S3 Bucket to Sage Maker-AWS

In [6]:
```python
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
```

In [2]:
```python
data=pd.read_csv("C:\\Users\\User\\Desktop\\train-1.csv")
```

In [3]:
```python
data
```

Out[3]:

	Id	MSSubClass	MSZoning	LotFrontage	LotArea	Street	Alley	LotShape	LandContour	Utilities	...	PoolArea	PoolQC	Fence	MiscFeature	MiscVal	MoSold	YrSold	SaleType	SaleCondit
0	1	60	RL	65.0	8450	Pave	NaN	Reg	Lvl	AllPub	...	0	NaN	NaN	NaN	0	2	2008	WD	Nor
1	2	20	RL	80.0	9600	Pave	NaN	Reg	Lvl	AllPub	...	0	NaN	NaN	NaN	0	5	2007	WD	Nor
2	3	60	RL	68.0	11250	Pave	NaN	IR1	Lvl	AllPub	...	0	NaN	NaN	NaN	0	9	2008	WD	Nor
3	4	70	RL	60.0	9550	Pave	NaN	IR1	Lvl	AllPub	...	0	NaN	NaN	NaN	0	2	2006	WD	Abno
4	5	60	RL	84.0	14260	Pave	NaN	IR1	Lvl	AllPub	...	0	NaN	NaN	NaN	0	12	2008	WD	Nor
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
1455	1456	60	RL	62.0	7917	Pave	NaN	Reg	Lvl	AllPub	...	0	NaN	NaN	NaN	0	8	2007	WD	Nor
1456	1457	20	RL	85.0	13175	Pave	NaN	Reg	Lvl	AllPub	...	0	NaN	MnPrv	NaN	0	2	2010	WD	Nor
1457	1458	70	RL	66.0	9042	Pave	NaN	Reg	Lvl	AllPub	...	0	NaN	GdPrv	Shed	2500	5	2010	WD	Nor
1458	1459	20	RL	68.0	9717	Pave	NaN	Reg	Lvl	AllPub	...	0	NaN	NaN	NaN	0	4	2010	WD	Nor
1459	1460	20	RL	75.0	9937	Pave	NaN	Reg	Lvl	AllPub	...	0	NaN	NaN	NaN	0	6	2008	WD	Nor

1460 rows × 81 columns

Data Cleaning

In [7]:
```python
data.shape
```

Out[7]:
```
(1460, 81)
```

In [4]:
```python
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1460 entries, 0 to 1459
Data columns (total 81 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   Id             1460 non-null   int64
 1   MSSubClass     1460 non-null   int64
 2   MSZoning       1460 non-null   object
 3   LotFrontage    1201 non-null   float64
 4   LotArea        1460 non-null   int64
 5   Street         1460 non-null   object
 6   Alley          91 non-null     object
 7   LotShape       1460 non-null   object
 8   LandContour    1460 non-null   object
 9   Utilities      1460 non-null   object
 10  LotConfig      1460 non-null   object
 11  LandSlope      1460 non-null   object
 12  Neighborhood   1460 non-null   object
 13  Condition1     1460 non-null   object
 14  Condition2     1460 non-null   object
 15  BldgType       1460 non-null   object
 16  HouseStyle     1460 non-null   object
 17  OverallQual    1460 non-null   int64
 18  OverallCond    1460 non-null   int64
 19  YearBuilt      1460 non-null   int64
 20  YearRemodAdd   1460 non-null   int64
 21  RoofStyle      1460 non-null   object
 22  RoofMatl       1460 non-null   object
 23  Exterior1st    1460 non-null   object
 24  Exterior2nd    1460 non-null   object
 25  MasVnrType     1452 non-null   object
 26  MasVnrArea     1452 non-null   float64
 27  ExterQual      1460 non-null   object
 28  ExterCond      1460 non-null   object
 29  Foundation     1460 non-null   object
 30  BsmtQual       1423 non-null   object
 31  BsmtCond       1423 non-null   object
 32  BsmtExposure   1422 non-null   object
 33  BsmtFinType1   1423 non-null   object
 34  BsmtFinSF1     1460 non-null   int64
 35  BsmtFinType2   1422 non-null   object
 36  BsmtFinSF2     1460 non-null   int64
 37  BsmtUnfSF      1460 non-null   int64
 38  TotalBsmtSF    1460 non-null   int64
 39  Heating        1460 non-null   object
 40  HeatingQC      1460 non-null   object
 41  CentralAir     1460 non-null   object
 42  Electrical     1459 non-null   object
 43  1stFlrSF       1460 non-null   int64
 44  2ndFlrSF       1460 non-null   int64
 45  LowQualFinSF   1460 non-null   int64
 46  GrLivArea      1460 non-null   int64
 47  BsmtFullBath   1460 non-null   int64
 48  BsmtHalfBath   1460 non-null   int64
 49  FullBath       1460 non-null   int64
 50  HalfBath       1460 non-null   int64
 51  BedroomAbvGr   1460 non-null   int64
 52  KitchenAbvGr   1460 non-null   int64
 53  KitchenQual    1460 non-null   object
 54  TotRmsAbvGrd   1460 non-null   int64
 55  Functional     1460 non-null   object
 56  Fireplaces     1460 non-null   int64
 57  FireplaceQu    770 non-null    object
 58  GarageType     1379 non-null   object
 59  GarageYrBlt    1379 non-null   float64
 60  GarageFinish   1379 non-null   object
 61  GarageCars     1460 non-null   int64
 62  GarageArea     1460 non-null   int64
 63  GarageQual     1379 non-null   object
 64  GarageCond     1379 non-null   object
 65  PavedDrive     1460 non-null   object
 66  WoodDeckSF     1460 non-null   int64
 67  OpenPorchSF    1460 non-null   int64
 68  EnclosedPorch  1460 non-null   int64
 69  3SsnPorch      1460 non-null   int64
 70  ScreenPorch    1460 non-null   int64
 71  PoolArea       1460 non-null   int64
 72  PoolQC         7 non-null      object
 73  Fence          281 non-null    object
 74  MiscFeature    54 non-null     object
 75  MiscVal        1460 non-null   int64
 76  MoSold         1460 non-null   int64
 77  YrSold         1460 non-null   int64
 78  SaleType       1460 non-null   object
 79  SaleCondition  1460 non-null   object
 80  SalePrice      1460 non-null   int64
dtypes: float64(3), int64(35), object(43)
memory usage: 924.0+ KB
```

