SURIYA S

225229140

house price prediction using LR with regularzation

In [1]:

```
import pandas as pd
```

In [2]:

```
df=pd.read_csv(r'Ames_House_Sales_Cropped.csv')
df
```

Out[2]:

	BldgType	CentralAir	1stFlrSF	2ndFlrSF	3SsnPorch	BedroomAbvGr	BsmtFinSF1	BsmtFinSF2	BsmtFullBath	BsmtHalfBath	...	OverallC
0	1Fam	Y	856.0	854.0	0.0	3	706.0	0.0	1	0	...	
1	1Fam	Y	1262.0	0.0	0.0	3	978.0	0.0	0	1	...	
2	1Fam	Y	920.0	866.0	0.0	3	486.0	0.0	1	0	...	
3	1Fam	Y	961.0	756.0	0.0	3	216.0	0.0	1	0	...	
4	1Fam	Y	1145.0	1053.0	0.0	4	655.0	0.0	1	0	...	
5	1Fam	Y	796.0	566.0	320.0	1	732.0	0.0	1	0	...	
6	1Fam	Y	1694.0	0.0	0.0	3	1369.0	0.0	1	0	...	
7	1Fam	Y	1107.0	983.0	0.0	3	859.0	32.0	1	0	...	
8	1Fam	Y	1022.0	752.0	0.0	2	0.0	0.0	0	0	...	
9	2fmCon	Y	1077.0	0.0	0.0	2	851.0	0.0	1	0	...	
10	1Fam	Y	1040.0	0.0	0.0	3	906.0	0.0	1	0	...	
11	1Fam	Y	1182.0	1142.0	0.0	4	998.0	0.0	1	0	...	
12	1Fam	Y	912.0	0.0	0.0	2	737.0	0.0	1	0	...	
13	1Fam	Y	1494.0	0.0	0.0	3	0.0	0.0	0	0	...	
14	1Fam	Y	1253.0	0.0	0.0	2	733.0	0.0	1	0	...	
15	1Fam	Y	854.0	0.0	0.0	2	0.0	0.0	0	0	...	
16	1Fam	Y	1004.0	0.0	0.0	2	578.0	0.0	1	0	...	
17	Duplex	Y	1296.0	0.0	0.0	2	0.0	0.0	0	0	...	
18	1Fam	Y	1114.0	0.0	0.0	3	646.0	0.0	1	0	...	
19	1Fam	Y	1339.0	0.0	0.0	3	504.0	0.0	0	0	...	
20	1Fam	Y	1158.0	1218.0	0.0	4	0.0	0.0	0	0	...	
21	1Fam	Y	1108.0	0.0	0.0	3	0.0	0.0	0	0	...	
22	1Fam	Y	1795.0	0.0	0.0	3	0.0	0.0	0	0	...	
23	TwnhsE	Y	1060.0	0.0	0.0	3	840.0	0.0	1	0	...	
24	1Fam	Y	1060.0	0.0	0.0	3	188.0	668.0	1	0	...	
25	1Fam	Y	1600.0	0.0	0.0	3	0.0	0.0	0	0	...	
26	1Fam	Y	900.0	0.0	0.0	3	234.0	486.0	0	1	...	
27	1Fam	Y	1704.0	0.0	0.0	3	1218.0	0.0	1	0	...	
28	1Fam	Y	1600.0	0.0	0.0	2	1277.0	0.0	1	0	...	
29	1Fam	N	520.0	0.0	0.0	1	0.0	0.0	0	0	...	
...	...	...	...	...	...	...	...	...	...	...	...	
1349	1Fam	Y	1048.0	510.0	0.0	3	580.0	0.0	1	0	...	
1350	1Fam	Y	804.0	0.0	0.0	2	510.0	0.0	1	0	...	
1351	1Fam	Y	1440.0	0.0	0.0	3	678.0	0.0	0	0	...	
1352	1Fam	Y	734.0	1104.0	0.0	4	0.0	0.0	0	0	...	
1353	TwnhsE	Y	958.0	0.0	0.0	2	958.0	0.0	0	0	...	
1354	1Fam	Y	968.0	0.0	0.0	4	0.0	0.0	0	0	...	
1355	1Fam	Y	962.0	830.0	0.0	3	0.0	0.0	1	0	...	
1356	1Fam	Y	1126.0	0.0	0.0	3	936.0	0.0	1	0	...	
1357	1Fam	Y	1537.0	0.0	0.0	3	0.0	0.0	1	0	...	
1358	1Fam	Y	864.0	0.0	0.0	3	616.0	0.0	0	0	...	
1359	1Fam	Y	1932.0	0.0	304.0	2	1336.0	0.0	1	0	...	
1360	1Fam	Y	1236.0	0.0	0.0	2	600.0	0.0	1	0	...	
1361	1Fam	Y	1040.0	685.0	0.0	3	315.0	110.0	0	0	...	
1362	1Fam	Y	1423.0	748.0	0.0	3	0.0	0.0	0	0	...	
1363	TwnhsE	Y	848.0	0.0	0.0	1	697.0	0.0	1	0	...	
1364	1Fam	Y	1026.0	981.0	0.0	3	765.0	0.0	1	0	...	
1365	1Fam	N	952.0	0.0	0.0	2	0.0	0.0	0	0	...	
1366	1Fam	Y	1422.0	0.0	0.0	3	0.0	0.0	0	0	...	
1367	1Fam	Y	913.0	0.0	0.0	3	187.0	627.0	1	0	...	
1368	1Fam	Y	1188.0	0.0	0.0	3	593.0	0.0	0	0	...	
1369	1Fam	Y	1220.0	870.0	0.0	3	1079.0	0.0	1	0	...	
1370	1Fam	N	796.0	550.0	0.0	2	0.0	0.0	0	0	...	
1371	1Fam	Y	1578.0	0.0	0.0	3	0.0	0.0	0	0	...	

	BldgType	CentralAir	1stFlrSF	2ndFlrSF	3SsnPorch	BedroomAbvGr	BsmtFinSF1	BsmtFinSF2	BsmtFullBath	BsmtHalfBath	...	OverallC
1372	TwnhsE	Y	1072.0	0.0	0.0	2	547.0	0.0	1	0	...	
1373	1Fam	Y	1221.0	0.0	0.0	2	410.0	0.0	1	0	...	
1374	1Fam	Y	953.0	694.0	0.0	3	0.0	0.0	0	0	...	
1375	1Fam	Y	2073.0	0.0	0.0	3	790.0	163.0	1	0	...	
1376	1Fam	Y	1188.0	1152.0	0.0	4	275.0	0.0	0	0	...	
1377	1Fam	Y	1078.0	0.0	0.0	2	49.0	1029.0	1	0	...	
1378	1Fam	Y	1256.0	0.0	0.0	3	830.0	290.0	1	0	...	

1379 rows × 39 columns

In [3]:

```
df.head()
```

Out[3]:

	BldgType	CentralAir	1stFlrSF	2ndFlrSF	3SsnPorch	BedroomAbvGr	BsmtFinSF1	BsmtFinSF2	BsmtFullBath	BsmtHalfBath	...	OverallQua
0	1Fam	Y	856.0	854.0	0.0	3	706.0	0.0	1	0	...	7
1	1Fam	Y	1262.0	0.0	0.0	3	978.0	0.0	0	1	...	6
2	1Fam	Y	920.0	866.0	0.0	3	486.0	0.0	1	0	...	7
3	1Fam	Y	961.0	756.0	0.0	3	216.0	0.0	1	0	...	7
4	1Fam	Y	1145.0	1053.0	0.0	4	655.0	0.0	1	0	...	8

5 rows × 39 columns

In [4]:

```
df.shape
```

Out[4]:

```
(1379, 39)
```

In [5]:

```
df.columns
```

Out[5]:

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

In [6]:

```
df.dtypes
```

Out[6]:

```
BldgType          object
CentralAir        object
1stFlrSF         float64
2ndFlrSF         float64
3SsnPorch        float64
BedroomAbvGr       int64
BsmtFinSF1       float64
BsmtFinSF2       float64
BsmtFullBath       int64
BsmtHalfBath       int64
BsmtUnfSF        float64
EnclosedPorch    float64
Fireplaces         int64
FullBath           int64
GarageArea       float64
GarageCars         int64
GarageYrBlt      float64
GrLivArea        float64
HalfBath           int64
KitchenAbvGr       int64
LotArea          float64
LotFrontage      float64
LowQualFinSF     float64
MSSubClass         int64
MasVnrArea       float64
MiscVal          float64
MoSold             int64
OpenPorchSF      float64
OverallCond        int64
OverallQual        int64
PoolArea         float64
ScreenPorch      float64
TotRmsAbvGrd       int64
TotalBsmtSF      float64
WoodDeckSF       float64
YearBuilt          int64
YearRemodAdd       int64
YrSold             int64
SalePrice        float64
dtype: object
```

In [7]:

```
df.info
```

Out[7]:

```
<bound method DataFrame.info of      BldgType CentralAir  1stFlrSF  2ndFlrSF  3SsnPorch  BedroomAbvGr  \
0        1Fam          Y     856.0     854.0        0.0             3
1        1Fam          Y    1262.0       0.0        0.0             3
2        1Fam          Y     920.0     866.0        0.0             3
3        1Fam          Y     961.0     756.0        0.0             3
4        1Fam          Y    1145.0    1053.0        0.0             4
5        1Fam          Y     796.0     566.0      320.0             1
6        1Fam          Y    1694.0       0.0        0.0             3
7        1Fam          Y    1107.0     983.0        0.0             3
8        1Fam          Y    1022.0     752.0        0.0             2
9      2fmCon          Y    1077.0       0.0        0.0             2
10       1Fam          Y    1040.0       0.0        0.0             3
11       1Fam          Y    1182.0    1142.0        0.0             4
12       1Fam          Y     912.0       0.0        0.0             2
13       1Fam          Y    1494.0       0.0        0.0             3
14       1Fam          Y    1253.0       0.0        0.0             2
15       1Fam          Y     854.0       0.0        0.0             2
16       1Fam          Y    1004.0       0.0        0.0             2
```

In [8]:

```
df.CentralAir.value_counts()
```

Out[8]:

```
Y    1310
N      69
Name: CentralAir, dtype: int64
```

In [9]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1379 entries, 0 to 1378
Data columns (total 39 columns):
BldgType         1379 non-null object
CentralAir       1379 non-null object
1stFlrSF         1379 non-null float64
2ndFlrSF         1379 non-null float64
3SsnPorch        1379 non-null float64
BedroomAbvGr     1379 non-null int64
BsmtFinSF1       1379 non-null float64
BsmtFinSF2       1379 non-null float64
BsmtFullBath     1379 non-null int64
BsmtHalfBath     1379 non-null int64
BsmtUnfSF        1379 non-null float64
EnclosedPorch    1379 non-null float64
Fireplaces       1379 non-null int64
FullBath         1379 non-null int64
GarageArea       1379 non-null float64
GarageCars       1379 non-null int64
GarageYrBlt      1379 non-null float64
GrLivArea        1379 non-null float64
HalfBath         1379 non-null int64
KitchenAbvGr     1379 non-null int64
LotArea          1379 non-null float64
LotFrontage      1379 non-null float64
LowQualFinSF     1379 non-null float64
MSSubClass       1379 non-null int64
MasVnrArea       1379 non-null float64
MiscVal          1379 non-null float64
MoSold           1379 non-null int64
OpenPorchSF      1379 non-null float64
OverallCond      1379 non-null int64
OverallQual      1379 non-null int64
PoolArea         1379 non-null float64
ScreenPorch      1379 non-null float64
TotRmsAbvGrd     1379 non-null int64
TotalBsmtSF      1379 non-null float64
WoodDeckSF       1379 non-null float64
YearBuilt        1379 non-null int64
YearRemodAdd     1379 non-null int64
YrSold           1379 non-null int64
SalePrice        1379 non-null float64
dtypes: float64(21), int64(16), object(2)
memory usage: 420.2+ KB
```

In [10]:

```
df1=df.drop("BldgType",axis=1)
df1
```

Out[10]:

	CentralAir	1stFlrSF	2ndFlrSF	3SsnPorch	BedroomAbvGr	BsmtFinSF1	BsmtFinSF2	BsmtFullBath	BsmtHalfBath	BsmtUnfSF	...	Overa
0	Y	856.0	854.0	0.0	3	706.0	0.0	1	0	150.0	...	
1	Y	1262.0	0.0	0.0	3	978.0	0.0	0	1	284.0	...	
2	Y	920.0	866.0	0.0	3	486.0	0.0	1	0	434.0	...	
3	Y	961.0	756.0	0.0	3	216.0	0.0	1	0	540.0	...	
4	Y	1145.0	1053.0	0.0	4	655.0	0.0	1	0	490.0	...	
5	Y	796.0	566.0	320.0	1	732.0	0.0	1	0	64.0	...	
6	Y	1694.0	0.0	0.0	3	1369.0	0.0	1	0	317.0	...	
7	Y	1107.0	983.0	0.0	3	859.0	32.0	1	0	216.0	...	
8	Y	1022.0	752.0	0.0	2	0.0	0.0	0	0	952.0	...	
9	Y	1077.0	0.0	0.0	2	851.0	0.0	1	0	140.0	...	
10	Y	1040.0	0.0	0.0	3	906.0	0.0	1	0	134.0	...	
11	Y	1182.0	1142.0	0.0	4	998.0	0.0	1	0	177.0	...	
12	Y	912.0	0.0	0.0	2	737.0	0.0	1	0	175.0	...	
13	Y	1494.0	0.0	0.0	3	0.0	0.0	0	0	1494.0	...	
14	Y	1253.0	0.0	0.0	2	733.0	0.0	1	0	520.0	...	
15	Y	854.0	0.0	0.0	2	0.0	0.0	0	0	832.0	...	
16	Y	1004.0	0.0	0.0	2	578.0	0.0	1	0	426.0	...	
17	Y	1296.0	0.0	0.0	2	0.0	0.0	0	0	0.0	...	
18	Y	1114.0	0.0	0.0	3	646.0	0.0	1	0	468.0	...	
19	Y	1339.0	0.0	0.0	3	504.0	0.0	0	0	525.0	...	
20	Y	1158.0	1218.0	0.0	4	0.0	0.0	0	0	1158.0	...	
21	Y	1108.0	0.0	0.0	3	0.0	0.0	0	0	637.0	...	
22	Y	1795.0	0.0	0.0	3	0.0	0.0	0	0	1777.0	...	
23	Y	1060.0	0.0	0.0	3	840.0	0.0	1	0	200.0	...	
24	Y	1060.0	0.0	0.0	3	188.0	668.0	1	0	204.0	...	
25	Y	1600.0	0.0	0.0	3	0.0	0.0	0	0	1566.0	...	
26	Y	900.0	0.0	0.0	3	234.0	486.0	0	1	180.0	...	
27	Y	1704.0	0.0	0.0	3	1218.0	0.0	1	0	486.0	...	
28	Y	1600.0	0.0	0.0	2	1277.0	0.0	1	0	207.0	...	
29	N	520.0	0.0	0.0	1	0.0	0.0	0	0	520.0	...	
...	...	...	...	...	...	...	...	...	...	...	...	
1349	Y	1048.0	510.0	0.0	3	580.0	0.0	1	0	333.0	...	
1350	Y	804.0	0.0	0.0	2	510.0	0.0	1	0	278.0	...	
1351	Y	1440.0	0.0	0.0	3	678.0	0.0	0	0	762.0	...	
1352	Y	734.0	1104.0	0.0	4	0.0	0.0	0	0	732.0	...	
1353	Y	958.0	0.0	0.0	2	958.0	0.0	0	0	0.0	...	
1354	Y	968.0	0.0	0.0	4	0.0	0.0	0	0	656.0	...	
1355	Y	962.0	830.0	0.0	3	0.0	0.0	1	0	936.0	...	
1356	Y	1126.0	0.0	0.0	3	936.0	0.0	1	0	190.0	...	
1357	Y	1537.0	0.0	0.0	3	0.0	0.0	1	0	1319.0	...	
1358	Y	864.0	0.0	0.0	3	616.0	0.0	0	0	248.0	...	
1359	Y	1932.0	0.0	304.0	2	1336.0	0.0	1	0	596.0	...	
1360	Y	1236.0	0.0	0.0	2	600.0	0.0	1	0	312.0	...	
1361	Y	1040.0	685.0	0.0	3	315.0	110.0	0	0	114.0	...	
1362	Y	1423.0	748.0	0.0	3	0.0	0.0	0	0	588.0	...	
1363	Y	848.0	0.0	0.0	1	697.0	0.0	1	0	151.0	...	
1364	Y	1026.0	981.0	0.0	3	765.0	0.0	1	0	252.0	...	
1365	N	952.0	0.0	0.0	2	0.0	0.0	0	0	952.0	...	
1366	Y	1422.0	0.0	0.0	3	0.0	0.0	0	0	1422.0	...	
1367	Y	913.0	0.0	0.0	3	187.0	627.0	1	0	0.0	...	
1368	Y	1188.0	0.0	0.0	3	593.0	0.0	0	0	595.0	...	
1369	Y	1220.0	870.0	0.0	3	1079.0	0.0	1	0	141.0	...	
1370	N	796.0	550.0	0.0	2	0.0	0.0	0	0	560.0	...	
1371	Y	1578.0	0.0	0.0	3	0.0	0.0	0	0	1573.0	...	

	CentralAir	1stFlrSF	2ndFlrSF	3SsnPorch	BedroomAbvGr	BsmtFinSF1	BsmtFinSF2	BsmtFullBath	BsmtHalfBath	BsmtUnfSF	...	Overa
1372	Y	1072.0	0.0	0.0	2	547.0	0.0	1	0	0.0	...	
1373	Y	1221.0	0.0	0.0	2	410.0	0.0	1	0	811.0	...	
1374	Y	953.0	694.0	0.0	3	0.0	0.0	0	0	953.0	...	
1375	Y	2073.0	0.0	0.0	3	790.0	163.0	1	0	589.0	...	
1376	Y	1188.0	1152.0	0.0	4	275.0	0.0	0	0	877.0	...	
1377	Y	1078.0	0.0	0.0	2	49.0	1029.0	1	0	0.0	...	
1378	Y	1256.0	0.0	0.0	3	830.0	290.0	1	0	136.0	...	

1379 rows × 38 columns

In [11]:

