17	tr_Err 1901.2596	ts_Err 2133.9882
alpha 21.18	tr_Err 1901.9665	ts_Err 2134.7573
```

```
alpha 21.19     tr_Err 1902.6733        ts_Err 2135.5262
alpha 21.2      tr_Err 1903.38  ts_Err 2136.2949
alpha 21.21     tr_Err 1904.0865        ts_Err 2137.0635
alpha 21.22     tr_Err 1904.7929        ts_Err 2137.832
alpha 21.23     tr_Err 1905.4992        ts_Err 2138.6003
alpha 21.24     tr_Err 1906.2053        ts_Err 2139.3685
alpha 21.25     tr_Err 1906.9113        ts_Err 2140.1365
alpha 21.26     tr_Err 1907.6172        ts_Err 2140.9044
alpha 21.27     tr_Err 1908.323         ts_Err 2141.6721
alpha 21.28     tr_Err 1909.0287        ts_Err 2142.4397
alpha 21.29     tr_Err 1909.7342        ts_Err 2143.2071
alpha 21.3      tr_Err 1910.4396        ts_Err 2143.9744
alpha 21.31     tr_Err 1911.1449        ts_Err 2144.7416
alpha 21.32     tr_Err 1911.85  ts_Err 2145.5086
alpha 21.33     tr_Err 1912.5551        ts_Err 2146.2754
alpha 21.34     tr_Err 1913.2602        ts_Err 2147.0421
alpha 21.35     tr_Err 1913.9651        ts_Err 2147.8087
alpha 21.36     tr_Err 1914.6699        ts_Err 2148.5751
alpha 21.37     tr_Err 1915.3745        ts_Err 2149.3414
alpha 21.38     tr_Err 1916.0791        ts_Err 2150.1075
alpha 21.39     tr_Err 1916.7835        ts_Err 2150.8735
alpha 21.4      tr_Err 1917.4878        ts_Err 2151.6393
alpha 21.41     tr_Err 1918.192         ts_Err 2152.405
alpha 21.42     tr_Err 1918.896         ts_Err 2153.1705
alpha 21.43     tr_Err 1919.5999        ts_Err 2153.9359
alpha 21.44     tr_Err 1920.3037        ts_Err 2154.7012
alpha 21.45     tr_Err 1921.0074        ts_Err 2155.4663
alpha 21.46     tr_Err 1921.711         ts_Err 2156.2313
alpha 21.47     tr_Err 1922.4144        ts_Err 2156.9961
alpha 21.48     tr_Err 1923.1177        ts_Err 2157.7608
alpha 21.49     tr_Err 1923.8209        ts_Err 2158.5253
alpha 21.5      tr_Err 1924.524         ts_Err 2159.2897
```

In [32]:

```
Q=[]
x=0.03
for i in range(0,100,1):
    x=x+0.001
    x=round(x,4)
    Q.append(x)
```

In [33]:

```
from sklearn.linear_model import Lasso
trerrs=[]
tserrs=[]
for i in Q:
    ls=Lasso(alpha=i)
    model=ls.fit(xtrain,ytrain)
    tr_pred=model.predict(xtrain)
    ts_pred=model.predict(xtest)
    from sklearn.metrics import mean_absolute_error
    ts_err = mean_absolute_error(ytest,ts_pred)
    tserrs.append(ts_err)
    tr_err = mean_absolute_error(ytrain,tr_pred)
    trerrs.append(tr_err)
    print("alpha",i,"\ttr_Err",round(tr_err,4),"\tts_Err",round(ts_err,4))
```

