PML_lab-4_205229118_Mahalakshmi.S

March 25, 2021

0.0.1 Lab4. House Price Prediction using LR with Regularization

https://www.dataquest.io/blog/kaggle-getting-started/

0.0.2 Step1. [Import dataset]. Using Pandas, import "Ames_House_Sales_Cropped.csv" file and print properties such as head, shape, columns, dtype, info and value_counts.

[31]:
```
import pandas as pd
import csv
```

[32]:
```
house=pd.read_csv("Ames_House_Sales_Cropped.csv")
house
```

[32]:

	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	
…	…	…	…	…	…	…	
1374	1Fam	Y	953.0	694.0	0.0	3	
1375	1Fam	Y	2073.0	0.0	0.0	3	
1376	1Fam	Y	1188.0	1152.0	0.0	4	
1377	1Fam	Y	1078.0	0.0	0.0	2	
1378	1Fam	Y	1256.0	0.0	0.0	3	

	BsmtFinSF1	BsmtFinSF2	BsmtFullBath	BsmtHalfBath	…	OverallQual	\
0	706.0	0.0	1	0	…	7	
1	978.0	0.0	0	1	…	6	
2	486.0	0.0	1	0	…	7	
3	216.0	0.0	1	0	…	7	
4	655.0	0.0	1	0	…	8	
…	…	…	…	…	…	…	
1374	0.0	0.0	0	0	…	6	
1375	790.0	163.0	1	0	…	6	
1376	275.0	0.0	0	0	…	7	
1377	49.0	1029.0	1	0	…	5	

```
1378       830.0       290.0             1             0  …            5

      PoolArea  ScreenPorch  TotRmsAbvGrd  TotalBsmtSF  WoodDeckSF  YearBuilt  \
0          0.0          0.0             8        856.0         0.0       2003
1          0.0          0.0             6       1262.0       298.0       1976
2          0.0          0.0             6        920.0         0.0       2001
3          0.0          0.0             7        756.0         0.0       1915
4          0.0          0.0             9       1145.0       192.0       2000
…          …            …               …          …          …           …
1374       0.0          0.0             7        953.0         0.0       1999
1375       0.0          0.0             7       1542.0       349.0       1978
1376       0.0          0.0             9       1152.0         0.0       1941
1377       0.0          0.0             5       1078.0       366.0       1950
1378       0.0          0.0             6       1256.0       736.0       1965

      YearRemodAdd  YrSold  SalePrice
0             2003    2008   208500.0
1             1976    2007   181500.0
2             2002    2008   223500.0
3             1970    2006   140000.0
4             2000    2008   250000.0
…              …       …         …
1374          2000    2007   175000.0
1375          1988    2010   210000.0
1376          2006    2010   266500.0
1377          1996    2010   142125.0
1378          1965    2008   147500.0

[1379 rows x 39 columns]
```

[33]:
```
house.head()
```

[33]:
```
  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

   BsmtFinSF1  BsmtFinSF2  BsmtFullBath  BsmtHalfBath  …  OverallQual  \
0       706.0         0.0             1             0  …            7
1       978.0         0.0             0             1  …            6
2       486.0         0.0             1             0  …            7
3       216.0         0.0             1             0  …            7
4       655.0         0.0             1             0  …            8

   PoolArea  ScreenPorch  TotRmsAbvGrd  TotalBsmtSF  WoodDeckSF  YearBuilt  \
```

```
0       0.0          0.0             8        856.0         0.0        2003
1       0.0          0.0             6       1262.0       298.0        1976
2       0.0          0.0             6        920.0         0.0        2001
3       0.0          0.0             7        756.0         0.0        1915
4       0.0          0.0             9       1145.0       192.0        2000

   YearRemodAdd  YrSold  SalePrice
0          2003    2008   208500.0
1          1976    2007   181500.0
2          2002    2008   223500.0
3          1970    2006   140000.0
4          2000    2008   250000.0

[5 rows x 39 columns]
```

[34]:
```
house.shape
```

[34]:
```
(1379, 39)
```

[35]:
```
df=pd.read_csv("Ames_House_Sales_Cropped.csv")
df
```

[35]:
```
     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
…         …          …        …         …          …              …
1374     1Fam          Y     953.0     694.0        0.0             3
1375     1Fam          Y    2073.0       0.0        0.0             3
1376     1Fam          Y    1188.0    1152.0        0.0             4
1377     1Fam          Y    1078.0       0.0        0.0             2
1378     1Fam          Y    1256.0       0.0        0.0             3

      BsmtFinSF1  BsmtFinSF2  BsmtFullBath  BsmtHalfBath  …  OverallQual  \
0          706.0         0.0             1             0  …            7
1          978.0         0.0             0             1  …            6
2          486.0         0.0             1             0  …            7
3          216.0         0.0             1             0  …            7
4          655.0         0.0             1             0  …            8
…          …          …           …            …  …  …          …
1374         0.0         0.0             0             0  …            6
1375       790.0       163.0             1             0  …            6
1376       275.0         0.0             0             0  …            7
1377        49.0      1029.0             1             0  …            5
1378       830.0       290.0             1             0  …            5
```

```
      PoolArea  ScreenPorch  TotRmsAbvGrd  TotalBsmtSF  WoodDeckSF  YearBuilt  \
0          0.0          0.0             8        856.0         0.0       2003
1          0.0          0.0             6       1262.0       298.0       1976
2          0.0          0.0             6        920.0         0.0       2001
3          0.0          0.0             7        756.0         0.0       1915
4          0.0          0.0             9       1145.0       192.0       2000
…          …            …               …          …          …
1374       0.0          0.0             7        953.0         0.0       1999
1375       0.0          0.0             7       1542.0       349.0       1978
1376       0.0          0.0             9       1152.0         0.0       1941
1377       0.0          0.0             5       1078.0       366.0       1950
1378       0.0          0.0             6       1256.0       736.0       1965

      YearRemodAdd  YrSold  SalePrice
0             2003    2008   208500.0
1             1976    2007   181500.0
2             2002    2008   223500.0
3             1970    2006   140000.0
4             2000    2008   250000.0
…             …       …      …
1374          2000    2007   175000.0
1375          1988    2010   210000.0
1376          2006    2010   266500.0
1377          1996    2010   142125.0
1378          1965    2008   147500.0

[1379 rows x 39 columns]
```

[36]:
```
d=df.columns
d
```

[36]:
```
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')
```

[37]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1379 entries, 0 to 1378
Data columns (total 39 columns):
 #   Column         Non-Null Count  Dtype
```

```
---  ------          --------------  -----
 0   BldgType        1379 non-null   object
 1   CentralAir      1379 non-null   object
 2   1stFlrSF        1379 non-null   float64
 3   2ndFlrSF        1379 non-null   float64
 4   3SsnPorch       1379 non-null   float64
 5   BedroomAbvGr    1379 non-null   int64
 6   BsmtFinSF1      1379 non-null   float64
 7   BsmtFinSF2      1379 non-null   float64
 8   BsmtFullBath    1379 non-null   int64
 9   BsmtHalfBath    1379 non-null   int64
 10  BsmtUnfSF       1379 non-null   float64
 11  EnclosedPorch   1379 non-null   float64
 12  Fireplaces      1379 non-null   int64
 13  FullBath        1379 non-null   int64
 14  GarageArea      1379 non-null   float64
 15  GarageCars      1379 non-null   int64
 16  GarageYrBlt     1379 non-null   float64
 17  GrLivArea       1379 non-null   float64
 18  HalfBath        1379 non-null   int64
 19  KitchenAbvGr    1379 non-null   int64
 20  LotArea         1379 non-null   float64
 21  LotFrontage     1379 non-null   float64
 22  LowQualFinSF    1379 non-null   float64
 23  MSSubClass      1379 non-null   int64
 24  MasVnrArea      1379 non-null   float64
 25  MiscVal         1379 non-null   float64
 26  MoSold          1379 non-null   int64
 27  OpenPorchSF     1379 non-null   float64
 28  OverallCond     1379 non-null   int64
 29  OverallQual     1379 non-null   int64
 30  PoolArea        1379 non-null   float64
 31  ScreenPorch     1379 non-null   float64
 32  TotRmsAbvGrd    1379 non-null   int64
 33  TotalBsmtSF     1379 non-null   float64
 34  WoodDeckSF      1379 non-null   float64
 35  YearBuilt       1379 non-null   int64
 36  YearRemodAdd    1379 non-null   int64
 37  YrSold          1379 non-null   int64
 38  SalePrice       1379 non-null   float64
dtypes: float64(21), int64(16), object(2)
memory usage: 420.3+ KB
```

[38]:
```
df.dtypes
```

[38]:
```
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
```

[39]:
```
df.dtypes.value_counts()
```

[39]:
```
float64    21
int64      16
object      2
dtype: int64
```

0.0.3 Step2. [Predict Sale Price without Categorical features].

Drop both categorical features – BldgType and CentralAir (USE drop() and pop() methods)

[40]:
```
house.drop(['BldgType'],axis=1)  ###drop() method
```

[40]:

	CentralAir	1stFlrSF	2ndFlrSF	3SsnPorch	BedroomAbvGr	BsmtFinSF1	\
0	Y	856.0	854.0	0.0	3	706.0	
1	Y	1262.0	0.0	0.0	3	978.0	
2	Y	920.0	866.0	0.0	3	486.0	
3	Y	961.0	756.0	0.0	3	216.0	
4	Y	1145.0	1053.0	0.0	4	655.0	
…	…	…	…	…	…	…	
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	…	OverallQual	\
0	0.0	1	0	150.0	…	7	
1	0.0	0	1	284.0	…	6	
2	0.0	1	0	434.0	…	7	
3	0.0	1	0	540.0	…	7	
4	0.0	1	0	490.0	…	8	
…	…	…	…	…	…	…	
1374	0.0	0	0	953.0	…	6	
1375	163.0	1	0	589.0	…	6	
1376	0.0	0	0	877.0	…	7	
1377	1029.0	1	0	0.0	…	5	
1378	290.0	1	0	136.0	…	5	

	PoolArea	ScreenPorch	TotRmsAbvGrd	TotalBsmtSF	WoodDeckSF	YearBuilt	\
0	0.0	0.0	8	856.0	0.0	2003	
1	0.0	0.0	6	1262.0	298.0	1976	
2	0.0	0.0	6	920.0	0.0	2001	
3	0.0	0.0	7	756.0	0.0	1915	
4	0.0	0.0	9	1145.0	192.0	2000	
…	…	…	…	…	…	…	
1374	0.0	0.0	7	953.0	0.0	1999	
1375	0.0	0.0	7	1542.0	349.0	1978	
1376	0.0	0.0	9	1152.0	0.0	1941	
1377	0.0	0.0	5	1078.0	366.0	1950	
1378	0.0	0.0	6	1256.0	736.0	1965	

	YearRemodAdd	YrSold	SalePrice
0	2003	2008	208500.0
1	1976	2007	181500.0