```
df2=df1.drop("CentralAir",axis=1)
df2
```

Out[11]:

	1stFlrSF	2ndFlrSF	3SsnPorch	BedroomAbvGr	BsmtFinSF1	BsmtFinSF2	BsmtFullBath	BsmtHalfBath	BsmtUnfSF	EnclosedPorch	...
0	856.0	854.0	0.0	3	706.0	0.0	1	0	150.0	0.0	...
1	1262.0	0.0	0.0	3	978.0	0.0	0	1	284.0	0.0	...
2	920.0	866.0	0.0	3	486.0	0.0	1	0	434.0	0.0	...
3	961.0	756.0	0.0	3	216.0	0.0	1	0	540.0	272.0	...
4	1145.0	1053.0	0.0	4	655.0	0.0	1	0	490.0	0.0	...
5	796.0	566.0	320.0	1	732.0	0.0	1	0	64.0	0.0	...
6	1694.0	0.0	0.0	3	1369.0	0.0	1	0	317.0	0.0	...
7	1107.0	983.0	0.0	3	859.0	32.0	1	0	216.0	228.0	...
8	1022.0	752.0	0.0	2	0.0	0.0	0	0	952.0	205.0	...
9	1077.0	0.0	0.0	2	851.0	0.0	1	0	140.0	0.0	...
10	1040.0	0.0	0.0	3	906.0	0.0	1	0	134.0	0.0	...
11	1182.0	1142.0	0.0	4	998.0	0.0	1	0	177.0	0.0	...
12	912.0	0.0	0.0	2	737.0	0.0	1	0	175.0	0.0	...
13	1494.0	0.0	0.0	3	0.0	0.0	0	0	1494.0	0.0	...
14	1253.0	0.0	0.0	2	733.0	0.0	1	0	520.0	176.0	...
15	854.0	0.0	0.0	2	0.0	0.0	0	0	832.0	0.0	...
16	1004.0	0.0	0.0	2	578.0	0.0	1	0	426.0	0.0	...
17	1296.0	0.0	0.0	2	0.0	0.0	0	0	0.0	0.0	...
18	1114.0	0.0	0.0	3	646.0	0.0	1	0	468.0	0.0	...
19	1339.0	0.0	0.0	3	504.0	0.0	0	0	525.0	0.0	...
20	1158.0	1218.0	0.0	4	0.0	0.0	0	0	1158.0	0.0	...
21	1108.0	0.0	0.0	3	0.0	0.0	0	0	637.0	205.0	...
22	1795.0	0.0	0.0	3	0.0	0.0	0	0	1777.0	0.0	...
23	1060.0	0.0	0.0	3	840.0	0.0	1	0	200.0	0.0	...
24	1060.0	0.0	0.0	3	188.0	668.0	1	0	204.0	0.0	...
25	1600.0	0.0	0.0	3	0.0	0.0	0	0	1566.0	0.0	...
26	900.0	0.0	0.0	3	234.0	486.0	0	1	180.0	0.0	...
27	1704.0	0.0	0.0	3	1218.0	0.0	1	0	486.0	0.0	...
28	1600.0	0.0	0.0	2	1277.0	0.0	1	0	207.0	0.0	...
29	520.0	0.0	0.0	1	0.0	0.0	0	0	520.0	87.0	...
...	...	...	...	...	...	...	...	...	...	...	...
1349	1048.0	510.0	0.0	3	580.0	0.0	1	0	333.0	0.0	...
1350	804.0	0.0	0.0	2	510.0	0.0	1	0	278.0	154.0	...
1351	1440.0	0.0	0.0	3	678.0	0.0	0	0	762.0	99.0	...
1352	734.0	1104.0	0.0	4	0.0	0.0	0	0	732.0	0.0	...
1353	958.0	0.0	0.0	2	958.0	0.0	0	0	0.0	0.0	...
1354	968.0	0.0	0.0	4	0.0	0.0	0	0	656.0	0.0	...
1355	962.0	830.0	0.0	3	0.0	0.0	1	0	936.0	0.0	...
1356	1126.0	0.0	0.0	3	936.0	0.0	1	0	190.0	0.0	...
1357	1537.0	0.0	0.0	3	0.0	0.0	1	0	1319.0	0.0	...
1358	864.0	0.0	0.0	3	616.0	0.0	0	0	248.0	0.0	...
1359	1932.0	0.0	304.0	2	1336.0	0.0	1	0	596.0	0.0	...
1360	1236.0	0.0	0.0	2	600.0	0.0	1	0	312.0	158.0	...
1361	1040.0	685.0	0.0	3	315.0	110.0	0	0	114.0	216.0	...
1362	1423.0	748.0	0.0	3	0.0	0.0	0	0	588.0	0.0	...
1363	848.0	0.0	0.0	1	697.0	0.0	1	0	151.0	0.0	...
1364	1026.0	981.0	0.0	3	765.0	0.0	1	0	252.0	0.0	...
1365	952.0	0.0	0.0	2	0.0	0.0	0	0	952.0	0.0	...
1366	1422.0	0.0	0.0	3	0.0	0.0	0	0	1422.0	0.0	...
1367	913.0	0.0	0.0	3	187.0	627.0	1	0	0.0	252.0	...
1368	1188.0	0.0	0.0	3	593.0	0.0	0	0	595.0	0.0	...
1369	1220.0	870.0	0.0	3	1079.0	0.0	1	0	141.0	0.0	...
1370	796.0	550.0	0.0	2	0.0	0.0	0	0	560.0	0.0	...
1371	1578.0	0.0	0.0	3	0.0	0.0	0	0	1573.0	0.0	...

	1stFlrSF	2ndFlrSF	3SsnPorch	BedroomAbvGr	BsmtFinSF1	BsmtFinSF2	BsmtFullBath	BsmtHalfBath	BsmtUnfSF	EnclosedPorch	...
1372	1072.0	0.0	0.0	2	547.0	0.0	1	0	0.0	0.0	...
1373	1221.0	0.0	0.0	2	410.0	0.0	1	0	811.0	0.0	...
1374	953.0	694.0	0.0	3	0.0	0.0	0	0	953.0	0.0	...
1375	2073.0	0.0	0.0	3	790.0	163.0	1	0	589.0	0.0	...
1376	1188.0	1152.0	0.0	4	275.0	0.0	0	0	877.0	0.0	...
1377	1078.0	0.0	0.0	2	49.0	1029.0	1	0	0.0	112.0	...
1378	1256.0	0.0	0.0	3	830.0	290.0	1	0	136.0	0.0	...

1379 rows × 37 columns

In [12]:

```
p=df.pop("BldgType")
p
```

Out[12]:

```
0         1Fam
1         1Fam
2         1Fam
3         1Fam
4         1Fam
5         1Fam
6         1Fam
7         1Fam
8         1Fam
9       2fmCon
10        1Fam
11        1Fam
12        1Fam
13        1Fam
14        1Fam
15        1Fam
16        1Fam
17      Duplex
18        1Fam
19        1Fam
20        1Fam
21        1Fam
22        1Fam
23      TwnhsE
24        1Fam
25        1Fam
26        1Fam
27        1Fam
28        1Fam
29        1Fam
         ...
1349      1Fam
1350      1Fam
1351      1Fam
1352      1Fam
1353    TwnhsE
1354      1Fam
1355      1Fam
1356      1Fam
1357      1Fam
1358      1Fam
1359      1Fam
1360      1Fam
1361      1Fam
1362      1Fam
1363    TwnhsE
1364      1Fam
1365      1Fam
1366      1Fam
1367      1Fam
1368      1Fam
1369      1Fam
1370      1Fam
1371      1Fam
1372    TwnhsE
1373      1Fam
1374      1Fam
1375      1Fam
1376      1Fam
1377      1Fam
1378      1Fam
Name: BldgType, Length: 1379, dtype: object
```

In [13]:

```
p1=df.pop("CentralAir")
p
```

Out[13]:

```
0         1Fam
1         1Fam
2         1Fam
3         1Fam
4         1Fam
5         1Fam
6         1Fam
7         1Fam
8         1Fam
9       2fmCon
10        1Fam
11        1Fam
12        1Fam
13        1Fam
14        1Fam
15        1Fam
16        1Fam
17      Duplex
18        1Fam
19        1Fam
20        1Fam
21        1Fam
22        1Fam
23      TwnhsE
24        1Fam
25        1Fam
26        1Fam
27        1Fam
28        1Fam
29        1Fam
         ...
1349      1Fam
1350      1Fam
1351      1Fam
1352      1Fam
1353    TwnhsE
1354      1Fam
1355      1Fam
1356      1Fam
1357      1Fam
1358      1Fam
1359      1Fam
1360      1Fam
1361      1Fam
1362      1Fam
1363    TwnhsE
1364      1Fam
1365      1Fam
1366      1Fam
1367      1Fam
1368      1Fam
1369      1Fam
1370      1Fam
1371      1Fam
1372    TwnhsE
1373      1Fam
1374      1Fam
1375      1Fam
1376      1Fam
1377      1Fam
1378      1Fam
Name: BldgType, Length: 1379, dtype: object
```

In [14]:

```
df
```

Out[14]:

	1stFlrSF	2ndFlrSF	3SsnPorch	BedroomAbvGr	BsmtFinSF1	BsmtFinSF2	BsmtFullBath	BsmtHalfBath	BsmtUnfSF	EnclosedPorch	...
0	856.0	854.0	0.0	3	706.0	0.0	1	0	150.0	0.0	...
1	1262.0	0.0	0.0	3	978.0	0.0	0	1	284.0	0.0	...
2	920.0	866.0	0.0	3	486.0	0.0	1	0	434.0	0.0	...
3	961.0	756.0	0.0	3	216.0	0.0	1	0	540.0	272.0	...
4	1145.0	1053.0	0.0	4	655.0	0.0	1	0	490.0	0.0	...
5	796.0	566.0	320.0	1	732.0	0.0	1	0	64.0	0.0	...
6	1694.0	0.0	0.0	3	1369.0	0.0	1	0	317.0	0.0	...
7	1107.0	983.0	0.0	3	859.0	32.0	1	0	216.0	228.0	...
8	1022.0	752.0	0.0	2	0.0	0.0	0	0	952.0	205.0	...
9	1077.0	0.0	0.0	2	851.0	0.0	1	0	140.0	0.0	...
10	1040.0	0.0	0.0	3	906.0	0.0	1	0	134.0	0.0	...
11	1182.0	1142.0	0.0	4	998.0	0.0	1	0	177.0	0.0	...
12	912.0	0.0	0.0	2	737.0	0.0	1	0	175.0	0.0	...
13	1494.0	0.0	0.0	3	0.0	0.0	0	0	1494.0	0.0	...
14	1253.0	0.0	0.0	2	733.0	0.0	1	0	520.0	176.0	...
15	854.0	0.0	0.0	2	0.0	0.0	0	0	832.0	0.0	...
16	1004.0	0.0	0.0	2	578.0	0.0	1	0	426.0	0.0	...
17	1296.0	0.0	0.0	2	0.0	0.0	0	0	0.0	0.0	...
18	1114.0	0.0	0.0	3	646.0	0.0	1	0	468.0	0.0	...
19	1339.0	0.0	0.0	3	504.0	0.0	0	0	525.0	0.0	...
20	1158.0	1218.0	0.0	4	0.0	0.0	0	0	1158.0	0.0	...
21	1108.0	0.0	0.0	3	0.0	0.0	0	0	637.0	205.0	...
22	1795.0	0.0	0.0	3	0.0	0.0	0	0	1777.0	0.0	...
23	1060.0	0.0	0.0	3	840.0	0.0	1	0	200.0	0.0	...
24	1060.0	0.0	0.0	3	188.0	668.0	1	0	204.0	0.0	...
25	1600.0	0.0	0.0	3	0.0	0.0	0	0	1566.0	0.0	...
26	900.0	0.0	0.0	3	234.0	486.0	0	1	180.0	0.0	...
27	1704.0	0.0	0.0	3	1218.0	0.0	1	0	486.0	0.0	...
28	1600.0	0.0	0.0	2	1277.0	0.0	1	0	207.0	0.0	...
29	520.0	0.0	0.0	1	0.0	0.0	0	0	520.0	87.0	...
...	...	...	...	...	...	...	...	...	...	...	...
1349	1048.0	510.0	0.0	3	580.0	0.0	1	0	333.0	0.0	...
1350	804.0	0.0	0.0	2	510.0	0.0	1	0	278.0	154.0	...
1351	1440.0	0.0	0.0	3	678.0	0.0	0	0	762.0	99.0	...
1352	734.0	1104.0	0.0	4	0.0	0.0	0	0	732.0	0.0	...
1353	958.0	0.0	0.0	2	958.0	0.0	0	0	0.0	0.0	...
1354	968.0	0.0	0.0	4	0.0	0.0	0	0	656.0	0.0	...
1355	962.0	830.0	0.0	3	0.0	0.0	1	0	936.0	0.0	...
1356	1126.0	0.0	0.0	3	936.0	0.0	1	0	190.0	0.0	...
1357	1537.0	0.0	0.0	3	0.0	0.0	1	0	1319.0	0.0	...
1358	864.0	0.0	0.0	3	616.0	0.0	0	0	248.0	0.0	...
1359	1932.0	0.0	304.0	2	1336.0	0.0	1	0	596.0	0.0	...
1360	1236.0	0.0	0.0	2	600.0	0.0	1	0	312.0	158.0	...
1361	1040.0	685.0	0.0	3	315.0	110.0	0	0	114.0	216.0	...
1362	1423.0	748.0	0.0	3	0.0	0.0	0	0	588.0	0.0	...
1363	848.0	0.0	0.0	1	697.0	0.0	1	0	151.0	0.0	...
1364	1026.0	981.0	0.0	3	765.0	0.0	1	0	252.0	0.0	...
1365	952.0	0.0	0.0	2	0.0	0.0	0	0	952.0	0.0	...
1366	1422.0	0.0	0.0	3	0.0	0.0	0	0	1422.0	0.0	...
1367	913.0	0.0	0.0	3	187.0	627.0	1	0	0.0	252.0	...
1368	1188.0	0.0	0.0	3	593.0	0.0	0	0	595.0	0.0	...
1369	1220.0	870.0	0.0	3	1079.0	0.0	1	0	141.0	0.0	...
1370	796.0	550.0	0.0	2	0.0	0.0	0	0	560.0	0.0	...
1371	1578.0	0.0	0.0	3	0.0	0.0	0	0	1573.0	0.0	...

	1stFlrSF	2ndFlrSF	3SsnPorch	BedroomAbvGr	BsmtFinSF1	BsmtFinSF2	BsmtFullBath	BsmtHalfBath	BsmtUnfSF	EnclosedPorch	...	C
1372	1072.0	0.0	0.0	2	547.0	0.0	1	0	0.0	0.0	...	
1373	1221.0	0.0	0.0	2	410.0	0.0	1	0	811.0	0.0	...	
1374	953.0	694.0	0.0	3	0.0	0.0	0	0	953.0	0.0	...	
1375	2073.0	0.0	0.0	3	790.0	163.0	1	0	589.0	0.0	...	
1376	1188.0	1152.0	0.0	4	275.0	0.0	0	0	877.0	0.0	...	
1377	1078.0	0.0	0.0	2	49.0	1029.0	1	0	0.0	112.0	...	
1378	1256.0	0.0	0.0	3	830.0	290.0	1	0	136.0	0.0	...	

1379 rows × 37 columns

In [15]:

```
y=df[["SalePrice"]]
y
```

Out[15]:

	SalePrice
0	208500.0
1	181500.0
2	223500.0
3	140000.0
4	250000.0
5	143000.0
6	307000.0
7	200000.0
8	129900.0
9	118000.0
10	129500.0
11	345000.0
12	144000.0
13	279500.0
14	157000.0
15	132000.0
16	149000.0
17	90000.0
18	159000.0
19	139000.0
20	325300.0
21	139400.0
22	230000.0
23	129900.0
24	154000.0
25	256300.0
26	134800.0
27	306000.0
28	207500.0
29	68500.0
...	...
1349	140000.0
1350	119000.0
1351	182900.0
1352	192140.0
1353	143750.0
1354	64500.0
1355	186500.0
1356	160000.0
1357	174000.0
1358	120500.0
1359	394617.0
1360	149700.0
1361	197000.0
1362	191000.0
1363	149300.0
1364	310000.0
1365	121000.0
1366	179600.0
1367	129000.0
1368	157900.0
1369	240000.0
1370	112000.0
1371	287090.0

	SalePrice
1372	145000.0
1373	185000.0
1374	175000.0
1375	210000.0
1376	266500.0
1377	142125.0
1378	147500.0

1379 rows × 1 columns

In [16]:

```
X=df2.drop("SalePrice",axis=1)
X
```

Out[16]:

	1stFlrSF	2ndFlrSF	3SsnPorch	BedroomAbvGr	BsmtFinSF1	BsmtFinSF2	BsmtFullBath	BsmtHalfBath	BsmtUnfSF	EnclosedPorch	...
0	856.0	854.0	0.0	3	706.0	0.0	1	0	150.0	0.0	...
1	1262.0	0.0	0.0	3	978.0	0.0	0	1	284.0	0.0	...
2	920.0	866.0	0.0	3	486.0	0.0	1	0	434.0	0.0	...
3	961.0	756.0	0.0	3	216.0	0.0	1	0	540.0	272.0	...
4	1145.0	1053.0	0.0	4	655.0	0.0	1	0	490.0	0.0	...
5	796.0	566.0	320.0	1	732.0	0.0	1	0	64.0	0.0	...
6	1694.0	0.0	0.0	3	1369.0	0.0	1	0	317.0	0.0	...
7	1107.0	983.0	0.0	3	859.0	32.0	1	0	216.0	228.0	...
8	1022.0	752.0	0.0	2	0.0	0.0	0	0	952.0	205.0	...
9	1077.0	0.0	0.0	2	851.0	0.0	1	0	140.0	0.0	...
10	1040.0	0.0	0.0	3	906.0	0.0	1	0	134.0	0.0	...
11	1182.0	1142.0	0.0	4	998.0	0.0	1	0	177.0	0.0	...
12	912.0	0.0	0.0	2	737.0	0.0	1	0	175.0	0.0	...
13	1494.0	0.0	0.0	3	0.0	0.0	0	0	1494.0	0.0	...
14	1253.0	0.0	0.0	2	733.0	0.0	1	0	520.0	176.0	...
15	854.0	0.0	0.0	2	0.0	0.0	0	0	832.0	0.0	...
16	1004.0	0.0	0.0	2	578.0	0.0	1	0	426.0	0.0	...
17	1296.0	0.0	0.0	2	0.0	0.0	0	0	0.0	0.0	...
18	1114.0	0.0	0.0	3	646.0	0.0	1	0	468.0	0.0	...
19	1339.0	0.0	0.0	3	504.0	0.0	0	0	525.0	0.0	...
20	1158.0	1218.0	0.0	4	0.0	0.0	0	0	1158.0	0.0	...
21	1108.0	0.0	0.0	3	0.0	0.0	0	0	637.0	205.0	...
22	1795.0	0.0	0.0	3	0.0	0.0	0	0	1777.0	0.0	...
23	1060.0	0.0	0.0	3	840.0	0.0	1	0	200.0	0.0	...
24	1060.0	0.0	0.0	3	188.0	668.0	1	0	204.0	0.0	...
25	1600.0	0.0	0.0	3	0.0	0.0	0	0	1566.0	0.0	...
26	900.0	0.0	0.0	3	234.0	486.0	0	1	180.0	0.0	...
27	1704.0	0.0	0.0	3	1218.0	0.0	1	0	486.0	0.0	...
28	1600.0	0.0	0.0	2	1277.0	0.0	1	0	207.0	0.0	...
29	520.0	0.0	0.0	1	0.0	0.0	0	0	520.0	87.0	...
...	...	...	...	...	...	...	...	...	...	...	...
1349	1048.0	510.0	0.0	3	580.0	0.0	1	0	333.0	0.0	...
1350	804.0	0.0	0.0	2	510.0	0.0	1	0	278.0	154.0	...
1351	1440.0	0.0	0.0	3	678.0	0.0	0	0	762.0	99.0	...
1352	734.0	1104.0	0.0	4	0.0	0.0	0	0	732.0	0.0	...
1353	958.0	0.0	0.0	2	958.0	0.0	0	0	0.0	0.0	...
1354	968.0	0.0	0.0	4	0.0	0.0	0	0	656.0	0.0	...
1355	962.0	830.0	0.0	3	0.0	0.0	1	0	936.0	0.0	...
1356	1126.0	0.0	0.0	3	936.0	0.0	1	0	190.0	0.0	...
1357	1537.0	0.0	0.0	3	0.0	0.0	1	0	1319.0	0.0	...
1358	864.0	0.0	0.0	3	616.0	0.0	0	0	248.0	0.0	...
1359	1932.0	0.0	304.0	2	1336.0	0.0	1	0	596.0	0.0	...
1360	1236.0	0.0	0.0	2	600.0	0.0	1	0	312.0	158.0	...
1361	1040.0	685.0	0.0	3	315.0	110.0	0	0	114.0	216.0	...
1362	1423.0	748.0	0.0	3	0.0	0.0	0	0	588.0	0.0	...
1363	848.0	0.0	0.0	1	697.0	0.0	1	0	151.0	0.0	...
1364	1026.0	981.0	0.0	3	765.0	0.0	1	0	252.0	0.0	...
1365	952.0	0.0	0.0	2	0.0	0.0	0	0	952.0	0.0	...
1366	1422.0	0.0	0.0	3	0.0	0.0	0	0	1422.0	0.0	...
1367	913.0	0.0	0.0	3	187.0	627.0	1	0	0.0	252.0	...
1368	1188.0	0.0	0.0	3	593.0	0.0	0	0	595.0	0.0	...
1369	1220.0	870.0	0.0	3	1079.0	0.0	1	0	141.0	0.0	...
1370	796.0	550.0	0.0	2	0.0	0.0	0	0	560.0	0.0	...
1371	1578.0	0.0	0.0	3	0.0	0.0	0	0	1573.0	0.0	...

	1stFlrSF	2ndFlrSF	3SsnPorch	BedroomAbvGr	BsmtFinSF1	BsmtFinSF2	BsmtFullBath	BsmtHalfBath	BsmtUnfSF	EnclosedPorch	...
1372	1072.0	0.0	0.0	2	547.0	0.0	1	0	0.0	0.0	...
1373	1221.0	0.0	0.0	2	410.0	0.0	1	0	811.0	0.0	...
1374	953.0	694.0	0.0	3	0.0	0.0	0	0	953.0	0.0	...
1375	2073.0	0.0	0.0	3	790.0	163.0	1	0	589.0	0.0	...
1376	1188.0	1152.0	0.0	4	275.0	0.0	0	0	877.0	0.0	...
1377	1078.0	0.0	0.0	2	49.0	1029.0	1	0	0.0	112.0	...
1378	1256.0	0.0	0.0	3	830.0	290.0	1	0	136.0	0.0	...

1379 rows × 36 columns

In [17]:

```
from sklearn.model_selection import train_test_split
```

In [18]:

```
X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.25,random_state=42)
```

In [19]:

```
X_train
```

Out[19]:

	1stFlrSF	2ndFlrSF	3SsnPorch	BedroomAbvGr	BsmtFinSF1	BsmtFinSF2	BsmtFullBath	BsmtHalfBath	BsmtUnfSF	EnclosedPorch	...
1193	1337.0	0.0	0.0	2	266.0	0.0	1	0	1139.0	0.0	...
910	1800.0	0.0	0.0	2	0.0	0.0	0	0	1800.0	0.0	...
1068	1328.0	653.0	0.0	4	622.0	0.0	1	0	500.0	0.0	...
1196	2018.0	0.0	0.0	3	0.0	0.0	0	0	2002.0	0.0	...
1102	959.0	712.0	0.0	3	786.0	0.0	1	0	173.0	0.0	...
447	970.0	0.0	0.0	2	630.0	0.0	1	0	340.0	0.0	...
796	1701.0	0.0	0.0	3	1390.0	0.0	1	0	0.0	0.0	...
543	1320.0	0.0	0.0	3	328.0	551.0	1	0	285.0	240.0	...
901	768.0	0.0	0.0	2	660.0	0.0	0	1	108.0	0.0	...
968	1264.0	0.0	0.0	3	697.0	0.0	1	0	571.0	0.0	...
411	904.0	0.0	0.0	2	0.0	0.0	0	0	884.0	105.0	...
96	1535.0	0.0	0.0	4	0.0	0.0	0	0	0.0	0.0	...
429	624.0	720.0	0.0	4	0.0	0.0	0	0	624.0	96.0	...
361	784.0	0.0	0.0	2	0.0	0.0	0	0	784.0	91.0	...
933	778.0	798.0	0.0	3	0.0	0.0	0	0	770.0	0.0	...
588	1422.0	0.0	0.0	3	0.0	0.0	0	0	978.0	36.0	...
156	854.0	0.0	0.0	2	360.0	0.0	0	0	360.0	0.0	...
528	1389.0	0.0	0.0	2	1071.0	123.0	1	0	195.0	0.0	...
654	616.0	0.0	0.0	2	616.0	0.0	0	0	0.0	129.0	...
857	1636.0	0.0	0.0	3	63.0	0.0	1	0	1560.0	0.0	...
1237	1294.0	0.0	0.0	3	1200.0	0.0	1	0	78.0	0.0	...
534	1496.0	636.0	0.0	1	1441.0	0.0	1	0	55.0	0.0	...
1078	1466.0	1362.0	0.0	4	1150.0	0.0	1	0	316.0	0.0	...
1190	1050.0	0.0	0.0	2	504.0	0.0	0	0	546.0	0.0	...
1312	869.0	349.0	0.0	3	375.0	0.0	0	1	360.0	0.0	...
371	1144.0	0.0	0.0	3	739.0	0.0	1	0	405.0	0.0	...
677	848.0	0.0	0.0	1	662.0	0.0	1	0	186.0	0.0	...
308	596.0	596.0	0.0	3	0.0	0.0	0	0	596.0	137.0	...
1371	1578.0	0.0	0.0	3	0.0	0.0	0	0	1573.0	0.0	...
710	866.0	902.0	0.0	3	20.0	0.0	0	0	846.0	0.0	...
...	...	...	...	...	...	...	...	...	...	...	...
474	1801.0	0.0	0.0	1	1247.0	0.0	1	0	254.0	0.0	...
856	1063.0	0.0	0.0	3	354.0	290.0	1	0	412.0	164.0	...
747	1086.0	809.0	0.0	3	0.0	0.0	0	0	712.0	0.0	...
252	1095.0	844.0	0.0	3	0.0	0.0	0	0	1095.0	0.0	...
21	1108.0	0.0	0.0	3	0.0	0.0	0	0	637.0	205.0	...
1337	1569.0	0.0	0.0	1	988.0	0.0	0	1	398.0	0.0	...
459	1484.0	0.0	0.0	3	998.0	0.0	0	0	486.0	0.0	...
1184	760.0	896.0	0.0	3	0.0	0.0	0	0	746.0	0.0	...
276	910.0	648.0	0.0	4	420.0	0.0	0	0	490.0	0.0	...
955	1022.0	0.0	0.0	2	247.0	465.0	1	0	310.0	226.0	...
1215	1582.0	0.0	0.0	4	812.0	0.0	0	1	812.0	0.0	...
385	1050.0	1028.0	0.0	3	789.0	0.0	1	0	245.0	0.0	...
805	1779.0	0.0	0.0	3	306.0	1085.0	1	0	372.0	0.0	...
343	790.0	784.0	0.0	3	712.0	0.0	1	0	84.0	0.0	...
769	1008.0	0.0	0.0	2	486.0	0.0	0	0	522.0	120.0	...
1332	944.0	896.0	0.0	3	666.0	0.0	1	0	278.0	0.0	...
130	928.0	836.0	0.0	3	821.0	0.0	1	0	107.0	0.0	...
871	936.0	785.0	0.0	3	814.0	0.0	0	1	114.0	0.0	...
1123	1622.0	0.0	0.0	3	1159.0	0.0	1	0	90.0	0.0	...
87	964.0	0.0	0.0	2	713.0	0.0	1	0	163.0	44.0	...
330	1453.0	0.0	0.0	2	1082.0	0.0	1	0	371.0	0.0	...
1238	1902.0	0.0	0.0	3	1406.0	0.0	1	0	496.0	162.0	...
466	886.0	0.0	0.0	2	0.0	0.0	0	0	190.0	80.0	...

	1stFlrSF	2ndFlrSF	3SsnPorch	BedroomAbvGr	BsmtFinSF1	BsmtFinSF2	BsmtFullBath	BsmtHalfBath	BsmtUnfSF	EnclosedPorch	...	C
121	1216.0	941.0	0.0	4	445.0	0.0	0	0	479.0	0.0	...	
1044	1500.0	1122.0	0.0	3	1032.0	0.0	1	0	431.0	0.0	...	
1095	855.0	601.0	0.0	3	311.0	0.0	0	0	544.0	0.0	...	
1130	815.0	875.0	0.0	3	0.0	0.0	0	0	815.0	330.0	...	
1294	1661.0	0.0	0.0	3	831.0	0.0	1	0	161.0	0.0	...	
860	742.0	742.0	0.0	3	0.0	0.0	0	0	742.0	0.0	...	
1126	1224.0	0.0	0.0	2	883.0	0.0	1	0	341.0	0.0	...	

1034 rows × 36 columns

In [20]:

```
X_test
```

Out[20]:

	1stFlrSF	2ndFlrSF	3SsnPorch	BedroomAbvGr	BsmtFinSF1	BsmtFinSF2	BsmtFullBath	BsmtHalfBath	BsmtUnfSF	EnclosedPorch	...
599	1518.0	0.0	0.0	1	1218.0	0.0	0	0	300.0	0.0	...
881	925.0	0.0	0.0	2	338.0	466.0	0	1	121.0	0.0	...
634	1095.0	679.0	0.0	4	0.0	0.0	1	0	1095.0	90.0	...
425	888.0	868.0	0.0	3	742.0	0.0	1	0	130.0	0.0	...
906	1337.0	0.0	0.0	3	699.0	0.0	1	0	638.0	0.0	...
1079	672.0	252.0	0.0	2	348.0	0.0	1	0	324.0	0.0	...
65	1479.0	0.0	0.0	3	1013.0	0.0	1	0	440.0	0.0	...
1351	1440.0	0.0	0.0	3	678.0	0.0	0	0	762.0	99.0	...
479	689.0	689.0	0.0	3	141.0	0.0	0	0	548.0	116.0	...
67	1304.0	983.0	0.0	3	603.0	0.0	0	0	701.0	114.0	...
939	774.0	456.0	0.0	3	384.0	0.0	1	0	363.0	0.0	...
573	1940.0	1254.0	0.0	4	428.0	0.0	0	0	537.0	0.0	...
917	918.0	0.0	0.0	2	0.0	0.0	0	0	918.0	0.0	...
1054	1734.0	0.0	0.0	3	1004.0	0.0	1	0	730.0	0.0	...
941	1442.0	0.0	0.0	2	0.0	0.0	0	0	1442.0	0.0	...
1116	1130.0	0.0	0.0	2	821.0	0.0	1	0	299.0	0.0	...
237	1005.0	1286.0	0.0	4	0.0	0.0	0	0	975.0	0.0	...
578	1054.0	0.0	0.0	3	763.0	0.0	1	0	173.0	0.0	...
772	1358.0	0.0	0.0	2	733.0	0.0	1	0	625.0	0.0	...
303	1898.0	1080.0	0.0	5	0.0	0.0	0	0	710.0	0.0	...
953	720.0	551.0	0.0	4	0.0	0.0	0	0	720.0	108.0	...
783	520.0	600.0	0.0	2	0.0	0.0	0	0	600.0	0.0	...
339	912.0	0.0	0.0	2	773.0	0.0	1	0	115.0	0.0	...
718	1494.0	0.0	0.0	2	437.0	1057.0	1	0	0.0	0.0	...
208	2392.0	0.0	0.0	3	56.0	0.0	0	0	2336.0	0.0	...
624	1465.0	915.0	0.0	3	187.0	723.0	0	0	111.0	0.0	...
1075	1167.0	0.0	0.0	3	645.0	0.0	0	0	270.0	216.0	...
1040	950.0	0.0	0.0	3	412.0	287.0	0	0	251.0	0.0	...
575	1476.0	677.0	0.0	3	904.0	0.0	1	0	536.0	0.0	...
244	1212.0	0.0	0.0	3	506.0	0.0	1	0	0.0	0.0	...
...	...	...	...	...	...	...	...	...	...	...	...
1334	1040.0	0.0	0.0	2	0.0	0.0	0	0	0.0	0.0	...
199	1236.0	0.0	0.0	2	360.0	0.0	0	1	710.0	0.0	...
367	961.0	406.0	0.0	4	241.0	391.0	1	0	229.0	112.0	...
723	1690.0	1589.0	0.0	4	1416.0	0.0	1	0	234.0	0.0	...
354	914.0	0.0	0.0	2	298.0	0.0	0	0	572.0	0.0	...
10	1040.0	0.0	0.0	3	906.0	0.0	1	0	134.0	0.0	...
147	1392.0	1070.0	168.0	4	57.0	0.0	1	0	1335.0	0.0	...
538	846.0	846.0	0.0	3	0.0	0.0	0	0	846.0	0.0	...
282	1541.0	0.0	0.0	3	0.0	0.0	0	0	1541.0	0.0	...
298	1472.0	0.0	0.0	3	1036.0	0.0	1	0	336.0	0.0	...
522	1048.0	0.0	0.0	2	0.0	0.0	0	0	993.0	116.0	...
291	793.0	325.0	0.0	3	507.0	0.0	1	0	286.0	0.0	...
503	880.0	844.0	0.0	3	0.0	0.0	0	0	880.0	0.0	...
903	979.0	979.0	0.0	4	484.0	0.0	0	0	495.0	0.0	...
930	1001.0	634.0	0.0	2	0.0	0.0	0	0	485.0	0.0	...
439	888.0	756.0	0.0	3	386.0	0.0	0	0	342.0	0.0	...
1033	1200.0	0.0	0.0	1	661.0	0.0	1	0	203.0	0.0	...
331	616.0	495.0	0.0	3	236.0	380.0	0	1	0.0	0.0	...
527	1392.0	0.0	0.0	3	1302.0	0.0	1	0	90.0	0.0	...
462	616.0	688.0	0.0	3	0.0	0.0	0	0	264.0	0.0	...
861	1105.0	1169.0	0.0	5	443.0	0.0	0	0	662.0	0.0	...
630	1208.0	0.0	0.0	3	767.0	0.0	1	0	441.0	0.0	...
135	970.0	739.0	0.0	3	0.0	0.0	0	0	970.0	0.0	...

	1stFlrSF	2ndFlrSF	3SsnPorch	BedroomAbvGr	BsmtFinSF1	BsmtFinSF2	BsmtFullBath	BsmtHalfBath	BsmtUnfSF	EnclosedPorch	...
358	1026.0	665.0	0.0	3	218.0	0.0	0	0	808.0	242.0	...
363	1269.0	0.0	0.0	2	24.0	0.0	0	0	1232.0	0.0	...
618	1142.0	793.0	0.0	3	0.0	0.0	0	0	793.0	252.0	...
561	684.0	684.0	0.0	3	0.0	0.0	0	0	684.0	0.0	...
529	1163.0	511.0	0.0	4	0.0	0.0	0	0	1163.0	0.0	...
567	927.0	988.0	0.0	3	789.0	0.0	1	0	119.0	0.0	...
158	1064.0	703.0	0.0	2	495.0	215.0	1	0	354.0	0.0	...

345 rows × 36 columns

In [21]:

```
from sklearn.linear_model import LinearRegression
lr=LinearRegression()
lr.fit(X_train,y_train)
```

Out[21]:

```
LinearRegression(copy_X=True, fit_intercept=True, n_jobs=1, normalize=False)
```

In [22]:

```
y_pred=lr.predict(X_test)
y_pred
```

Out[22]:

```
array([[257434.93050745],
       [111083.7347476 ],
       [100018.05832303],
       [204028.53821314],
       [207319.25418312],
       [ 38036.30928932],
       [234153.38582868],
       [205076.12689608],
       [187014.1265524 ],
       [235636.78799031],
       [100976.50943769],
       [304119.53646983],
       [101769.64600713],
       [288758.46001593],
       [204767.89338328],
       [145002.47279616],
       [248322.97436852],
       [151533.68038634],
```

In [23]:

```
from sklearn.metrics import mean_squared_error
```

In [24]:

```
mse_In=mean_squared_error(y_test,y_pred)
mse_In
```

Out[24]:

```
1474827325.5975406
```

In [25]:

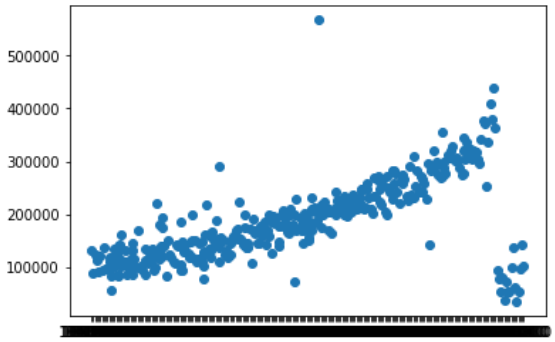
```
import matplotlib.pyplot as plt
```

In [26]:

```
plt.scatter(y_test,y_pred)
```

Out[26]:

```
<matplotlib.collections.PathCollection at 0x28633c249b0>
```



In [27]:

```
gd=pd.get_dummies(df)
gd
```

Out[27]:

	1stFlrSF	2ndFlrSF	3SsnPorch	BedroomAbvGr	BsmtFinSF1	BsmtFinSF2	BsmtFullBath	BsmtHalfBath	BsmtUnfSF	EnclosedPorch	...
0	856.0	854.0	0.0	3	706.0	0.0	1	0	150.0	0.0	...
1	1262.0	0.0	0.0	3	978.0	0.0	0	1	284.0	0.0	...
2	920.0	866.0	0.0	3	486.0	0.0	1	0	434.0	0.0	...
3	961.0	756.0	0.0	3	216.0	0.0	1	0	540.0	272.0	...
4	1145.0	1053.0	0.0	4	655.0	0.0	1	0	490.0	0.0	...
5	796.0	566.0	320.0	1	732.0	0.0	1	0	64.0	0.0	...
6	1694.0	0.0	0.0	3	1369.0	0.0	1	0	317.0	0.0	...
7	1107.0	983.0	0.0	3	859.0	32.0	1	0	216.0	228.0	...
8	1022.0	752.0	0.0	2	0.0	0.0	0	0	952.0	205.0	...
9	1077.0	0.0	0.0	2	851.0	0.0	1	0	140.0	0.0	...
10	1040.0	0.0	0.0	3	906.0	0.0	1	0	134.0	0.0	...
11	1182.0	1142.0	0.0	4	998.0	0.0	1	0	177.0	0.0	...
12	912.0	0.0	0.0	2	737.0	0.0	1	0	175.0	0.0	...
13	1494.0	0.0	0.0	3	0.0	0.0	0	0	1494.0	0.0	...
14	1253.0	0.0	0.0	2	733.0	0.0	1	0	520.0	176.0	...
15	854.0	0.0	0.0	2	0.0	0.0	0	0	832.0	0.0	...
16	1004.0	0.0	0.0	2	578.0	0.0	1	0	426.0	0.0	...
17	1296.0	0.0	0.0	2	0.0	0.0	0	0	0.0	0.0	...
18	1114.0	0.0	0.0	3	646.0	0.0	1	0	468.0	0.0	...
19	1339.0	0.0	0.0	3	504.0	0.0	0	0	525.0	0.0	...
20	1158.0	1218.0	0.0	4	0.0	0.0	0	0	1158.0	0.0	...
21	1108.0	0.0	0.0	3	0.0	0.0	0	0	637.0	205.0	...
22	1795.0	0.0	0.0	3	0.0	0.0	0	0	1777.0	0.0	...
23	1060.0	0.0	0.0	3	840.0	0.0	1	0	200.0	0.0	...
24	1060.0	0.0	0.0	3	188.0	668.0	1	0	204.0	0.0	...
25	1600.0	0.0	0.0	3	0.0	0.0	0	0	1566.0	0.0	...
26	900.0	0.0	0.0	3	234.0	486.0	0	1	180.0	0.0	...
27	1704.0	0.0	0.0	3	1218.0	0.0	1	0	486.0	0.0	...
28	1600.0	0.0	0.0	2	1277.0	0.0	1	0	207.0	0.0	...
29	520.0	0.0	0.0	1	0.0	0.0	0	0	520.0	87.0	...
...	...	...	...	...	...	...	...	...	...	...	...
1349	1048.0	510.0	0.0	3	580.0	0.0	1	0	333.0	0.0	...
1350	804.0	0.0	0.0	2	510.0	0.0	1	0	278.0	154.0	...
1351	1440.0	0.0	0.0	3	678.0	0.0	0	0	762.0	99.0	...
1352	734.0	1104.0	0.0	4	0.0	0.0	0	0	732.0	0.0	...
1353	958.0	0.0	0.0	2	958.0	0.0	0	0	0.0	0.0	...
1354	968.0	0.0	0.0	4	0.0	0.0	0	0	656.0	0.0	...
1355	962.0	830.0	0.0	3	0.0	0.0	1	0	936.0	0.0	...
1356	1126.0	0.0	0.0	3	936.0	0.0	1	0	190.0	0.0	...
1357	1537.0	0.0	0.0	3	0.0	0.0	1	0	1319.0	0.0	...
1358	864.0	0.0	0.0	3	616.0	0.0	0	0	248.0	0.0	...
1359	1932.0	0.0	304.0	2	1336.0	0.0	1	0	596.0	0.0	...
1360	1236.0	0.0	0.0	2	600.0	0.0	1	0	312.0	158.0	...
1361	1040.0	685.0	0.0	3	315.0	110.0	0	0	114.0	216.0	...
1362	1423.0	748.0	0.0	3	0.0	0.0	0	0	588.0	0.0	...
1363	848.0	0.0	0.0	1	697.0	0.0	1	0	151.0	0.0	...
1364	1026.0	981.0	0.0	3	765.0	0.0	1	0	252.0	0.0	...
1365	952.0	0.0	0.0	2	0.0	0.0	0	0	952.0	0.0	...
1366	1422.0	0.0	0.0	3	0.0	0.0	0	0	1422.0	0.0	...
1367	913.0	0.0	0.0	3	187.0	627.0	1	0	0.0	252.0	...
1368	1188.0	0.0	0.0	3	593.0	0.0	0	0	595.0	0.0	...
1369	1220.0	870.0	0.0	3	1079.0	0.0	1	0	141.0	0.0	...
1370	796.0	550.0	0.0	2	0.0	0.0	0	0	560.0	0.0	...
1371	1578.0	0.0	0.0	3	0.0	0.0	0	0	1573.0	0.0	...

	1stFlrSF	2ndFlrSF	3SsnPorch	BedroomAbvGr	BsmtFinSF1	BsmtFinSF2	BsmtFullBath	BsmtHalfBath	BsmtUnfSF	EnclosedPorch	...	C
1372	1072.0	0.0	0.0	2	547.0	0.0	1	0	0.0	0.0	...	
1373	1221.0	0.0	0.0	2	410.0	0.0	1	0	811.0	0.0	...	
1374	953.0	694.0	0.0	3	0.0	0.0	0	0	953.0	0.0	...	
1375	2073.0	0.0	0.0	3	790.0	163.0	1	0	589.0	0.0	...	
1376	1188.0	1152.0	0.0	4	275.0	0.0	0	0	877.0	0.0	...	
1377	1078.0	0.0	0.0	2	49.0	1029.0	1	0	0.0	112.0	...	
1378	1256.0	0.0	0.0	3	830.0	290.0	1	0	136.0	0.0	...	

1379 rows × 37 columns

In [28]:

```
x=gd.drop("SalePrice",axis=1)
x
```

Out[28]:

	1stFlrSF	2ndFlrSF	3SsnPorch	BedroomAbvGr	BsmtFinSF1	BsmtFinSF2	BsmtFullBath	BsmtHalfBath	BsmtUnfSF	EnclosedPorch	...
0	856.0	854.0	0.0	3	706.0	0.0	1	0	150.0	0.0	...
1	1262.0	0.0	0.0	3	978.0	0.0	0	1	284.0	0.0	...
2	920.0	866.0	0.0	3	486.0	0.0	1	0	434.0	0.0	...
3	961.0	756.0	0.0	3	216.0	0.0	1	0	540.0	272.0	...
4	1145.0	1053.0	0.0	4	655.0	0.0	1	0	490.0	0.0	...
5	796.0	566.0	320.0	1	732.0	0.0	1	0	64.0	0.0	...
6	1694.0	0.0	0.0	3	1369.0	0.0	1	0	317.0	0.0	...
7	1107.0	983.0	0.0	3	859.0	32.0	1	0	216.0	228.0	...
8	1022.0	752.0	0.0	2	0.0	0.0	0	0	952.0	205.0	...
9	1077.0	0.0	0.0	2	851.0	0.0	1	0	140.0	0.0	...
10	1040.0	0.0	0.0	3	906.0	0.0	1	0	134.0	0.0	...
11	1182.0	1142.0	0.0	4	998.0	0.0	1	0	177.0	0.0	...
12	912.0	0.0	0.0	2	737.0	0.0	1	0	175.0	0.0	...
13	1494.0	0.0	0.0	3	0.0	0.0	0	0	1494.0	0.0	...
14	1253.0	0.0	0.0	2	733.0	0.0	1	0	520.0	176.0	...
15	854.0	0.0	0.0	2	0.0	0.0	0	0	832.0	0.0	...
16	1004.0	0.0	0.0	2	578.0	0.0	1	0	426.0	0.0	...
17	1296.0	0.0	0.0	2	0.0	0.0	0	0	0.0	0.0	...
18	1114.0	0.0	0.0	3	646.0	0.0	1	0	468.0	0.0	...
19	1339.0	0.0	0.0	3	504.0	0.0	0	0	525.0	0.0	...
20	1158.0	1218.0	0.0	4	0.0	0.0	0	0	1158.0	0.0	...
21	1108.0	0.0	0.0	3	0.0	0.0	0	0	637.0	205.0	...
22	1795.0	0.0	0.0	3	0.0	0.0	0	0	1777.0	0.0	...
23	1060.0	0.0	0.0	3	840.0	0.0	1	0	200.0	0.0	...
24	1060.0	0.0	0.0	3	188.0	668.0	1	0	204.0	0.0	...
25	1600.0	0.0	0.0	3	0.0	0.0	0	0	1566.0	0.0	...
26	900.0	0.0	0.0	3	234.0	486.0	0	1	180.0	0.0	...
27	1704.0	0.0	0.0	3	1218.0	0.0	1	0	486.0	0.0	...
28	1600.0	0.0	0.0	2	1277.0	0.0	1	0	207.0	0.0	...
29	520.0	0.0	0.0	1	0.0	0.0	0	0	520.0	87.0	...
...	...	...	...	...	...	...	...	...	...	...	...
1349	1048.0	510.0	0.0	3	580.0	0.0	1	0	333.0	0.0	...
1350	804.0	0.0	0.0	2	510.0	0.0	1	0	278.0	154.0	...
1351	1440.0	0.0	0.0	3	678.0	0.0	0	0	762.0	99.0	...
1352	734.0	1104.0	0.0	4	0.0	0.0	0	0	732.0	0.0	...
1353	958.0	0.0	0.0	2	958.0	0.0	0	0	0.0	0.0	...
1354	968.0	0.0	0.0	4	0.0	0.0	0	0	656.0	0.0	...
1355	962.0	830.0	0.0	3	0.0	0.0	1	0	936.0	0.0	...
1356	1126.0	0.0	0.0	3	936.0	0.0	1	0	190.0	0.0	...
1357	1537.0	0.0	0.0	3	0.0	0.0	1	0	1319.0	0.0	...
1358	864.0	0.0	0.0	3	616.0	0.0	0	0	248.0	0.0	...
1359	1932.0	0.0	304.0	2	1336.0	0.0	1	0	596.0	0.0	...
1360	1236.0	0.0	0.0	2	600.0	0.0	1	0	312.0	158.0	...
1361	1040.0	685.0	0.0	3	315.0	110.0	0	0	114.0	216.0	...
1362	1423.0	748.0	0.0	3	0.0	0.0	0	0	588.0	0.0	...
1363	848.0	0.0	0.0	1	697.0	0.0	1	0	151.0	0.0	...
1364	1026.0	981.0	0.0	3	765.0	0.0	1	0	252.0	0.0	...
1365	952.0	0.0	0.0	2	0.0	0.0	0	0	952.0	0.0	...
1366	1422.0	0.0	0.0	3	0.0	0.0	0	0	1422.0	0.0	...
1367	913.0	0.0	0.0	3	187.0	627.0	1	0	0.0	252.0	...
1368	1188.0	0.0	0.0	3	593.0	0.0	0	0	595.0	0.0	...
1369	1220.0	870.0	0.0	3	1079.0	0.0	1	0	141.0	0.0	...
1370	796.0	550.0	0.0	2	0.0	0.0	0	0	560.0	0.0	...
1371	1578.0	0.0	0.0	3	0.0	0.0	0	0	1573.0	0.0	...

	1stFlrSF	2ndFlrSF	3SsnPorch	BedroomAbvGr	BsmtFinSF1	BsmtFinSF2	BsmtFullBath	BsmtHalfBath	BsmtUnfSF	EnclosedPorch	...
1372	1072.0	0.0	0.0	2	547.0	0.0	1	0	0.0	0.0	...
1373	1221.0	0.0	0.0	2	410.0	0.0	1	0	811.0	0.0	...
1374	953.0	694.0	0.0	3	0.0	0.0	0	0	953.0	0.0	...
1375	2073.0	0.0	0.0	3	790.0	163.0	1	0	589.0	0.0	...
1376	1188.0	1152.0	0.0	4	275.0	0.0	0	0	877.0	0.0	...
1377	1078.0	0.0	0.0	2	49.0	1029.0	1	0	0.0	112.0	...
1378	1256.0	0.0	0.0	3	830.0	290.0	1	0	136.0	0.0	...

1379 rows × 36 columns

In [29]:

```
Y=gd.pop('SalePrice')
```

In [30]:

```
Y
```

Out[30]:

```
0       208500.0
1       181500.0
2       223500.0
3       140000.0
4       250000.0
5       143000.0
6       307000.0
7       200000.0
8       129900.0
9       118000.0
10      129500.0
11      345000.0
12      144000.0
13      279500.0
14      157000.0
15      132000.0
16      149000.0
17       90000.0
18      159000.0
19      139000.0
20      325300.0
21      139400.0
22      230000.0
23      129900.0
24      154000.0
25      256300.0
26      134800.0
27      306000.0
28      207500.0
29       68500.0
          ...
1349    140000.0
1350    119000.0
1351    182900.0
1352    192140.0
1353    143750.0
1354     64500.0
1355    186500.0
1356    160000.0
1357    174000.0
1358    120500.0
1359    394617.0
1360    149700.0
1361    197000.0
1362    191000.0
1363    149300.0
1364    310000.0
1365    121000.0
1366    179600.0
1367    129000.0
1368    157900.0
1369    240000.0
1370    112000.0
1371    287090.0
1372    145000.0
1373    185000.0
1374    175000.0
1375    210000.0
1376    266500.0
1377    142125.0
1378    147500.0
Name: SalePrice, Length: 1379, dtype: float64
```

In [31]:

```
from sklearn.model_selection import train_test_split
```

In [32]:

```
X_train,X_test,y_train,y_test=train_test_split(x,Y,test_size=0.25,random_state=42)
```

In [33]:

```
X_train
```

Out[33]:

	1stFlrSF	2ndFlrSF	3SsnPorch	BedroomAbvGr	BsmtFinSF1	BsmtFinSF2	BsmtFullBath	BsmtHalfBath	BsmtUnfSF	EnclosedPorch	...	C
1193	1337.0	0.0	0.0	2	266.0	0.0	1	0	1139.0	0.0	...	
910	1800.0	0.0	0.0	2	0.0	0.0	0	0	1800.0	0.0	...	
1068	1328.0	653.0	0.0	4	622.0	0.0	1	0	500.0	0.0	...	
1196	2018.0	0.0	0.0	3	0.0	0.0	0	0	2002.0	0.0	...	
1102	959.0	712.0	0.0	3	786.0	0.0	1	0	173.0	0.0	...	
447	970.0	0.0	0.0	2	630.0	0.0	1	0	340.0	0.0	...	
796	1701.0	0.0	0.0	3	1390.0	0.0	1	0	0.0	0.0	...	
543	1320.0	0.0	0.0	3	328.0	551.0	1	0	285.0	240.0	...	
901	768.0	0.0	0.0	2	660.0	0.0	0	1	108.0	0.0	...	
968	1264.0	0.0	0.0	3	697.0	0.0	1	0	571.0	0.0	...	
411	904.0	0.0	0.0	2	0.0	0.0	0	0	884.0	105.0	...	
96	1535.0	0.0	0.0	4	0.0	0.0	0	0	0.0	0.0	...	
429	624.0	720.0	0.0	4	0.0	0.0	0	0	624.0	96.0	...	
361	784.0	0.0	0.0	2	0.0	0.0	0	0	784.0	91.0	...	
933	778.0	798.0	0.0	3	0.0	0.0	0	0	770.0	0.0	...	
588	1422.0	0.0	0.0	3	0.0	0.0	0	0	978.0	36.0	...	
156	854.0	0.0	0.0	2	360.0	0.0	0	0	360.0	0.0	...	
528	1389.0	0.0	0.0	2	1071.0	123.0	1	0	195.0	0.0	...	
654	616.0	0.0	0.0	2	616.0	0.0	0	0	0.0	129.0	...	
857	1636.0	0.0	0.0	3	63.0	0.0	1	0	1560.0	0.0	...	
1237	1294.0	0.0	0.0	3	1200.0	0.0	1	0	78.0	0.0	...	
534	1496.0	636.0	0.0	1	1441.0	0.0	1	0	55.0	0.0	...	
1078	1466.0	1362.0	0.0	4	1150.0	0.0	1	0	316.0	0.0	...	
1190	1050.0	0.0	0.0	2	504.0	0.0	0	0	546.0	0.0	...	
1312	869.0	349.0	0.0	3	375.0	0.0	0	1	360.0	0.0	...	
371	1144.0	0.0	0.0	3	739.0	0.0	1	0	405.0	0.0	...	
677	848.0	0.0	0.0	1	662.0	0.0	1	0	186.0	0.0	...	
308	596.0	596.0	0.0	3	0.0	0.0	0	0	596.0	137.0	...	
1371	1578.0	0.0	0.0	3	0.0	0.0	0	0	1573.0	0.0	...	
710	866.0	902.0	0.0	3	20.0	0.0	0	0	846.0	0.0	...	
...	...	...	...	...	...	...	...	...	...	...	...	
474	1801.0	0.0	0.0	1	1247.0	0.0	1	0	254.0	0.0	...	
856	1063.0	0.0	0.0	3	354.0	290.0	1	0	412.0	164.0	...	
747	1086.0	809.0	0.0	3	0.0	0.0	0	0	712.0	0.0	...	
252	1095.0	844.0	0.0	3	0.0	0.0	0	0	1095.0	0.0	...	
21	1108.0	0.0	0.0	3	0.0	0.0	0	0	637.0	205.0	...	
1337	1569.0	0.0	0.0	1	988.0	0.0	0	1	398.0	0.0	...	
459	1484.0	0.0	0.0	3	998.0	0.0	0	0	486.0	0.0	...	
1184	760.0	896.0	0.0	3	0.0	0.0	0	0	746.0	0.0	...	
276	910.0	648.0	0.0	4	420.0	0.0	0	0	490.0	0.0	...	
955	1022.0	0.0	0.0	2	247.0	465.0	1	0	310.0	226.0	...	
1215	1582.0	0.0	0.0	4	812.0	0.0	0	1	812.0	0.0	...	
385	1050.0	1028.0	0.0	3	789.0	0.0	1	0	245.0	0.0	...	
805	1779.0	0.0	0.0	3	306.0	1085.0	1	0	372.0	0.0	...	
343	790.0	784.0	0.0	3	712.0	0.0	1	0	84.0	0.0	...	
769	1008.0	0.0	0.0	2	486.0	0.0	0	0	522.0	120.0	...	
1332	944.0	896.0	0.0	3	666.0	0.0	1	0	278.0	0.0	...	
130	928.0	836.0	0.0	3	821.0	0.0	1	0	107.0	0.0	...	
871	936.0	785.0	0.0	3	814.0	0.0	0	1	114.0	0.0	...	
1123	1622.0	0.0	0.0	3	1159.0	0.0	1	0	90.0	0.0	...	
87	964.0	0.0	0.0	2	713.0	0.0	1	0	163.0	44.0	...	
330	1453.0	0.0	0.0	2	1082.0	0.0	1	0	371.0	0.0	...	
1238	1902.0	0.0	0.0	3	1406.0	0.0	1	0	496.0	162.0	...	
466	886.0	0.0	0.0	2	0.0	0.0	0	0	190.0	80.0	...	

	1stFlrSF	2ndFlrSF	3SsnPorch	BedroomAbvGr	BsmtFinSF1	BsmtFinSF2	BsmtFullBath	BsmtHalfBath	BsmtUnfSF	EnclosedPorch	...	C
121	1216.0	941.0	0.0	4	445.0	0.0	0	0	479.0	0.0	...	
1044	1500.0	1122.0	0.0	3	1032.0	0.0	1	0	431.0	0.0	...	
1095	855.0	601.0	0.0	3	311.0	0.0	0	0	544.0	0.0	...	
1130	815.0	875.0	0.0	3	0.0	0.0	0	0	815.0	330.0	...	
1294	1661.0	0.0	0.0	3	831.0	0.0	1	0	161.0	0.0	...	
860	742.0	742.0	0.0	3	0.0	0.0	0	0	742.0	0.0	...	
1126	1224.0	0.0	0.0	2	883.0	0.0	1	0	341.0	0.0	...	

1034 rows × 36 columns

In [34]:

```
from sklearn.linear_model import LinearRegression
lrr=LinearRegression()
lrr.fit(X_train,y_train)
```

Out[34]:

```
LinearRegression(copy_X=True, fit_intercept=True, n_jobs=1, normalize=False)
```

In [35]:

```
y_pred=lrr.predict(X_test)
y_pred
```

Out[35]:

```
array([257434.93050745, 111083.7347476 , 100018.05832303, 204028.53821314,
       207319.25418312,  38036.30928932, 234153.38582868, 205076.12689608,
       187014.1265524 , 235636.78799031, 100976.50943769, 304119.53646983,
       101769.64600713, 288758.46001593, 204767.89338328, 145002.47279616,
       248322.97436852, 151533.68038634, 209697.39458712, 278312.23574148,
       [illegible]93329.85601474, 151784.14868375, 163237.87451138, 241495.9121289 ,
       372429.94726945, 221672.07367609, 119658.75685737, 100039.78137073,
       319435.0289208 , 172088.64665031, 258772.64470117, 199597.36129642,
       156349.69283212, 142311.22500101, 204807.33161321, 379608.17785443,
       124176.23377551, 141127.25006141, 273832.43247645, 212547.04205553,
       169474.57176142, 123305.65719311, 200341.24016791, 377957.32307114,
       146941.7171239 , 283693.48760668, 106694.19786718, 221733.02172508,
        87733.47014004, 295005.66546305, 125250.62019285,  53569.11256022,
       133831.0508612 , 170901.4292971 , 213273.17757301, 110818.14723232,
       103893.81849706, 200665.21921033, 137764.84464902, 311840.64900748,
        53035.52267594, 184866.47797858, 154675.79963879, 298477.53600006,
        78162.5407468 , 323728.93535803, 169417.44138514, 112718.07937635,
       200547.87648954, 205739.30910765, 130292.88475647, 271581.37221152,
       [illegible], [illegible],  77972.26780485, 138563.37305272,
        35555.51251487, 163787.98080893, 271929.13690186, 115260.62742896,
       319958.55004404, 218515.16541529, 291730.97232824, 211176.53228423,
        93761.81278837, 263031.10150322, 141886.3966699 , 118491.50416092,
       143580.34191288, 278395.85370477, 265558.84177945, 300839.51324225,
       182673.3443797 , 168604.5492604 , 148322.24627425, 213781.03244613,
       247750.58741   , 221693.90686786, 274474.96290575, 242912.73279161,
       111041.26737678, 100160.66573767, 193531.44933265, 206050.88712632,
       154584.47386041, 217440.52365094, 210904.23539265, 125864.27688695,
       171322.66919858, 224459.70210717, 155548.12334499, 111942.7871775 ,
       316624.83474109, 298116.31992277, 335573.77762719,  86738.88506816,
       146692.58771903, 228471.5403292 , 123374.07354542, 204867.33148103,
       217343.21720916, 120569.43114727, 134354.1238425 , 223860.01643704,
       138360.76267459, 210765.75439351, 145423.97229834, 291155.26974376,
       183089.17904138,  98716.55524167, 323896.46706846, 170447.30657102,
       123643.15346299, 134345.24799853, 309423.49789818, 136836.97808937,
       282083.02092584, 132621.2237255 ,  90821.15043377, 164193.14328588,
       147333.60026797, 182030.92949963, 177621.20328218, 254054.45691355,
       119092.92298357, 140081.84030886, 236509.39317212, 318578.43961646,
       105538.81578042, 192023.58779383, 174531.98294663, 146314.75164225,
        95146.46236358, 250936.32825846, 304056.57810773,  55008.42319119,
       281810.05395732,  93402.68236907, 138283.4460375 , 166415.3192293 ,
       209758.26998556, 205602.16976218, 177026.58819395, 248435.62725108,
       145190.55457313, 155283.94785764, 171593.4339452 , 344925.23412401,
       181194.80288331,  98431.28117762, 131287.46561773, 110849.96802284,
       131283.81966769, 194221.20494195, 156966.39268757, 268340.58471943,
       211804.83660278, 151971.72881374, 115946.37551381, 243618.89320955,
       136292.7606826 , 265335.41730193, 302175.59064213,  85054.45272476,
       146007.43532809, 150439.28821412, 159145.83079286, 232246.23120855,
       [illegible]009, 139178.99313677, 355030.619324  , 127659.29864106,
       235339.23724177, 116707.21916901, 103103.75191051, 159574.23418457,
       149485.12469048, 408898.40096809, 335368.65026123, 290500.91914899,
       281845.43486756, 275341.8171841 , 321705.52591338, 307549.31796899,
       201661.19536934, 142047.81494961, 157876.45367574, 258695.10810332,
       207592.92972056, 146121.60182471, 107032.8224214 , 151245.32235213,
       100804.2036383 , 296980.23153195, 341413.30991677, 255158.86136987,
       147006.3495448 , 199721.99255174, 127291.41093944, 264030.58756771,
       121760.19396639, 134138.79148625, 307661.79229119, 170331.68886912,
       173593.89056669, 567240.20119424, 182277.72265598, 144159.48526101,
       199515.17896325, 107691.20503754, 181383.00130498, 328401.72331   ,
       139225.20214766, 197660.10401731, 116340.37479629,  54949.57465389,
       [illegible].81604289,
       230[illegible]822715, 128709.5470232 , 289158.80585494,
       [illegible].7225403 ,
       22[illegible]659817,  93631.91767228, 178506.48207587,
       [illegible].41948308,
       24[illegible]030912, 114633.38186071,  89093.07024776,
       208503.15479642, 172828.98634139, 270797.22136173, 253693.44924276,
       [illegible].90296985,
       15[illegible]839967,  85972.23828462, 151323.12661355,
       [illegible].69902625,
        70[illegible]49164 , 110000.03909838, 231518.89807261,
       [illegible].65743007,
       20[illegible]5704, 194633.36256922,  99160.35922065,
       230240.81594374, 211484.77477044, 198342.97693781, 109749.28794511,
        95564.23930705, 235396.44527254, 184775.90385427, 188806.96222903,
       265328.01705249, 245862.13668724, 107422.00402518, 164786.63106786,
       268303.4442141 , 116075.01592356, 222914.33666726, 102600.4329218 ,
       226400.49839176, 156093.55758651, 112003.82907499, 218089.37220887,
        81085.92660263,  85015.51341434,  98208.34263398, 304262.27895919,
       143757.79043406, 195367.3512202 , 241835.21911123,  60944.84144938,
       143731.15831778, 110459.58351734, 438412.24654518, 123265.07239636,
       116119.87930811, 277961.71617821, 201687.81739715, 237947.02580917,
       206117.84987333,  99453.60459331, 115124.57035037, 209795.31353811,
       167537.15563245, 162970.26517749, 151279.43057796, 189724.86274087,
        96455.47006118, 168652.82586918,  83051.31596953, 171340.42959854,
       176868.33711878, 175644.68961803, 131414.67048231, 204543.74960377,
```

In [36]:

```
from sklearn.metrics import mean_squared_error
```

In [37]:

```
MSE=mean_squared_error(y_test,y_pred)
MSE
```

Out[37]:

```
1474827325.5975406
```

In [38]:

```
from sklearn.preprocessing import StandardScaler
```

In [39]:

```
scaler=StandardScaler()
```

In [40]:

```
scaled_X_train=scaler.fit_transform(X_train)
scaled_X_train
```

Out[40]:

```
array([[ 0.39851037,  0.79290427,  0.11340519, ...,  0.84304574,
         0.65341548,  0.11447318],
       [ 1.57467708,  0.79290427,  0.11340519, ...,  1.14796951,
         1.04559751,  0.8683921 ],
       [ 0.37564751,  0.70143387,  0.11340519, ...,  1.52858358,
         1.74869949,  0.8683921 ],
       ...,
       [ 1.22157303,  0.79290427,  0.11340519, ...,  0.61381227,
         0.50634721,  0.11447318],
       [ 1.11297817,  0.90510323,  0.11340519, ...,  1.08020867,
         0.947552  ,  0.8683921 ],
       [ 0.11145456,  0.79290427,  0.11340519, ...,  0.87692616,
         0.65341548,  0.8683921 ]])
```

In [41]:

```
scaled_X_train.shape
```

Out[41]:

```
(1034, 36)
```

In [42]:

```
scaled_X_test=scaler.transform(X_test)
scaled_X_test
```

Out[42]:

```
array([[ [illegible]
```

In [43]:

```
from sklearn.metrics import mean_squared_error
```

```
       233147.40044683, 125478.90203711, 175060.50167495, 258877.30470763,
       229115.65129666])
```

In [44]:

```
Scaler=StandardScaler()
scaled_X_train=Scaler.fit_transform(X_train)
scaled_X_train
```

Out[44]:

```
array([[ 0.39851037, -0.79290427, -0.11340519, ...,  0.84304574,
         0.65341548,  0.11447318],
       [ 1.57467708, -0.79290427, -0.11340519, ...,  1.14796951,
         1.04559751,  0.8683921 ],
       [ 0.37564751,  0.70143387, -0.11340519, ..., -1.52858358,
        -1.74869949,  0.8683921 ],
       ...,
       [ 1.22157303, -0.79290427, -0.11340519, ..., -0.61381227,
         0.50634721,  0.11447318],
       [-1.11297817,  0.90510323, -0.11340519, ...,  1.08020867,
         0.947552  ,  0.8683921 ],
       [ 0.11145456, -0.79290427, -0.11340519, ...,  0.87692616,
         0.65341548,  0.8683921 ]])
```

In [45]:

```
scaled_X_test=Scaler.transform(X_test)
scaled_X_test
```

Out[45]:

```
array([[ 0.85830772, -0.79290427, -0.11340519, ...,  1.01244784,
         0.89852925,  0.11447318],
       [-0.64810018, -0.79290427, -0.11340519, ..., -0.27500808,
        -1.01335818,  0.8683921 ],
       [-0.21624632,  0.76093278, -0.11340519, ..., -2.47723531,
        -1.74869949, -1.39336465],
       ...,
       [-0.04350477,  0.37647826, -0.11340519, ..., -1.86738777,
        -1.74869949,  0.11447318],
       [-0.64301955,  1.46805451, -0.11340519, ...,  0.63976323,
         0.31025619, -1.39336465],
       [-0.29499614,  0.81585486, -0.11340519, ...,  0.47036113,
         0.06514242, -1.39336465]])
```

In [46]:

```
model1=LinearRegression()
model1.fit(scaled_X_train,y_train)
sy_pred=model1.predict(scaled_X_test)
sy_pred
```

Out[46]:

```
array([257434.93050748, 111083.73474764, 100018.0583229 , 204028.53821316,
       207319.25418314,  38036.3092894 , 234153.38582869, 205076.12689603,
       187014.12655231, 235636.78799029, 100976.5094378 , 304119.53646984,
       101769.64600715, 288758.46001591, 204767.89338331, 145002.4727962 ,
       248322.97436853, 151533.68038638, 209697.39458715, 278312.23574155,
       103329.85601463, 157784.1486038 , 163237.87451135, 241495.91212888,
       172429.94726951, 221672.07367603, 119658.75685733, 100039.78137072,
       194435.02892076, 172088.64665032, 258772.64470141, 199597.36129638,
       156349.69283197, 142311.22500108, 204807.33161322, 379608.17785439,
       124176.23377553, 141127.25006143, 273832.43247648, 212547.0420556 ,
       169474.5717514 , 123505.6571939 , 200541.2401877 , 377957.32307109,
       146941.7171239 , 283693.48760665, 106694.19786715, 221733.02172498,
        87733.4701401 , 295005.66546303, 125250.6201929 ,  53569.11256027,
       133831.05086115, 170901.42929711, 213273.17757298, 110818.14723239,
       103893.84840705, 190665.21910141, 137764.84464906, 311840.64900749,
       153035.52267581, 184866.47797854, 154675.79963876, 298477.53600002,
        78162.54074685, 323728.93535801, 169417.44138502, 112718.07937644,
       200547.87648959, 205739.30910765, 130292.88475649, 271581.37221157,
       119489.76459236, 126635.29232493,  77972.26780485, 138563.37305274,
        35555.51251474, 163787.98080894, 371929.1369019 , 115260.62742894,
       319958.55004398, 218515.16541523, 291730.97232826, 211176.53228417,
        93761.81278832, 263031.10150316, 141886.39666991, 118491.50416098,
       143580.34191292, 278395.8537048 , 265558.84177935, 300839.51324223,
       182673.3443797 , 168604.54926019, 148322.24627427, 213781.03244608,
       247750.58740996, 221693.90686781, 274474.96290584, 242912.73279158,
       111041.26737678, 100160.6057377 , 193531.44933267, 206050.88712628,
       154584.47386049, 217440.52365088, 210904.23539268, 125864.27688701,
       171322.6831988 , 224459.70210705, 155548.12334504, 111942.78717749,
       316624.83474108, 298116.31992275, 335573.77762719,  86738.8850682 ,
       146692.58771907, 228471.54032911, 123374.07354541, 204867.33148108,
       217343.2172092 , 120569.43114714, 134354.12384256, 223860.01643707,
       138360.76267459, 210765.75439351, 145423.97229839, 291155.26974382,
       183089.17904137,  98716.55524168, 323896.46706842, 170447.30057099,
       123643.15346305, 134345.24799852, 309423.49789816, 136836.97808939,
       253083.02092585, 132621.22372555,  90821.15043383, 164193.14528597,
       147333.60026799, 162030.92949967, 177621.20328223, 254054.45691356,
       119092.92298358, 140081.84030887, 236509.39317217, 318578.4396165 ,
       105538.81578044, 192023.58779378, 174531.98294676, 146314.75164223,
        95146.4623636 , 250936.32825849, 304056.57810764,  55008.42319126,
       281810.05395731,  93402.68236917, 138283.44603758, 166415.31922933,
       209758.26998557, 205602.16976193, 177026.58819392, 248435.62725108,
       145120.55457312, 195263.94785761, 171593.43394517, 344925.23412389,
       181194.80288335,  98431.28117765, 131287.46561778, 110849.96802293,
       131283.81966778, 194221.20494201, 156966.39268746, 268340.58471954,
       211804.83660278, 151971.72881378, 115946.37551381, 243618.89320954,
       136292.76068278, 265335.41730192, 302175.59064211,  85054.45272485,
       146007.425328  6, 150439.28821419, 159145.83029284, 232246.23120847,
       221571.1353201 , 139178.99313682, 355030.61932406, 127659.29864107,
       235239.23724176, 116707.21916903, 103103.75191053, 159574.2341845 ,
       149485.12469047, 408898.40096808, 335368.65026121, 290500.91914897,
       281845.43487677, 275341.81718413, 321705.52591335, 307549.31796902,
       201661.19536927, 142047.81494963, 157876.45367581, 258695.10810312,
       207592.92972056, 146121.60182467, 107032.82242136, 151245.32235212,
       100804.20263822, 296980.21553202, 341413.30991675, 255158.86136984,
       147006.34954488, 199721.9925516 , 127291.41093952, 264030.58756771,
       121760.1939664 , 134138.78148626, 307661.79220113, 170331.68886901,
       173593.89056667, 567240.20119423, 182277.7226559 , 144159.48526094,
       109515.1789637 , 107681.20503763, 181383.00130506, 228401.72330995,
       129225.20214767, 197660.10401735, 116340.37479633,  54949.57465393,
       199375.67314464, 272915.38777825, 119902.62390782, 197573.81604287,
       231881.98496931, 119555.98822724, 128709.54702316, 289158.8058549 ,
       200733.61408615, 364228.39366111, 119308.94568442, 124086.72254034,
       225161.62739516, 171005.22659816,  93631.9176723 , 178506.48207582,
       215661.6458802 , 226225.3003896 , 132391.85608605,  73443.41948312,
       242913.00968347, 141042.39030918, 114633.38186074,  89093.07024775,
       208503.15479643, 172828.98634139, 270797.22136179, 253693.44924279,
       103518.85057202, 220521.27460717, 185034.32129264, 169518.90296988,
       180628.92554339, 200106.98029839, 179499.81727936, 223755.69902623,
        73639.52734136,  86011.81491626, 110000.03909849, 231518.89807255,
       153369.35976788, 320899.35432164, 194412.83170321,  57741.65743008,
       203877.63015227, 109803.05085706, 194633.36256922,  99160.35922074,
       230240.81594383, 211484.77477047, 198342.97693778, 109749.28794505,
        95564.23930695, 235396.44527242, 184775.90385426, 188806.96222902,
       265328.01705249, 245862.13668722, 107422.0040252 , 164786.63106781,
       268303.44421406, 116075.01592358, 222914.33666728, 102600.43292176,
       226400.49839176, 156093.55758654, 112003.82907507, 218089.37220885,
        81085.92660268,  85015.51341445,  98208.34263402, 304262.27895917,
       143757.79043401, 195367.35122022, 241835.21911119,  60944.84144924,
       143731.15831783, 110459.58351738, 438412.24654495, 123265.07239639,
       116119.87930815, 277961.71617821, 201687.81739716, 237947.02580915,
       206117.84987327,  99453.60459335, 115124.57035038, 209795.31353812,
       167537.15563228, 162970.2651776 , 151279.43057794, 189724.86274099,
        96455.47006121, 168652.8258692 ,  83051.31596961, 171340.42959833,
       176868.3371188 , 175644.68961809, 131414.67048232, 204543.74960378,
```

In [47]:

```
mse_In1=mean_squared_error(y_test,y_pred)
mse_In1
print("mean squared error using standard scalar",mse_In1)
```

```
mean squared error using standard scalar 1474827325.15975823
```

In [48]:

```
from sklearn.preprocessing import MinMaxScaler
mm_scaler=MinMaxScaler()
```

In [49]:

```
mmX_train=mm_scaler.fit_transform(X_train)
mmX_train
```

Out[49]:

```
array([[0.21133051, 0.        , 0.        , 0.90769231, 0.81666667,
        0.5       ],
       [0.32016225, 0.        , 0.        , 0.87692308, 0.95      ,
        0.75      ],
       [0.20921486, 0.31622276, 0.        , 0.36923077, 0.        ,
        0.75      ],
       [0.28749412, 0.        , 0.        , 0.87692308, 0.76666667,
        0.5       ],
       [0.07146215, 0.35932203, 0.        , 0.96153846, 0.91666667,
        0.75      ],
       [0.18476728, 0.        , 0.        , 0.91538462, 0.81666667,
        1.        ],
       [0.2538787 , 0.        , 0.        , 0.94615385, 0.9       ,
        0.5       ],
       [0.11448049, 0.        , 0.        , 0.85384615, 0.25      ,
        0.75      ],
       [0.13442288, 0.32881356, 0.        , 0.15384615, 0.        ,
        0.        ],
       [0.17042783, 0.24745763, 0.        , 0.29230769, 0.        ,
        0.75      ],
       [0.11495063, 0.47845038, 0.        , 0.86153846, 0.7       ,
        0.        ],
       [0.14715582, 0.34043384, 0.        , 0.82307692, 0.61666667,
        0.        ]])
```

In [50]:

```
mmX_test=mm_scaler.transform(X_test)
mmX_test
```

Out[50]:

```
array([[0.2538787 , 0.        , 0.        , 0.94615385, 0.9       ,
        0.5       ],
       [0.11448049, 0.        , 0.        , 0.85384615, 0.25      ,
        0.75      ],
       [0.13442288, 0.32881356, 0.        , 0.15384615, 0.        ,
        0.        ],
       [0.17042783, 0.24745763, 0.        , 0.29230769, 0.        ,
        0.75      ],
       [0.11495063, 0.47845038, 0.        , 0.86153846, 0.7       ,
        0.        ],
       [0.14715582, 0.34043384, 0.        , 0.82307692, 0.61666667,
        0.        ]])
```

In [51]:

```
model3=LinearRegression()
model3.fit(mmX_train,y_train)
```

Out[51]:

```
LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None, normalize=False)
```

```
       233147.40044682, 125478.90203713, 175060.50167486, 258877.30470761,
       229115.6512967 ])
```

In [52]:

```
mms_y_pred=model3.predict(mmX_test)
mms_y_pred
```

Out[52]:

```
array([257434.93050748, 111083.73474764, 100018.0583229 , 204028.53821316,
       207319.25418314,  38036.3092894 , 234153.38582869, 205076.12689603,
       187014.12655231, 235636.78799029, 100976.5094378 , 304119.53646984,
       101769.64600715, 288758.46001591, 204767.89338331, 145002.4727962 ,
       248322.97436853, 151533.68038638, 209697.39458715, 278312.23574155,
       293329.85601463, 153784.1486838 , 153237.87451135, 241495.91212888,
       372429.94726951, 221672.07367603, 119658.75685733, 100039.78137072,
       319435.02892076, 172088.64665032, 258772.64470115, 199597.36129638,
       156349.69283197, 142311.22500108, 204807.33161322, 379608.17785439,
       124176.23377553, 141127.25006143, 273832.43247648, 212547.0420556 ,
       169474.57176143, 123305.65719309, 200341.24016795, 377957.32307109,
       146041.7171239 , 282683.48760665, 196692.19786715, 221733.02172498,
        87733.4701401 , 295005.66546303, 125250.6201929 ,  53569.11256027,
       133831.05086115, 170901.42929711, 213273.17757298, 110818.14723239,
       103893.81849705, 200665.21921041, 137764.84464906, 311840.64900749,
        53035.52267581, 184866.47797854, 154675.79963876, 298477.53600001,
        78162.54074685, 323728.93535801, 169417.44138502, 112718.07937644,
       200547.87648959, 205739.30910765, 130292.88475649, 271581.37221157,
       119489.76459236, 126635.29232493,  77972.26780485, 138563.37305274,
        35555.51251474, 163787.98080894, 271929.1369019 , 115260.62742894,
       319958.55004398, 218515.16541523, 291730.97232826, 211176.53228417,
        93761.81278832, 263031.10150316, 141886.39666991, 118491.50416098,
       143580.34191292, 278395.8537048 , 265558.84177935, 300839.51324223,
       182673.3443797 , 168604.54926019, 148322.24627427, 213781.03244608,
       247750.58740996, 221693.90686781, 274474.96290584, 242912.73279158,
       111041.26737678, 100160.6657377 , 193531.44933267, 206050.88712628,
       154584.47386049, 217440.52365088, 210904.23539268, 125864.27688701,
       171322.66919868, 224459.70210705, 155548.12334504, 111942.78717749,
       316624.83474108, 298116.31992275, 335573.77762719,  86738.8850682 ,
       146692.58771907, 228471.54032911, 123374.07354541, 204867.33148108,
       217343.2172092 , 120569.43114714, 134354.12384257, 223860.01643707,
       138360.76267459, 210765.75439351, 145423.97229839, 291155.26974381,
       183089.17904137,  98716.55524168, 323896.46706842, 170447.30657099,
       123643.15346305, 134345.24799852, 309423.49789816, 136836.97808939,
       282083.02092585, 132621.22372555,  90821.15043383, 164193.14328597,
       147333.60026799, 162030.92949967, 177621.20328223, 254054.45691356,
       119092.92298358, 140081.84030887, 236509.39317217, 318578.4396165 ,
       105538.81578044, 192023.58779378, 174531.98294676, 146314.75164223,
        95146.4623636 , 250936.32825849, 304056.57810764,  55008.42319126,
       281810.05395731,  93402.68236917, 138283.44603758, 166415.31922933,
       209758.26998557, 205602.16976193, 177026.58819392, 248435.62725108,
       145190.55457312, 133283.94785761, 171593.43394517, 344925.23412389,
       181194.80288335,  98431.28117765, 131287.46561778, 110849.96802293,
       131283.81966778, 194221.20494201, 156966.39268746, 268340.58471954,
       211804.83660278, 151971.72881378, 115946.37551381, 243618.89320954,
       136292.76068278, 265335.41730192, 302175.59064211,  85054.45272485,
       146007.43532806, 150439.28821419, 159145.83079284, 232246.23120847,
       231571.1353201 , 139178.99313682, 355030.61932406, 127659.29864107,
       235339.23724176, 116707.21916903, 103103.75191053, 159574.2341845 ,
       149485.12469047, 408898.40096808, 335368.65026121, 290500.91914897,
       281845.43486757, 275341.81718413, 321705.52591335, 307549.31796902,
       201661.19536927, 142047.81494963, 157876.45367581, 258695.10810312,
       207592.92972056, 146121.60182467, 107032.82242136, 151245.32235212,
       100804.20363832, 296980.23153202, 341413.30991675, 255158.86136984,
       147006.34954488, 199721.9925516 , 127291.41093952, 264030.58756771,
       121760.1939664 , 134138.79148626, 307661.79229112, 170331.68886901,
       173593.89056667, 567240.20119423, 182277.7226559 , 144159.48526094,
       199515.17896317, 107691.20503763, 181383.00130506, 328401.72330995,
       139225.20214767, 197660.10401735, 116340.37479633,  54949.57465393,
       199375.67314464, 272915.38777825, 119902.62390782, 197573.81604287,
       231881.98496931, 119555.98822724, 128709.54702316, 289158.8058549 ,
       200733.61408615, 364228.39366111, 119308.94568442, 124086.72254035,
       225161.62739516, 171005.22659816,  93631.9176723 , 178506.48207582,
       215661.64584802, 226225.3003896 , 132391.85608605,  73443.41948312,
       242913.00968347, 141042.39030918, 114633.38186074,  89093.07024775,
       208503.15479643, 172828.98634139, 270797.22136179, 253693.44924279,
       103518.85057202, 220521.27460717, 185034.32129264, 169518.90296988,
       154846.17459054, 224605.38839966,  85972.23828463, 151323.12661365,
       180628.92554339, 200106.98029839, 179499.81727936, 223755.69902623,
        73639.52734136,  86011.81491626, 110000.03909849, 231518.89807255,
       153369.35976788, 320899.35432164, 194412.83170321,  57741.65743008,
       203877.63015227, 109803.05085706, 194633.36256922,  99160.35922074,
       230240.81594383, 211484.77477047, 198342.97693778, 109749.28794505,
        95564.23930695, 235396.44527242, 184775.90385426, 188806.96222902,
       265328.01705249, 245862.13668722, 107422.0040252 , 164786.63106781,
       268303.44421406, 116075.01592358, 222914.33666728, 102600.43292176,
       226400.49839176, 156093.55758654, 112003.82907507, 218089.37220885,
        81085.92660268,  85015.51341445,  98208.34263402, 304262.27895917,
       143757.79043401, 195367.35122022, 241835.21911119,  60944.84144924,
       143731.15831783, 110459.58351738, 438412.24654495, 123265.07239639,
       116119.87930815, 277961.71617821, 201687.81739716, 237947.02580915,
       206117.84987327,  99453.60459335, 115124.57035038, 209795.31353812,
       167537.15563228, 162970.2651776 , 151279.43057794, 189724.86274099,
        96455.47006121, 168652.8258692 ,  83051.31596961, 171340.42959833,
       176868.3371188 , 175644.68961809, 131414.67048232, 204543.74960378,
```

In [53]:

```
mmMSE=mean_squared_error(y_test,mms_y_pred)
```

In [54]:

```
print("mean sqaure error using min max scaler: ",mmMSE)
```

```
mean sqaure error using min max scaler:  1496827.1975973811
```

```
       233147.40044682, 125478.90203713, 175060.50167486, 258877.30470761,
       229115.6512967 ])
```

In [59]:

```
from sklearn.linear_model import SGDRegressor
sgd=SGDRegressor()
sgd.fit(scaled_X_train, y_train)
sgd_y_pred=sgd.predict(scaled_X_test)
print("Predictions of scaled data using SGDRegressor:", sgd_y_pred)
```

```
Predictions of scaled data using SGDRegressor: [257091.46017544 106042.08998592 108527.94507204 202082.10522505
 209461.79410059  34624.20874776 237505.19048492 205316.91070978
 184574.32449253 234355.59450454  99064.57498927 303589.07540494
  99842.30604406 297124.37065545 207637.28875766 153366.24734498
 243671.45462599 148898.94494287 211232.91988745 275472.15072131
  93161.11656339 148683.12761984 164493.34758947 243611.4647815
 380124.99802158 220234.33335125 127514.53727007  95807.60639554
 325803.77661287 169247.77012666 256933.67701027 203871.86224417
 148184.84701439 137129.53135912 207241.15957308 380444.32176964
 116429.51656084 144200.26028454 278186.87870156 206508.20390402
 164452.37071611 121101.04348929 194938.46711798 378343.71825541
 152860.62436276 294898.33338617 100262.45052073 217296.82171397
  84671.37317478 306363.41868878 122161.80106562  49992.57160709
 140857.408276   171828.69308789 208599.71565883 112668.16180724
 104684.68393158 205536.21069115 138469.91982267 320153.29814189
  44354.33021495 190552.06305323 153212.35111395 295301.26000775
  80347.84126854 328508.78287058 173915.68672164 109499.46132528
 202226.78501919 203366.69962687 134788.93568887 272183.22678348
 116858.07032497 130957.37718956  72411.60710761 134695.93239109
  30572.99263561 164514.95690301 280650.50219178 109535.5900692
 327585.31675134 218325.82191808 297626.18029214 212988.10182392
  92465.48963296 258249.18271566 139216.75988593 113485.77000912
 135463.75598291 284405.44201711 269586.32015817 298771.09156656
 179465.23228256 165258.45024036 153295.06104484 209546.13527539
 244792.56400298 216364.40553588 278683.69935871 241254.48205473
 106814.47776045 103230.9025873  198674.84368017 204112.02917113
 154158.59405109 214836.16592503 213350.50922919 122645.26961372
 170387.93409207 242174.29669993 150261.39184498 112659.35141691
 321247.63817056 292917.30326744 340449.84022278  81783.4446023
 147309.76779032 220989.06912362 119027.50200628 199488.64851674
 212467.32432461 126434.23374257 127464.15240973 224147.22304139
 135705.59823899 205175.2748297  144254.88759841 293848.48115179
 179304.39585605  91604.60660368 318768.69682153 169935.56164524
 128278.49048291 130490.0613867  310027.99991702 143294.69195508
 279622.26741856 136418.45104488  85944.11424722 166739.04254859
 142962.01656619 159571.59807289 179403.18670514 252050.33884594
 122282.62836538 138186.32874456 235585.81782428 323527.700349
 108023.59907487 188255.43441317 176524.68581296 145665.76169633
  89191.14544062 250932.54948365 315584.91955657  47709.51287799
 286351.39888427  94242.95244734 142417.40810733 174290.57780999
 211190.90182635 201482.60022211 173702.04460812 258541.25610725
 153591.86481917 130646.73292271 167560.57803169 346329.67598867
 179715.09410279  92358.68346385 126416.84343899 104951.83235673
 124085.96460448 183950.49776423 154443.49141401 272167.66635679
 207764.44588155 151512.3162005  114081.74884333 246982.18016784
 123955.84743517 265007.18060352 308072.56445138  86684.13328435
 140075.94157924 145443.29940937 158703.83748616 236050.25625423
 234134.61506582 138149.18984999 354814.80382633 127969.62964284
 237039.41604837 113975.828479    97908.50638558 164554.65291558
 152292.45349462 409160.06963737 345385.62104493 291704.60047706
 280544.42875864 278549.47647318 317512.72409123 311700.7648852
 203030.70408781 140418.36362891 160206.15357253 247942.77263629
 207795.75900907 147039.00651544 109144.76939253 147817.38875364
 101339.60540492 302613.19479123 350551.81462856 256416.38114419
 144452.07006526 200736.18573187 129254.99918914 266972.14853924
 126807.61500302 138344.93920847 313605.39964357 171819.79687815
 166929.80606203 572472.50854343 188173.35743687 149316.39580374
 203112.65760409 106550.26154247 181719.11284177 333844.68781357
 135464.93989785 194101.83171039 120324.55798416  49328.028965
 197748.57038181 277187.99541648 113748.20408655 206110.86761143
 237418.6626885  118378.5154093  131250.25475187 286436.67082311
 205256.33787212 372828.17620961 116082.20000064 126076.48208182
 219384.63870375 168127.87911494  86227.82937477 176475.9818725
 217190.18843869 227441.55028985 133924.50094772  80115.4828712
 245477.88282586 140125.45790217 112114.87319751  97232.4242616
 212849.68440365 164220.32518404 274728.07235669 257650.82963116
  99866.95372333 220581.21466529 191383.37393815 164765.06794613
 158989.31786931 230024.1774638   82481.71667627 150509.9626051
 179495.91881948 199235.45115608 177404.99733298 218157.20774049
  43840.61193095  88728.58910575 104694.59773146 232567.8221411
 152324.92207438 317846.40177822 192810.50804307  58607.02288426
 204514.71655964 110495.73611478 197189.33139686 100096.14770364
 230756.61362831 215525.5338806  211677.79020572 108479.9823298
  94879.5039108  242952.67541366 186697.30621821 188883.87087968
 264344.3310676  240931.38286349 105284.78972523 157045.84250287
 270426.68454519 111978.10147353 220114.03481324 103181.92961962
 229266.70214924 154664.83753039 114030.95479455 223274.24307793
  83258.37611191  82210.60312013  89948.16237422 312758.36443855
  97730.32221594 185504.12697839 249534.20156044  53625.99588916
 143790.06257189 115497.34005841 445741.41598326 115734.91059775
 117277.22440407 280346.65983349 197299.65614799 239759.20614456
 215277.20039685  97658.70733114 115539.09621145 204529.07866149
 161093.15903442 157834.74048346 151315.2054891  191975.49188287
  88856.43009346 179434.78145205  74903.79387818 163937.04426513
 179968.77346957 164729.01999777 131458.03558089 203822.15215231
```

```
 229572.48033751 122237.12639169 179041.34990555 259582.94225157
 221962.96112881]
C:\Program Files (x86)\Microsoft Visual Studio\Shared\Anaconda3_64\lib\site-packages\sklearn\linear_model\stochasti
c_gradient.py:128: FutureWarning: max_iter and tol parameters have been added in <class 'sklearn.linear_model.stoch
astic_gradient.SGDRegressor'> in 0.19. If both are left unset, they default to max_iter=5 and tol=None. If tol is n
ot None, max_iter defaults to max_iter=1000. From 0.21, default max_iter will be 1000, and default tol will be 1e-
3.
  "and default tol will be 1e-3." % type(self), FutureWarning)
```

In [60]:

```
sgd_mse=mean_squared_error(y_test, sgd_y_pred)
print("SGD_MSE:",sgd_mse)
```

```
SGD_MSE: 1572075620.9101768
```

In [65]:

```
from sklearn.linear_model import Ridge
ridge=Ridge()
ridge.fit(scaled_X_train, y_train)
ridge_y_pred=ridge.predict(scaled_X_test)
print("Predictions of scaled data using RIDGERegression:", ridge_y_pred)
```

```
Predictions of scaled data using RIDGERegression: [257421.40327467 111048.93900918 100148.22166502 204007.69016285
 207318.26438778  38130.16627273 234150.19679274 205083.79807217
 186990.35389928 235496.28665193 100966.52935332 303934.63164847
 101890.94872879 288779.98473831 204801.28875069 145108.44985327
 248228.38505616 151485.3620995  209733.2274515  278197.4394422
  93455.6609326  153804.93005254 163194.49635267 241430.2463605
 372368.42646834 221632.71135779 119760.90284387 100047.58843771
 319413.64552184 172029.21920287 258742.7512024  199493.44526659
 156253.40634584 142260.62082583 204826.1806726  379453.76863261
 124049.03967105 140938.68021813 273742.38050574 212502.29963116
 169496.97954299 123279.27640286 200131.43501379 377624.34223268
 147067.67507791 283730.24445742 106654.11168627 221721.98335033
  87859.6015935  294987.18055758 125202.71399397  53519.12559123
 133851.65120566 170814.11457786 213257.89409118 110890.21052619
 103962.51733658 200705.10120865 137825.92554828 311801.15478353
  53110.79651703 184799.6096954  154734.86644008 298380.40493944
  78246.19078408 323702.02510789 169466.06274122 112823.93682159
 200532.47961727 205732.7434815  130430.90213571 271576.54141384
 119517.69643887 126813.06031022  77948.07913159 138459.94428071
  35594.69872621 163851.24303414 271911.16049004 115170.8409182
 319926.92800417 218539.48490804 291693.99915374 211226.11308583
  93921.33416786 262946.95574916 141809.88774323 118408.82479328
 143529.08575669 278395.45721621 265527.26500154 300767.84214392
 182712.18425801 168624.1947174  148468.0112008  213828.16526345
 247699.43190119 221655.98604126 274448.97443032 242853.53739016
 111008.88869611 100159.22278121 193584.21304243 205986.02391731
 154621.7915883  217316.85068503 210896.31853698 125867.74499667
 171334.06377096 224552.40663195 155561.4891094  112029.96523929
 316553.30396313 297970.38898717 335452.56642894  86702.91494695
 146732.96845793 228454.197212   123467.16116323 204859.18689786
 217316.45502954 120778.23382002 134362.92386445 223895.62311255
 138354.30270403 210759.85184416 145371.25618944 291104.98829883
 183135.45289874  98714.38796483 323759.01921323 170426.52509926
 123724.57076007 134394.63803712 309274.63604218 136992.66409229
 282021.23633084 132696.1999706   90813.11962894 164237.83443573
 147281.26846666 162003.41169996 177647.14470987 253988.2363156
 119205.33260286 140118.15986393 236442.68491103 318486.93596214
 105671.08096883 192080.32594035 174466.08136758 146377.97611372
  95151.72242267 250913.00451295 304117.39635736  55131.83869741
 281794.68069314  93471.49702787 138431.7751144  166584.56732248
 209797.30479015 205691.94925356 177054.65626513 248576.02992629
 145449.73519361 133220.86441506 171632.89844021 344767.21487893
 181092.55310315  98456.02843269 131282.04325445 110934.36779116
 131317.332356   194129.22012668 156885.58294314 268253.70749367
 211776.05309354 151997.19209824 115885.87157944 243633.12876265
 136046.12092605 265371.76336608 302122.63871315  85249.62268514
 145934.87875522 150476.83298288 159184.2915899  232268.61259225
 231568.20715322 139168.63211663 354705.36600225 127677.9652855
 235350.87769917 116731.80224207 103066.91017833 159648.98348223
 149516.96210387 408710.91797528 335360.95624801 290491.24871895
 281759.86238466 275280.43869755 321581.48463028 307495.11977823
 201708.6866945  142026.52881773 157911.49475873 258946.36167454
 207616.54676893 146189.84392674 107127.69428021 151193.37312002
 100806.45474285 297016.85500457 341332.96001968 255156.65509949
 147042.32133388 199835.1635117  127396.28236212 264028.87507205
 121842.35752249 134224.48204824 307690.96067651 170299.19516249
 173634.61888396 566895.37162083 182267.3152515  144230.71243381
 199542.41967282 107725.24929291 181340.3055302  328346.05768061
 139169.93510577 197676.26414789 116515.40028976  54920.97367886
 199359.79732577 272909.48207651 119964.15243506 197635.14098683
 231846.26227277 119652.57513788 128795.15181617 289062.25535611
 200794.72224425 364283.2369707  119300.12544367 124055.74846497
 225088.06389983 170924.12061461  93546.44521775 178549.05783466
 215651.32553685 226256.01720285 132397.67484753  73733.38519087
 242947.50810971 141056.27144097 114625.16811511  89328.50332163
 208394.9758949  172744.77142958 270816.44537974 253664.42153481
 103542.72018456 220477.69002152 185184.89845883 169549.94526138
 154899.03211341 224612.6907629   86017.93200485 151268.69678584
 180569.55851468 200116.32845338 179525.24701369 223517.0089862
  73881.92160492  86307.0983211  109982.15609129 231440.35424211
 153338.62588424 320803.87482713 194414.65718501  57804.18747402
 203909.81175009 109852.78794991 194673.94964359  99307.14089046
 230248.65801279 211480.2526221  198415.83220873 109779.16815645
  95603.43692129 235431.54757453 184887.17845958 188858.81438927
 265348.1010974  245750.11638204 107449.72804741 164526.92609266
 268206.85567299 116000.37230721 222832.75187152 102708.53949781
 226446.39169676 156039.73352165 112118.2621359  218009.48577344
  81138.79394674  85183.53240143  98120.58664156 304221.85064095
 143825.10439664 195254.9013155  241807.65208589  60898.21659707
 143757.7855566  110569.79788548 438249.62394314 123258.66869765
 116123.99585904 277966.01915293 201690.38568323 237978.5356511
 206247.6433917   99449.19364681 115124.23894833 209778.46658561
 167434.01104678 162852.00375468 151360.56894059 189760.65510995
  96349.80598172 168767.97321727  83076.25780314 171281.0121331
 176931.55688827 175568.0530263  131382.65018492 204547.73197275
```

```
 233033.33528768 125501.91800535 175116.80170643 258825.79733597
 228967.00891845]
```

In [66]:

```
ridge_mse=mean_squared_error(y_test, ridge_y_pred)
print("RIDGE_MSE:",ridge_mse)
```

```
RIDGE_MSE: 1473019452.6343954
```

In [67]:

```
from sklearn.linear_model import Lasso
lasso=Lasso()
lasso.fit(scaled_X_train, y_train)
lasso_y_pred=lasso.predict(scaled_X_test)
print("Predictions of scaled data using LASSORegression:", lasso_y_pred)
```

```
Predictions of scaled data using LASSORegression: [257421.66197135 111077.71879469 100035.6191829  204024.7169887
 207324.54758555  38037.76044559 234149.07629388 205067.06537765
 186995.76236831 235644.9892929  100980.81814107 304124.32600883
 101772.17918346 288771.22252734 204759.49619808 145026.2155349
 248321.68441805 151528.08346291 209694.08183559 278309.14448672
  93335.15311431 153783.56780017 163228.65012312 241483.60508861
 372429.51050608 221679.58601399 119671.76786021 100045.46561762
 319437.725821   172082.33265962 258777.40797254 199570.17239129
 156363.52609123 142320.7170313  204809.98100164 379595.39611426
 124166.56112343 141099.38504814 273833.85728474 212527.67447459
 169476.63411535 123302.15760227 200345.31641435 377933.4095104
 146945.90334484 283683.39848157 106687.15479997 221733.41564452
  87730.76613378 295006.80358316 125244.58525479  53575.54835405
 133831.34643938 170889.49976251 213275.46196983 110829.42033236
 103891.6332743  200669.15236764 137760.96717638 311838.89428256
  53043.94004306 184860.11037264 154678.2527721  298472.04195713
  78160.84398476 323727.20193672 169432.06791697 112719.49362593
 200537.03858638 205730.23822059 130319.03385979 271572.18419671
 119472.76012401 126676.8179564   77973.93013485 138543.96745588
  35566.47418905 163776.26481031 271932.10184352 115245.1018397
 319947.80376808 218524.65113731 291728.64199643 211184.25132808
  93766.24418463 263013.07200756 141884.14887891 118479.94316
 143579.25976741 278387.09016183 265542.4010783  300840.57586394
 182676.22637    168614.55293982 148325.13384874 213772.6534435
 247744.3651599  221694.07398868 274473.40032123 242910.07733687
 111024.32538033 100163.63124117 193529.83383857 206051.99840087
 154580.62661267 217424.37479503 210897.8675803  125862.78260242
 171328.72206529 224472.85497059 155547.87084065 111948.33451172
 316614.91783726 298110.80068024 335577.64106164  86728.4538159
 146691.87441899 228462.03993473 123369.93579067 204857.54373339
 217328.94220496 120601.69683214 134345.84299432 223863.77784552
 138369.30181601 210754.70687606 145421.12522723 291149.08503414
 183093.71939896  98717.31648601 323879.66306981 170451.65428263
 123658.89486739 134346.04127115 309415.83992248 136856.15012265
 282080.39764236 132642.24782298  90805.08020318 164197.20983374
 147307.78749655 162024.48429843 177631.27200207 254051.12619419
 119093.87836429 140079.94586752 236494.44620744 318591.37268904
 105548.04487938 192017.47219519 174529.12056029 146312.3192297
  95155.25228485 250937.19138626 304068.78612049  55008.97537921
 281813.60681016  93421.72719713 138282.43504312 166444.38175778
 209754.51753527 205585.33877568 177023.0120615  248437.29278052
 145210.40127476 133272.48067222 171592.92483253 344930.02937206
 181190.92645123  98431.12271156 131285.74545319 110846.83896709
 131266.92367927 194210.32162317 156959.82444794 268332.77077295
 211799.49315843 151961.57337152 115947.01195725 243618.37820482
 136267.49311244 265336.52498987 302183.29729299  85066.61513023
 145993.91535619 150453.5803261  159131.11701994 232247.04116354
 231568.64435712 139159.53154749 355010.16172113 127649.28853894
 235347.25114993 116723.15594985 103097.80873158 159579.48633352
 149485.86637871 408877.80842203 335364.79787407 290491.82929326
 281827.17369039 275344.01943175 321689.95267985 307550.06260338
 201657.13893689 142043.29571062 157877.79211304 258686.36229281
 207597.50455903 146126.19536972 107060.5836089  151209.63196943
 100803.88138449 296986.5536651  341413.68242682 255159.09225146
 147004.67858323 199741.60256684 127286.60854202 264039.47903905
 121757.76018412 134168.47750067 307664.59299118 170333.21476955
 173582.30534653 567237.57815443 182257.07700069 144169.57742366
 199532.70579598 107682.44334241 181381.68902942 328403.51993725
 139213.92957091 197651.36277817 116359.69944472  54926.04103464
 199373.73822273 272930.85953348 119891.94030613 197577.32950931
 231874.9035266  119554.80329438 128727.34148942 289156.98065855
 200732.19876921 364217.55430551 119318.00170502 124076.51847075
 225155.15159498 170995.84911163  93628.56726571 178508.96243962
 215656.81266785 226226.64669321 132395.31450583  73477.28786308
 242922.64107088 141045.5898111  114624.54008242  89110.5621172
 208500.76357774 172822.2226141  270785.25838424 253704.15606858
 103518.18195189 220525.08538562 185034.26794754 169525.90813566
 154842.46818282 224609.79313203  85978.13407565 151315.30593555
 180620.69731956 200112.16486667 179501.34198625 223731.81293301
  73647.74981614  86044.36289061 109989.31336578 231531.36662467
 153351.49297766 320891.11855445 194414.35048782  57747.24284655
 203879.31819891 109783.81220237 194641.16474284  99165.08860284
 230240.64044644 211487.44994193 198349.64311181 109741.69039496
  95564.46870871 235405.4273042  184786.80800297 188804.34018409
 265336.71871244 245859.81354339 107410.41353388 164771.90203723
 268301.56371188 116072.68714602 222929.57590988 102598.5478327
 226401.18603924 156092.60471447 112036.64841012 218083.15785584
  81100.88899446  85030.47954969  98197.57559804 304263.87834769
 143748.4550891  195361.44665619 241819.00146474  60942.15059443
 143722.83401197 110480.74297331 438384.9892158  123261.18298137
 116126.99598144 277962.95998679 201695.29400144 237947.35392237
 206133.53793285  99446.01705384 115122.09776666 209784.11403448
 167541.59670271 162969.03673934 151277.96604473 189720.09910618
  96447.55044808 168669.905925    83070.12662493 171326.97842054
 176873.7024297  175633.95029986 131412.8139454  204533.41322714
```

```
 233136.38634834 125468.90236908 175066.41285203 258869.95615198
 229111.23932376]
C:\Program Files (x86)\Microsoft Visual Studio\Shared\Anaconda3_64\lib\site-packages\sklearn\linear_model\coordinat
e_descent.py:491: ConvergenceWarning: Objective did not converge. You might want to increase the number of iteratio
ns. Fitting data with very small alpha may cause precision problems.
  ConvergenceWarning)
```

In [68]:

```
lasso_mse=mean_squared_error(y_test, lasso_y_pred)
print("LASSO_MSE:",lasso_mse)
```

```
LASSO_MSE: 1474731226.7767062
```

In [69]:

```
import numpy as np
#RMSE without CD
print("RMSE without CD: ",np.sqrt(mse_In))
#RMSE with CD
print("RMSE with CD: ",np.sqrt(MSE))
#RMSE with CD and Standard Scaling
print("RMSE with CD and SS: ",np.sqrt(mse_In1))
#RMSE with CD and MinMaxScaling
print("RMSE with CD and MnMaxScaling: ",np.sqrt(mmMSE))
#RMSE of SGDRegressor with CD and StandardScaler
print("RMSE of SGDRegressor with CD and StandardScaler: ",np.sqrt(sgd_mse))
#RMSE of Ridgecv with CD and Standard Scaler
print("RMSE of Ridgecv with CD and Standard Scaler: ",np.sqrt(ridge_mse))
#RMSE of LassoCV with CD and StandardScaler
print("RMSE of LassoCV with CD and StandardScaler",np.sqrt(lasso_mse))
```

```
RMSE without CD:  38403.48064430541
RMSE with CD:  38403.48064430541
RMSE with CD and SS:  38403.48064430596
RMSE with CD and MnMaxScaling:  38403.48064430594
RMSE of SGDRegressor with CD and StandardScaler:  39649.40883430895
RMSE of Ridgecv with CD and Standard Scaler:  38379.93554755395
RMSE of LassoCV with CD and StandardScaler 38402.229450602295
```

In [ ]: