Roll no : 225229110

Lab4.House Price Prediction uding LR with Regularization

Step1.Import dataset

In [180]:
```
import pandas as pd
```

In [181]:
```
data=pd.read_csv("Ames_House_Sales_Cropped.csv")
data
```

											...		
8	1Fam	Y	1022.0	752.0	0.0	2	0.0	0.0	0	0	...	7	0.0
9	2fmCon	Y	1077.0	0.0	0.0	2	851.0	0.0	1	0	...	5	0.0
10	1Fam	Y	1040.0	0.0	0.0	3	906.0	0.0	1	0	...	5	0.0
11	1Fam	Y	1182.0	1142.0	0.0	4	998.0	0.0	1	0	...	9	0.0
12	1Fam	Y	912.0	0.0	0.0	2	737.0	0.0	1	0	...	5	0.0
13	1Fam	Y	1494.0	0.0	0.0	3	0.0	0.0	0	0	...	7	0.0
14	1Fam	Y	1253.0	0.0	0.0	2	733.0	0.0	1	0	...	6	0.0
15	1Fam	Y	854.0	0.0	0.0	2	0.0	0.0	0	0	...	7	0.0
16	1Fam	Y	1004.0	0.0	0.0	2	578.0	0.0	1	0	...	6	0.0
17	Duplex	Y	1296.0	0.0	0.0	2	0.0	0.0	0	0	...	4	0.0
18	1Fam	Y	1114.0	0.0	0.0	3	646.0	0.0	1	0	...	5	0.0
19	1Fam	Y	1339.0	0.0	0.0	3	504.0	0.0	0	0	...	5	0.0

In [182]:
```
#head
data.head()
```

Out[182]:

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

5 rows × 39 columns

In [183]:
```
#shape
data.shape
```

Out[183]: (1379, 39)

In [184]:
```
#columns
data.shape[1]
```

Out[184]: 39

In [185]:
```
#dtype
data.dtypes
```

Out[185]:
```
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 [186]:
```
#info
data.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 [187]:
```
#value_counts
data.CentralAir.value_counts()
```

Out[187]:
```
Y    1310
N      69
Name: CentralAir, dtype: int64
```

Step2.Predict Sale Price without Categorical features

In [188]:
```
df=data.drop("BldgType",axis=1)
print(df)
```

```
1367          Y     913.0       0.0        0.0            3       187.0
1368          Y    1188.0       0.0        0.0            3       593.0
1369          Y    1220.0     870.0        0.0            3      1079.0
1370          N     796.0     550.0        0.0            2         0.0
1371          Y    1578.0       0.0        0.0            3         0.0
1372          Y    1072.0       0.0        0.0            2       547.0
1373          Y    1221.0       0.0        0.0            2       410.0
1374          Y     953.0     694.0        0.0            3         0.0
1375          Y    2073.0       0.0        0.0            3       790.0
1376          Y    1188.0    1152.0        0.0            4       275.0
1377          Y    1078.0       0.0        0.0            2        49.0
1378          Y    1256.0       0.0        0.0            3       830.0

      BsmtFinSF2  BsmtFullBath  BsmtHalfBath  BsmtUnfSF    ...      \
0            0.0             1             0      150.0    ...
1            0.0             0             1      284.0    ...
2            0.0             1             0      434.0    ...
3            0.0             1             0      540.0    ...
4            0.0             1             0      490.0    ...
5            0.0             1             0       64.0    ...
```

In [189]:
```
df.drop("CentralAir",axis=1)
```

Out[189]:

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

	1stFlrSF	2ndFlrSF	3SsnPorch	BedroomAbvGr	BsmtFinSF1	BsmtFinSF2	BsmtFullBath	BsmtHalfBath	BsmtUnfSF	EnclosedPorch	...	OverallQual	Poo
1372	1072.0	0.0	0.0	2	547.0	0.0	1	0	0.0	0.0	...	5	
1373	1221.0	0.0	0.0	2	410.0	0.0	1	0	811.0	0.0	...	7	
1374	953.0	694.0	0.0	3	0.0	0.0	0	0	953.0	0.0	...	6	
1375	2073.0	0.0	0.0	3	790.0	163.0	1	0	589.0	0.0	...	6	
1376	1188.0	1152.0	0.0	4	275.0	0.0	0	0	877.0	0.0	...	7	
1377	1078.0	0.0	0.0	2	49.0	1029.0	1	0	0.0	112.0	...	5	
1378	1256.0	0.0	0.0	3	830.0	290.0	1	0	136.0	0.0	...	5	

1379 rows × 37 columns

In [190]:
```
df3=data.pop("BldgType")
print(df3)
```

```
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 [191]:
```
df=data.pop("CentralAir")
df
```

Out[191]:
```
0       Y
1       Y
2       Y
3       Y
4       Y
5       Y
6       Y
7       Y
8       Y
9       Y
10      Y
11      Y
12      Y
13      Y
14      Y
15      Y
16      Y
17      Y
18      Y
19      Y
20      Y
21      Y
22      Y
23      Y
24      Y
25      Y
26      Y
27      Y
28      Y
29      N
       ..
1349    Y
1350    Y
1351    Y
1352    Y
1353    Y
1354    Y
1355    Y
1356    Y
1357    Y
1358    Y
1359    Y
1360    Y
1361    Y
1362    Y
1363    Y
1364    Y
1365    N
1366    Y
1367    Y
1368    Y
1369    Y
1370    N
1371    Y
1372    Y
1373    Y
1374    Y
1375    Y
1376    Y
1377    Y
1378    Y
Name: CentralAir, Length: 1379, dtype: object
```

In [192]:
```
data
```

Out[192]:

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

	1stFlrSF	2ndFlrSF	3SsnPorch	BedroomAbvGr	BsmtFinSF1	BsmtFinSF2	BsmtFullBath	BsmtHalfBath	BsmtUnfSF	EnclosedPorch	...	OverallQual	Poo
1372	1072.0	0.0	0.0	2	547.0	0.0	1	0	0.0	0.0	...	5	
1373	1221.0	0.0	0.0	2	410.0	0.0	1	0	811.0	0.0	...	7	
1374	953.0	694.0	0.0	3	0.0	0.0	0	0	953.0	0.0	...	6	
1375	2073.0	0.0	0.0	3	790.0	163.0	1	0	589.0	0.0	...	6	
1376	1188.0	1152.0	0.0	4	275.0	0.0	0	0	877.0	0.0	...	7	
1377	1078.0	0.0	0.0	2	49.0	1029.0	1	0	0.0	112.0	...	5	
1378	1256.0	0.0	0.0	3	830.0	290.0	1	0	136.0	0.0	...	5	

1379 rows × 37 columns

In [193]:
```
X=df1.drop("SalePrice",axis=1)
print(X)
```