```
2             2002    2008   223500.0
3             1970    2006   140000.0
4             2000    2008   250000.0
…              …       …         …
1374          2000    2007   175000.0
1375          1988    2010   210000.0
1376          2006    2010   266500.0
1377          1996    2010   142125.0
1378          1965    2008   147500.0

[1379 rows x 38 columns]
```

[41]:
```
df.pop('CentralAir')
```

[41]:
```
0       Y
1       Y
2       Y
3       Y
4       Y
       ..
1374    Y
1375    Y
1376    Y
1377    Y
1378    Y
Name: CentralAir, Length: 1379, dtype: object
```

Prepare X matrix (36 feature columns) and y vector (ie., SalePrice column)

[42]:
```
data1 = ['1stFlrSF', '2ndFlrSF',␣
 ↪'3SsnPorch','BedroomAbvGr','BsmtFinSF1','BsmtFinSF2','BsmtFullBath','BsmtHalfBath','BsmtUnf
 ↪'GrLivArea','HalfBath','KitchenAbvGr','LotArea','LotFrontage','LowQualFinSF','MSSubClass','
 ↪'MoSold','OpenPorchSF','OverallCond','OverallQual','PoolArea','ScreenPorch','TotRmsAbvGrd',
X=house[data1]
data2 = ['SalePrice']
y=house.SalePrice
```

[43]:
```
X
```

[43]:
```
      1stFlrSF  2ndFlrSF  3SsnPorch  BedroomAbvGr  BsmtFinSF1  BsmtFinSF2  \
0        856.0     854.0        0.0             3       706.0         0.0
1       1262.0       0.0        0.0             3       978.0         0.0
2        920.0     866.0        0.0             3       486.0         0.0
3        961.0     756.0        0.0             3       216.0         0.0
4       1145.0    1053.0        0.0             4       655.0         0.0
…          …         …          …             …          …           …
1374     953.0     694.0        0.0             3         0.0         0.0
1375    2073.0       0.0        0.0             3       790.0       163.0
```

```
1376    1188.0    1152.0        0.0             4       275.0         0.0
1377    1078.0       0.0        0.0             2        49.0      1029.0
1378    1256.0       0.0        0.0             3       830.0       290.0

      BsmtFullBath  BsmtHalfBath  BsmtUnfSF  EnclosedPorch  …  OverallCond  \
0                1             0      150.0            0.0  …            5
1                0             1      284.0            0.0  …            8
2                1             0      434.0            0.0  …            5
3                1             0      540.0          272.0  …            5
4                1             0      490.0            0.0  …            5
…              …             …        …              …  …          …
1374             0             0      953.0            0.0  …            5
1375             1             0      589.0            0.0  …            6
1376             0             0      877.0            0.0  …            9
1377             1             0        0.0          112.0  …            6
1378             1             0      136.0            0.0  …            6

      OverallQual  PoolArea  ScreenPorch  TotRmsAbvGrd  TotalBsmtSF  \
0               7       0.0          0.0             8        856.0
1               6       0.0          0.0             6       1262.0
2               7       0.0          0.0             6        920.0
3               7       0.0          0.0             7        756.0
4               8       0.0          0.0             9       1145.0
…             …       …          …             …          …
1374            6       0.0          0.0             7        953.0
1375            6       0.0          0.0             7       1542.0
1376            7       0.0          0.0             9       1152.0
1377            5       0.0          0.0             5       1078.0
1378            5       0.0          0.0             6       1256.0

      WoodDeckSF  YearBuilt  YearRemodAdd  YrSold
0            0.0       2003          2003    2008
1          298.0       1976          1976    2007
2            0.0       2001          2002    2008
3            0.0       1915          1970    2006
4          192.0       2000          2000    2008
…          …         …           …       …
1374         0.0       1999          2000    2007
1375       349.0       1978          1988    2010
1376         0.0       1941          2006    2010
1377       366.0       1950          1996    2010
1378       736.0       1965          1965    2008

[1379 rows x 36 columns]
```

[44]:
```
y
```

[44]:
```
0       208500.0
1       181500.0
2       223500.0
3       140000.0
4       250000.0
          …
1374    175000.0
1375    210000.0
1376    266500.0
1377    142125.0
1378    147500.0
Name: SalePrice, Length: 1379, dtype: float64
```

Split dataset for training and testing as X_train, X_test, y_train, y_test (use 25% test size).

[45]:
```
from sklearn.model_selection import train_test_split
```

[46]:
```
X_train, X_test, y_train, y_test = train_test_split(X, y, train_size=0.
↪75,test_size=0.25)
```

[47]:
```
X_train, X_test, y_train, y_test
```

[47]:
```
(      1stFlrSF  2ndFlrSF  3SsnPorch  BedroomAbvGr  BsmtFinSF1  BsmtFinSF2  \
 903      979.0     979.0        0.0             4       484.0         0.0
 582     1482.0     780.0        0.0             4       871.0         0.0
 151     1149.0     467.0        0.0             3       370.0         0.0
 523     1092.0       0.0        0.0             2       895.0         0.0
 940     1050.0       0.0        0.0             3       915.0         0.0
 …          …         …          …             …          …
 833      808.0     785.0        0.0             3         0.0         0.0
 982     1306.0       0.0        0.0             1       900.0         0.0
 1096    1640.0       0.0      216.0             3       728.0         0.0
 884     2069.0       0.0        0.0             4       425.0         0.0
 1091    1265.0       0.0       96.0             3       633.0         0.0

       BsmtFullBath  BsmtHalfBath  BsmtUnfSF  EnclosedPorch  …  OverallCond  \
 903              0             0      495.0            0.0  …            6
 582              1             0      611.0            0.0  …            5
 151              0             0      779.0          183.0  …            7
 523              1             0      197.0            0.0  …            5
 940              1             0      135.0            0.0  …            6
 …              …             …          …              …  …          …
 833              0             0      808.0            0.0  …            5
 982              1             0      406.0            0.0  …            5
 1096             1             0      568.0            0.0  …            7
 884              1             0      160.0            0.0  …            7
 1091             0             1      586.0            0.0  …            8
```

```
      OverallQual  PoolArea  ScreenPorch  TotRmsAbvGrd  TotalBsmtSF  \
903             6       0.0          0.0             8        979.0
582             8       0.0          0.0            10       1482.0
151             6       0.0          0.0             5       1149.0
523             5       0.0        122.0             6       1092.0
940             5       0.0          0.0             6       1050.0
…             …       …          …             …           …
833             6       0.0          0.0             7        808.0
982             6       0.0          0.0             5       1306.0
1096            6       0.0          0.0             7       1296.0
884             7       0.0          0.0             9        585.0
1091            5       0.0          0.0             6       1219.0

      WoodDeckSF  YearBuilt  YearRemodAdd  YrSold
903          0.0       1946          1950    2007
582        168.0       2003          2003    2008
151          0.0       1926          2004    2007
523        268.0       1957          1957    2006
940          0.0       1961          1961    2006
…          …         …             …       …
833        342.0       1992          1993    2009
982        170.0       2005          2005    2009
1096       108.0       1954          2006    2008
884          0.0       1960          2007    2008
1091         0.0       1965          1999    2008

[1034 rows x 36 columns],
      1stFlrSF  2ndFlrSF  3SsnPorch  BedroomAbvGr  BsmtFinSF1  BsmtFinSF2  \
1173    2076.0       0.0        0.0             2      1386.0         0.0
759      980.0       0.0        0.0             3       400.0       480.0
860      742.0     742.0        0.0             3         0.0         0.0
363     1269.0       0.0        0.0             2        24.0         0.0
32      1234.0       0.0        0.0             3         0.0         0.0
…         …       …          …             …         …           …
788     1559.0       0.0        0.0             2       338.0         0.0
539      774.0     656.0        0.0             3         0.0         0.0
818     1120.0       0.0        0.0             3       932.0         0.0
333     1192.0     403.0        0.0             2       388.0         0.0
1377    1078.0       0.0        0.0             2        49.0      1029.0

      BsmtFullBath  BsmtHalfBath  BsmtUnfSF  EnclosedPorch  …  OverallCond  \
1173             1             0      690.0            0.0  …            5
759              0             0      100.0            0.0  …            5
860              0             0      742.0            0.0  …            5
363              0             0     1232.0            0.0  …            5
32               0             0     1234.0            0.0  …            5
```

```
…             …            …         …              …  …          …
788             1            0    1221.0          230.0  …          6
539             0            0     384.0            0.0  …          5
818             1            0     108.0            0.0  …          5
333             0            0     552.0          108.0  …          5
1377            1            0       0.0          112.0  …          6