In [5]:
```python
data.describe()
```

Out[5]:

	Id	MSSubClass	LotFrontage	LotArea	OverallQual	OverallCond	YearBuilt	YearRemodAdd	MasVnrArea	BsmtFinSF1	...	WoodDeckSF	OpenPorchSF	EnclosedPorch
count	1460.000000	1460.000000	1201.000000	1460.000000	1460.000000	1460.000000	1460.000000	1460.000000	1452.000000	1460.000000	...	1460.000000	1460.000000	1460.000000
mean	730.500000	56.897260	70.049958	10516.828082	6.099315	5.575342	1971.267808	1984.865753	103.685262	443.639726	...	94.244521	46.660274	21.954110
std	421.610009	42.300571	24.284752	9981.264932	1.382997	1.112799	30.202904	20.645407	181.066207	456.098091	...	125.338794	66.256028	61.119149
min	1.000000	20.000000	21.000000	1300.000000	1.000000	1.000000	1872.000000	1950.000000	0.000000	0.000000	...	0.000000	0.000000	0.000000
25%	365.750000	20.000000	59.000000	7553.500000	5.000000	5.000000	1954.000000	1967.000000	0.000000	0.000000	...	0.000000	0.000000	0.000000
50%	730.500000	50.000000	69.000000	9478.500000	6.000000	5.000000	1973.000000	1994.000000	0.000000	383.500000	...	0.000000	25.000000	0.000000
75%	1095.250000	70.000000	80.000000	11601.500000	7.000000	6.000000	2000.000000	2004.000000	166.000000	712.250000	...	168.000000	68.000000	0.000000
max	1460.000000	190.000000	313.000000	215245.000000	10.000000	9.000000	2010.000000	2010.000000	1600.000000	5644.000000	...	857.000000	547.000000	552.000000

8 rows × 38 columns

In [9]:
```python
print(data.SalePrice.value_counts())
data['SalePrice'].value_counts(normalize=True)
```

```
140000    20
135000    17
155000    14
145000    14
190000    13
          ..
202665     1
164900     1
208300     1
181500     1
147500     1
Name: SalePrice, Length: 663, dtype: int64
```

Out[9]:
```
140000    0.013699
135000    0.011644
155000    0.009589
145000    0.009589
190000    0.008904
            ...
202665    0.000685
164900    0.000685
208300    0.000685
181500    0.000685
147500    0.000685
Name: SalePrice, Length: 663, dtype: float64
```