```
      BsmtFullBath  BsmtHalfBath  BsmtUnfSF  EnclosedPorch   ...    \
0                1             0      150.0            0.0   ...
1                0             1      284.0            0.0   ...
2                1             0      434.0            0.0   ...
3                1             0      540.0          272.0   ...
4                1             0      490.0            0.0   ...
5                1             0       64.0            0.0   ...
6                1             0      317.0            0.0   ...
7                1             0      216.0          228.0   ...
8                0             0      952.0          205.0   ...
9                1             0      140.0            0.0   ...
10               1             0      134.0            0.0   ...
11               1             0      177.0            0.0   ...
12               1             0      175.0            0.0   ...
13               0             0     1494.0            0.0   ...
14               1             0      520.0          176.0   ...
15               0             0      832.0            0.0   ...
16               1             0      426.0            0.0   ...
17               0             0        0.0            0.0
```

In [194]:
```
y=data[["SalePrice"]]
y
```

	SalePrice
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

In [195]:
```
from sklearn.model_selection import train_test_split
```

In [196]:
```
X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.25,random_state=42)
```

In [197]:
```
X_train
```

Out[197]:

	1stFlrSF	2ndFlrSF	3SsnPorch	BedroomAbvGr	BsmtFinSF1	BsmtFinSF2	BsmtFullBath	BsmtHalfBath	BsmtUnfSF	EnclosedPorch	...	OverallCond	Ov
1193	1337.0	0.0	0.0	2	266.0	0.0	1	0	1139.0	0.0	...	5	
910	1800.0	0.0	0.0	2	0.0	0.0	0	0	1800.0	0.0	...	5	
1068	1328.0	653.0	0.0	4	622.0	0.0	1	0	500.0	0.0	...	3	
1196	2018.0	0.0	0.0	3	0.0	0.0	0	0	2002.0	0.0	...	5	
1102	959.0	712.0	0.0	3	786.0	0.0	1	0	173.0	0.0	...	5	
447	970.0	0.0	0.0	2	630.0	0.0	1	0	340.0	0.0	...	6	
796	1701.0	0.0	0.0	3	1390.0	0.0	1	0	0.0	0.0	...	5	
543	1320.0	0.0	0.0	3	328.0	551.0	1	0	285.0	240.0	...	6	
901	768.0	0.0	0.0	2	660.0	0.0	0	1	108.0	0.0	...	8	
968	1264.0	0.0	0.0	3	697.0	0.0	1	0	571.0	0.0	...	5	
411	904.0	0.0	0.0	2	0.0	0.0	0	0	884.0	105.0	...	7	
96	1535.0	0.0	0.0	4	0.0	0.0	0	0	0.0	0.0	...	5	
429	624.0	720.0	0.0	4	0.0	0.0	0	0	624.0	96.0	...	5	
361	784.0	0.0	0.0	2	0.0	0.0	0	0	784.0	91.0	...	3	
933	778.0	798.0	0.0	3	0.0	0.0	0	0	770.0	0.0	...	5	
588	1422.0	0.0	0.0	3	0.0	0.0	0	0	978.0	36.0	...	5	
156	854.0	0.0	0.0	2	360.0	0.0	0	0	360.0	0.0	...	6	
528	1389.0	0.0	0.0	2	1071.0	123.0	1	0	195.0	0.0	...	5	
654	616.0	0.0	0.0	2	616.0	0.0	0	0	0.0	129.0	...	7	
857	1636.0	0.0	0.0	3	63.0	0.0	1	0	1560.0	0.0	...	5	
1237	1294.0	0.0	0.0	3	1200.0	0.0	1	0	78.0	0.0	...	5	
534	1496.0	636.0	0.0	1	1441.0	0.0	1	0	55.0	0.0	...	8	
1078	1466.0	1362.0	0.0	4	1150.0	0.0	1	0	316.0	0.0	...	5	
1190	1050.0	0.0	0.0	2	504.0	0.0	0	0	546.0	0.0	...	6	
1312	869.0	349.0	0.0	3	375.0	0.0	0	1	360.0	0.0	...	6	
371	1144.0	0.0	0.0	3	739.0	0.0	1	0	405.0	0.0	...	6	
677	848.0	0.0	0.0	1	662.0	0.0	1	0	186.0	0.0	...	5	
308	596.0	596.0	0.0	3	0.0	0.0	0	0	596.0	137.0	...	5	
1371	1578.0	0.0	0.0	3	0.0	0.0	0	0	1573.0	0.0	...	5	
710	866.0	902.0	0.0	3	20.0	0.0	0	0	846.0	0.0	...	5	
...	...	...	...	...	...	...	...	...	...	...	...	...	...
474	1801.0	0.0	0.0	1	1247.0	0.0	1	0	254.0	0.0	...	8	
856	1063.0	0.0	0.0	3	354.0	290.0	1	0	412.0	164.0	...	5	
747	1086.0	809.0	0.0	3	0.0	0.0	0	0	712.0	0.0	...	5	
252	1095.0	844.0	0.0	3	0.0	0.0	0	0	1095.0	0.0	...	5	
21	1108.0	0.0	0.0	3	0.0	0.0	0	0	637.0	205.0	...	7	
1337	1569.0	0.0	0.0	1	988.0	0.0	0	1	398.0	0.0	...	5	
459	1484.0	0.0	0.0	3	998.0	0.0	0	0	486.0	0.0	...	6	
1184	760.0	896.0	0.0	3	0.0	0.0	0	0	746.0	0.0	...	5	
276	910.0	648.0	0.0	4	420.0	0.0	0	0	490.0	0.0	...	5	
955	1022.0	0.0	0.0	2	247.0	465.0	1	0	310.0	226.0	...	4	
1215	1582.0	0.0	0.0	4	812.0	0.0	0	1	812.0	0.0	...	5	
385	1050.0	1028.0	0.0	3	789.0	0.0	1	0	245.0	0.0	...	5	
805	1779.0	0.0	0.0	3	306.0	1085.0	1	0	372.0	0.0	...	4	
343	790.0	784.0	0.0	3	712.0	0.0	1	0	84.0	0.0	...	5	
769	1008.0	0.0	0.0	2	486.0	0.0	0	0	522.0	120.0	...	6	
1332	944.0	896.0	0.0	3	666.0	0.0	1	0	278.0	0.0	...	5	
130	928.0	836.0	0.0	3	821.0	0.0	1	0	107.0	0.0	...	5	
871	936.0	785.0	0.0	3	814.0	0.0	0	1	114.0	0.0	...	5	
1123	1622.0	0.0	0.0	3	1159.0	0.0	1	0	90.0	0.0	...	4	
87	964.0	0.0	0.0	2	713.0	0.0	1	0	163.0	44.0	...	7	
330	1453.0	0.0	0.0	2	1082.0	0.0	1	0	371.0	0.0	...	5	