      OverallQual  PoolArea  ScreenPorch  TotRmsAbvGrd  TotalBsmtSF  \
1173           10       0.0          0.0             7       2076.0
759             5       0.0          0.0             6        980.0
860             6       0.0          0.0             8        742.0
363             8       0.0        144.0             6       1256.0
32              8       0.0          0.0             7       1234.0
…             …       …          …           …           …
788             5       0.0          0.0             5       1559.0
539             7       0.0          0.0             8        384.0
818             4       0.0          0.0             5       1040.0
333             6       0.0          0.0             6        940.0
1377            5       0.0          0.0             5       1078.0

      WoodDeckSF  YearBuilt  YearRemodAdd  YrSold
1173       216.0       2006          2006    2006
759         68.0       1967          1967    2006
860         36.0       2005          2005    2009
363        146.0       2004          2005    2010
32           0.0       2007          2007    2008
…          …         …           …     …
788          0.0       1948          1973    2007
539        100.0       2000          2000    2007
818        168.0       1961          1961    2007
333          0.0       1940          2000    2006
1377       366.0       1950          1996    2010

[345 rows x 36 columns],
903     145000.0
582     305000.0
151     152000.0
523     141000.0
940     136500.0
         …
833     178000.0
982     196000.0
1096    224000.0
884     242000.0
1091    179900.0
Name: SalePrice, Length: 1034, dtype: float64,
1173    465000.0
```

```
759     135500.0
860     174000.0
363     192000.0
32      179900.0
          …
788     153500.0
539     170000.0
818     129000.0
333     140000.0
1377    142125.0
Name: SalePrice, Length: 345, dtype: float64)
```

Create LinearRegression model, fit on training set and predict on test set

[48]:
```
from sklearn.linear_model import LinearRegression
```

[49]:
```
#create a linear regression object
model = LinearRegression()
#train a model
model.fit( X_train,y_train)
```

[49]:
```
LinearRegression()
```

[50]:
```
y_pred = model.predict(X_test)
y_pred
```

[50]:
```
array([343843.65754748,  95029.38549773, 169794.24889062, 211304.43611183,
       210398.63861163,  75073.75727441, 111613.78843825, 144962.28248309,
       178539.06759803, 113796.00153267,  91671.80644439, 112023.44969957,
       106306.57435984, 222129.07385228, 174818.66894032, 108839.68591425,
       180008.73409479, 136905.97584528, 101504.92408718, 141319.2703993 ,
       169394.68535725, 141770.09535832,  49816.7114211 , 209753.53862673,
       220739.27697146, 102349.50412943, 100955.66549767,  88788.24616048,
       179979.31978142, 229207.93369507, 241499.59810583, 188573.8133879 ,
       138375.20136656, 112016.38806082, 198329.03194489, 213189.51413244,
       231224.16110581, 154516.5968845 , 128722.31576005, 151342.18758167,
       173073.15762992, 194854.4806387 , 283405.38148664, 117922.26895843,
        73247.28522753, 112162.98929892, 170780.47241076, 324588.27230496,
       240759.96079044, 376707.91509053, 265351.68182457,  52797.83793952,
       207565.38162927, 137146.55832762, 191801.41285861, 204542.17349544,
       240015.20819539, 191464.28519076,  88148.39470493, 223662.22798159,
       218008.19170056, 252613.22316689, 211216.46271609, 121252.1408625 ,
       189858.25835928, 114656.39533588, 311195.87421964, 160167.35381274,
       161506.36275148, 145894.30802898, 209508.32111458, 244821.78260324,
        44364.86688781, 142116.28241324, 219838.56665944, 127267.1121915 ,
       129126.64201766,  71205.84496587, 316361.93753265, 111323.07906176,
       172203.96169718, 118218.47248644, 375364.49984387, 186482.14424936,
       133975.88760024, 195839.25614804, 252274.12652459,  95868.28330894,
```

 96467.0065489 , 193469.76575237, 237346.83053708, 317316.53283583,
104338.1839894 , 169324.18407076, 147120.54908734, 162379.31436992,
114252.41919604, 241078.17358412, 180049.15726857, 374334.07520483,
182872.24886861, 389594.21545611, 100083.54193546, 179894.28103745,
 86343.17988349, 161582.77420357, 190120.30064037, 370966.34888382,
217181.34609185, 193768.54543017, 268317.70720833, 180129.69249009,
 84368.38112288, 203167.6505222 , 136923.11025027, 120723.16582559,
 78874.36006872, 240952.34883125, 302603.0658477 , 197655.47810502,
132960.04668215, 227244.22062148, 145305.85479897, 160068.57599372,
263402.82296716, 223641.15730607, 233353.84989055, 236842.74634309,
123161.64295409, 318429.78602217, 236085.05579585, 125656.71370325,
201346.2142906 , 212772.07888998, 232278.40725318, 241399.84028217,
339662.57382198, 165932.85963536, 249188.74082173, 148432.75729727,
313322.94694106, 281458.57291332, 198250.88898276, 307062.99875764,
212830.2678777 , 195185.84256723, 198296.71248287, 241384.50337242,
278124.91272708, 293537.34106261, 138422.97188358, 199705.71261725,
122685.60179981, 134101.98706842, 233156.2788514 , 173867.95240058,
127079.30119989, 275272.46587755, 149614.4985271 , 116838.56773206,
179349.89036903, 205296.45988448, 280704.42302814, 220472.76452976,
181736.32767933, 72906.5192921 , 209955.36277415, 162241.12368289,
168417.29546066, 187516.28570373, 178130.39219583, 283402.3917727 ,
266787.97378909, 235228.03906513, 175463.25683058, 208772.99037848,
181954.46424814, 96087.44546314, 97373.3039676 , 156479.21529055,
246842.73204126, 84439.97994498, 207358.2905316 , 183480.20272491,
295485.06817171, 110016.25396967, 280948.15927398, 150143.37721436,
140699.35578697, 111522.96936896, 203426.31539381, 309617.66605417,
222605.94078697, 152126.4755989 , 208944.97909766, 200269.93244918,
129421.54124901, 327464.78729045, 150540.69768937, 138957.65876576,
240236.91935131, 120076.1806853 , 152807.2110174 , 116919.20757723,
269626.12282073, 193157.06313123, 109296.41387479, 155573.91414951,
168412.70691209, 192130.8872861 , 258759.02562345, 146189.29338464,
193676.5935051 , 185695.54792075, 205471.4708046 , 204388.63726063,
147153.80828935, 149691.45422258, 127272.57565986, 109813.26239063,
 97890.08593742, 66559.03971253, 182196.79195973, 121240.38793064,
124492.85617282, 187414.63011091, 233925.97634536, 121505.6186186 ,
151713.93743585, 248904.82020454, 223745.3477359 , 206925.87759428,
311348.59321594, 114916.79869721, 98359.79222636, 104630.75578839,
238044.2406506 , 127425.30219648, 107660.43604895, 139091.94792935,
161006.3049412 , 149546.31747296, 191680.71710645, 212247.00270499,
 71997.50881137, 253388.24859653, 161542.76759104, 163304.93672006,
164299.49390882, 81040.29144344, 130689.37322539, 175833.68795763,
312826.74380163, 261361.54029388, 222710.0253466 , 236803.75840854,
125591.75631497, 118622.07676837, 185032.35923665, 156047.64563767,
181953.94767967, 185024.21450639, 159390.20954353, 217728.91375373,
173461.00212053, 149525.83708286, 271526.4382511 , 189721.30939484,
247996.35077442, 123758.83075548, 146413.20190542, 206782.80814967,
145757.39891842, 235760.080671 , 255195.70046381, 83013.32995505,