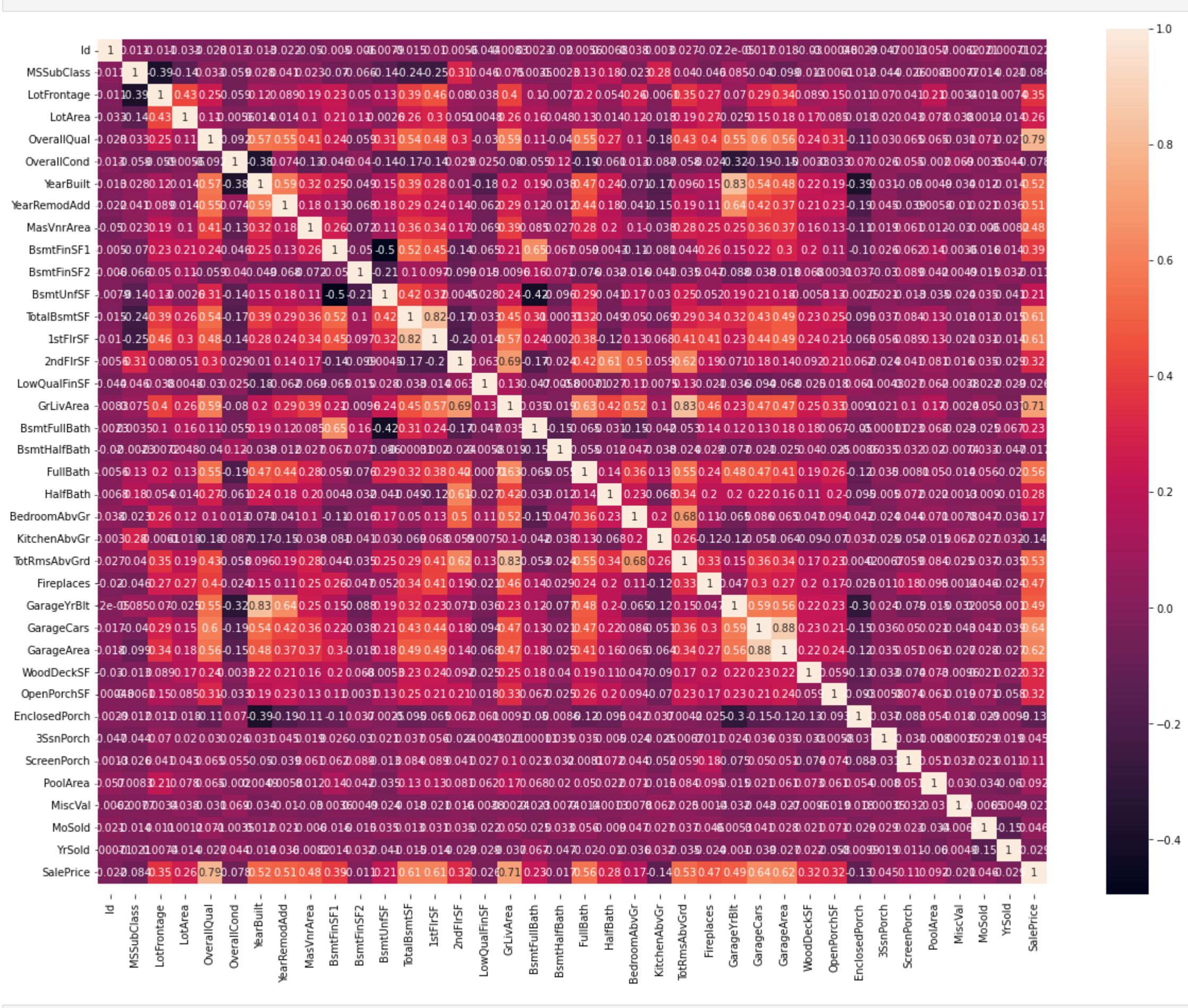
In [11]:
```python
data.isnull().sum()
```

Out[11]:
```
Id                 0
MSSubClass         0
MSZoning           0
LotFrontage      259
LotArea            0
                ...
MoSold             0
YrSold             0
SaleType           0
SaleCondition      0
SalePrice          0
Length: 81, dtype: int64
```