1238	1902.0	0.0	0.0	3	1406.0	0.0	1	0	496.0	162.0	...	8	
466	886.0	0.0	0.0	2	0.0	0.0	0	0	190.0	80.0	...	8	

	1stFlrSF	2ndFlrSF	3SsnPorch	BedroomAbvGr	BsmtFinSF1	BsmtFinSF2	BsmtFullBath	BsmtHalfBath	BsmtUnfSF	EnclosedPorch	...	OverallCond	Ov
121	1216.0	941.0	0.0	4	445.0	0.0	0	0	479.0	0.0	...	6	
1044	1500.0	1122.0	0.0	3	1032.0	0.0	1	0	431.0	0.0	...	5	
1095	855.0	601.0	0.0	3	311.0	0.0	0	0	544.0	0.0	...	5	
1130	815.0	875.0	0.0	3	0.0	0.0	0	0	815.0	330.0	...	6	
1294	1661.0	0.0	0.0	3	831.0	0.0	1	0	161.0	0.0	...	6	
860	742.0	742.0	0.0	3	0.0	0.0	0	0	742.0	0.0	...	5	
1126	1224.0	0.0	0.0	2	883.0	0.0	1	0	341.0	0.0	...	5	

1034 rows × 36 columns

In [198]:
```
X_test
```

Out[198]:

	1stFlrSF	2ndFlrSF	3SsnPorch	BedroomAbvGr	BsmtFinSF1	BsmtFinSF2	BsmtFullBath	BsmtHalfBath	BsmtUnfSF	EnclosedPorch	...	OverallCond	Ov
599	1518.0	0.0	0.0	1	1218.0	0.0	0	0	300.0	0.0	...	5	
881	925.0	0.0	0.0	2	338.0	466.0	0	1	121.0	0.0	...	6	
634	1095.0	679.0	0.0	4	0.0	0.0	1	0	1095.0	90.0	...	2	
425	888.0	868.0	0.0	3	742.0	0.0	1	0	130.0	0.0	...	5	
906	1337.0	0.0	0.0	3	699.0	0.0	1	0	638.0	0.0	...	5	
1079	672.0	252.0	0.0	2	348.0	0.0	1	0	324.0	0.0	...	4	
65	1479.0	0.0	0.0	3	1013.0	0.0	1	0	440.0	0.0	...	5	
1351	1440.0	0.0	0.0	3	678.0	0.0	0	0	762.0	99.0	...	7	
479	689.0	689.0	0.0	3	141.0	0.0	0	0	548.0	116.0	...	9	
67	1304.0	983.0	0.0	3	603.0	0.0	0	0	701.0	114.0	...	4	
939	774.0	456.0	0.0	3	384.0	0.0	1	0	363.0	0.0	...	7	
573	1940.0	1254.0	0.0	4	428.0	0.0	0	0	537.0	0.0	...	6	
917	918.0	0.0	0.0	2	0.0	0.0	0	0	918.0	0.0	...	5	
1054	1734.0	0.0	0.0	3	1004.0	0.0	1	0	730.0	0.0	...	5	
941	1442.0	0.0	0.0	2	0.0	0.0	0	0	1442.0	0.0	...	6	
1116	1130.0	0.0	0.0	2	821.0	0.0	1	0	299.0	0.0	...	6	
237	1005.0	1286.0	0.0	4	0.0	0.0	0	0	975.0	0.0	...	5	
578	1054.0	0.0	0.0	3	763.0	0.0	1	0	173.0	0.0	...	7	
772	1358.0	0.0	0.0	2	733.0	0.0	1	0	625.0	0.0	...	5	
303	1898.0	1080.0	0.0	5	0.0	0.0	0	0	710.0	0.0	...	7	
953	720.0	551.0	0.0	4	0.0	0.0	0	0	720.0	108.0	...	5	
783	520.0	600.0	0.0	2	0.0	0.0	0	0	600.0	0.0	...	5	
339	912.0	0.0	0.0	2	773.0	0.0	1	0	115.0	0.0	...	6	
718	1494.0	0.0	0.0	2	437.0	1057.0	1	0	0.0	0.0	...	5	
208	2392.0	0.0	0.0	3	56.0	0.0	0	0	2336.0	0.0	...	5	
624	1465.0	915.0	0.0	3	187.0	723.0	0	0	111.0	0.0	...	5	
1075	1167.0	0.0	0.0	3	645.0	0.0	0	0	270.0	216.0	...	5	
1040	950.0	0.0	0.0	3	412.0	287.0	0	0	251.0	0.0	...	5	
575	1476.0	677.0	0.0	3	904.0	0.0	1	0	536.0	0.0	...	5	
244	1212.0	0.0	0.0	3	506.0	0.0	1	0	0.0	0.0	...	7	
...	...	...	...	...	...	...	...	...	...	...	...	...	
1334	1040.0	0.0	0.0	2	0.0	0.0	0	0	0.0	0.0	...	5	
199	1236.0	0.0	0.0	2	360.0	0.0	0	1	710.0	0.0	...	6	
367	961.0	406.0	0.0	4	241.0	391.0	1	0	229.0	112.0	...	8	
723	1690.0	1589.0	0.0	4	1416.0	0.0	1	0	234.0	0.0	...	5	
354	914.0	0.0	0.0	2	298.0	0.0	0	0	572.0	0.0	...	5	
10	1040.0	0.0	0.0	3	906.0	0.0	1	0	134.0	0.0	...	5	
147	1392.0	1070.0	168.0	4	57.0	0.0	1	0	1335.0	0.0	...	5	
538	846.0	846.0	0.0	3	0.0	0.0	0	0	846.0	0.0	...	5	
282	1541.0	0.0	0.0	3	0.0	0.0	0	0	1541.0	0.0	...	5	
298	1472.0	0.0	0.0	3	1036.0	0.0	1	0	336.0	0.0	...	5	
522	1048.0	0.0	0.0	2	0.0	0.0	0	0	993.0	116.0	...	6	
291	793.0	325.0	0.0	3	507.0	0.0	1	0	286.0	0.0	...	7	
503	880.0	844.0	0.0	3	0.0	0.0	0	0	880.0	0.0	...	5	
903	979.0	979.0	0.0	4	484.0	0.0	0	0	495.0	0.0	...	6	
930	1001.0	634.0	0.0	2	0.0	0.0	0	0	485.0	0.0	...	8	
439	888.0	756.0	0.0	3	386.0	0.0	0	0	342.0	0.0	...	7	
1033	1200.0	0.0	0.0	1	661.0	0.0	1	0	203.0	0.0	...	8	
331	616.0	495.0	0.0	3	236.0	380.0	0	1	0.0	0.0	...	6	
527	1392.0	0.0	0.0	3	1302.0	0.0	1	0	90.0	0.0	...	6	
462	616.0	688.0	0.0	3	0.0	0.0	0	0	264.0	0.0	...	6	
861	1105.0	1169.0	0.0	5	443.0	0.0	0	0	662.0	0.0	...	5	
630	1208.0	0.0	0.0	3	767.0	0.0	1	0	441.0	0.0	...	6	
135	970.0	739.0	0.0	3	0.0	0.0	0	0	970.0	0.0	...	5	

	1stFlrSF	2ndFlrSF	3SsnPorch	BedroomAbvGr	BsmtFinSF1	BsmtFinSF2	BsmtFullBath	BsmtHalfBath	BsmtUnfSF	EnclosedPorch	...	OverallCond	Ov
358	1026.0	665.0	0.0	3	218.0	0.0	0	0	808.0	242.0	...	6	
363	1269.0	0.0	0.0	2	24.0	0.0	0	0	1232.0	0.0	...	5	
618	1142.0	793.0	0.0	3	0.0	0.0	0	0	793.0	252.0	...	7	
561	684.0	684.0	0.0	3	0.0	0.0	0	0	684.0	0.0	...	7	
529	1163.0	511.0	0.0	4	0.0	0.0	0	0	1163.0	0.0	...	7	
567	927.0	988.0	0.0	3	789.0	0.0	1	0	119.0	0.0	...	5	
158	1064.0	703.0	0.0	2	495.0	215.0	1	0	354.0	0.0	...	7	

345 rows × 36 columns

In [199]:
```
from sklearn.linear_model import LinearRegression
lr=LinearRegression()
lr.fit(X_train,y_train)
```

Out[199]:
```
LinearRegression(copy_X=True, fit_intercept=True, n_jobs=1, normalize=False)
```

In [200]:
```
y_pred=lr.predict(X_test)
y_pred
```

```
[ 93329.85601474],
[153784.14868375],
[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],
```

In [201]:
```
from sklearn.metrics import mean_squared_error
```

In [202]:
```
mse_In=mean_squared_error(y_test,y_pred)
mse_In
```

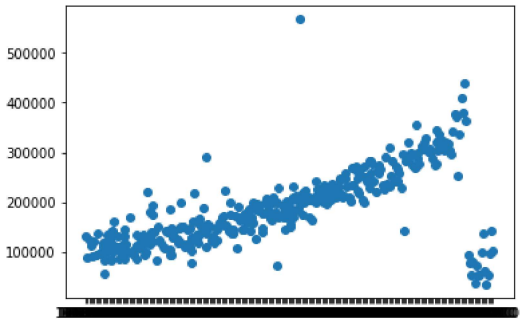
Out[202]:
```
1474827325.5975406
```

Step3.Create Scatter Plot

In [203]:
```
import matplotlib.pyplot as plt
```

In [204]:
```
plt.scatter(y_test,y_pred)
```

Out[204]:
```
<matplotlib.collections.PathCollection at 0x2a83441b5c0>
```



Step4.Encode Categorical columns

In [205]:
```
gd=pd.get_dummies(data)
gd
```

Out[205]:

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

	1stFlrSF	2ndFlrSF	3SsnPorch	BedroomAbvGr	BsmtFinSF1	BsmtFinSF2	BsmtFullBath	BsmtHalfBath	BsmtUnfSF	EnclosedPorch	...	OverallQual	Poo
1372	1072.0	0.0	0.0	2	547.0	0.0	1	0	0.0	0.0	...	5	
1373	1221.0	0.0	0.0	2	410.0	0.0	1	0	811.0	0.0	...	7	
1374	953.0	694.0	0.0	3	0.0	0.0	0	0	953.0	0.0	...	6	
1375	2073.0	0.0	0.0	3	790.0	163.0	1	0	589.0	0.0	...	6	
1376	1188.0	1152.0	0.0	4	275.0	0.0	0	0	877.0	0.0	...	7	
1377	1078.0	0.0	0.0	2	49.0	1029.0	1	0	0.0	112.0	...	5	
1378	1256.0	0.0	0.0	3	830.0	290.0	1	0	136.0	0.0	...	5	

1379 rows × 37 columns

Step5.Predict Sale Price with Categorical features

In [206]:
```
x=gd.drop("SalePrice",axis=1)
x
```

Out[206]:

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

	1stFlrSF	2ndFlrSF	3SsnPorch	BedroomAbvGr	BsmtFinSF1	BsmtFinSF2	BsmtFullBath	BsmtHalfBath	BsmtUnfSF	EnclosedPorch	...	OverallCond	Ov
1372	1072.0	0.0	0.0	2	547.0	0.0	1	0	0.0	0.0	...	5	
1373	1221.0	0.0	0.0	2	410.0	0.0	1	0	811.0	0.0	...	5	
1374	953.0	694.0	0.0	3	0.0	0.0	0	0	953.0	0.0	...	5	
1375	2073.0	0.0	0.0	3	790.0	163.0	1	0	589.0	0.0	...	6	
1376	1188.0	1152.0	0.0	4	275.0	0.0	0	0	877.0	0.0	...	9	
1377	1078.0	0.0	0.0	2	49.0	1029.0	1	0	0.0	112.0	...	6	
1378	1256.0	0.0	0.0	3	830.0	290.0	1	0	136.0	0.0	...	6	

1379 rows × 36 columns

In [207]:
```
Y=gd.pop('SalePrice')
Y
```

Out[207]:
```
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 [208]:
```
from sklearn.model_selection import train_test_split
```

In [209]:
```
X_train,X_test,y_train,y_test=train_test_split(x,Y,test_size=0.25,random_state=42)
```

In [210]:
```
X_train
```

Out[210]:

	1stFlrSF	2ndFlrSF	3SsnPorch	BedroomAbvGr	BsmtFinSF1	BsmtFinSF2	BsmtFullBath	BsmtHalfBath	BsmtUnfSF	EnclosedPorch	...	OverallCond	Ov
1193	1337.0	0.0	0.0	2	266.0	0.0	1	0	1139.0	0.0	...	5	
910	1800.0	0.0	0.0	2	0.0	0.0	0	0	1800.0	0.0	...	5	
1068	1328.0	653.0	0.0	4	622.0	0.0	1	0	500.0	0.0	...	3	
1196	2018.0	0.0	0.0	3	0.0	0.0	0	0	2002.0	0.0	...	5	
1102	959.0	712.0	0.0	3	786.0	0.0	1	0	173.0	0.0	...	5	
447	970.0	0.0	0.0	2	630.0	0.0	1	0	340.0	0.0	...	6	
796	1701.0	0.0	0.0	3	1390.0	0.0	1	0	0.0	0.0	...	5	
543	1320.0	0.0	0.0	3	328.0	551.0	1	0	285.0	240.0	...	6	
901	768.0	0.0	0.0	2	660.0	0.0	0	1	108.0	0.0	...	8	
968	1264.0	0.0	0.0	3	697.0	0.0	1	0	571.0	0.0	...	5	
411	904.0	0.0	0.0	2	0.0	0.0	0	0	884.0	105.0	...	7	
96	1535.0	0.0	0.0	4	0.0	0.0	0	0	0.0	0.0	...	5	
429	624.0	720.0	0.0	4	0.0	0.0	0	0	624.0	96.0	...	5	
361	784.0	0.0	0.0	2	0.0	0.0	0	0	784.0	91.0	...	3	
933	778.0	798.0	0.0	3	0.0	0.0	0	0	770.0	0.0	...	5	
588	1422.0	0.0	0.0	3	0.0	0.0	0	0	978.0	36.0	...	5	
156	854.0	0.0	0.0	2	360.0	0.0	0	0	360.0	0.0	...	6	
528	1389.0	0.0	0.0	2	1071.0	123.0	1	0	195.0	0.0	...	5	
654	616.0	0.0	0.0	2	616.0	0.0	0	0	0.0	129.0	...	7	
857	1636.0	0.0	0.0	3	63.0	0.0	1	0	1560.0	0.0	...	5	
1237	1294.0	0.0	0.0	3	1200.0	0.0	1	0	78.0	0.0	...	5	
534	1496.0	636.0	0.0	1	1441.0	0.0	1	0	55.0	0.0	...	8	
1078	1466.0	1362.0	0.0	4	1150.0	0.0	1	0	316.0	0.0	...	5	
1190	1050.0	0.0	0.0	2	504.0	0.0	0	0	546.0	0.0	...	6	
1312	869.0	349.0	0.0	3	375.0	0.0	0	1	360.0	0.0	...	6	
371	1144.0	0.0	0.0	3	739.0	0.0	1	0	405.0	0.0	...	6	
677	848.0	0.0	0.0	1	662.0	0.0	1	0	186.0	0.0	...	5	
308	596.0	596.0	0.0	3	0.0	0.0	0	0	596.0	137.0	...	5	
1371	1578.0	0.0	0.0	3	0.0	0.0	0	0	1573.0	0.0	...	5	
710	866.0	902.0	0.0	3	20.0	0.0	0	0	846.0	0.0	...	5	
...	...	...	...	...	...	...	...	...	...	...	...	...	...
474	1801.0	0.0	0.0	1	1247.0	0.0	1	0	254.0	0.0	...	8	
856	1063.0	0.0	0.0	3	354.0	290.0	1	0	412.0	164.0	...	5	
747	1086.0	809.0	0.0	3	0.0	0.0	0	0	712.0	0.0	...	5	
252	1095.0	844.0	0.0	3	0.0	0.0	0	0	1095.0	0.0	...	5	
21	1108.0	0.0	0.0	3	0.0	0.0	0	0	637.0	205.0	...	7	
1337	1569.0	0.0	0.0	1	988.0	0.0	0	1	398.0	0.0	...	5	
459	1484.0	0.0	0.0	3	998.0	0.0	0	0	486.0	0.0	...	6	
1184	760.0	896.0	0.0	3	0.0	0.0	0	0	746.0	0.0	...	5	
276	910.0	648.0	0.0	4	420.0	0.0	0	0	490.0	0.0	...	5	
955	1022.0	0.0	0.0	2	247.0	465.0	1	0	310.0	226.0	...	4	
1215	1582.0	0.0	0.0	4	812.0	0.0	0	1	812.0	0.0	...	5	
385	1050.0	1028.0	0.0	3	789.0	0.0	1	0	245.0	0.0	...	5	
805	1779.0	0.0	0.0	3	306.0	1085.0	1	0	372.0	0.0	...	4	
343	790.0	784.0	0.0	3	712.0	0.0	1	0	84.0	0.0	...	5	
769	1008.0	0.0	0.0	2	486.0	0.0	0	0	522.0	120.0	...	6	
1332	944.0	896.0	0.0	3	666.0	0.0	1	0	278.0	0.0	...	5	
130	928.0	836.0	0.0	3	821.0	0.0	1	0	107.0	0.0	...	5	
871	936.0	785.0	0.0	3	814.0	0.0	0	1	114.0	0.0	...	5	
1123	1622.0	0.0	0.0	3	1159.0	0.0	1	0	90.0	0.0	...	4	
87	964.0	0.0	0.0	2	713.0	0.0	1	0	163.0	44.0	...	7	
330	1453.0	0.0	0.0	2	1082.0	0.0	1	0	371.0	0.0	...	5	
1238	1902.0	0.0	0.0	3	1406.0	0.0	1	0	496.0	162.0	...	8	
466	886.0	0.0	0.0	2	0.0	0.0	0	0	190.0	80.0	...	8	

	1stFlrSF	2ndFlrSF	3SsnPorch	BedroomAbvGr	BsmtFinSF1	BsmtFinSF2	BsmtFullBath	BsmtHalfBath	BsmtUnfSF	EnclosedPorch	...	OverallCond	Ov
121	1216.0	941.0	0.0	4	445.0	0.0	0	0	479.0	0.0	...	6	
1044	1500.0	1122.0	0.0	3	1032.0	0.0	1	0	431.0	0.0	...	5	
1095	855.0	601.0	0.0	3	311.0	0.0	0	0	544.0	0.0	...	5	
1130	815.0	875.0	0.0	3	0.0	0.0	0	0	815.0	330.0	...	6	
1294	1661.0	0.0	0.0	3	831.0	0.0	1	0	161.0	0.0	...	6	
860	742.0	742.0	0.0	3	0.0	0.0	0	0	742.0	0.0	...	5	
1126	1224.0	0.0	0.0	2	883.0	0.0	1	0	341.0	0.0	...	5	

1034 rows × 36 columns

In [211]:
```
from sklearn.linear_model import LinearRegression
lrr=LinearRegression()
lrr.fit(X_train,y_train)
```

Out[211]:
```
LinearRegression(copy_X=True, fit_intercept=True, n_jobs=1, normalize=False)
```

In [212]:
```
y_pred=lrr.predict(X_test)
y_pred
```

Out[212]:
```
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,
        93329.85601474, 153784.14868375, 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,
       119489.76459232, 126635.29232505,  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, 162030.92949963, 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, 133283.94785764, 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,
       231571.13532009, 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,
       199375.67314461, 272915.38777822, 119902.62390774, 197573.81604289,
       231881.98496935, 119555.98822715, 128709.5470232 , 289158.80585494,
       200733.61408622, 364228.39366113, 119308.94568436, 124086.7225403 ,
       225161.62739518, 171005.22659817,  93631.91767228, 178506.48207587,
       215661.64584809, 226225.30038962, 132391.85608602,  73443.41948308,
       242913.00968351, 141042.39030912, 114633.38186071,  89093.07024776,
       208503.15479642, 172828.98634139, 270797.22136173, 253693.44924276,
       103518.85057198, 220521.27460707, 185034.32129269, 169518.90296985,
       154846.17459061, 224605.38839967,  85972.23828462, 151323.12661355,
       180628.92554331, 200106.98029834, 179499.8172794 , 223755.69902625,
        73639.52734133,  86011.8149164 , 110000.03909838, 231518.89807261,
       153369.35976787, 320899.35432162, 194412.83170324,  57741.65743007,
       203877.63015232, 109803.05085704, 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,
```

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

In [213]:
```
from sklearn.metrics import mean_squared_error
```

In [214]:
```
MSE=mean_squared_error(y_test,y_pred)
MSE
```

Out[214]:
```
1474827325.5975406
```

Step6.Normalize using StandardScaler and Predict Sale Price

In [215]:
```
from sklearn.preprocessing import StandardScaler
```

In [216]:
```
scaler=StandardScaler()
```

In [217]:
```
scaled_X_train=scaler.fit_transform(X_train)
scaled_X_train
```

Out[217]:
```
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 [218]:
```
scaled_X_train.shape
```

Out[218]:
```
(1034, 36)
```

In [219]:
```
scaled_X_test=scaler.transform(X_test)
scaled_X_test
```

Out[219]:
```
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 [220]:
```
from sklearn.metrics import mean_squared_error
```

In [221]:
```
Scaler=StandardScaler()
scaled_X_train=Scaler.fit_transform(X_train)
scaled_X_train
```

Out[221]:
```
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 [222]:
```
scaled_X_test=Scaler.transform(X_test)
scaled_X_test
```

Out[222]:
```
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 [223]:
```
model1=LinearRegression()
model1.fit(scaled_X_train,y_train)
sy_pred=model1.predict(scaled_X_test)
sy_pred
```

Out[223]:
```
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,
        93329.85601463, 153784.1486838 , 163237.87451135, 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,
       146941.7171239 , 283693.48760665, 106694.19786715, 221733.02172498,
        87733.4701401 , 295005.66546303, 125250.6201929 ,  53569.11256027,
       133831.05086115, 170901.42929711, 213273.17757298, 110818.14723239,
       103893.81849705, 200665.21921041, 137764.84464906, 311840.64900749,
        53035.52267581, 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, 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, 134354.12384256, 223860.01643707,
       138360.76267459, 210765.75439351, 145423.97229839, 291155.26974382,
       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, 124086.72254034,
       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,
```

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

In [224]:
```
mse_In1=mean_squared_error(y_test,sy_pred)
mse_In1
print("mean square error using standard scalar:",mse_In1)
```

```
mean square error using standard scalar: 1474827325.5975823
```

Step7.Normalize using MinMaxScaler and Predict Sale Price

In [225]:
```
from sklearn.preprocessing import MinMaxScaler
mm_scaler=MinMaxScaler()
```

In [226]:
```
mmX_train=mm_scaler.fit_transform(X_train)
mmX_train
```

Out[226]:
```
array([[0.21133051, 0.        , 0.        , ..., 0.90769231, 0.81666667,
        0.5       ],
       [0.32016925, 0.        , 0.        , ..., 0.97692308, 0.95      ,
        0.75      ],
       [0.20921486, 0.31622276, 0.        , ..., 0.36923077, 0.        ,
        0.75      ],
       ...,
       [0.28749412, 0.        , 0.        , ..., 0.57692308, 0.76666667,
        0.5       ],
       [0.07146215, 0.35932203, 0.        , ..., 0.96153846, 0.91666667,
        0.75      ],
       [0.18476728, 0.        , 0.        , ..., 0.91538462, 0.81666667,
        0.75      ]])
```

In [227]:
```
mmX_test=mm_scaler.transform(X_test)
mmX_test
```

Out[227]:
```
array([[0.2538787 , 0.        , 0.        , ..., 0.94615385, 0.9       ,
        0.5       ],
       [0.11448049, 0.        , 0.        , ..., 0.65384615, 0.25      ,
        0.75      ],
       [0.15444288, 0.32881356, 0.        , ..., 0.15384615, 0.        ,
        0.        ],
       ...,
       [0.17042783, 0.24745763, 0.        , ..., 0.29230769, 0.        ,
        0.5       ],
       [0.11495063, 0.47845036, 0.        , ..., 0.86153846, 0.7       ,
        0.        ],
       [0.14715562, 0.34043584, 0.        , ..., 0.82307692, 0.61666667,
        0.        ]])
```

In [228]:
```
model3=LinearRegression()
model3.fit(mmX_train,y_train)
```

Out[228]:
```
LinearRegression(copy_X=True, fit_intercept=True, n_jobs=1, normalize=False)
```

In [229]:
```
mms_y_pred=model3.predict(mmX_test)
mms_y_pred
```

Out[229]:
```
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,
        93329.85601463, 153784.1486838 , 163237.87451135, 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,
       146941.7171239 , 283693.48760665, 106694.19786715, 221733.02172498,
        87733.4701401 , 295005.66546303, 125250.6201929 ,  53569.11256027,
       133831.05086115, 170901.42929711, 213273.17757298, 110818.14723239,
       103893.81849705, 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,
```

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

In [230]:
```
mmMSE=mean_squared_error(y_test,mms_y_pred)
```

In [231]:
```
print("Mean sqaure error using MinMaxScaler:",mmMSE)
```

```
Mean sqaure error using MinMaxScaler: 1474827325.5975811
```

Step8.Predict using Ridge and SGD Regression

In [239]:
```
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: [250851.08701713 113046.15345709 110215.57231865 200690.21553301
 208255.36365841  47034.73843394 233831.98270146 203942.89751742
 194157.43704549 224848.84415329 104786.90478342 285927.60031221
 112247.39033602 284506.85909405 207296.47131102 155213.02630626
 238672.53242113 153962.43406735 211228.46968196 261423.72279589
 102083.28516987 159508.27265748 169538.87624881 244375.06010953
 357650.91616614 210668.96522354 128385.35232838 103249.78111936
 312982.4591815  175061.28026342 240682.60998631 200821.70956329
 147153.21088862 147074.20825085 209860.62512088 361489.66473804
 124028.33879649 145942.81316447 261494.13296702 210485.55011114
 172223.15038761 129287.01359267 190049.40028832 366128.94264503
 154790.3800747  280102.19680578 110626.22891403 215574.05135203
  91387.12921027 292956.23675168 129378.29586996  58512.92146235
 147302.83961466 170103.96783047 208611.35774562 120608.36551101
 110598.49859265 197978.31856349 143974.35357487 303499.19716547
  55792.63348257 186101.28978829 156979.00619636 285569.92561712
  88046.32972286 316345.2947259  167366.85389997 120377.04830319
 200130.20238792 208966.77505062 136069.7255166  268811.26584678
 129880.91612209 132967.66660034  80465.8903749  137372.56289361
  44326.47550692 170022.01948623 267422.33019354 121571.02025789
 317872.21119126 213457.14156838 284491.83489067 208188.5658527
 104589.08681713 254259.40900486 142080.207281   114679.12174299
 138771.34892194 279943.95166722 264542.67429881 287029.7359947
 181642.84115515 163919.30623042 156782.17218052 212155.13636597
 243886.96738247 216779.633019   271146.21510057 239350.94874882
 114853.2412123  105878.50295633 194687.61629086 204326.44333928
 155142.24170734 211848.77363691 214405.69140606 133673.21024797
 177332.33942174 226588.61585729 160694.13400918 115570.62170025
 310936.62379005 284274.70825307 320061.67925837  94696.046271
 150924.18558875 220288.84251664 128640.1859062  202629.943196
 218200.2209175  118158.78250045 132032.03214889 218754.26448825
 141270.17951156 209133.38881404 146056.69851842 285455.12082552
 180738.92546607 101071.55452799 310780.42733594 169374.92026214
 126888.52337069 138261.50936761 293510.68119291 145830.26697951
 275828.38379144 136348.70989642  96988.89570827 167741.12827323
 149815.22656343 166610.29401401 179073.4604493  246900.41121939
 126200.39958332 142955.01301409 231462.85665889 306077.22825801
 110960.46012431 195376.74523845 170904.51359999 149589.47488674
  96793.03093433 247305.30162233 301858.59424996  58470.47056675
 274701.74202198  99891.46462489 145400.82483068 170138.67703668
 212144.89527954 189688.88783278 180060.78829098 250137.04470811
 157893.77079671 135786.93208563 172092.81054341 322348.4373301
 177172.62907044 101467.71004957 128655.54678455 116644.14179487
 137300.92472209 188613.36402934 154389.34085197 262371.97406526
 208622.63232356 156261.96436979 115702.98827925 239987.23007589
 130205.14232415 262167.65778449 299207.69551207  93631.80238483
 143812.42373437 147260.99307113 163858.03328841 229811.39827074
 229155.50182926 145113.73390798 332648.82143072 130508.017802
 231574.34262925 119983.96267855 106500.12535036 168175.92844473
 149668.44573733 392875.07989813 328598.44653042 285366.70587729
 279372.54190014 271810.43323098 306855.13295791 296116.34997407
 203371.65069488 140470.43179008 160529.00474037 210736.52969394
 207209.28551848 147602.14042871 111316.76043823 161285.20930472
 109156.41329861 289430.46676731 335434.9364487  252977.85402853
 154075.15442073 198118.83308252 133422.89679149 260348.87335848
 138776.50709071 137913.3293974  301154.41307913 174043.83617789
 175488.36260327 514438.52637105 185566.93886465 147067.38826808
 194694.40296533 117018.58512198 178628.65636224 319560.00822801
 144699.83290919 199587.86153301 128118.42668569  68106.31436715
 197171.74587675 264074.55286023 124186.09050836 195989.55037768
 235074.02511108 122527.754431   133908.798196   273588.63869008
 206487.11255228 354315.22637195 122153.54673697 131217.51102153
 219384.95779473 174230.80708777  90791.36866263 180106.06094426
 214215.31947767 222484.26050932 141160.05421634  84739.65963994
 241383.14013834 136253.74122713 123577.63933239  97108.53691404
 209659.54305957 166997.7954833  272209.03224303 245662.56819911
 108992.12672892 220370.63307283 192757.17716878 169588.48264611
 161519.6287375  224453.25831433  91784.83682114 147818.66273271
 176408.10853678 198188.73533599 180300.47915672 215136.68965183
  28166.47470073  87168.40624526 116267.31356404 224008.35965364
 159548.28831737 303915.06645982 195997.88771663  64205.57781099
 207200.03099167 116344.76631827 195391.16665352 108370.69429338
 229711.97394503 208230.28607324 203300.59127753 112307.69302091
 101480.66718546 234651.71796636 178810.17637122 192263.38093019
 260863.18861466 238577.82545967 114247.79571307 158203.11788476
 261235.26266404 120018.84887211 214334.98726304 110589.04301375
 228812.33833055 159172.44159021 119745.22666805 215120.62661922
  91540.09096617  92658.23396281 101173.49068194 298492.5273976
  70175.60494488 187866.82349679 243849.19006854  68821.63500256
 145323.57229819 119357.88695377 420387.35497599 124631.63993908
 120143.45558428 267021.09989005 197973.50867011 238467.96943022
 210208.41791904 107050.33702569 122827.66352525 207193.6237502
 159442.62093416 163117.12613184 152691.33855964 193423.73169172
  98312.30708896 170324.771287    83724.66561554 159883.02163719
 181373.67130207 169625.21294556 132549.3443558  208320.59720371
```

```
 227792.9349017  129907.80790589 180653.05812503 257177.14200734
 224898.19144258]
```

```
C:\Program Files (x86)\Microsoft Visual Studio\Shared\Anaconda3_64\lib\site-packages\sklearn\linear_model\stochastic_gradient.p
y:128: FutureWarning: max_iter and tol parameters have been added in <class 'sklearn.linear_model.stochastic_gradient.SGDRegres
sor'> in 0.19. If both are left unset, they default to max_iter=5 and tol=None. If tol is not None, max_iter defaults to max_it
er=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 [240]:
```
sgd_mse=mean_squared_error(y_test, sgd_y_pred)
print("SGD_MSE:",sgd_mse)
```

```
SGD_MSE: 1487957680.751117
```

In [241]:
```
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 [242]:
```
ridge_mse=mean_squared_error(y_test, ridge_y_pred)
print("RIDGE_MSE:",ridge_mse)
```

```
RIDGE_MSE: 1473019452.6343954
```

In [ ]:
```
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)
```

In [243]:
```
lasso_mse=mean_squared_error(y_test, lasso_y_pred)
print("LASSO_MSE:",lasso_mse)
```

```
LASSO_MSE: 1474731226.7767062
```

Step9.RMSE

In [246]:
```
import numpy as np
#RMSE without CD
print("RMSE without one hot encoding: ",np.sqrt(mse_In))
#RMSE with CD
print("RMSE with one hot encoding: ",np.sqrt(MSE))
#RMSE with CD and Standard Scaling
print("RMSE with OHE and SS: ",np.sqrt(mse_In1))
#RMSE with CD and MinMaxScaling
print("RMSE with OHE and MinMaxScaling: ",np.sqrt(mmMSE))
#RMSE of SGDRegressor with CD and StandardScaler
print("RMSE of SGDRegressor with OHE and StandardScaler: ",np.sqrt(sgd_mse))
#RMSE of Ridgecv with CD and Standard Scaler
print("RMSE of Ridgecv with OHE and Standard Scaler: ",np.sqrt(ridge_mse))
#RMSE of LassoCV with CD and StandardScaler
print("RMSE of LassoCV with OHE and StandardScaler",np.sqrt(lasso_mse))
```

```
RMSE without one hot encoding:  38403.48064430541
RMSE with one hot encoding:  38403.48064430541
RMSE with OHE and SS:  38403.48064430596
RMSE with OHE and MinMaxScaling:  38403.48064430594
RMSE of SGDRegressor with OHE and StandardScaler:  38574.054502361
RMSE of Ridgecv with OHE and Standard Scaler:  38379.93554755395
RMSE of LassoCV with OHE and StandardScaler 38402.229450602295
```