```
274441.95664722, 152272.98095611, 173640.58390295, 122195.40672676,
 96642.84610201, 112395.07266538, 155844.40737419, 234640.57994914,
 82659.12415209, 165264.69402393, 123807.15190975, 117234.8014548 ,
 63526.9835117 , 164142.7821134 , 119562.68646113, 133563.35601578,
180300.01635249, 194468.87415699, 112137.51810579, 175908.09300241,
198497.30947856, 211527.53277822,   -887.83369126, 149940.98575298,
223437.7602771 , 322575.87534534, 214588.14225161, 269683.6773258 ,
250148.10809315, 210958.46363519, 221064.3637554 , 145134.69029348,
314669.0756796 , 202093.96054347, 153567.04196271, 195343.53299302,
187135.53691539, 212421.32205353, 131766.79441047, 160504.86756532,
221870.87604526, 180381.69525633, 181513.70086275, 270279.43930549,
376513.47803224, 215030.37472794, 227078.69915926, 271347.61961927,
307094.34305625, 124269.9338672 , 198860.50918028, 132753.38157498,
154140.18509407, 240226.46491656, 253087.2949071 , 223504.51287066,
212863.77291139, 191172.96194414, 225182.43049873, 202435.23860725,
225215.69134337, 168757.20189302,  89133.2624629 , 113800.09896732,
164258.34732077, 192132.29723948, 116433.33223496, 159725.8773468 ,
134803.88016751])
```

Compute Mean Squared Error (MSE) on actual values and predicted values (you will get output as 1474827326.0).

[51]:
```
from sklearn.metrics import mean_squared_error
```

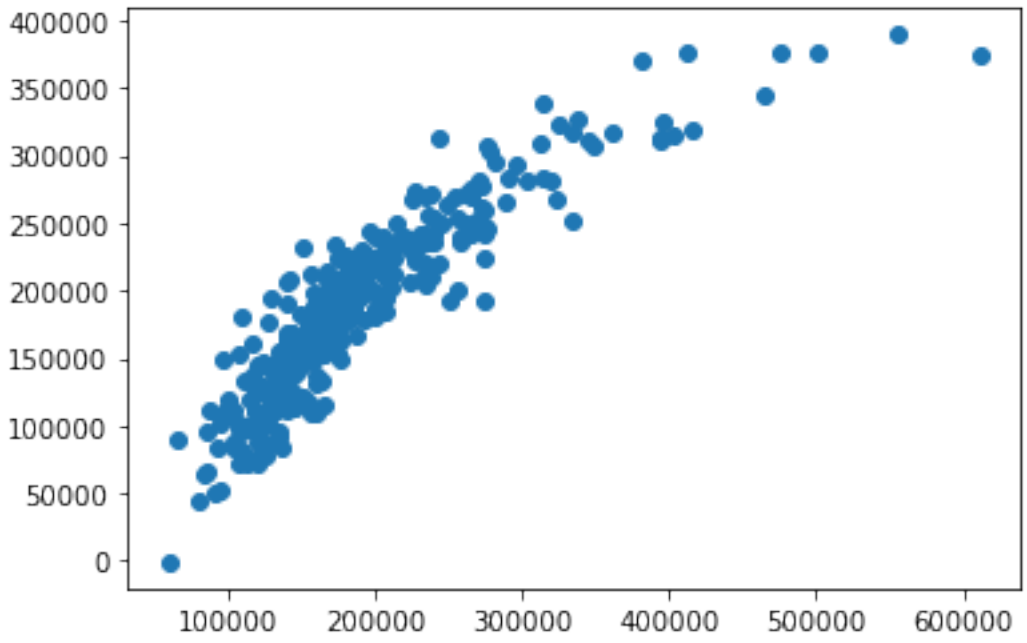
[52]:
```
mse=mean_squared_error(y_test,y_pred)
mse
```

[52]:
```
1127595525.476533
```

0.0.4 Step3. [Create Scatter Plot]. Plot Scatterplot between y_test and y_pred.

[53]:
```
import matplotlib.pyplot as plt
plt.scatter(y_test,y_pred)
```

[53]:
```
<matplotlib.collections.PathCollection at 0x1f1906ff7c0>
```



0.0.5 Step4. [Encode Categorical columns]. Using get_dummies() method, perform one hot encoding on the two categorical columns, BldgType and CentralAir. Now, you will get 5 columns for BldgType variable and 2 columns for CentralAir column. So, now you have 43 independent variable and 1 dependent variable.

[62]:
```
house=pd.read_csv("Ames_House_Sales_Cropped.csv")
house
```

[62]:

	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	
…	…	…	…	…	…	…	
1374	1Fam	Y	953.0	694.0	0.0	3	
1375	1Fam	Y	2073.0	0.0	0.0	3	
1376	1Fam	Y	1188.0	1152.0	0.0	4	
1377	1Fam	Y	1078.0	0.0	0.0	2	
1378	1Fam	Y	1256.0	0.0	0.0	3	

	BsmtFinSF1	BsmtFinSF2	BsmtFullBath	BsmtHalfBath	…	OverallQual	\
0	706.0	0.0	1	0	…	7	
1	978.0	0.0	0	1	…	6	
2	486.0	0.0	1	0	…	7	

```
3          216.0         0.0             1             0  …            7
4          655.0         0.0             1             0  …            8
…            …           …               …             …  …          …
1374         0.0         0.0             0             0  …            6
1375       790.0       163.0             1             0  …            6
1376       275.0         0.0             0             0  …            7
1377        49.0      1029.0             1             0  …            5
1378       830.0       290.0             1             0  …            5

      PoolArea  ScreenPorch  TotRmsAbvGrd  TotalBsmtSF  WoodDeckSF  YearBuilt  \
0          0.0          0.0             8        856.0         0.0       2003
1          0.0          0.0             6       1262.0       298.0       1976
2          0.0          0.0             6        920.0         0.0       2001
3          0.0          0.0             7        756.0         0.0       1915
4          0.0          0.0             9       1145.0       192.0       2000
…          …            …               …          …          …           …
1374       0.0          0.0             7        953.0         0.0       1999
1375       0.0          0.0             7       1542.0       349.0       1978
1376       0.0          0.0             9       1152.0         0.0       1941
1377       0.0          0.0             5       1078.0       366.0       1950
1378       0.0          0.0             6       1256.0       736.0       1965

      YearRemodAdd  YrSold  SalePrice
0             2003    2008   208500.0
1             1976    2007   181500.0
2             2002    2008   223500.0
3             1970    2006   140000.0
4             2000    2008   250000.0
…              …      …         …
1374          2000    2007   175000.0
1375          1988    2010   210000.0
1376          2006    2010   266500.0
1377          1996    2010   142125.0
1378          1965    2008   147500.0

[1379 rows x 39 columns]
```

[63]:
```
encode_house=pd.get_dummies(house, columns=["CentralAir","BldgType"])
encode_house.head()
```

[63]:
```
   1stFlrSF  2ndFlrSF  3SsnPorch  BedroomAbvGr  BsmtFinSF1  BsmtFinSF2  \
0     856.0     854.0        0.0             3       706.0         0.0
1    1262.0       0.0        0.0             3       978.0         0.0
2     920.0     866.0        0.0             3       486.0         0.0
3     961.0     756.0        0.0             3       216.0         0.0
4    1145.0    1053.0        0.0             4       655.0         0.0
```

```
   BsmtFullBath  BsmtHalfBath  BsmtUnfSF  EnclosedPorch  …  YearRemodAdd  \
0             1             0      150.0            0.0  …          2003
1             0             1      284.0            0.0  …          1976
2             1             0      434.0            0.0  …          2002
3             1             0      540.0          272.0  …          1970
4             1             0      490.0            0.0  …          2000

   YrSold  SalePrice  CentralAir_N  CentralAir_Y  BldgType_1Fam  \
0    2008   208500.0             0             1              1
1    2007   181500.0             0             1              1
2    2008   223500.0             0             1              1
3    2006   140000.0             0             1              1
4    2008   250000.0             0             1              1

   BldgType_2fmCon  BldgType_Duplex  BldgType_Twnhs  BldgType_TwnhsE
0                0                0               0                0
1                0                0               0                0
2                0                0               0                0
3                0                0               0                0
4                0                0               0                0

[5 rows x 44 columns]
```

[64]:
```
encode_house.columns
```

[64]:
```
Index(['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', 'CentralAir_N', 'CentralAir_Y',
       'BldgType_1Fam', 'BldgType_2fmCon', 'BldgType_Duplex', 'BldgType_Twnhs',
       'BldgType_TwnhsE'],
      dtype='object')
```

[65]:
```
encode_house.shape
```

[65]:
```
(1379, 44)
```

0.0.6 Step5. [Predict Sale Price with Categorical features]

Prepare X matrix (43 feature columns) and y vector (ie., SalePrice column)

[75]:
```
data3 = ['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', 'CentralAir_N', 'CentralAir_Y',
       'BldgType_1Fam', 'BldgType_2fmCon', 'BldgType_Duplex', 'BldgType_Twnhs',
       'BldgType_TwnhsE']
encode_X=house[data1]
data4 = ['SalePrice']
encode_y=house.SalePrice
```

[76]:
```
encode_X
```

[76]:

	1stFlrSF	2ndFlrSF	3SsnPorch	BedroomAbvGr	BsmtFinSF1	BsmtFinSF2	\
0	856.0	854.0	0.0	3	706.0	0.0	
1	1262.0	0.0	0.0	3	978.0	0.0	
2	920.0	866.0	0.0	3	486.0	0.0	
3	961.0	756.0	0.0	3	216.0	0.0	
4	1145.0	1053.0	0.0	4	655.0	0.0	
…	…	…	…	…	…	…	
1374	953.0	694.0	0.0	3	0.0	0.0	
1375	2073.0	0.0	0.0	3	790.0	163.0	
1376	1188.0	1152.0	0.0	4	275.0	0.0	
1377	1078.0	0.0	0.0	2	49.0	1029.0	
1378	1256.0	0.0	0.0	3	830.0	290.0	

	BsmtFullBath	BsmtHalfBath	BsmtUnfSF	EnclosedPorch	…	OverallCond	\
0	1	0	150.0	0.0	…	5	
1	0	1	284.0	0.0	…	8	
2	1	0	434.0	0.0	…	5	
3	1	0	540.0	272.0	…	5	
4	1	0	490.0	0.0	…	5	
…	…	…	…	…	…	…	
1374	0	0	953.0	0.0	…	5	
1375	1	0	589.0	0.0	…	6	
1376	0	0	877.0	0.0	…	9	
1377	1	0	0.0	112.0	…	6	
1378	1	0	136.0	0.0	…	6	

	OverallQual	PoolArea	ScreenPorch	TotRmsAbvGrd	TotalBsmtSF	\
0	7	0.0	0.0	8	856.0	
1	6	0.0	0.0	6	1262.0	
2	7	0.0	0.0	6	920.0	
3	7	0.0	0.0	7	756.0	
4	8	0.0	0.0	9	1145.0	
…	…	…	…	…	…	
1374	6	0.0	0.0	7	953.0	

```
1375            6       0.0           0.0             7       1542.0
1376            7       0.0           0.0             9       1152.0
1377            5       0.0           0.0             5       1078.0
1378            5       0.0           0.0             6       1256.0

      WoodDeckSF  YearBuilt  YearRemodAdd  YrSold
0            0.0       2003          2003    2008
1          298.0       1976          1976    2007
2            0.0       2001          2002    2008
3            0.0       1915          1970    2006
4          192.0       2000          2000    2008
…            …          …             …       …
1374         0.0       1999          2000    2007
1375       349.0       1978          1988    2010
1376         0.0       1941          2006    2010
1377       366.0       1950          1996    2010
1378       736.0       1965          1965    2008

[1379 rows x 36 columns]
```

[77]:
```
encode_y
```

[77]:
```
0       208500.0
1       181500.0
2       223500.0
3       140000.0
4       250000.0
          …
1374    175000.0
1375    210000.0
1376    266500.0
1377    142125.0
1378    147500.0
Name: SalePrice, Length: 1379, dtype: float64
```

Split dataset for training and testing

[78]:
```
encode_X_train, encode_X_test, encode_y_train, encode_y_test =␣
 ↪train_test_split(encode_X, encode_y, train_size=0.75,test_size=0.25)
encode_X_train, encode_X_test, encode_y_train, encode_y_test
```

[78]:
```
(     1stFlrSF  2ndFlrSF  3SsnPorch  BedroomAbvGr  BsmtFinSF1  BsmtFinSF2  \
 223     798.0     689.0        0.0             3        94.0         0.0
 673    1178.0       0.0        0.0             3      1084.0         0.0
 718    1494.0       0.0        0.0             2       437.0      1057.0
 136     875.0       0.0        0.0             2       209.0         0.0
 684    1776.0       0.0        0.0             4         0.0         0.0
 ..        …         …          …               …         …           …
```

```
965    1504.0       0.0        0.0             2        16.0         0.0
482    1368.0       0.0        0.0             2         0.0         0.0
562    1402.0       0.0        0.0             2         0.0         0.0
581    1828.0       0.0        0.0             3        48.0         0.0
949     985.0       0.0        0.0             3       595.0         0.0

     BsmtFullBath  BsmtHalfBath  BsmtUnfSF  EnclosedPorch  …  OverallCond  \
223             0             0      641.0            0.0  …            4
673             0             1       92.0            0.0  …            6
718             1             0        0.0            0.0  …            5
136             1             0      506.0            0.0  …            7
684             1             0     1584.0            0.0  …            5
..            …             …         …              …  …          …
965             0             0     1330.0            0.0  …            5
482             0             0     1368.0            0.0  …            5
562             0             2     1258.0            0.0  …            5
581             0             0     1774.0            0.0  …            5
949             0             0      390.0            0.0  …            8

     OverallQual  PoolArea  ScreenPorch  TotRmsAbvGrd  TotalBsmtSF  \
223            6       0.0          0.0             7        735.0
673            5       0.0          0.0             5       1176.0
718            8       0.0        216.0             6       1494.0
136            5       0.0          0.0             5        715.0
684            5       0.0          0.0             9       1584.0
..           …       …            …             …          …
965            7       0.0          0.0             7       1346.0
482            7       0.0          0.0             6       1368.0
562            7       0.0          0.0             7       1258.0
581            9       0.0        260.0             9       1822.0
949            5       0.0          0.0             6        985.0

     WoodDeckSF  YearBuilt  YearRemodAdd  YrSold
223         0.0       1945          1950    2010
673       224.0       1973          2000    2008
718         0.0       1995          1996    2006
136        48.0       1931          1993    2009
684         0.0       1958          1958    2009
..          …         …             …       …
965       156.0       2005          2006    2008
482       132.0       2005          2006    2006
562       120.0       2006          2007    2007
581         0.0       2007          2007    2007
949       210.0       1977          1977    2008

[1034 rows x 36 columns],
      1stFlrSF  2ndFlrSF  3SsnPorch  BedroomAbvGr  BsmtFinSF1  BsmtFinSF2  \
```

```
646     1419.0       0.0        0.0             2       945.0         0.0
571     1040.0       0.0        0.0             3       732.0         0.0
262     1175.0     807.0      196.0             3       421.0         0.0
285     2000.0       0.0        0.0             3      1464.0         0.0
1360    1236.0       0.0        0.0             2       600.0         0.0
…          …         …          …             …          …
1184     760.0     896.0        0.0             3         0.0         0.0
1372    1072.0       0.0        0.0             2       547.0         0.0
1167     847.0    1101.0        0.0             4         0.0         0.0
432      979.0     224.0        0.0             3       185.0         0.0
1269     960.0       0.0        0.0             2       250.0         0.0

      BsmtFullBath  BsmtHalfBath  BsmtUnfSF  EnclosedPorch  …  OverallCond  \
646              1             0      474.0            0.0  …            6
571              1             0      308.0            0.0  …            5
262              0             0      386.0            0.0  …            6
285              1             0      536.0            0.0  …            5
1360             1             0      312.0          158.0  …            7
…              …             …          …              …  …          …
1184             0             0      746.0            0.0  …            5
1372             1             0        0.0            0.0  …            5
1167             0             0      847.0            0.0  …            5
432              1             0      524.0          248.0  …            4
1269             0             0      710.0          168.0  …            4

      OverallQual  PoolArea  ScreenPorch  TotRmsAbvGrd  TotalBsmtSF  \
646             8       0.0          0.0             7       1419.0
571             5       0.0          0.0             6       1040.0
262             7       0.0          0.0             7        807.0
285             8       0.0          0.0             8       2000.0
1360            6       0.0          0.0             6        912.0
…             …         …            …             …          …
1184            7       0.0          0.0             7        746.0
1372            5       0.0          0.0             5        547.0
1167            7       0.0          0.0             8        847.0
432             5       0.0          0.0             5        709.0
1269            4       0.0          0.0             5        960.0

      WoodDeckSF  YearBuilt  YearRemodAdd  YrSold
646        140.0       2007          2007    2007
571        168.0       1996          1996    2009
262          0.0       1989          1989    2007
285        168.0       2004          2005    2007
1360         0.0       1957          1996    2010
…            …         …             …          …
1184       178.0       2003          2004    2008
1372         0.0       2005          2005    2006
```

```
1167       100.0       2004          2005    2006
432          0.0       1950          1950    2009
1269         0.0       1920          1950    2007

[345 rows x 36 columns],
223    113000.0
673    157000.0
718    270000.0
136    105000.0
684    110000.0
         …
965    191000.0
482    202665.0
562    194201.0
581    314813.0
949    149900.0
Name: SalePrice, Length: 1034, dtype: float64,
646     392000.0
571     152000.0
262     228500.0
285     305900.0
1360    149700.0
          …
1184    165400.0
1372    145000.0
1167    195000.0
432     110000.0
1269    108500.0
Name: SalePrice, Length: 345, dtype: float64)
```

Create LinearRegression model, fit on training set and predict on test set

[79]:
```
#create a linear regression object
model3 = LinearRegression()
#train a model
model3.fit( encode_X_train,encode_y_train)
```

[79]:
```
LinearRegression()
```

[80]:
```
encode_y_pred3 = model3.predict(encode_X_test)
encode_y_pred3
```

[80]:
```
array([263011.17348096, 151278.69406732, 237521.15280445, 306444.83659023,
       179589.12045248, 115295.24967863, 203596.65767938, 176617.38472468,
       268631.19291751, 182056.65942805, 280772.96016023, 107016.90710112,
       116173.93325518, 120909.63455584, 165041.44922353, 176959.05623095,
       168691.0385779 , 297130.05957958, 211041.07273794, 229225.62046386,
       204372.74722721, 125472.33582051, 210250.28184771, 128133.06143696,
```

160466.20410665, 143316.83598233, 113662.70546705, 128384.85053326,
204005.67809678, 254170.40185392, 111373.05179766, 117721.75287262,
 89979.83990709, 53289.42641272, 227546.2766431 , 174599.88447911,
363855.5618764 , 112180.99083582, 174408.02921482, 320263.70850029,
170015.30933011, 226691.10446723, 121966.20214686, 145222.19936231,
172222.59936543, 90329.90726443, 134316.70375444, 134398.98814105,
170693.92853795, 295137.06691152, 192412.42730159, 202648.90969701,
114478.17984398, 91631.32842667, 316425.49712862, 77870.01585279,
279912.95630235, 345131.7259968 , 354245.95943962, 124802.71354199,
221822.48322739, 58512.9545638 , 242066.2632748 , 280446.35727271,
125903.62241077, 200033.66968395, 188384.13418649, 187014.74534527,
366840.92399515, 171340.68202929, 180659.96318462, 241796.97393946,
178984.99265562, 256187.75060533, 238896.11051133, 127986.90388687,
127945.75954876, 136560.17497404, 347308.27837608, 220818.50833088,
319258.77269899, 151892.02301402, 332118.0268508 , 200334.08430189,
 96927.44804144, 258774.95707355, 134845.09515714, 236906.54301218,
125484.87168967, 160332.82904559, 167751.71389238, 212448.64215522,
219695.3821903 , 109280.55617755, 256299.31063149, 113250.09453629,
193447.00826941, 250918.88471371, 165991.81257587, 174378.35619732,
138691.16327863, 246565.48680294, 114628.78766436, 118028.51327917,
157226.47275121, 234567.00576567, 244633.4325637 , 203512.75897764,
188172.48748111, 300578.67352358, 125936.78644275, 152772.71209275,
317321.56611107, 204630.72168036, 265764.30817731, 318292.02460144,
218103.1205221 , 227570.84723725, 144899.41628243, 162528.96358485,
113675.08061916, 99646.87295535, 175831.98050102, 279607.81356017,
 80388.45142476, 157558.57514864, 168716.53749412, 158511.1143292 ,
268159.52493592, 236030.91601818, 259218.29023872, 306769.60415783,
312268.68976816, 247398.57461256, 58478.58513448, 256977.277043 ,
 99767.96631978, 185131.37204413, 191154.40856538, 73535.38345415,
286114.33914278, 194183.96819515, 322951.28345957, 155545.05636391,
205482.5522098 , 108365.17269122, 240811.73295928, 219289.99754991,
187107.91381672, 176435.51802209, 198062.33666498, 173945.49975203,
222380.04466889, 107961.12387109, 189053.34508695, 218586.62766183,
272959.17668754, 210270.16586046, 109073.96530544, 201890.12203778,
152070.40540525, 243804.4681767 , 132567.33466158, 205569.63752122,
310253.08207241, 219023.30086696, 71434.11407795, 232625.9493711 ,
116663.18756643, 194412.47892819, 153498.99227512, 87922.04070131,
160958.73940293, 173844.05321751, 144703.03984228, 176244.49900156,
214601.32799234, 147770.63989154, 212838.86885315, 132436.55317839,
182066.90605849, 172966.84537452, 115902.78084216, 198990.76950787,
100858.42124779, 124087.51174644, 191911.25416032, 280290.68776879,
168058.4776116 , 95870.47684534, 236900.8889987 , 249720.00396569,
 91038.74809643, 159546.57107196, 146685.83943646, 253455.91290233,
 39809.78528565, 216266.91371523, 134516.02029511, 153774.77814121,
213863.55081859, 146365.420822 , 238660.89531225, 116989.66446081,
104169.22099331, 269343.66639209, 137006.01403918, 105856.31425958,
136728.36963521, 195616.73994141, 307220.9427874 , 135630.1097641 ,

```
123819.66544811, 233493.80313421, 126293.91297419, 269723.82669657,
187050.69751067, 182010.5803106 , 174415.96370449, 212096.77060329,
167451.55099717, 235941.70161552, 108186.15889551,  84014.25633019,
 91509.91102211, 191364.53226892, 222172.6976128 , 304879.0636896 ,
117378.50173638, 166442.96518088, 224278.64544932, 173874.75971112,
 51320.87049418, 245464.72312831, 117695.87260639, 153068.51183363,
185521.95186163, 263951.5687599 , 192974.30487573, 151554.40638456,
289869.79849727, 114332.9567395 , 325686.37376872, 183617.37524352,
170375.61864669, 265660.95274134, 338058.80404854,  90432.41646941,
130697.29943982, 322330.7171025 , 206494.67765573, 217801.94377649,
197440.34901259, 188853.62981309, 141799.12382521, 188725.00017183,
202593.17419429, 154981.98807933, 215951.57937482, 156892.08838412,
108413.50885758, 175793.81847454, 190290.43772055, 276491.19707614,
 90684.53922019, 178990.13616749, 204272.18383693, 395103.81203293,
250330.88058944, 216749.38023034, 338149.81739905, 129065.56432815,
114852.59813526, 183704.56915614,  77724.39581912, 119295.61866632,
 85125.19311684, 112336.27193449, 136734.87988703, 189230.78678898,
140351.79140137, 167472.87343677, 255651.38742537, 250560.48788194,
128348.35137954, 198812.52588592, 198973.62468277, 282407.75408133,
204156.88903463, 118201.02786527, 259503.7059336 , 210497.32464545,
144318.01028186, 218206.35075101, 168758.05219649, 192116.25796623,
 84124.35128754, 174483.07513333, 114455.60670438, 162952.99027018,
163370.15493512, 185867.67412221, 143442.1821607 , 133335.64273741,
232920.80084234, 130420.20688356, 265221.32286707, 180656.98578366,
268759.68205338, 187263.82446352, 128223.98511918, 209432.10302999,
229927.56623541, 100692.22043756, 167147.5715183 , 204197.84577366,
222393.52363762, 237642.86126765, 188461.28661532, 266329.18727241,
184233.43166737, 154538.6997815 , 211658.36389192, 246958.88902765,
332224.4655521 , 198189.98332438, 139392.87862161, 130614.43000603,
108086.23253089, 136853.92269232, 211668.28402068,  77139.51516822,
171526.78910715, 153455.57979293, 113005.2184186 , 211979.73979886,
308238.95500522, 145949.37373025, 134900.0070884 ,  95068.31565341,
211776.03931884, 122288.44223574, 222255.09745693, 107166.74408712,
 56209.2667581 ])
```

Compute Mean Squared Error on actual values and predicted values (you will get output as 1461036570.0).

[82]:
```
mse1=mean_squared_error(encode_y_test,encode_y_pred3)
mse1
```

[82]:
```
840762442.6361699
```

0.0.7 Step6. [Normalize using StandardScaler and Predict Sale Price]

Using StandardScaler, perform fit_transform() on X_train and transform() on X_test matrix that you already splitted.

[83]:
```
from sklearn.preprocessing import StandardScaler
```

[84]:
```
scaler = StandardScaler()
ss3=scaler.fit_transform(encode_X_train)    # scale X_train using␣
 ↪fit_transform() method
print(ss3)
```

```
[[-0.97719621  0.76025618 -0.13151388 … -0.96495393 -1.72036703
   1.68452317]
 [-0.01783362 -0.79993851 -0.13151388 …  0.00341162  0.72527132
   0.16130764]
 [ 0.77995212 -0.79993851 -0.13151388 …  0.76427027  0.52962025
  -1.36190789]
 …
 [ 0.54768539 -0.79993851 -0.13151388 …  1.1446996   1.06766068
  -0.60030012]
 [ 1.62318135 -0.79993851 -0.13151388 …  1.17928408  1.06766068
  -0.60030012]
 [-0.50508883 -0.79993851 -0.13151388 …  0.14174956 -0.39972232
   0.16130764]]
```

[85]:
```
ss5=scaler.transform(encode_X_test)      #X_test using transform() method
print(ss5)
```

```
[[ 0.59060424 -0.79993851 -0.13151388 …  1.17928408  1.06766068
  -0.60030012]
 [-0.36623372 -0.79993851 -0.13151388 …  0.79885476  0.52962025
   0.9229154 ]
 [-0.02540753  1.02745933  5.70578985 …  0.55676337  0.18723088
  -0.60030012]
 …
 [-0.85348893  1.69320278 -0.13151388 …  1.07553063  0.96983515
  -1.36190789]
 [-0.52023666 -0.29270541 -0.13151388 … -0.79203151 -1.72036703
   0.9229154 ]
 [-0.56820479 -0.79993851 -0.13151388 … -1.82956604 -1.72036703
  -0.60030012]]
```

Create a new Linear Regression model, fit on scaled X_train and y_train and predict on scaled X_test.

[90]:
```
#create a linear regression object
model4 = LinearRegression()
#train a model
model4.fit(ss3,encode_y_train)
```

[90]:
```
LinearRegression()
```

[91]:
```
s_y_pred = model4.predict(encode_X_test)
s_y_pred
```

```
[91]: array([9.16641201e+07, 1.01960588e+08, 1.17921652e+08, 1.18784635e+08,
             8.13404241e+07, 8.94732382e+07, 9.40521718e+07, 8.47305101e+07,
             1.11879748e+08, 6.47857271e+07, 1.39809605e+08, 8.72764978e+07,
             9.85254414e+07, 7.72023423e+07, 9.66862338e+07, 1.11663252e+08,
             9.90062546e+07, 1.29149393e+08, 1.53985324e+08, 1.13938057e+08,
             9.44874523e+07, 8.04543375e+07, 9.55535295e+07, 1.12715683e+08,
             8.45654831e+07, 9.46792646e+07, 9.30213017e+07, 9.51049792e+07,
             1.00308638e+08, 1.01110706e+08, 8.46569143e+07, 7.87684559e+07,
             9.51238547e+07, 8.44224001e+07, 9.64064389e+07, 7.14143473e+07,
             2.99870872e+08, 9.00930628e+07, 7.98589158e+07, 1.23314146e+08,
             9.47740853e+07, 1.36703065e+08, 8.92381830e+07, 8.29015280e+07,
             6.47684408e+07, 7.28783855e+07, 8.35825411e+07, 9.62460522e+07,
             1.56130989e+08, 1.30085628e+08, 9.41808575e+07, 1.37399172e+08,
             9.56875697e+07, 6.71284621e+07, 1.39568890e+08, 7.19868076e+07,
             1.40921072e+08, 1.36362965e+08, 1.27910779e+08, 1.02474794e+08,
             1.24479957e+08, 8.92606595e+07, 1.09407801e+08, 1.25688788e+08,
             9.02957006e+07, 9.37778793e+07, 9.30732934e+07, 9.02820844e+07,
             1.99388307e+08, 9.42320616e+07, 8.34053312e+07, 7.72624437e+07,
             1.10582394e+08, 9.87283474e+07, 1.15513216e+08, 7.25534410e+07,
             1.09310637e+08, 9.20374785e+07, 1.31179967e+08, 1.02381062e+08,
             1.24246165e+08, 9.98933331e+07, 1.22981220e+08, 1.14885993e+08,
             6.76937183e+07, 1.47239733e+08, 1.04789960e+08, 1.29851531e+08,
             8.07059974e+07, 9.80360183e+07, 1.01462812e+08, 9.19190997e+07,
             1.20607654e+08, 8.94451470e+07, 7.92758343e+07, 5.21040469e+07,
             9.70979648e+07, 1.33645982e+08, 9.75844215e+07, 1.16932286e+08,
             9.05383680e+07, 1.13395704e+08, 8.17611341e+07, 9.26553145e+07,
             7.30816587e+07, 1.02899776e+08, 9.39256261e+07, 1.10572573e+08,
             7.98994838e+07, 1.38512152e+08, 7.28357281e+07, 6.07388723e+07,
             1.20411275e+08, 1.24206979e+08, 1.15181569e+08, 1.56402983e+08,
             8.07768788e+07, 1.04060733e+08, 6.61875041e+07, 9.07279922e+07,
             7.65473156e+07, 8.37475103e+07, 9.16628008e+07, 1.14966565e+08,
             6.68177663e+07, 1.17358911e+08, 6.85002864e+07, 9.43845460e+07,
             1.65833402e+08, 1.04614663e+08, 1.14491876e+08, 1.13956884e+08,
             1.56405781e+08, 1.26236143e+08, 5.36109947e+07, 1.12691125e+08,
             9.23031865e+07, 2.28469833e+08, 1.16744290e+08, 7.67268299e+07,
             1.05680716e+08, 1.00118862e+08, 1.25669415e+08, 8.39872429e+07,
             1.09373936e+08, 7.01473628e+07, 1.09704528e+08, 1.01774303e+08,
             7.16728335e+07, 1.00959615e+08, 1.08175447e+08, 1.26176587e+08,
             1.99590228e+08, 5.92470972e+07, 9.66054120e+07, 9.24548203e+07,
             1.15713043e+08, 9.89009719e+07, 6.46164542e+07, 1.38205260e+08,
             8.06373480e+07, 1.10955137e+08, 6.51196433e+07, 6.58506181e+07,
             1.45993206e+08, 1.19771278e+08, 8.07381134e+07, 1.05959802e+08,
             8.59720795e+07, 8.23026924e+07, 6.08504443e+07, 6.63097150e+07,
             1.07485905e+08, 1.08918601e+08, 1.01261217e+08, 1.02021569e+08,
             7.29412137e+07, 5.95393189e+07, 8.83747718e+07, 7.86319076e+07,
             1.01661929e+08, 6.87087724e+07, 8.55632040e+07, 1.33760274e+08,
             7.09432761e+07, 7.60087911e+07, 7.35824812e+07, 1.03657494e+08,
```

```
6.49093376e+07, 8.58657662e+07, 7.44920196e+07, 1.07741758e+08,
6.97833653e+07, 1.04687913e+08, 8.21479326e+07, 1.18079129e+08,
6.32165786e+07, 1.02204371e+08, 6.36160742e+07, 8.38877131e+07,
9.87039930e+07, 9.19352380e+07, 1.30043791e+08, 7.07709388e+07,
5.18607555e+07, 1.04338825e+08, 1.64968639e+08, 7.17895141e+07,
6.00582080e+07, 8.67773943e+07, 1.62080488e+08, 7.24391264e+07,
8.89251420e+07, 8.16662594e+08, 8.81034385e+07, 1.34670048e+08,
9.22457377e+07, 1.19223531e+08, 1.18382877e+08, 9.39139341e+07,
1.19511333e+08, 9.68588909e+07, 1.21804997e+08, 7.92371538e+07,
9.11865528e+07, 1.15920774e+08, 1.05730375e+08, 1.26427349e+08,
7.99437311e+07, 9.32777696e+07, 1.04156997e+08, 8.48304495e+07,
8.72083166e+07, 1.15087361e+08, 7.97450318e+07, 1.08030183e+08,
1.07363092e+08, 1.29637269e+08, 1.17782376e+08, 9.84558173e+07,
1.07157171e+08, 1.59256698e+08, 1.68249069e+08, 9.85401439e+07,
1.23929236e+08, 1.05170368e+08, 1.49520016e+08, 9.43043341e+07,
7.32764182e+07, 1.06782692e+08, 1.05960393e+08, 9.59927249e+07,
9.08892952e+07, 1.04143658e+08, 1.02963385e+08, 7.80821520e+07,
1.05110574e+08, 1.08717449e+08, 1.02391918e+08, 1.05734320e+08,
4.89999891e+07, 7.22827480e+07, 9.74740462e+07, 1.29681685e+08,
7.46262426e+07, 1.06663755e+08, 9.31611382e+07, 1.52783153e+08,
9.97074504e+07, 1.14506874e+08, 1.18258264e+08, 7.47109510e+07,
7.50098534e+07, 9.95254797e+07, 9.28027774e+07, 8.30553627e+07,
7.24690740e+07, 7.14928073e+07, 7.99921243e+07, 1.00937130e+08,
5.94354435e+07, 6.58244375e+07, 1.10367044e+08, 1.22011916e+08,
7.93460638e+07, 3.03733902e+08, 8.47470568e+07, 1.39044001e+08,
1.07214468e+08, 8.95826846e+07, 2.82530972e+08, 9.23072760e+07,
8.85673716e+07, 1.17266244e+08, 9.41489065e+07, 9.01380777e+07,
8.43275899e+07, 1.01344499e+08, 7.35218838e+07, 1.11560783e+08,
8.01922500e+07, 9.94324541e+07, 1.00859252e+08, 7.45208247e+07,
8.12282741e+07, 6.89538819e+07, 1.30218132e+08, 9.01186100e+07,
9.29986175e+07, 1.06019768e+08, 6.49795908e+07, 7.83716852e+07,
1.15652778e+08, 8.66608872e+07, 9.38367180e+07, 1.17935343e+08,
1.02700950e+08, 1.02576625e+08, 9.58117017e+07, 2.03837209e+08,
9.78755175e+07, 6.52359539e+07, 9.76479776e+07, 1.07303616e+08,
1.18225721e+08, 1.11322943e+08, 6.19466718e+07, 9.03351056e+07,
6.86394862e+07, 6.87814966e+07, 2.00793971e+08, 9.00403543e+07,
9.94373605e+07, 6.15647734e+07, 2.03275965e+08, 1.01790898e+08,
1.36709677e+08, 7.66373011e+07, 8.83327752e+07, 7.92059860e+07,
8.90455807e+07, 5.96399320e+07, 1.19966136e+08, 7.85258597e+07,
6.77285855e+07])
```

Compute Mean Squared Error (MSE) on actual values and predicted values (you will get output as 1461036570.0).

[92]:
```
s_mse=mean_squared_error(encode_y_test,s_y_pred)
s_mse
```

[92]:
```
1.3194590569635376e+16
```

0.0.8 Step7. [Normalize using MinMaxScaler and Predict Sale Price]

Repeat Step6 using MinMaxScaler

[93]:
```
from sklearn.preprocessing import MinMaxScaler
mm_scaler = MinMaxScaler()
```

[94]:
```
mm_ss = mm_scaler.fit_transform(encode_X_train)
mm_ss
```

[94]:
```
array([[0.08462623, 0.33365617, 0.        , …, 0.50387597, 0.        ,
        1.        ],
       [0.17395393, 0.        , 0.        , …, 0.72093023, 0.83333333,
        0.5       ],
       [0.24823695, 0.        , 0.        , …, 0.89147287, 0.76666667,
        0.        ],
       …,
       [0.22661025, 0.        , 0.        , …, 0.97674419, 0.95      ,
        0.25      ],
       [0.32675129, 0.        , 0.        , …, 0.98449612, 0.95      ,
        0.25      ],
       [0.12858486, 0.        , 0.        , …, 0.75193798, 0.45      ,
        0.5       ]])
```

[95]:
```
mm_ss5 = mm_scaler.transform(X_test)
mm_ss5
```

[95]:
```
array([[0.12858486, 0.        , 0.        , …, 0.75193798, 0.45      ,
        0.5       ],
       [0.37141514, 0.        , 0.        , …, 0.99224806, 0.98333333,
        1.        ],
       [0.1274095 , 0.        , 0.        , …, 0.75193798, 0.45      ,
        0.75      ],
       …,
       [0.20122238, 0.        , 0.        , …, 0.97674419, 0.93333333,
        0.        ],
       [0.14621533, 0.        , 0.        , …, 0.68217054, 0.85      ,
        1.        ],
       [0.07898449, 0.31767554, 0.        , …, 0.93023256, 0.83333333,
        0.25      ]])
```

[96]:
```
model5 =LinearRegression()
model5.fit(mm_ss,encode_y_train)
```

[96]:
```
LinearRegression()
```

[97]:
```
mm_y_pred = model5.predict(mm_ss5)
mm_y_pred
```

```
[97]: array([131797.95467908, 311704.86657981, 139200.22539657, 313443.20643999,
             252797.86648437, 123046.91784846, 117378.5017364 , 222172.6976128 ,
             225919.4630138 , 180879.68083942, 173522.13201178, 147080.98290307,
             204272.18383693, 100657.93059886,  97695.61667541, 191181.57760294,
             308238.9550052 , 132857.42712151, 141577.62792228, 113675.08061916,
             133983.75273388, 106062.11127522, 235032.39810745, 125903.62241075,
             189230.78678897, 199644.5909249 , 184328.28307383, 147043.47428562,
             204828.93843517, 191484.14923426, 114409.9774009 , 171845.43746774,
             176599.78087299, 204242.60775792, 250560.48788192, 216913.12810681,
             232734.08171018, 211273.28920557, 205569.63752126, 167472.87343678,
              58478.58513446, 113005.21841861, 157012.7838008 , 265221.32286705,
             168950.64885348, 152213.169075  , 167451.55099716, 328199.39199748,
             182010.58031063, 203642.03567921, 147973.13197864,  96325.89094823,
             227570.84723723,  87289.27444502, 169799.61515644, 202276.66487611,
             158903.08731367, 236695.18810547, 200704.90091373,  70912.58735497,
             372414.87062263, 120904.75499932,  96877.38457711, 180410.18529927,
             131992.24757059, 112748.40036357, 374947.25696151, 208122.54514936,
             245464.72312833, 171561.09113784, 135713.20259672, 201627.87982328,
             190963.78357114, 280290.68776881, 134900.00708843, 152486.91542099,
             168058.4776116 , 263951.56875983, 208821.17236003, 108191.58736302,
             119009.30046201, 229225.62046382, 171340.68202927, 301868.29481478,
             168344.11865204, 124946.7784666 , 202924.04278927, 283129.01806473,
              90321.09801381, 111434.79492568, 188279.584864  , 310253.08207238,
             124177.88215722, 191364.53226893, 240510.01635049, 137097.71622446,
             224450.7257173 , 186186.58033816, 174408.02921481, 106497.17048662,
             222255.09745691, 187107.9138167 , 171013.26852221, 286300.5193322 ,
             151816.48733704, 192330.6818937 , 121339.24136301, 187943.44089951,
              69672.21423892, 226013.06200258, 157736.36367454, 125614.14239237,
              82370.14313929, 234633.38466123, 174849.87891931, 228994.29928491,
             302419.37850495, 372850.36173382, 148534.95382617, 125161.76106928,
             160958.73940291, 244633.43256371, 185104.44245956, 109348.07455719,
             142219.41722934, 170693.92853793, 289372.03297905, 203836.88086247,
             162952.99027016,  87353.3701625 , 217253.70326651, 229022.16508795,
             137336.7806884 ,  45810.62635894,  83810.06542091, 292457.11129938,
             106603.3551778 , 205243.88581606, 241796.97393948, 259925.58173828,
             190073.21382609,  60190.08832769, 179326.57329592, 268759.6820534 ,
             219695.3821903 , 110159.98639961, 151730.09618353, 189467.95297045,
             238896.11051135, 154538.69978151, 179535.24405334, 157226.4727512 ,
             206920.7651633 ,  87488.18342567, 318281.55475284, 121592.17926646,
             225079.95899261,  99206.43007536, 350886.78442964,  64911.93346957,
             199162.10444576, 165041.44922353, 198408.78376549, 342284.15682558,
             253629.30319572, 141702.38177368, 223936.79704981, 221763.0218073 ,
              71748.62376294, 216749.38023036, 209819.305251  , 247398.57461258,
             148818.45871333, 134299.956664  , 208529.84307596, 121718.46776888,
             246796.42571128, 221822.48322739, 150426.82848835,  91038.74809643,
             210250.2818477 , 193469.14479176, 126743.41868673, 159628.50146886,
             154222.3116162 , 164828.96002419,  58346.59238042, 236030.91601818,
```

```
163337.84431464, 197506.33333978, 189900.67153842, 202065.55393313,
143427.31590278,  91941.75422128, 194033.123235  , 122288.44223573,
219835.3266294 , 203509.58311323, 277486.32626813, 113393.43516722,
 71434.11407794, 128133.0614369 , 232835.91042458, 126345.58693362,
213757.2855931 , 188461.28661533, 228138.77777231, 113286.08499329,
199954.19844488, 229168.14499803, 126662.70717147, 151343.46012082,
132467.27566252, 119722.60839977, 166888.64391534, 170015.30933012,
211776.03931884, 249508.84112791, 225821.34852512, 237642.86126766,
134845.09515712, 139693.30522558, 166225.66762986, 167412.08988042,
197306.88021357, 128962.76955742, 149439.48347799, 137780.58869745,
113250.09453629, 211294.82771477, 256438.20380864,  91527.76302953,
315648.3103501 , 153349.04140675, 223484.37770866, 220406.03575001,
176244.49900157, 326154.33124785, 240139.27500634, 246194.02367624,
163370.15493512, 158774.43897576, 168062.08340555, 150938.68882643,
237521.15280443, 118350.86924399, 152772.7120928 , 189182.40557759,
 94156.03258368, 136728.36963522, 323253.92392125, 306444.83659023,
197364.56585763, 191515.20856002, 162528.96358485, 158805.30420469,
174483.07513334, 339453.10792647, 259218.29023872, 256269.4913154 ,
 81078.92710764, 179216.09079246, 196053.7999349 ,  69925.56921821,
 77524.44997807, 278729.65010917, 136737.19550606, 159011.75015581,
141549.47303786, 107016.90710112,  56188.48328662, 257486.98554138,
283512.88170101, 110833.17828138, 140185.37989928, 208750.68153392,
146653.62396668,  86351.99804667, 366840.92399517, 210497.32464546,
160332.82904559, 345131.72599683, 209895.385998  , 170754.79172982,
196763.59104331, 183495.97878296, 125440.12321561, 210270.16586046,
157561.02922355, 183617.37524351, 124347.42275114, 209237.40519505,
146365.42082202, 128348.35137953, 228287.30600781,  58502.20696902,
202053.93165781,  60633.65728011, 327693.85423899, 220571.15035579,
266599.32182306, 205775.00759204, 126943.75499451, 328475.86147282,
174599.88447912, 153277.84921259, 198858.71075878, 153370.43576296,
210658.84725672,  42328.15085915, 165274.73178474, 311684.57942758,
282407.75408125, 137053.12462343, 115903.56228679, 201504.45607005,
201221.62291035, 217801.94377652, 143893.6090452 , 117069.72214884,
196117.41625152, 260669.23754427, 120909.63455586, 221838.58604191,
232625.94937109, 181106.71333204, 319054.33392657, 186768.41427705,
286314.57583226, 135045.22806807, 203596.65767939, 199064.41622639,
116994.40277252,  76661.25164569, 203813.8622668 , 186125.05188484,
295016.0751565 , 148796.46199888, 108086.23253092, 232920.80084236,
138691.16327863, 215651.2658567 , 209239.9661726 , 150208.0263511 ,
194551.01171507])
```

Mean Squared Error will be: 1461036570.0

[98]:
```
mse_mm=mean_squared_error(encode_y_test,mm_y_pred)
mse_mm
```

[98]: 9404508368.107819

0.0.9 Step8. [Predict using SGD Regressor]

Use scaled X_train and X_test using StandardScaler that you computed before

[99]:
```
from sklearn.linear_model import SGDRegressor
from sklearn.pipeline import make_pipeline
```

Create SGDRegressor, fit and predict

[100]:
```
seg = make_pipeline(StandardScaler(), SGDRegressor(max_iter=1000, tol=1e-3))
seg.fit(X, y)
```

[100]:
```
Pipeline(steps=[('standardscaler', StandardScaler()),
                ('sgdregressor', SGDRegressor())])
```

[101]:
```
re_y_pred = seg.predict(X)
re_y_pred
```

[101]:
```
array([228671.31817897, 200777.00895064, 222178.96664092, …,
       229832.92538152, 128254.43107227, 155640.32874484])
```

Compute MSE on y_test and y_pred this time. You will get output as 1592430104.0

[103]:
```
mse_seg=mean_squared_error(y,re_y_pred)
mse_seg
```

[103]:
```
1207136617.6570287
```

0.0.10 Step8. [Predict using Ridge Regression]

Create RidgeCV, fit and predict

[118]:
```
from sklearn.linear_model import Ridge
```

[119]:
```
RidgeCV = Ridge(alpha=1.0)
RidgeCV.fit(ss3,encode_y_train)
rid_y_pred = RidgeCV.predict(ss3)
rid_y_pred
```

[119]:
```
array([129474.25435239, 154299.40507943, 226646.37404798, …,
       213069.56190215, 319814.40859995, 131879.06092969])
```

Compute MSE on y_test and y_pred this time. You will get output as 1442196000.3367693.

[120]:
```
mse_rid=mean_squared_error(encode_y_train,rid_y_pred)
mse_rid
```

[120]:
```
1316556019.1346939
```

0.0.11 Step8. [Predict using Lasso Regression]

[121]:
```
from sklearn.linear_model import Lasso
```

Create LassoCV, fit and predict

[122]:
```
LasCV = Lasso(alpha=1.0)
LasCV.fit(ss3,encode_y_train)
```

```
C:\ProgramData\Anaconda3\lib\site-
packages\sklearn\linear_model\_coordinate_descent.py:529: ConvergenceWarning:
Objective did not converge. You might want to increase the number of iterations.
Duality gap: 305643652788.40485, tolerance: 675327095.0964903
  model = cd_fast.enet_coordinate_descent(
```

[122]:
```
Lasso()
```

[123]:
```
las_y_pred = LasCV.predict(ss3)
las_y_pred
```

[123]:
```
array([129497.73854696, 154226.11412444, 226690.0456038 , …,
       213090.74498072, 319851.56030583, 131789.02164197])
```

Compute MSE on y_test and y_pred this time. You will get output as 1409368613.5329669

[124]:
```
mse_las=mean_squared_error(encode_y_train,las_y_pred)
mse_las
```

[124]:
```
1316552641.500684
```

0.0.12 Step9.[RMSE]. Print Root Mean Squared Error values (use numpy.sqrt() method) as below and compare error values.

RMSE without one hot encoding: 38403.0

RMSE with One hot encoding: 38224.0

RMSE with OHE and Standard Scaling: 38224.0

RMSE with OHE and MinMax Scaling: 38224.0

RMSE of SGDRegressor with OHE and Standard Scaler: 38528.0

RMSE of RidgeCV with OHE and Standard Scaler: 37976.0

RMSE of LassoCV with OHE and Standard Scaler: 37542.0

[125]:
```
from math import sqrt
```

[126]:
```
print("RMSE without one hot encoding:  ",sqrt(mse))
print("RMSE with one hot encoding:  ",sqrt(mse1))
print("RMSE with one and Standard Scaling:  ",sqrt(s_mse))
print("RMSE with one and MinMax Scaling:  ",sqrt(mse_mm))
print("RMSE of SGDRegressor with one and Standard Scaler:  ",sqrt(mse_seg))
print("RMSE of RigdCV with one and Standard Scaler:  ",sqrt(mse_rid))
print("RMSE of LassoCV with one and Standard Scaler:  ",sqrt(mse_las))
```

```
RMSE without one hot encoding:   33579.689180761234
RMSE with one hot encoding:   28995.903894104937
RMSE with one and Standard Scaling:   114867708.99445751
RMSE with one and MinMax Scaling:   96976.8444944865
RMSE of SGDRegressor with one and Standard Scaler:   34743.87165612129
RMSE of RigdCV with one and Standard Scaler:   36284.37706692363
RMSE of LassoCV with one and Standard Scaler:   36284.3305229776
```