```
alpha 0.031     tr_Err 10.9821  ts_Err 14.5597
alpha 0.032     tr_Err 10.9352  ts_Err 14.4751
alpha 0.033     tr_Err 10.8889  ts_Err 14.3892
alpha 0.034     tr_Err 10.8437  ts_Err 14.3098
alpha 0.035     tr_Err 10.8014  ts_Err 14.2377
alpha 0.036     tr_Err 10.7612  ts_Err 14.1701
alpha 0.037     tr_Err 10.7227  ts_Err 14.1022
alpha 0.038     tr_Err 10.6851  ts_Err 14.0301
alpha 0.039     tr_Err 10.6496  ts_Err 13.9635
alpha 0.04      tr_Err 10.616   ts_Err 13.9009
alpha 0.041     tr_Err 10.5838  ts_Err 13.8415
alpha 0.042     tr_Err 10.5527  ts_Err 13.7842
alpha 0.043     tr_Err 10.5218  ts_Err 13.7282
alpha 0.044     tr_Err 10.4911  ts_Err 13.6732
alpha 0.045     tr_Err 10.4612  ts_Err 13.6151
alpha 0.046     tr_Err 10.4308  ts_Err 13.5644
alpha 0.047     tr_Err 10.3999  ts_Err 13.5128
alpha 0.048     tr_Err 10.3694  ts_Err 13.4616
alpha 0.049     tr_Err 10.3388  ts_Err 13.412
alpha 0.05      tr_Err 10.308   ts_Err 13.3645
alpha 0.051     tr_Err 10.2774  ts_Err 13.3179
alpha 0.052     tr_Err 10.2472  ts_Err 13.2732
alpha 0.053     tr_Err 10.2172  ts_Err 13.229
alpha 0.054     tr_Err 10.1877  ts_Err 13.1852
alpha 0.055     tr_Err 10.159   ts_Err 13.1439
alpha 0.056     tr_Err 10.1296  ts_Err 13.1013
alpha 0.057     tr_Err 10.1006  ts_Err 13.0595
alpha 0.058     tr_Err 10.0722  ts_Err 13.019
alpha 0.059     tr_Err 10.0434  ts_Err 12.9794
alpha 0.06      tr_Err 10.0154  ts_Err 12.9408
alpha 0.061     tr_Err 9.987    ts_Err 12.9013
alpha 0.062     tr_Err 9.9591   ts_Err 12.8644
```

```
alpha 0.063     tr_Err 9.93     ts_Err 12.8271
alpha 0.064     tr_Err 9.9026   ts_Err 12.7913
alpha 0.065     tr_Err 9.8741   ts_Err 12.7535
alpha 0.066     tr_Err 9.847    ts_Err 12.7184
alpha 0.067     tr_Err 9.8189   ts_Err 12.682
alpha 0.068     tr_Err 9.7923   ts_Err 12.6477
alpha 0.069     tr_Err 9.7644   ts_Err 12.6116
alpha 0.07      tr_Err 9.7379   ts_Err 12.577
alpha 0.071     tr_Err 9.7114   ts_Err 12.542
alpha 0.072     tr_Err 9.6823   ts_Err 12.5043
alpha 0.073     tr_Err 9.6548   ts_Err 12.4685
alpha 0.074     tr_Err 9.6274   ts_Err 12.4311
alpha 0.075     tr_Err 9.6014   ts_Err 12.3934
alpha 0.076     tr_Err 9.5737   ts_Err 12.3532
alpha 0.077     tr_Err 9.5469   ts_Err 12.3135
alpha 0.078     tr_Err 9.5183   ts_Err 12.27
alpha 0.079     tr_Err 9.4893   ts_Err 12.2276
alpha 0.08      tr_Err 9.4585   ts_Err 12.1851
alpha 0.081     tr_Err 9.4284   ts_Err 12.1433
alpha 0.082     tr_Err 9.3984   ts_Err 12.1007
alpha 0.083     tr_Err 9.3694   ts_Err 12.0578
alpha 0.084     tr_Err 9.3403   ts_Err 12.0168
alpha 0.085     tr_Err 9.3111   ts_Err 11.9739
alpha 0.086     tr_Err 9.2814   ts_Err 11.9322
alpha 0.087     tr_Err 9.2518   ts_Err 11.8915
alpha 0.088     tr_Err 9.2221   ts_Err 11.8491
alpha 0.089     tr_Err 9.1925   ts_Err 11.8078
alpha 0.09      tr_Err 9.4204   ts_Err 12.1058
alpha 0.091     tr_Err 9.3906   ts_Err 12.0632
alpha 0.092     tr_Err 9.3613   ts_Err 12.0226
alpha 0.093     tr_Err 9.3315   ts_Err 11.9833
alpha 0.094     tr_Err 9.3016   ts_Err 11.9433
alpha 0.095     tr_Err 9.2721   ts_Err 11.9045
alpha 0.096     tr_Err 9.2424   ts_Err 11.8661
alpha 0.097     tr_Err 9.213    ts_Err 11.8274
alpha 0.098     tr_Err 9.1834   ts_Err 11.79
alpha 0.099     tr_Err 9.1517   ts_Err 11.7509
alpha 0.1       tr_Err 9.1209   ts_Err 11.7125
alpha 0.101     tr_Err 9.0914   ts_Err 11.6759
alpha 0.102     tr_Err 9.061    ts_Err 11.6388
alpha 0.103     tr_Err 9.0312   ts_Err 11.6022
alpha 0.104     tr_Err 9.0026   ts_Err 11.567
alpha 0.105     tr_Err 8.9737   ts_Err 11.5314
alpha 0.106     tr_Err 8.9443   ts_Err 11.4951
alpha 0.107     tr_Err 8.9162   ts_Err 11.4599
alpha 0.108     tr_Err 8.8869   ts_Err 11.422
alpha 0.109     tr_Err 8.8552   ts_Err 11.3811
alpha 0.11      tr_Err 8.8225   ts_Err 11.339
alpha 0.111     tr_Err 8.791    ts_Err 11.299
alpha 0.112     tr_Err 8.7594   ts_Err 11.2592
alpha 0.113     tr_Err 8.7262   ts_Err 11.2174
alpha 0.114     tr_Err 8.6932   ts_Err 11.1745
alpha 0.115     tr_Err 8.6625   ts_Err 11.1339
alpha 0.116     tr_Err 8.6316   ts_Err 11.0928
alpha 0.117     tr_Err 8.5994   ts_Err 11.05
alpha 0.118     tr_Err 8.5676   ts_Err 11.0076
alpha 0.119     tr_Err 8.5377   ts_Err 10.9672
alpha 0.12      tr_Err 8.5075   ts_Err 10.9262
alpha 0.121     tr_Err 8.4758   ts_Err 10.8831
alpha 0.122     tr_Err 8.708    ts_Err 11.1856
alpha 0.123     tr_Err 8.6783   ts_Err 11.145
alpha 0.124     tr_Err 8.6486   ts_Err 11.1043
alpha 0.125     tr_Err 8.6173   ts_Err 11.0616
alpha 0.126     tr_Err 8.5863   ts_Err 11.0191
alpha 0.127     tr_Err 8.5567   ts_Err 10.9782
alpha 0.128     tr_Err 8.5279   ts_Err 10.9378
alpha 0.129     tr_Err 8.4982   ts_Err 10.8958
alpha 0.13      tr_Err 8.4679   ts_Err 10.8534
```

In [34]:

```python
from sklearn.model_selection import train_test_split
xtrain,xtest,ytrain,ytest=train_test_split(Xnew,Y,test_size=0.2,random_state=19)
ls=Lasso(alpha=0.99)
model=ls.fit(xtrain,ytrain)
tr_pred=model.predict(xtrain)
ts_pred=model.predict(xtest)
from sklearn.metrics import mean_absolute_error
ts_err=mean_absolute_error(ytest,ts_pred)
tr_err=mean_absolute_error(ytrain,tr_pred)
print("alpha",1.99,"\ttr_Err=",round(tr_err,4),"\tts_Err=",round(ts_err,4))
```

```
alpha 1.99      tr_Err= 0.6463  ts_Err= 0.5397
```

Create a Tuning grid

In [35]:

```python
Q=[]
x=0.99
for i in range(0,100,1):
    x=x+0.001
```

```
    x=round(x,4)
    Q.append(x)
```

In [36]:
```
tuning_grid={"alpha":Q}
ls=Lasso()
from sklearn.model_selection import GridSearchCV
cv=GridSearchCV(ls,tuning_grid,scoring="neg_mean_squared_error",cv=4)
cvmodel=cv.fit(Xnew,Y)
cvmodel.best_params_
```

Out[36]:
```
{'alpha': 0.991}
```

In [37]:
```
test=pd.read_csv("C:/Users/ASUS/Downloads/testing_set.csv")
```

In [38]:
```
test.head()
```

Out[38]:

	Id	MSSubClass	MSZoning	LotFrontage	LotArea	Street	Alley	LotShape	LandContour	Utilities	...	ScreenPorch	PoolArea	PoolQC
0	1461	20	RH	80.0	11622	Pave	NaN	Reg	Lvl	AllPub	...	120	0	NaN
1	1462	20	RL	81.0	14267	Pave	NaN	IR1	Lvl	AllPub	...	0	0	NaN
2	1463	60	RL	74.0	13830	Pave	NaN	IR1	Lvl	AllPub	...	0	0	NaN
3	1464	60	RL	78.0	9978	Pave	NaN	IR1	Lvl	AllPub	...	0	0	NaN
4	1465	120	RL	43.0	5005	Pave	NaN	IR1	HLS	AllPub	...	144	0	NaN

5 rows × 80 columns

In [39]:
```
from preprocesser import replacer
replacer(test)
```

In [40]:
```
test.isna().sum()
```

Out[40]:
```
Id               0
MSSubClass       0
MSZoning         0
LotFrontage      0
LotArea          0
                ..
MiscVal          0
MoSold           0
YrSold           0
SaleType         0
SaleCondition    0
Length: 80, dtype: int64
```

In [41]:
```
cols_keep=list(xtrain.columns)
```

In [42]:
```
cols_keep
```

Out[42]:
```
['MSSubClass',
 'LotArea',
 'OverallQual',
 'OverallCond',
 'MasVnrArea',
 'BsmtFinSF1',
 'BsmtFinSF2',
 'BsmtUnfSF',
 'TotalBsmtSF',
 'LowQualFinSF',
 'GrLivArea',
 'BsmtFullBath',
 'BsmtHalfBath',
 'BedroomAbvGr',
 'KitchenAbvGr',
 'Fireplaces',
 'GarageArea',
 'OpenPorchSF',
 'PoolArea',
 'YrSold',
 'SalePrice',
 'Street_Grvl',
 'Street_Pave',
 'LotShape_IR1',
 'LotShape_IR2',
 'LotShape_IR3',
 'LotShape_Reg',
 'LandContour_Bnk',
 'LandContour_HLS',
 'LandContour_Low',
 'LandContour_Lvl',
 'Utilities_AllPub',
 'Utilities_NoSeWa',
 'LotConfig_Corner',
```

```
'LotConfig_CulDSac',
'LotConfig_FR2',
'LotConfig_FR3',
'LotConfig_Inside',
'Neighborhood_Blmngtn',
'Neighborhood_Blueste',
'Neighborhood_BrDale',
'Neighborhood_BrkSide',
'Neighborhood_ClearCr',
'Neighborhood_CollgCr',
'Neighborhood_Crawfor',
'Neighborhood_Edwards',
'Neighborhood_Gilbert',
'Neighborhood_IDOTRR',
'Neighborhood_MeadowV',
'Neighborhood_Mitchel',
'Neighborhood_NAmes',
'Neighborhood_NPkVill',
'Neighborhood_NWAmes',
'Neighborhood_NoRidge',
'Neighborhood_NridgHt',
'Neighborhood_OldTown',
'Neighborhood_SWISU',
'Neighborhood_Sawyer',
'Neighborhood_SawyerW',
'Neighborhood_Somerst',
'Neighborhood_StoneBr',
'Neighborhood_Timber',
'Neighborhood_Veenker',
'Condition1_Artery',
'Condition1_Feedr',
'Condition1_Norm',
'Condition1_PosA',
'Condition1_PosN',
'Condition1_RRAe',
'Condition1_RRAn',
'Condition1_RRNe',
'Condition1_RRNn',
'Condition2_Artery',
'Condition2_Feedr',
'Condition2_Norm',
'Condition2_PosA',
'Condition2_PosN',
'Condition2_RRAe',
'Condition2_RRAn',
'Condition2_RRNn',
'BldgType_1Fam',
'BldgType_2fmCon',
'BldgType_Duplex',
'BldgType_Twnhs',
'BldgType_TwnhsE',
'HouseStyle_1.5Fin',
'HouseStyle_1.5Unf',
'HouseStyle_1Story',
'HouseStyle_2.5Fin',
'HouseStyle_2.5Unf',
'HouseStyle_2Story',
'HouseStyle_SFoyer',
'HouseStyle_SLvl',
'ExterCond_Ex',
'ExterCond_Fa',
'ExterCond_Gd',
'ExterCond_Po',
'ExterCond_TA',
'Foundation_BrkTil',
'Foundation_CBlock',
'Foundation_PConc',
'Foundation_Slab',
'Foundation_Stone',
'Foundation_Wood',
'BsmtQual_Ex',
'BsmtQual_Fa',
'BsmtQual_Gd',
'BsmtQual_TA',
'BsmtCond_Fa',
'BsmtCond_Gd',
'BsmtCond_Po',
'BsmtCond_TA',
'BsmtExposure_Av',
'BsmtExposure_Gd',
'BsmtExposure_Mn',
'BsmtExposure_No',
'BsmtFinType1_ALQ',
'BsmtFinType1_BLQ',
'BsmtFinType1_GLQ',
'BsmtFinType1_LwQ',
'BsmtFinType1_Rec',
'BsmtFinType1_Unf',
'BsmtFinType2_ALQ',
```

```
 'BsmtFinType2_BLQ',
 'BsmtFinType2_GLQ',
 'BsmtFinType2_LwQ',
 'BsmtFinType2_Rec',
 'BsmtFinType2_Unf',
 'CentralAir_N',
 'CentralAir_Y',
 'Electrical_FuseA',
 'Electrical_FuseF',
 'Electrical_FuseP',
 'Electrical_Mix',
 'Electrical_SBrkr',
 'KitchenQual_Ex',
 'KitchenQual_Fa',
 'KitchenQual_Gd',
 'KitchenQual_TA',
 'Functional_Maj1',
 'Functional_Maj2',
 'Functional_Min1',
 'Functional_Min2',
 'Functional_Mod',
 'Functional_Sev',
 'Functional_Typ',
 'GarageQual_Ex',
 'GarageQual_Fa',
 'GarageQual_Gd',
 'GarageQual_Po',
 'GarageQual_TA',
 'PavedDrive_N',
 'PavedDrive_P',
 'PavedDrive_Y',
 'PoolQC_Ex',
 'PoolQC_Fa',
 'PoolQC_Gd',
 'Fence_GdPrv',
 'Fence_GdWo',
 'Fence_MnPrv',
 'Fence_MnWw',
 'MiscFeature_Gar2',
 'MiscFeature_Othr',
 'MiscFeature_Shed',
 'MiscFeature_TenC',
 'SaleType_COD',
 'SaleType_CWD',
 'SaleType_Con',
 'SaleType_ConLD',
 'SaleType_ConLI',
 'SaleType_ConLw',
 'SaleType_New',
 'SaleType_Oth',
 'SaleType_WD',
 'SaleCondition_Abnorml',
 'SaleCondition_AdjLand',
 'SaleCondition_Alloca',
 'SaleCondition_Family',
 'SaleCondition_Normal',
 'SaleCondition_Partial']
```

In [43]:

```
cat=[]
con=[]
for i in test.columns:
    if test[i].dtypes=="object":
        cat.append(i)
    else:
        con.append(i)
```

In [44]:

```
cat
```

Out[44]:
```
['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',
 'Functional',
 'FireplaceQu',
 'GarageType',
 'GarageFinish',
 'GarageQual',
 'GarageCond',
 'PavedDrive',
 'PoolQC',
 'Fence',
 'MiscFeature',
 'SaleType',
 'SaleCondition']
```

In [45]:
```
con
```

Out[45]:
```
['Id',
 'MSSubClass',
 'LotFrontage',
 'LotArea',
 'OverallQual',
 'OverallCond',
 'YearBuilt',
 'YearRemodAdd',
 'MasVnrArea',
 'BsmtFinSF1',
 'BsmtFinSF2',
 'BsmtUnfSF',
 'TotalBsmtSF',
 '1stFlrSF',
 '2ndFlrSF',
 'LowQualFinSF',
 'GrLivArea',
 'BsmtFullBath',
 'BsmtHalfBath',
 'FullBath',
 'HalfBath',
 'BedroomAbvGr',
 'KitchenAbvGr',
 'TotRmsAbvGrd',
 'Fireplaces',
 'GarageYrBlt',
 'GarageCars',
 'GarageArea',
 'WoodDeckSF',
 'OpenPorchSF',
 'EnclosedPorch',
 '3SsnPorch',
 'ScreenPorch',
 'PoolArea',
 'MiscVal',
 'MoSold',
 'YrSold']
```

In [46]:
```
X1=pd.DataFrame(ss.fit_transform(test[con]))
```

In [47]:
```
X2=pd.get_dummies(test[cat],dtype='int')
```

In [48]:
```
X=X1.join(X2)
```

In [48]: X=X1.join(X2)

```
In [49]: X
```

Out[49]:

	0	1	2	3	4	5	6	7	8	9	...	SaleType_ConLw	SaleTy
0	-1.730864	-0.874711	0.555587	0.363929	-0.751101	0.400766	-0.340945	-1.072885	-0.570108	0.063295	...	0	
1	-1.728490	-0.874711	0.604239	0.897861	-0.054877	0.400766	-0.439695	-1.214908	0.041273	1.063392	...	0	
2	-1.726115	0.061351	0.263676	0.809646	-0.751101	-0.497418	0.844059	0.678742	-0.570108	0.773254	...	0	
3	-1.723741	0.061351	0.458284	0.032064	-0.054877	0.400766	0.876976	0.678742	-0.456889	0.357829	...	0	
4	-1.721367	1.465443	-1.244533	-0.971808	1.337571	-0.497418	0.679475	0.394694	-0.570108	-0.387298	...	0	
...	...	...	...	...	...	...	...	...	...	...	...	...	
1454	1.721367	2.401505	-2.314875	-1.591330	-1.447325	1.298950	-0.044694	-0.646813	-0.570108	-0.965376	...	0	
1455	1.723741	2.401505	-2.314875	-1.599808	-1.447325	-0.497418	-0.044694	-0.646813	-0.570108	-0.411477	...	0	
1456	1.726115	-0.874711	4.447740	2.055150	-0.751101	1.298950	-0.373861	0.584059	-0.570108	1.724994	...	0	
1457	1.728490	0.646389	-0.320147	0.125527	-0.751101	-0.497418	0.679475	0.394694	-0.570108	-0.224645	...	0	
1458	1.730864	0.061351	0.263676	-0.038790	0.641347	-0.497418	0.712392	0.489377	-0.037980	0.700719	...	0	

1459 rows × 270 columns

```
In [50]: cols_to_add=['MSSubClass', 'LotArea', 'OverallQual', 'OverallCond', 'MasVnrArea', 'BsmtFinSF1', 'BsmtFinSF2', '
```

```
In [51]: for i in cols_to_add:
             X[i]=0
```

```
In [52]: final_preds=model.predict(X[cols_keep])
```

```
In [53]: final_preds
```

```
Out[53]: array([180921.21450504, 180921.21450504, 180921.21450504, ...,
                180921.21450504, 180921.21450504, 180921.21450504])
```

```
In [54]: test["Predicted_sale_Price"]=final_preds
```

```
In [55]: test.head()
```

Out[55]:

	Id	MSSubClass	MSZoning	LotFrontage	LotArea	Street	Alley	LotShape	LandContour	Utilities	...	PoolArea	PoolQC	Fence	MiscF
0	1461	20	RH	80.0	11622	Pave	Grvl	Reg	Lvl	AllPub	...	0	Ex	MnPrv	
1	1462	20	RL	81.0	14267	Pave	Grvl	IR1	Lvl	AllPub	...	0	Ex	MnPrv	
2	1463	60	RL	74.0	13830	Pave	Grvl	IR1	Lvl	AllPub	...	0	Ex	MnPrv	
3	1464	60	RL	78.0	9978	Pave	Grvl	IR1	Lvl	AllPub	...	0	Ex	MnPrv	
4	1465	120	RL	43.0	5005	Pave	Grvl	IR1	HLS	AllPub	...	0	Ex	MnPrv	

5 rows × 81 columns

```
In [56]: test[["Id","Predicted_sale_Price"]].to_csv("C:/Users/ASUS/Downloads/Submission.csv")
```

```
In [57]: pd.set_option("display.max_rows",5000)
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js