In [13]:
```python
data['Alley'].isnull().sum()
```

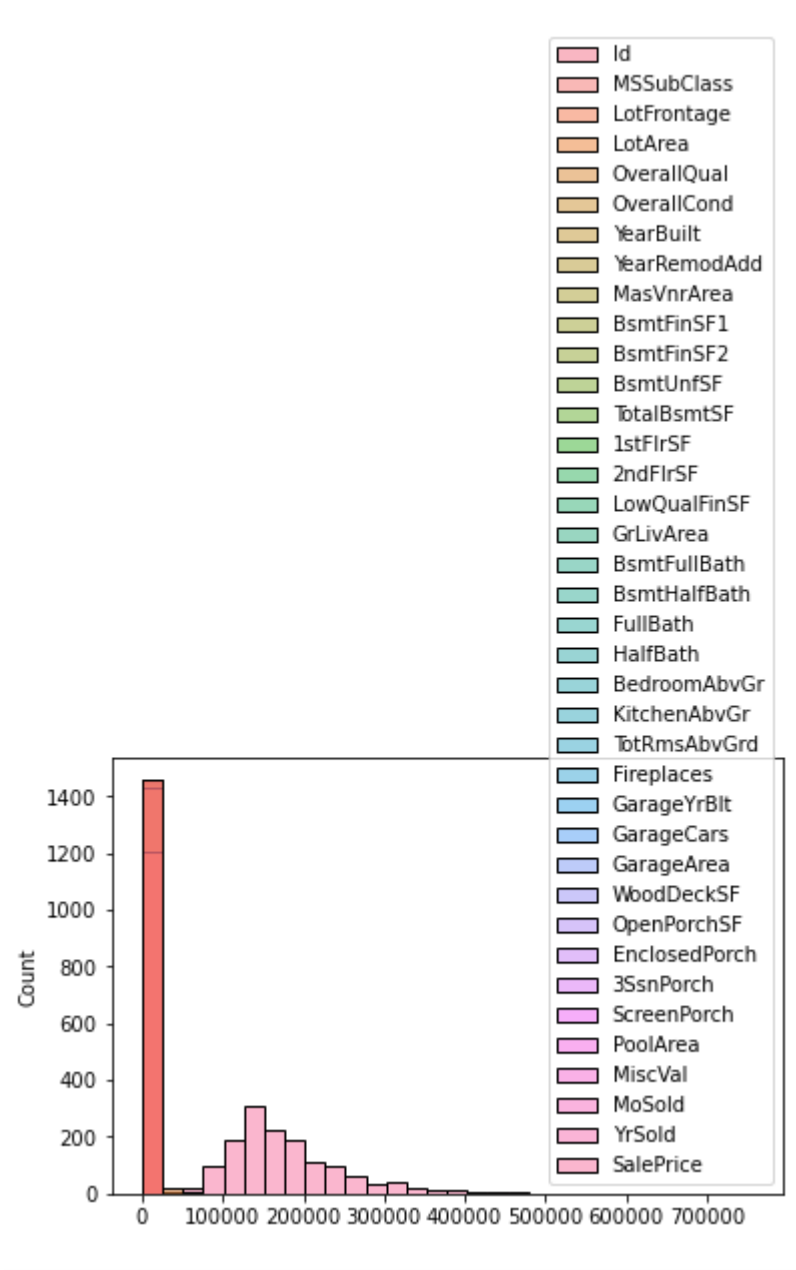
Out[13]:
```
1369
```

In [15]:
```python
#Heatmap to show the correlation between various variables of the dataset

plt.figure(figsize=(20, 15))
cor = data.corr()
ax= sns.heatmap(cor,annot=True)
bottom, top= ax.get_ylim()
ax.set_ylim(bottom + 0.5, top - 0.5)
plt.show()
```

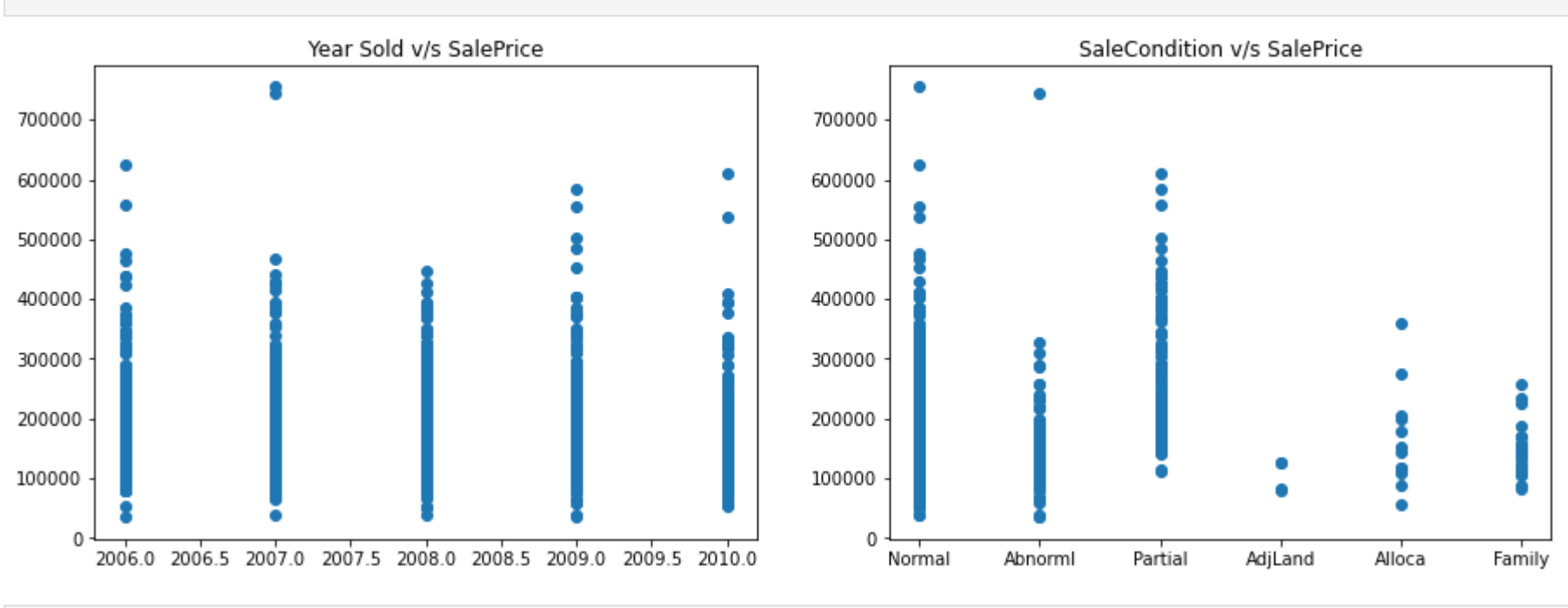


In [17]:
```python
sns.histplot(data=data,bins=30)
```

Out[17]:
```
<AxesSubplot:ylabel='Count'>
```



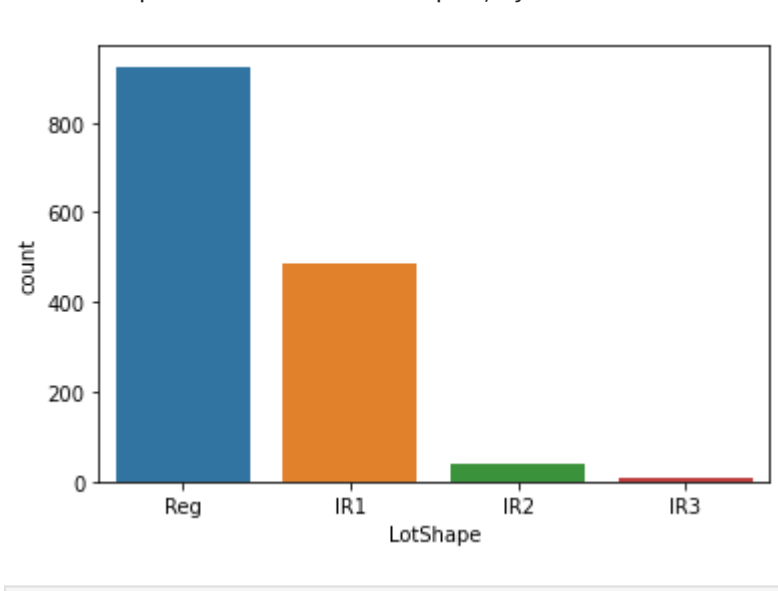
In [21]:
```python
fig, (axl, ax2) = plt.subplots(1,2,figsize = (15,5))

#scatter plot 1
axl.scatter(x=data['YrSold'],y= data['SalePrice'])
axl.set_title('Year Sold v/s SalePrice')

#scatter plot 2
ax2.scatter(x=data['SaleCondition'],y=data['SalePrice'])
ax2.set_title('SaleCondition v/s SalePrice')

plt.draw()
```



In [23]:
```python
sns.countplot(data['LotShape'])
```

```
C:\Users\User\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variable as a keyword arg: x. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```

Out[23]:
```
<AxesSubplot:xlabel='LotShape', ylabel='count'>
```



In [24]:
```python
data.isna().any()[lambda x: x]
```

Out[24]:
```
LotFrontage     True
Alley           True
MasVnrType      True
MasVnrArea      True
BsmtQual        True
BsmtCond        True
BsmtExposure    True
BsmtFinType1    True
BsmtFinType2    True
Electrical      True
FireplaceQu     True
GarageType      True
GarageYrBlt     True
GarageFinish    True
GarageQual      True
GarageCond      True
PoolQC          True
Fence           True
MiscFeature     True
dtype: bool
```

In [26]:
```python
data[data.columns[data.isnull().any()]]
```

Out[26]:

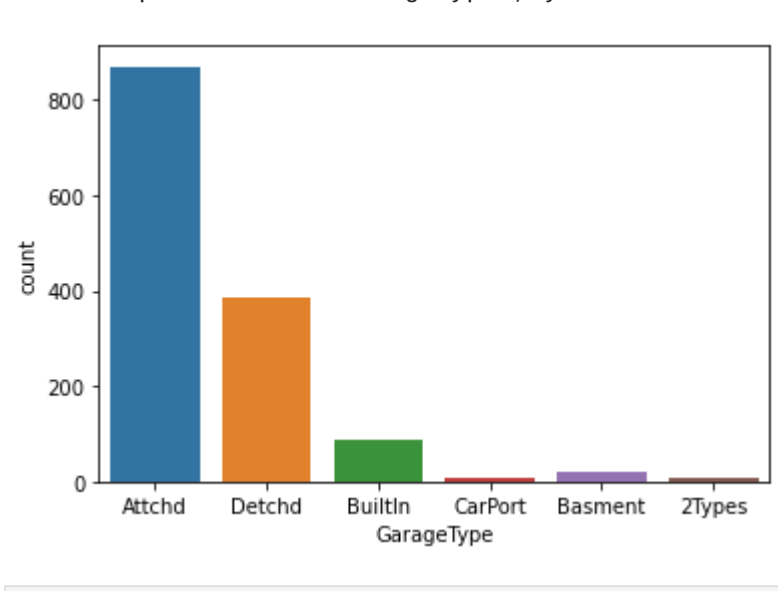
	LotFrontage	Alley	MasVnrType	MasVnrArea	BsmtQual	BsmtCond	BsmtExposure	BsmtFinType1	BsmtFinType2	Electrical	FireplaceQu	GarageType	GarageYrBlt	GarageFinish	GarageQual
0	65.0	NaN	BrkFace	196.0	Gd	TA	No	GLQ	Unf	SBrkr	NaN	Attchd	2003.0	RFn	TA
1	80.0	NaN	None	0.0	Gd	TA	Gd	ALQ	Unf	SBrkr	TA	Attchd	1976.0	RFn	TA
2	68.0	NaN	BrkFace	162.0	Gd	TA	Mn	GLQ	Unf	SBrkr	TA	Attchd	2001.0	RFn	TA
3	60.0	NaN	None	0.0	TA	Gd	No	ALQ	Unf	SBrkr	Gd	Detchd	1998.0	Unf	TA
4	84.0	NaN	BrkFace	350.0	Gd	TA	Av	GLQ	Unf	SBrkr	TA	Attchd	2000.0	RFn	TA
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
1455	62.0	NaN	None	0.0	Gd	TA	No	Unf	Unf	SBrkr	TA	Attchd	1999.0	RFn	TA
1456	85.0	NaN	Stone	119.0	Gd	TA	No	ALQ	Rec	SBrkr	TA	Attchd	1978.0	Unf	TA
1457	66.0	NaN	None	0.0	TA	Gd	No	GLQ	Unf	SBrkr	Gd	Attchd	1941.0	RFn	TA
1458	68.0	NaN	None	0.0	TA	TA	Mn	GLQ	Rec	FuseA	NaN	Attchd	1950.0	Unf	TA
1459	75.0	NaN	None	0.0	TA	TA	No	BLQ	LwQ	SBrkr	NaN	Attchd	1965.0	Fin	TA

1460 rows × 19 columns

In [28]:
```python
sns.countplot(data['GarageType'])
```

```
C:\Users\User\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variable as a keyword arg: x. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```

Out[28]:
```
<AxesSubplot:xlabel='GarageType', ylabel='count'>
```



In [29]:
```python
data=data.drop(['Alley','PoolQC','Fence','MiscFeature'],axis=1)
data.head()
```

Out[29]:

	Id	MSSubClass	MSZoning	LotFrontage	LotArea	Street	LotShape	LandContour	Utilities	LotConfig	...	EnclosedPorch	3SsnPorch	ScreenPorch	PoolArea	MiscVal	MoSold	YrSold	SaleType
0	1	60	RL	65.0	8450	Pave	Reg	Lvl	AllPub	Inside	...	0	0	0	0	0	2	2008	WD
1	2	20	RL	80.0	9600	Pave	Reg	Lvl	AllPub	FR2	...	0	0	0	0	0	5	2007	WD
2	3	60	RL	68.0	11250	Pave	IR1	Lvl	AllPub	Inside	...	0	0	0	0	0	9	2008	WD
3	4	70	RL	60.0	9550	Pave	IR1	Lvl	AllPub	Corner	...	272	0	0	0	0	2	2006	WD
4	5	60	RL	84.0	14260	Pave	IR1	Lvl	AllPub	FR2	...	0	0	0	0	0	12	2008	WD

5 rows × 77 columns

In [31]:
```python
data1=pd.get_dummies(data)
data1
```

Out[31]:

	Id	MSSubClass	LotFrontage	LotArea	OverallQual	OverallCond	YearBuilt	YearRemodAdd	MasVnrArea	BsmtFinSF1	...	SaleType_ConLw	SaleType_New	SaleType_Oth	SaleType_WD
0	1	60	65.0	8450	7	5	2003	2003	196.0	706	...	0	0	0	1
1	2	20	80.0	9600	6	8	1976	1976	0.0	978	...	0	0	0	1
2	3	60	68.0	11250	7	5	2001	2002	162.0	486	...	0	0	0	1
3	4	70	60.0	9550	7	5	1915	1970	0.0	216	...	0	0	0	1
4	5	60	84.0	14260	8	5	2000	2000	350.0	655	...	0	0	0	1
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
1455	1456	60	62.0	7917	6	5	1999	2000	0.0	0	...	0	0	0	1
1456	1457	20	85.0	13175	6	6	1978	1988	119.0	790	...	0	0	0	1
1457	1458	70	66.0	9042	7	9	1941	2006	0.0	275	...	0	0	0	1
1458	1459	20	68.0	9717	5	6	1950	1996	0.0	49	...	0	0	0	1
1459	1460	20	75.0	9937	5	6	1965	1965	0.0	830	...	0	0	0	1

1460 rows × 277 columns

In [34]:
```python
X=data1.dropna(axis=1)
y=data['SalePrice']
```

Train and test

In [36]:
```python
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X,y,test_size = 0.30, random_state=101)
print(X_train.shape)
print(X_test.shape)
print(y_train.shape)
print(y_test.shape)
```

```
(1022, 274)
(438, 274)
(1022,)
(438,)
```

In [37]:
```python
from sklearn.linear_model import LinearRegression
lm = LinearRegression()
lm.fit(X_train,y_train)
```

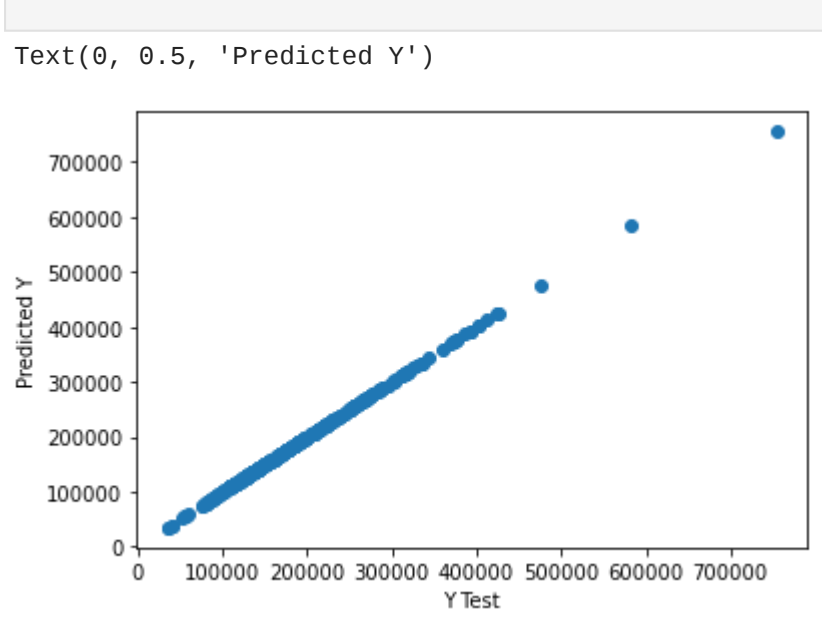
Out[37]:
```
LinearRegression
LinearRegression()
```

In [38]:
```python
#The coefficients
print('Coefficients: \n', lm.coef_)
```

```
Coefficients:
 [-3.77798998e-16  4.28546088e-13  1.55431223e-15  3.07819499e-11
  1.69110834e-11  5.63302009e-13 -8.16700006e-13 -6.86299975e-15
  1.47970286e-14  2.58130919e-14  3.37263854e-14 -1.40072415e-14
  7.50469565e-14 -3.03499088e-14  3.07008379e-14  1.22244827e-11
 -1.05081409e-12 -8.57495911e-12  1.73012189e-12  1.29700339e-12
  4.34313444e-13  6.40625184e-13 -1.03116041e-11  6.37297638e-12
 -4.91306214e-14  4.76145273e-14 -5.83365550e-14 -4.80981899e-14
  3.72313861e-13 -2.52549175e-14 -1.19631105e-13 -4.99534225e-14
  1.24271303e-12 -4.48370283e-12  1.00000000e+00 -3.80196935e-12
  4.94809626e-12 -2.24818297e-11  3.15449688e-12  1.81812058e-11
 -4.71952013e-12  4.71952013e-12 -7.93444132e-12  4.05583911e-11
 -2.44657599e-11 -8.15818996e-12  5.30933829e-12  4.95668186e-12
 -1.63316439e-11  6.06562369e-12  1.97379498e-11 -1.97379498e-11
 -8.93766052e-13  9.34829954e-12  5.66519847e-12 -1.54339855e-11
  1.31425354e-12 -7.16932474e-12  1.02066505e-12  6.14865970e-12
  1.72997615e-11  1.47439682e-11 -4.62759266e-12 -8.32636266e-12
 -6.84151769e-12 -9.04470126e-13 -2.41125335e-11 -7.39446638e-12
  8.83063487e-13 -1.06083208e-11  4.92969023e-12  1.12663604e-11
 -1.88620125e-11 -9.35310565e-12  1.60892203e-11  8.53372169e-12
 -4.75870821e-12 -1.87972440e-11  1.31275244e-11 -1.21033841e-11
  1.93785281e-11  1.63999799e-11  6.57776834e-13  2.72955120e-12
  6.50572086e-13  1.06681412e-11 -5.48155353e-12 -8.16941725e-14
 -4.16645474e-11  1.02280226e-10 -1.33982799e-11 -2.56984692e-12
 -4.00285085e-11 -9.72393655e-12 -1.05943866e-10  5.56530129e-11
  3.94728613e-11 -1.65152038e-10 -6.73974534e-11  1.80956261e-10
  8.95495098e-12  5.34562719e-11  1.38256106e-12 -1.10570984e-11
  7.56335748e-12 -2.63920322e-12  4.75038315e-12 -2.72187429e-12
  1.02438204e-11  4.11648608e-12 -1.56970533e-11  1.85318344e-12
  6.85833973e-12 -3.97386898e-12 -6.79032951e-13 -7.65663208e-11
  1.35070381e-11  2.10768819e-11  1.34728838e-11  6.74299775e-12
  2.17665192e-11  3.17637355e-22 -3.58259223e-11  1.32271119e-10
  1.26762408e-11 -1.61116938e-11  3.08971576e-11 -4.86278195e-11
 -7.52790820e-11  4.22851218e-11 -1.25119647e-11 -2.65340113e-11
  6.42521577e-11 -1.59961757e-11  4.27243360e-13 -5.14993167e-11
  3.84473715e-11 -7.96717818e-12 -9.93482609e-12 -1.58818678e-22
  1.14291437e-13  2.14307916e-12 -5.50041711e-12 -1.77253752e-11
 -3.52143652e-11  1.06369736e-11  2.46743071e-11 -6.52311422e-11
 -1.59961757e-11  8.24460102e-12  5.12598070e-11 -2.78162194e-11
  1.27808975e-11  3.47351747e-11  8.73083226e-12 -1.29853456e-11
 -8.95961644e-12 -2.68219319e-13  4.99351168e-12  1.04149790e-11
 -1.07254030e-11 -2.17060567e-11 -1.05708763e-11 -1.67821471e-11
  4.82642734e-12  1.41370604e-11 -1.22989321e-11 -6.66455564e-12
  6.42486816e-12  2.03709251e-12  9.58447771e-12 -2.46599036e-11
  6.61346522e-12  4.97454961e-12  8.14612310e-12 -4.99283319e-13
 -1.85124562e-12  2.60263425e-11 -3.67964863e-11  2.94861728e-12
  1.94833708e-11 -1.18541387e-11  7.62355453e-13  1.93047601e-11
  1.63443948e-11 -3.06717757e-11  6.36282566e-12 -3.48399784e-11
 -3.48420095e-11 -5.26726807e-11 -3.60228426e-11 -2.14045971e-12
  1.74192942e-11  3.43600966e-12 -6.76725407e-12  4.61922982e-12
 -5.22661487e-12  2.64919403e-11  2.58840763e-11  1.04907496e-12
  2.63141167e-11  6.00804257e-11  1.88729904e-11  7.11234760e-12
 -1.03609921e-11 -2.07663804e-11  1.63092825e-11 -7.68259341e-13
  8.47400167e-12 -9.22185197e-12  3.85029665e-12 -2.59389343e-12
 -8.27180613e-23  7.96544885e-12 -8.07463054e-12  8.07463057e-12
 -1.33154989e-11  1.74775779e-12 -3.93146678e-11  2.08564617e-11
 -2.43803932e-12 -5.66253187e-12  2.71303647e-12  1.80803282e-12
  1.14146253e-12 -1.45397773e-12 -8.26852044e-12 -3.85746344e-12
  5.16775594e-13  9.25038070e-12  0.00000000e+00  3.81280533e-12
 -4.79207683e-12  7.32356868e-12  1.34075637e-11 -3.97466436e-12
  3.12631669e-12 -3.54058398e-11  5.91400953e-13  1.76403264e-11
  1.17225459e-11  2.46127223e-11  8.95546579e-12  1.19690739e-11
  1.87658435e-11 -2.61829555e-12  1.62780997e-11 -4.01082492e-11
  1.34166372e-10 -4.86038392e-11 -3.36157617e-11  1.62780997e-11
  2.93108494e-11 -7.38792727e-11  2.92212609e-11  2.71856845e-11
  8.65333290e-12 -1.57751065e-11  7.12177360e-12 -2.02972012e-12
  3.00119119e-11  4.32041058e-11  9.13743618e-12  1.90277189e-11
 -3.76253069e-11 -6.81803025e-11  2.96522445e-13  6.15763433e-12
 -1.39194064e-11  1.65062383e-11 -2.18144117e-11 -1.83593340e-11
 -9.99553837e-12  4.75824522e-11]
```

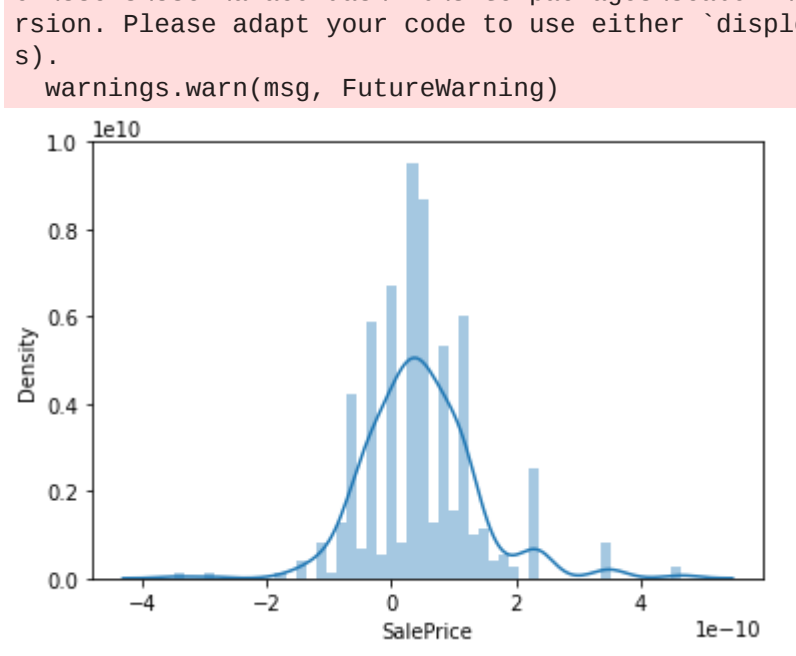
In [39]:
```python
predictions= lm.predict( X_test)
plt.scatter(y_test,predictions)
plt.xlabel('Y Test')
plt.ylabel('Predicted Y')
```

Out[39]:
```
Text(0, 0.5, 'Predicted Y')
```



In [40]:
```python
sns.distplot((y_test-predictions),bins=50);
```

```
C:\Users\User\anaconda3\lib\site-packages\seaborn\distributions.py:2619: FutureWarning: `distplot` is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```



In [41]:
```python
coeffecients = pd.DataFrame(lm.coef_,X.columns)
coeffecients.columns = ['Coeffecient']
coeffecients
```

Out[41]:

	Coeffecient
Id	-3.777990e-16
MSSubClass	4.285461e-13
LotArea	1.554312e-15
OverallQual	3.078195e-11
OverallCond	1.691108e-11
...	...
SaleCondition_AdjLand	1.650624e-11
SaleCondition_Alloca	-2.181441e-11
SaleCondition_Family	-1.835933e-11
SaleCondition_Normal	-9.995538e-12
SaleCondition_Partial	4.758245e-11

274 rows × 1 columns

In [42]:
```python
from sklearn import metrics

print('MAE:', metrics.mean_absolute_error(y_test, predictions))
print('MSE:', metrics.mean_squared_error(y_test, predictions))
print('RMSE:', np.sqrt(metrics.mean_squared_error(y_test, predictions)))
```

```
MAE: 7.453703610694299e-11
MSE: 1.0534367948607742e-20
RMSE: 1.0263706907646838e-10
```

In [ ]: