In [1]:
```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
```

In [2]:
```
mydata=pd.read_csv("houseprice.csv")
pd.set_option("max_columns",90)
mydata
```

Out[2]:

	Id	MSSubClass	MSZoning	LotFrontage	LotArea	Street	Alley	LotShape	LandContour	Utilities	LotConfig	LandSlope	Neighb
0	1	60	RL	65.0	8450	Pave	NaN	Reg	Lvl	AllPub	Inside	Gtl	
1	2	20	RL	80.0	9600	Pave	NaN	Reg	Lvl	AllPub	FR2	Gtl	
2	3	60	RL	68.0	11250	Pave	NaN	IR1	Lvl	AllPub	Inside	Gtl	
3	4	70	RL	60.0	9550	Pave	NaN	IR1	Lvl	AllPub	Corner	Gtl	
4	5	60	RL	84.0	14260	Pave	NaN	IR1	Lvl	AllPub	FR2	Gtl	N
...	...	...	...	...	...	...	...	...	...	...	...	...	
1455	1456	60	RL	62.0	7917	Pave	NaN	Reg	Lvl	AllPub	Inside	Gtl	
1456	1457	20	RL	85.0	13175	Pave	NaN	Reg	Lvl	AllPub	Inside	Gtl	N
1457	1458	70	RL	66.0	9042	Pave	NaN	Reg	Lvl	AllPub	Inside	Gtl	
1458	1459	20	RL	68.0	9717	Pave	NaN	Reg	Lvl	AllPub	Inside	Gtl	
1459	1460	20	RL	75.0	9937	Pave	NaN	Reg	Lvl	AllPub	Inside	Gtl	E

1460 rows × 81 columns

In [3]:
```
1 mydata.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 [4]:
```
1 mydata.describe()
```

Out[4]:

	Id	MSSubClass	LotFrontage	LotArea	OverallQual	OverallCond	YearBuilt	YearRemodAdd	MasVnrArea	Bsm
count	1460.000000	1460.000000	1201.000000	1460.000000	1460.000000	1460.000000	1460.000000	1460.000000	1452.000000	1460
mean	730.500000	56.897260	70.049958	10516.828082	6.099315	5.575342	1971.267808	1984.865753	103.685262	443
std	421.610009	42.300571	24.284752	9981.264932	1.382997	1.112799	30.202904	20.645407	181.066207	456
min	1.000000	20.000000	21.000000	1300.000000	1.000000	1.000000	1872.000000	1950.000000	0.000000	0
25%	365.750000	20.000000	59.000000	7553.500000	5.000000	5.000000	1954.000000	1967.000000	0.000000	0
50%	730.500000	50.000000	69.000000	9478.500000	6.000000	5.000000	1973.000000	1994.000000	0.000000	383
75%	1095.250000	70.000000	80.000000	11601.500000	7.000000	6.000000	2000.000000	2004.000000	166.000000	712
max	1460.000000	190.000000	313.000000	215245.000000	10.000000	9.000000	2010.000000	2010.000000	1600.000000	5644

In [5]:
```
1 mydata.isnull().sum()
```

Out[5]:
```
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 [6]:
```
1 mydata['LotFrontage'].fillna(mydata['LotFrontage'].mean(),inplace=True)
```

In [7]:
```
mydata['Alley'].fillna('mis',inplace=True)
```

In [8]:
```
mydata['BsmtQual'].fillna(mydata['BsmtQual'].mode()[0],inplace=True)
```

In [9]:
```
mydata['BsmtCond'].fillna(mydata['BsmtCond'].mode()[0],inplace=True)
```

In [10]:
```
mydata['BsmtExposure'].fillna(mydata['BsmtExposure'].mode()[0],inplace=True)
```

In [11]:
```
mydata['BsmtFinType1'].fillna(mydata['BsmtFinType1'].mode()[0],inplace=True)
```

In [12]:
```
mydata['BsmtFinType2'].fillna(mydata['BsmtFinType2'].mode()[0],inplace=True)
```

In [13]:
```
mydata['Electrical'].fillna(mydata['Electrical'].mode()[0],inplace=True)
```

In [14]:
```
mydata['FireplaceQu'].fillna('mis',inplace=True)
```

In [15]:
```
mydata['GarageType'].fillna(mydata['GarageType'].mode()[0],inplace=True)
```

In [16]:
```
mydata['GarageYrBlt'].fillna(mydata['GarageYrBlt'].mode()[0],inplace=True)
```

In [17]:
```
mydata['GarageFinish'].fillna(mydata['GarageFinish'].mode()[0],inplace=True)
```

In [18]:
```
mydata['GarageQual'].fillna(mydata['GarageQual'].mode()[0],inplace=True)
```

In [19]:
```
mydata['GarageCond'].fillna(mydata['GarageCond'].mode()[0],inplace=True)
```

In [20]:
```
mydata['PoolQC'].fillna('mis',inplace=True)
```

In [21]:
```
mydata['Fence'].fillna('mis',inplace=True)
```

In [22]:
```
mydata['MiscFeature'].fillna('mis',inplace=True)
```

In [23]:
```
mydata['MasVnrType'].fillna(mydata['MasVnrType'].mode()[0],inplace=True)
```

In [24]:
```
mydata['MasVnrArea'].fillna(mydata['MasVnrArea'].mode()[0],inplace=True)
```

In [25]:
```
1 mydata.isnull().sum().head(41)
```

Out[25]:
```
Id              0
MSSubClass      0
MSZoning        0
LotFrontage     0
LotArea         0
Street          0
Alley           0
LotShape        0
LandContour     0
Utilities       0
LotConfig       0
LandSlope       0
Neighborhood    0
Condition1      0
Condition2      0
BldgType        0
HouseStyle      0
OverallQual     0
OverallCond     0
YearBuilt       0
YearRemodAdd    0
RoofStyle       0
RoofMatl        0
Exterior1st     0
Exterior2nd     0
MasVnrType      0
MasVnrArea      0
ExterQual       0
ExterCond       0
Foundation      0
BsmtQual        0
BsmtCond        0
BsmtExposure    0
BsmtFinType1    0
BsmtFinSF1      0
BsmtFinType2    0
BsmtFinSF2      0
BsmtUnfSF       0
TotalBsmtSF     0
Heating         0
```

```
HeatingQC       0
dtype: int64
```

In [26]:
```
1 mydata.isnull().sum().tail(40)
```

Out[26]:
```
CentralAir       0
Electrical       0
1stFlrSF         0
2ndFlrSF         0
LowQualFinSF     0
GrLivArea        0
BsmtFullBath     0
BsmtHalfBath     0
FullBath         0
HalfBath         0
BedroomAbvGr     0
KitchenAbvGr     0
KitchenQual      0
TotRmsAbvGrd     0
Functional       0
Fireplaces       0
FireplaceQu      0
GarageType       0
GarageYrBlt      0
GarageFinish     0
GarageCars       0
GarageArea       0
GarageQual       0
GarageCond       0
PavedDrive       0
WoodDeckSF       0
OpenPorchSF      0
EnclosedPorch    0
3SsnPorch        0
ScreenPorch      0
PoolArea         0
PoolQC           0
Fence            0
MiscFeature      0
MiscVal          0
MoSold           0
YrSold           0
SaleType         0
SaleCondition    0
```

```
SalePrice        0
dtype: int64
```

In [27]:
```
# correlation
```

In [28]:
```
mydata_corr=mydata.corr()
```

In [29]:
```
1 mydata_corr
```

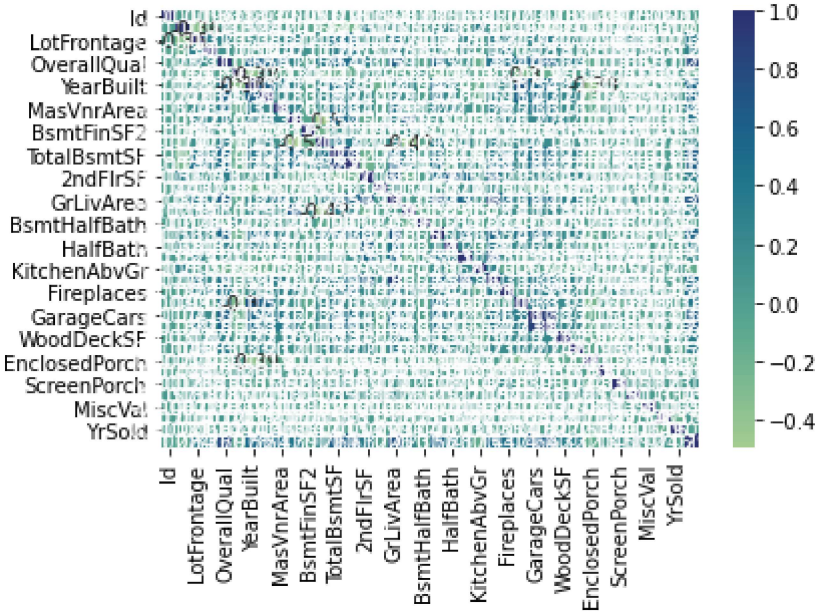
Out[29]:

	Id	MSSubClass	LotFrontage	LotArea	OverallQual	OverallCond	YearBuilt	YearRemodAdd	MasVnrArea	BsmtF
Id	1.000000	0.011156	-0.009601	-0.033226	-0.028365	0.012609	-0.012713	-0.021998	-0.051071	-0.0
MSSubClass	0.011156	1.000000	-0.357056	-0.139781	0.032628	-0.059316	0.027850	0.040581	0.023573	-0.0
LotFrontage	-0.009601	-0.357056	1.000000	0.306795	0.234196	-0.052820	0.117598	0.082746	0.178699	0.2
LotArea	-0.033226	-0.139781	0.306795	1.000000	0.105806	-0.005636	0.014228	0.013788	0.103321	0.2
OverallQual	-0.028365	0.032628	0.234196	0.105806	1.000000	-0.091932	0.572323	0.550684	0.407252	0.2
OverallCond	0.012609	-0.059316	-0.052820	-0.005636	-0.091932	1.000000	-0.375983	0.073741	-0.125694	-0.0
YearBuilt	-0.012713	0.027850	0.117598	0.014228	0.572323	-0.375983	1.000000	0.592855	0.311600	0.2
YearRemodAdd	-0.021998	0.040581	0.082746	0.013788	0.550684	0.073741	0.592855	1.000000	0.176529	0.1
MasVnrArea	-0.051071	0.023573	0.178699	0.103321	0.407252	-0.125694	0.311600	0.176529	1.000000	0.2
BsmtFinSF1	-0.005024	-0.069836	0.215828	0.214103	0.239666	-0.046231	0.249503	0.128451	0.261256	1.0
BsmtFinSF2	-0.005968	-0.065649	0.043340	0.111170	-0.059119	0.040229	-0.049107	-0.067759	-0.071330	-0.0
BsmtUnfSF	-0.007940	-0.140759	0.122156	-0.002618	0.308159	-0.136841	0.149040	0.181133	0.113862	-0.4
TotalBsmtSF	-0.015415	-0.238518	0.363358	0.260833	0.537808	-0.171098	0.391452	0.291066	0.360067	0.5
1stFlrSF	0.010496	-0.251758	0.414266	0.299475	0.476224	-0.144203	0.281986	0.240379	0.339850	0.4
2ndFlrSF	0.005590	0.307886	0.072483	0.050986	0.295493	0.028942	0.010308	0.140024	0.173800	-0.1
LowQualFinSF	-0.044230	0.046474	0.036849	0.004779	-0.030429	0.025494	-0.183784	-0.062419	-0.068628	-0.0
GrLivArea	0.008273	0.074853	0.368392	0.263116	0.593007	-0.079686	0.199010	0.287389	0.388052	0.2
BsmtFullBath	0.002289	0.003491	0.091481	0.158155	0.111098	-0.054942	0.187599	0.119470	0.083010	0.6
BsmtHalfBath	-0.020155	-0.002333	-0.006419	0.048046	-0.040150	0.117821	-0.038162	-0.012337	0.027403	0.0
FullBath	0.005587	0.131608	0.180424	0.126031	0.550600	-0.194149	0.468271	0.439046	0.272999	0.0
HalfBath	0.006784	0.177354	0.048258	0.014259	0.273458	-0.060769	0.242656	0.183331	0.199108	0.0
BedroomAbvGr	0.037719	-0.023438	0.237023	0.119690	0.101676	0.012980	-0.070651	-0.040581	0.102775	-0.1
KitchenAbvGr	0.002951	0.281721	-0.005805	-0.017784	-0.183882	-0.087001	-0.174800	-0.149598	-0.038450	-0.0
TotRmsAbvGrd	0.027239	0.040380	0.320146	0.190015	0.427452	-0.057583	0.095589	0.191740	0.279568	0.0
Fireplaces	-0.019772	-0.045569	0.235755	0.271364	0.396765	-0.023820	0.147716	0.112581	0.247015	0.2

	Id	MSSubClass	LotFrontage	LotArea	OverallQual	OverallCond	YearBuilt	YearRemodAdd	MasVnrArea	BsmtF
GarageYrBlt	-0.003243	0.098664	0.038586	-0.042198	0.437998	-0.299203	0.700098	0.571369	0.209289	0.1
GarageCars	0.016570	-0.040110	0.269729	0.154871	0.600671	-0.185758	0.537850	0.420622	0.361945	0.2
GarageArea	0.017634	-0.098672	0.323663	0.180403	0.562022	-0.151521	0.478954	0.371600	0.370884	0.2
WoodDeckSF	-0.029643	-0.012579	0.077106	0.171698	0.238923	-0.003334	0.224880	0.205726	0.159991	0.2
OpenPorchSF	-0.000477	-0.006100	0.137454	0.084774	0.308819	-0.032589	0.188686	0.226298	0.122528	0.1
EnclosedPorch	0.002889	-0.012037	0.009790	-0.018340	-0.113937	0.070356	-0.387268	-0.193919	-0.109907	-0.1
3SsnPorch	-0.046635	-0.043825	0.062335	0.020423	0.030371	0.025504	0.031355	0.045286	0.019144	0.0
ScreenPorch	0.001330	-0.026030	0.037684	0.043160	0.064886	0.054811	-0.050364	-0.038740	0.062248	0.0
PoolArea	0.057044	0.008283	0.180868	0.077672	0.065166	-0.001985	0.004950	0.005829	0.011928	0.1
MiscVal	-0.006242	-0.007683	0.001168	0.038068	-0.031406	0.068777	-0.034383	-0.010286	-0.029512	0.0
MoSold	0.021172	-0.013585	0.010158	0.001205	0.070815	-0.003511	0.012398	0.021490	-0.006723	-0.0
YrSold	0.000712	-0.021407	0.006768	-0.014261	-0.027347	0.043950	-0.013618	0.035743	-0.008317	0.0
SalePrice	-0.021917	-0.084284	0.334901	0.263843	0.790982	-0.077856	0.522897	0.507101	0.472614	0.3

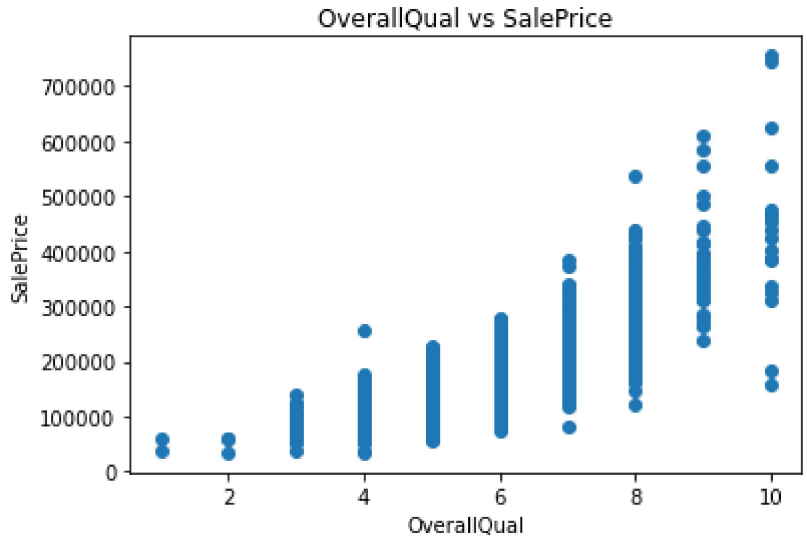
In [30]:
```
1 sns.heatmap(mydata_corr,annot=True,cmap='crest')
```

Out[30]: <AxesSubplot:>



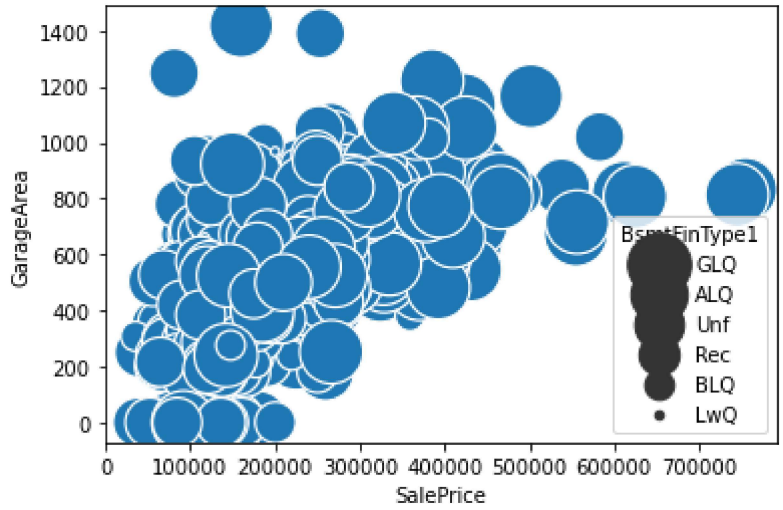
In [ ]:
```
1
```

In [31]:
```
plt.plot('OverallQual','SalePrice','', data=mydata, linestyle='', marker='o')
plt.xlabel('OverallQual')
plt.ylabel('SalePrice')
plt.title('OverallQual vs SalePrice')
plt.show()
```



In [32]:
```
1 sns.scatterplot(x='SalePrice', y='GarageArea', size='BsmtFinType1', legend=True, sizes=(20,1000), data=myda
```

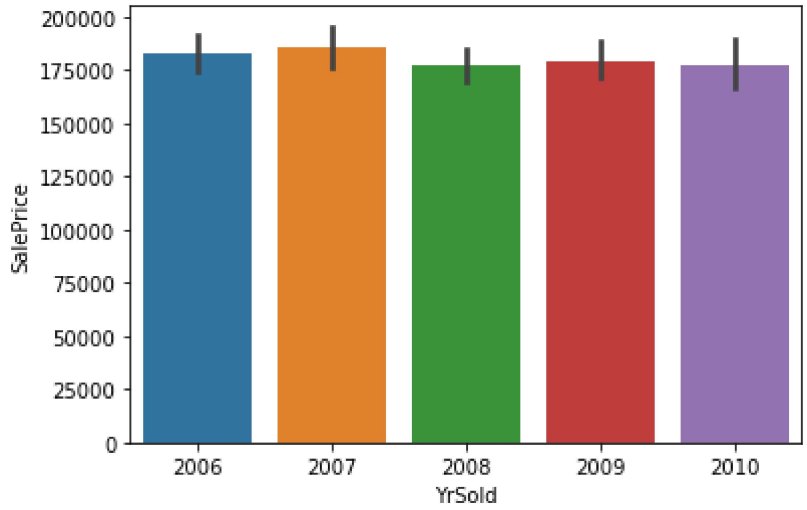
Out[32]: <AxesSubplot:xlabel='SalePrice', ylabel='GarageArea'>



In [ ]:
```
1
```

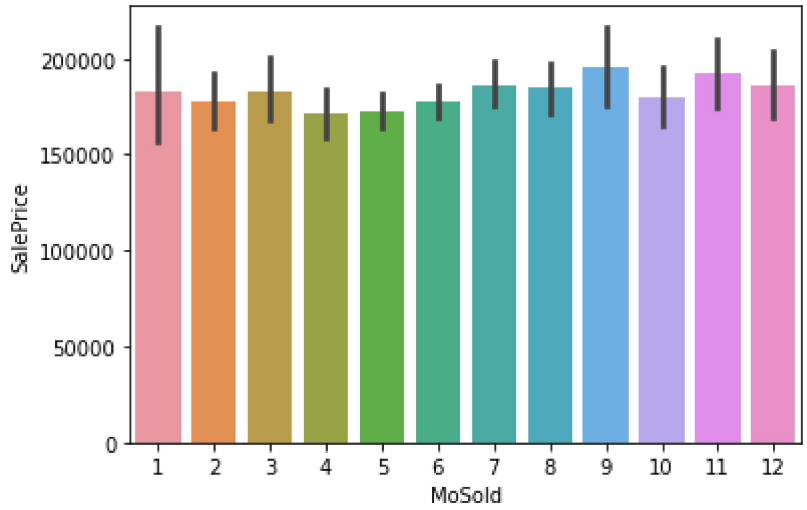
In [33]:
```
1 sns.barplot(x='YrSold',y='SalePrice',data=mydata)
```

Out[33]: <AxesSubplot:xlabel='YrSold', ylabel='SalePrice'>



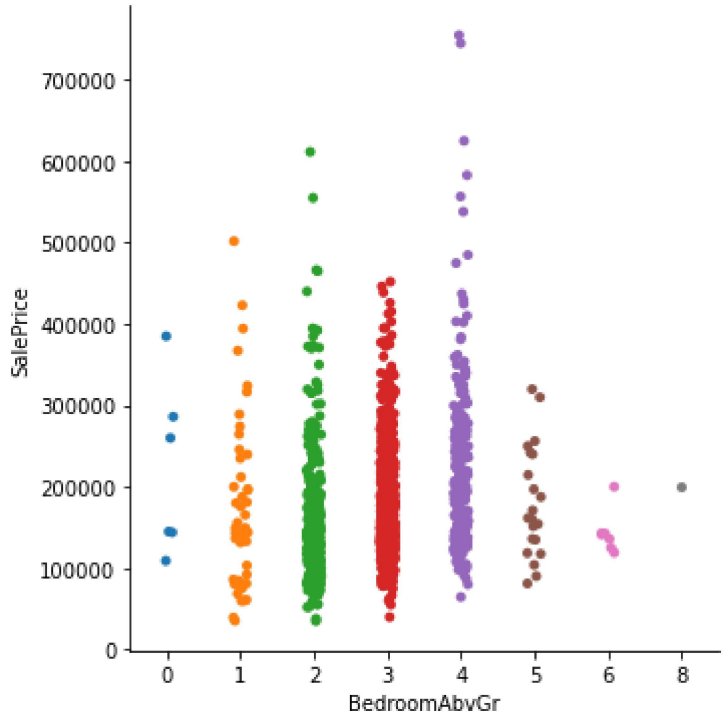
In [34]:
```
1 sns.barplot(x='MoSold',y='SalePrice',data=mydata)
```

Out[34]: <AxesSubplot:xlabel='MoSold', ylabel='SalePrice'>



In [35]:
```
1 sns.catplot(x='BedroomAbvGr',y='SalePrice',data=mydata)
```

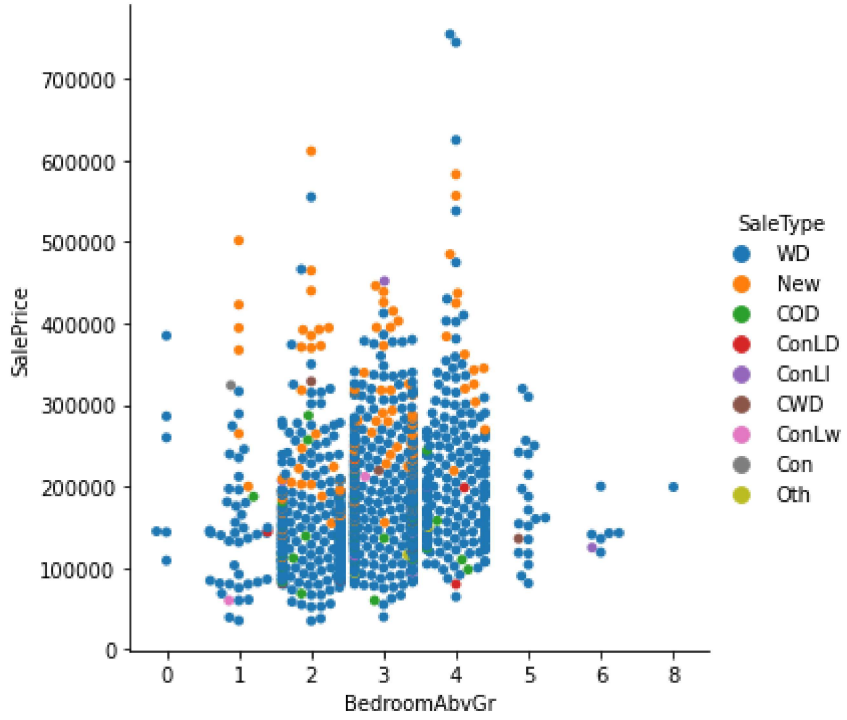
Out[35]: <seaborn.axisgrid.FacetGrid at 0x20137ca8ca0>



In [36]:
```
1 sns.catplot(x='BedroomAbvGr',y='SalePrice',kind='swarm',hue='SaleType',data=mydata)
```

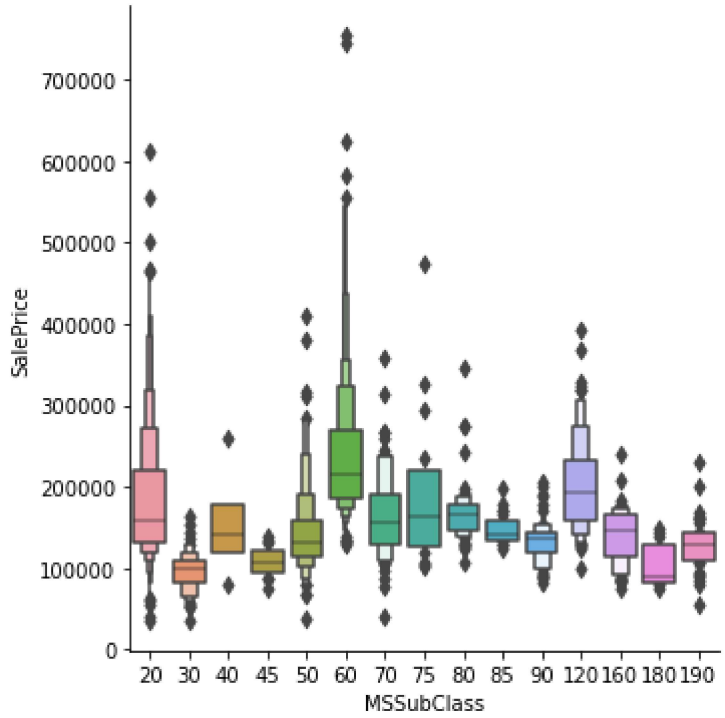
```
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\categorical.py:1296: UserWarning: 8.0% of the points cannot
be placed; you may want to decrease the size of the markers or use stripplot.
  warnings.warn(msg, UserWarning)
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\categorical.py:1296: UserWarning: 64.2% of the points canno
t be placed; you may want to decrease the size of the markers or use stripplot.
  warnings.warn(msg, UserWarning)
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\categorical.py:1296: UserWarning: 81.7% of the points canno
t be placed; you may want to decrease the size of the markers or use stripplot.
  warnings.warn(msg, UserWarning)
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\categorical.py:1296: UserWarning: 39.4% of the points canno
t be placed; you may want to decrease the size of the markers or use stripplot.
  warnings.warn(msg, UserWarning)
```

Out[36]: <seaborn.axisgrid.FacetGrid at 0x20137ca0550>



In [37]:
```
1 sns.catplot(x='MSSubClass',y='SalePrice',kind='boxen',data=mydata)
```

Out[37]: <seaborn.axisgrid.FacetGrid at 0x20137d2ae80>



In [38]:
```
1 x_ind=mydata.drop('SalePrice',axis=1)
```

In [39]:
```
1 x_ind
```

Out[39]:

	Id	MSSubClass	MSZoning	LotFrontage	LotArea	Street	Alley	LotShape	LandContour	Utilities	LotConfig	LandSlope	Neighb
0	1	60	RL	65.0	8450	Pave	mis	Reg	Lvl	AllPub	Inside	Gtl	
1	2	20	RL	80.0	9600	Pave	mis	Reg	Lvl	AllPub	FR2	Gtl	
2	3	60	RL	68.0	11250	Pave	mis	IR1	Lvl	AllPub	Inside	Gtl	
3	4	70	RL	60.0	9550	Pave	mis	IR1	Lvl	AllPub	Corner	Gtl	
4	5	60	RL	84.0	14260	Pave	mis	IR1	Lvl	AllPub	FR2	Gtl	N
...	...	...	...	...	...	...	...	...	...	...	...	...	
1455	1456	60	RL	62.0	7917	Pave	mis	Reg	Lvl	AllPub	Inside	Gtl	
1456	1457	20	RL	85.0	13175	Pave	mis	Reg	Lvl	AllPub	Inside	Gtl	N
1457	1458	70	RL	66.0	9042	Pave	mis	Reg	Lvl	AllPub	Inside	Gtl	
1458	1459	20	RL	68.0	9717	Pave	mis	Reg	Lvl	AllPub	Inside	Gtl	
1459	1460	20	RL	75.0	9937	Pave	mis	Reg	Lvl	AllPub	Inside	Gtl	E

1460 rows × 80 columns

In [40]:
```
1 y_dep=mydata.SalePrice
```

In [41]:
```
1 y_dep
```

Out[41]:
```
0       208500
1       181500
2       223500
3       140000
4       250000
         ...
1455    175000
1456    210000
1457    266500
1458    142125
1459    147500
Name: SalePrice, Length: 1460, dtype: int64
```

In [42]:
```
1 from sklearn.preprocessing import LabelEncoder
```

In [43]:
```
1 le=LabelEncoder()
```

In [44]:
```
1 for i in x_ind:
2     mydata[i]=le.fit_transform(mydata[i].astype(str).values)
```

In [45]:
```
1 mydata.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   int32
 1   MSSubClass     1460 non-null   int32
 2   MSZoning       1460 non-null   int32
 3   LotFrontage    1460 non-null   int32
 4   LotArea        1460 non-null   int32
 5   Street         1460 non-null   int32
 6   Alley          1460 non-null   int32
 7   LotShape       1460 non-null   int32
 8   LandContour    1460 non-null   int32
 9   Utilities      1460 non-null   int32
 10  LotConfig      1460 non-null   int32
 11  LandSlope      1460 non-null   int32
 12  Neighborhood   1460 non-null   int32
 13  Condition1     1460 non-null   int32
 14  Condition2     1460 non-null   int32
 15  BldgType       1460 non-null   int32
 16  HouseStyle     1460 non-null   int32
 17  OverallQual    1460 non-null   int32
 18  OverallCond    1460 non-null   int32
 19  YearBuilt      1460 non-null   int32
 20  YearRemodAdd   1460 non-null   int32
 21  RoofStyle      1460 non-null   int32
 22  RoofMatl       1460 non-null   int32
 23  Exterior1st    1460 non-null   int32
 24  Exterior2nd    1460 non-null   int32
 25  MasVnrType     1460 non-null   int32
 26  MasVnrArea     1460 non-null   int32
 27  ExterQual      1460 non-null   int32
 28  ExterCond      1460 non-null   int32
 29  Foundation     1460 non-null   int32
 30  BsmtQual       1460 non-null   int32
 31  BsmtCond       1460 non-null   int32
 32  BsmtExposure   1460 non-null   int32
 33  BsmtFinType1   1460 non-null   int32
 34  BsmtFinSF1     1460 non-null   int32
```

```
35  BsmtFinType2   1460 non-null   int32
36  BsmtFinSF2     1460 non-null   int32
37  BsmtUnfSF      1460 non-null   int32
38  TotalBsmtSF    1460 non-null   int32
39  Heating        1460 non-null   int32
40  HeatingQC      1460 non-null   int32
41  CentralAir     1460 non-null   int32
42  Electrical     1460 non-null   int32
43  1stFlrSF       1460 non-null   int32
44  2ndFlrSF       1460 non-null   int32
45  LowQualFinSF   1460 non-null   int32
46  GrLivArea      1460 non-null   int32
47  BsmtFullBath   1460 non-null   int32
48  BsmtHalfBath   1460 non-null   int32
49  FullBath       1460 non-null   int32
50  HalfBath       1460 non-null   int32
51  BedroomAbvGr   1460 non-null   int32
52  KitchenAbvGr   1460 non-null   int32
53  KitchenQual    1460 non-null   int32
54  TotRmsAbvGrd   1460 non-null   int32
55  Functional     1460 non-null   int32
56  Fireplaces     1460 non-null   int32
57  FireplaceQu    1460 non-null   int32
58  GarageType     1460 non-null   int32
59  GarageYrBlt    1460 non-null   int32
60  GarageFinish   1460 non-null   int32
61  GarageCars     1460 non-null   int32
62  GarageArea     1460 non-null   int32
63  GarageQual     1460 non-null   int32
64  GarageCond     1460 non-null   int32
65  PavedDrive     1460 non-null   int32
66  WoodDeckSF     1460 non-null   int32
67  OpenPorchSF    1460 non-null   int32
68  EnclosedPorch  1460 non-null   int32
69  3SsnPorch      1460 non-null   int32
70  ScreenPorch    1460 non-null   int32
71  PoolArea       1460 non-null   int32
72  PoolQC         1460 non-null   int32
73  Fence          1460 non-null   int32
74  MiscFeature    1460 non-null   int32
75  MiscVal        1460 non-null   int32
76  MoSold         1460 non-null   int32
77  YrSold         1460 non-null   int32
```

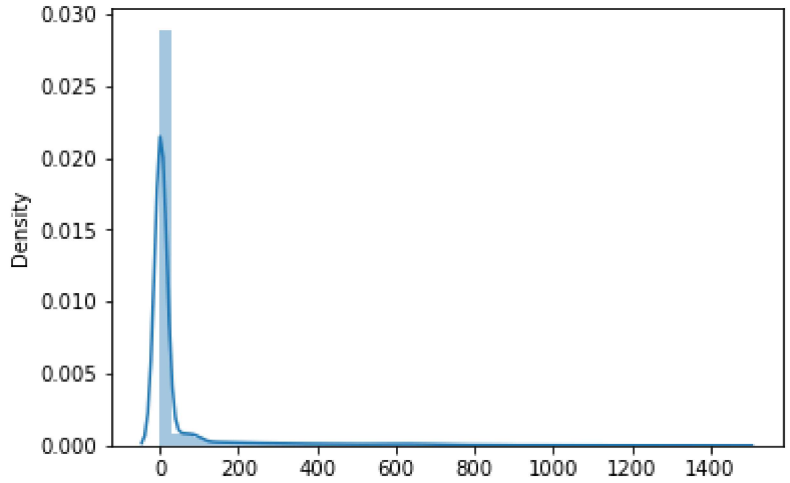
```
 78  SaleType       1460 non-null   int32
 79  SaleCondition  1460 non-null   int32
 80  SalePrice      1460 non-null   int64
dtypes: int32(80), int64(1)
memory usage: 467.8 KB
```

In [46]:
```
1 x_ind=mydata.drop('SalePrice',axis=1)
```

In [47]:
```
1 sns.distplot(x_ind)
```

```
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\distributions.py:2557: FutureWarning: `distplot` is a depre
cated function and will be removed in a future version. Please adapt your code to use either `displot` (a figu
re-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

Out[47]: <AxesSubplot:ylabel='Density'>



In [48]:
```
1 from sklearn.preprocessing import StandardScaler
```

In [49]:
```
1 norm=StandardScaler()
```

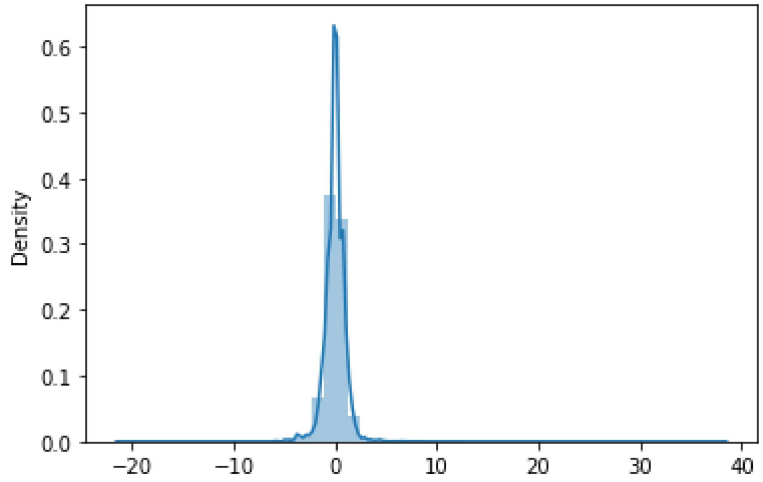
In [50]:
```
1 x_norm=norm.fit_transform(x_ind)
```

In [51]:
```
1 # after normalisation
```

In [52]:
```
1 sns.distplot(x_norm)
```

```
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\distributions.py:2557: FutureWarning: `distplot` is a depre
cated function and will be removed in a future version. Please adapt your code to use either `displot` (a figu
re-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

Out[52]: <AxesSubplot:ylabel='Density'>



In [54]:
```
1 from sklearn.decomposition import PCA
```

In [55]:
```
1 PCA_reduce=PCA()
```

In [56]:
```
1 x_new=PCA_reduce.fit_transform(x_norm)
```

In [57]:
```
1 x_new
```

Out[57]:
```
array([[-2.57418393e+00, -1.20259574e+00,  2.55234935e-01, ...,
        -3.88712555e-02, -1.01248104e-02,  1.42483337e-03],
       [ 3.38578513e-03,  2.59594681e+00,  4.50023042e-01, ...,
         2.18493887e-01, -6.65131798e-02,  1.39295184e-01],
       [-3.17475033e+00, -1.47760831e+00,  1.35976006e+00, ...,
        -1.23156599e-01, -8.79563616e-02,  4.84694645e-02],
       ...,
       [-5.47518677e-01, -5.33738404e-02,  2.52237530e+00, ...,
        -1.64137821e-01,  9.45582814e-01, -1.63526979e-01],
       [ 2.43344063e+00,  3.03251927e+00, -1.83892077e+00, ...,
         1.74688575e-01,  2.18239179e-01, -3.07549591e-01],
       [ 1.07764666e+00,  3.63462472e+00,  1.80736526e-01, ...,
         1.54177195e-01,  3.87245081e-02, -1.60552811e-01]])
```

In [58]:
```
1 # covariance matrix
```

In [59]:
```
1 cov_mat=np.cov(x_norm.T)
2 cov_mat
```

Out[59]:
```
array([[ 1.0006854 ,  0.03008051, -0.06594236, ...,  0.02619764,
        -0.01550434, -0.0199563 ],
       [ 0.03008051,  1.0006854 ,  0.01898742, ..., -0.01506027,
         0.07513055, -0.04838619],
       [-0.06594236,  0.01898742,  1.0006854 , ..., -0.02064171,
         0.09750405,  0.00950002],
       ...,
       [ 0.02619764, -0.01506027, -0.02064171, ...,  1.0006854 ,
        -0.00232889,  0.00388309],
       [-0.01550434,  0.07513055,  0.09750405, ..., -0.00232889,
         1.0006854 ,  0.18419272],
       [-0.0199563 , -0.04838619,  0.00950002, ...,  0.00388309,
         0.18419272,  1.0006854 ]])
```

In [60]:
```
1 eigen_vals,eigen_vecs=np.linalg.eig(cov_mat)
```

In [61]:
```
1 eigen_vals
```

Out[61]:
```
array([8.75857741, 3.56153554, 2.97225615, 2.54772542, 2.3805353 ,
       2.09569591, 2.01800551, 1.87646625, 1.77712012, 1.68552777,
       1.62039624, 1.5541197 , 1.4770776 , 1.42778565, 1.39704669,
       1.3702706 , 1.32414395, 0.10313573, 0.10612903, 0.13183563,
       0.14293388, 1.2047447 , 1.19003343, 1.17402432, 1.15199942,
       1.12654797, 1.12106924, 1.08852452, 1.07210071, 0.18918864,
       0.19952667, 0.20961631, 0.22397872, 0.23020878, 1.04665829,
       0.25590213, 0.2750926 , 1.01649008, 0.99747249, 0.99095934,
       0.29489146, 0.96150321, 0.3207635 , 0.93260652, 0.33790438,
       0.35544785, 0.37128134, 0.37844157, 0.39114044, 0.9163163 ,
       0.4153249 , 0.43210522, 0.44921328, 0.45498991, 0.47459818,
       0.49078104, 0.48833684, 0.50869186, 0.89653923, 0.87990089,
       0.55204578, 0.86230457, 0.56874714, 0.59001036, 0.84552451,
       0.83333944, 0.81371109, 0.79850918, 0.78674622, 0.76627911,
       0.74584947, 0.622571  , 0.63697579, 0.72208344, 0.70867951,
       0.69751059, 0.65000916, 0.66164303, 0.67510463, 0.67395171])
```

In [62]:
```
1 eigen_vecs
```

Out[62]:
```
array([[ 0.00273814,  0.0126241 ,  0.01246154, ...,  0.08558376,
        -0.10339955,  0.04358534],
       [-0.02921591, -0.18300789,  0.36240073, ..., -0.01260118,
        -0.01021385, -0.221791  ],
       [-0.09835612,  0.00428258,  0.02285393, ...,  0.4247864 ,
         0.10517381, -0.04057326],
       ...,
       [-0.00485275,  0.03466712, -0.02612444, ..., -0.00462879,
        -0.09490252,  0.00662549],
       [-0.01656567, -0.01842343,  0.05821199, ...,  0.06945318,
        -0.11754904,  0.07379673],
       [ 0.08869965, -0.03987056, -0.04174404, ..., -0.10064745,
         0.25683349,  0.02453257]])
```

In [63]:
```
1 # explained variance
```

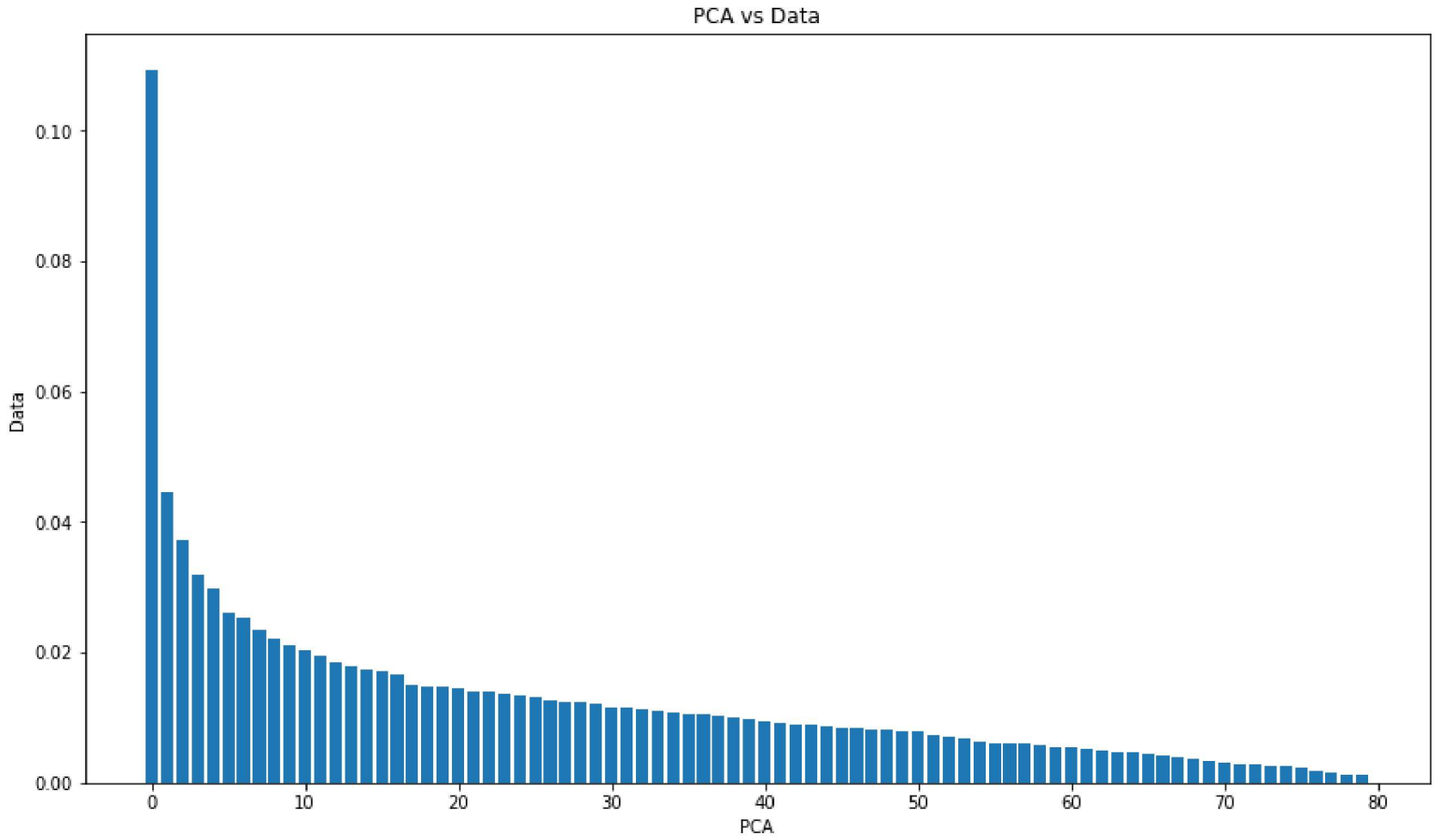
In [64]:
```
1 explained_variance=PCA_reduce.explained_variance_ratio_
```

In [65]:
```
1 explained_variance
```

Out[65]:
```
array([0.10940723, 0.0444887 , 0.03712775, 0.03182476, 0.02973631,
       0.02617826, 0.02520779, 0.02343976, 0.02219879, 0.02105467,
       0.02024108, 0.01941319, 0.01845082, 0.0178351 , 0.01745112,
       0.01711665, 0.01654046, 0.01504899, 0.01486523, 0.01466525,
       0.01439013, 0.0140722 , 0.01400377, 0.01359724, 0.01339208,
       0.01307427, 0.01269742, 0.01245987, 0.01237851, 0.01201056,
       0.0116496 , 0.01144611, 0.01119906, 0.01099123, 0.01077142,
       0.01056182, 0.01040961, 0.01016442, 0.00997453, 0.00982759,
       0.00957193, 0.00931673, 0.00901986, 0.00885243, 0.00871291,
       0.00843303, 0.00841863, 0.00826487, 0.00811955, 0.00795674,
       0.00777681, 0.00737008, 0.00710447, 0.00689585, 0.00635429,
       0.00613056, 0.00610003, 0.00592841, 0.00568348, 0.00561132,
       0.00539762, 0.00518801, 0.00488591, 0.00472728, 0.00463784,
       0.00444005, 0.00422091, 0.0040068 , 0.00368362, 0.0034363 ,
       0.00319659, 0.00287564, 0.00279782, 0.00261841, 0.00249238,
       0.00236324, 0.00178545, 0.00164682, 0.0013257 , 0.00128831])
```

In [66]:
```
plt.figure(figsize=(14,8))
plt.bar(range(80),explained_variance,label='Information gained by each PCA')
plt.xlabel('PCA')
plt.ylabel('Data')
plt.title('PCA vs Data')
```

Out[66]: Text(0.5, 1.0, 'PCA vs Data')



In [67]:
```
# as per the PCA 95% of data is enogh for building a model by checking with the explained  variance ratio,
# number of components that occupies 95% data= 63 PCA lines which occupies 95.359106 of this dataset
```

In [68]:
```
pca=PCA(n_components=63)
```

In [69]:
```
x_new_info=pca.fit_transform(x_norm)
```

In [70]:
```
x_new_info
```

Out[70]:
```
array([[-2.57418405e+00, -1.20259856e+00,  2.55236928e-01, ...,
        -3.25358104e-01,  5.54307143e-01, -3.71086416e-01],
       [ 3.38570389e-03,  2.59594516e+00,  4.50026230e-01, ...,
         8.67408010e-01,  6.21699956e-01,  8.96781489e-01],
       [-3.17475039e+00, -1.47760871e+00,  1.35976164e+00, ...,
        -4.26733648e-01,  7.59317394e-01, -4.45594933e-01],
       ...,
       [-5.47518697e-01, -5.33724904e-02,  2.52237376e+00, ...,
         1.40946129e+00, -9.55116459e-01,  5.23440362e-01],
       [ 2.43344062e+00,  3.03251718e+00, -1.83891922e+00, ...,
        -6.87853182e-01,  2.15665069e-01, -3.74042923e-01],
       [ 1.07764647e+00,  3.63462107e+00,  1.80742415e-01, ...,
         1.09349430e+00, -9.30339406e-01,  1.23235725e-01]])
```

In [71]:
```
1 from sklearn.model_selection import train_test_split
```

In [72]:
```
1 x_train,x_test,y_train,y_test=train_test_split(x_ind,y_dep,train_size=0.8,random_state=1)
```

In [73]:
```
1 # the dependent variable is contionus we gonna use regressor
```

In [74]:
```
1 # randorm forest regressor
```

In [76]:
```
1 from sklearn.ensemble import RandomForestRegressor
```

In [77]:
```
1 # build the model
```

In [78]:
```
1 model_rf=RandomForestRegressor()
```

In [79]:
```
1 model_rf.fit(x_train,y_train)
```

Out[79]:
```
RandomForestRegressor()
```

In [80]:
```
1 y_pred=model_rf.predict(x_test)
```

In [81]:
```
1 y_pred
```

Out[81]:
```
array([212649.  , 159403.2 , 119984.66,  81814.  , 148225.6 , 341165.37,
       312761.64, 153290.42, 222109.37, 227412.78, 166784.9 ,  92483.73,
       202067.67, 346459.61, 250295.9 , 111072.87, 122314.85, 138290.78,
       211641.6 , 134635.5 , 136230.  , 112005.79, 271085.52, 346858.27,
       114978.54, 196958.3 , 163131.17, 186068.08, 519530.67, 133565.86,
       143029.54, 107823.04, 124600.  ,  89303.61, 156600.57, 355544.53,
       126560.58,  92021.61, 250284.52, 105831.13, 136625.71, 145678.75,
       116002.62, 131107.06, 179823.96, 166717.05, 140925.02, 178021.19,
       268326.82, 274278.44, 104916.14, 328374.53, 116069.58, 249028.5 ,
       210892.56, 107721.04, 123015.5 , 172298.09, 129186.25, 184641.63,
       153102.23, 275151.41, 105023.18, 128502.1 , 155723.05, 129957.  ,
       131003.36, 213202.42, 160026.86, 157806.75, 160540.84, 107541.84,
       318067.45, 161690.85, 157303.85, 212386.16, 174800.19, 137313.9 ,
       388625.9 , 250622.  , 178601.55, 117141.93, 136566.73, 154042.14,
       191373.28, 144311.08, 153880.  , 153453.21, 188386.7 , 178088.  ,
       256289.41, 160553.  , 108838.16, 110476.92, 123528.04, 142605.44,
       119465.72, 135209.98, 145630.  , 144385.87, 181817.94, 136764.93,
       103685.12, 145874.89, 122561.75, 161653.87, 174427.56, 210653.01,
       132094.93, 293999.54, 160425.39, 158857.24, 156028.38, 193490.5 ,
       259540.95, 180821.22, 247205.09, 128018.1 , 173383.56, 260427.85,
       134909.64, 238149.19, 358133.51, 178473.93, 171843.74, 170328.95,
       351435.75, 138134.36, 240225.21, 216274.22, 301526.05, 107813.89,
       127172.75, 125632.84, 113024.24, 188373.6 , 393194.06, 369817.69,
       230816.64, 131646.04, 119953.33, 222907.7 , 188003.2 , 201876.13,
        97307.99, 221356.09, 107857.04, 192129.18, 217971.65, 124437.66,
       177309.09, 170371.83, 126673.64, 180654.24, 184864.9 , 388435.46,
        88692.2 , 142043.34,  97253.01, 131006.42,  72746.86, 120586.96,
       177645.7 , 151264.64, 128320.53, 141556.25, 167955.85, 143464.39,
       148012.03, 129556.12, 237982.  , 157100.52, 224953.02, 267020.61,
       180044.1 , 135346.95, 192048.44, 203736.24, 121428.26, 158595.82,
       136970.46, 181709.84, 141917.21, 139592.05, 319834.69, 116951.61,
       381323.32, 306793.7 , 156816.5 , 118944.  , 116701.66, 145478.5 ,
        96943.5 , 205435.36, 134341.08, 263945.9 , 208996.96, 151568.  ,
       142018.  ,  87650.21, 198307.07, 371946.24, 167837.5 , 178802.95,
       281007.98, 112011.46, 180545.64, 267304.96, 300904.  , 174187.49,
       194112.7 , 121678.74, 165687.04, 125433.58, 307018.42, 264615.83,
       122492.5 ,  95606.3 , 154931.5 ,  88577.68, 410006.43, 123522.58,
       161072.84, 217462.51, 119758.47, 133334.58, 222804.91, 162720.1 ,
       166854.  , 167652.  , 117529.72, 198219.35,  99467.99, 125967.71,
       334115.41, 126208.28, 302837.45, 118038.32, 134587.08, 307211.39,
```

```
       326181.05, 178752.12, 138861.55, 140332.07, 144298.14, 124785.66,
       149738.74, 137235.34, 162641.5 , 188350.1 , 120684.08, 117612.  ,
       202727.9 , 161567.19, 128847.5 , 131619.26, 180931.98,  93462.71,
       333771.07, 135056.25, 203758.17, 149203.27, 207177.82, 163938.42,
       121097.96, 175052.  , 137315.14, 240839.18, 147672.  , 101701.  ,
       197570.5 , 177888.04, 103400.88, 120840.27, 184167.78, 118124.  ,
       127018.53, 129730.58, 165836.43, 149177.  , 129396.29, 143138.82,
       116992.29, 149145.14, 249348.61, 250387.59, 125755.79, 104484.48,
       249001.27, 115255.66, 107308.02, 315917.23])
```

In [82]:
```
1 model_rf.score(x_test,y_test)
```

Out[82]: 0.8300874465406305

In [83]:
```
1 # mean square error
```

In [84]:
```
1 from sklearn.metrics import mean_squared_error
```

In [85]:
```
1 MSE=mean_squared_error(y_test,y_pred)
```

In [86]:
```
1 MSE
```

Out[86]: 1211806710.2329834

In [87]:
```
1 # root mean square
```

In [88]:
```
1 root_mean=np.sqrt(MSE)
```

In [89]:
```
1 root_mean
```

Out[89]: 34811.01420862345

In [91]:
```
1 f_com=pd.DataFrame({'Actual':y_test,'machine_predicted':y_pred})
```

In [92]:
```
1 f_com
```

Out[92]:

	Actual	machine_predicted
258	231500	212649.00
267	179500	159403.20
288	122000	119984.66
649	84500	81814.00
1233	142000	148225.60
...	...	...
163	103200	104484.48
47	249700	249001.27
1432	64500	115255.66
98	83000	107308.02
409	339750	315917.23

292 rows × 2 columns

In [93]:
```
1 Residual=y_test-y_pred
```

In [94]:
```
1 Residual
```

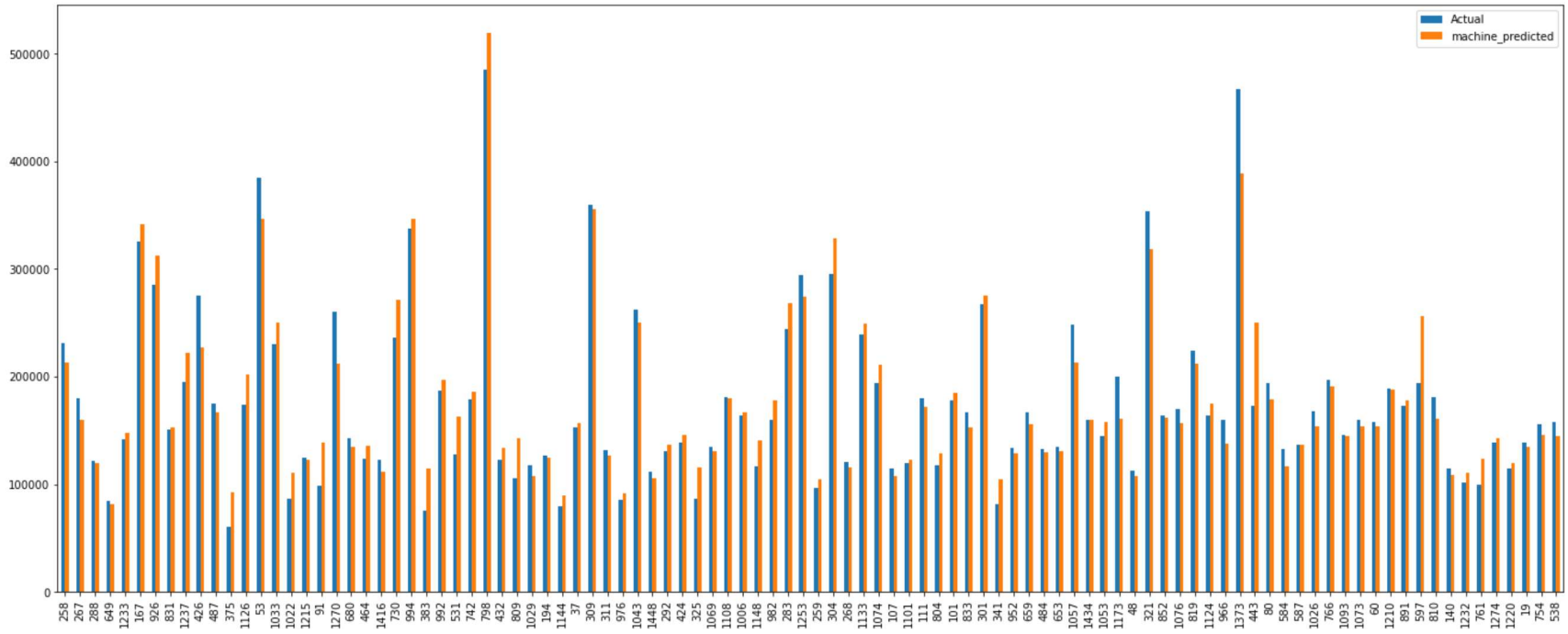
Out[94]:
```
258     18851.00
267     20096.80
288      2015.34
649      2686.00
1233    -6225.60
          ...
163     -1284.48
47        698.73
1432   -50755.66
98     -24308.02
409     23832.77
Name: SalePrice, Length: 292, dtype: float64
```

In [95]:
```
1 x_test.shape
```

Out[95]:
```
(292, 80)
```

In [96]:
```
1 comp_g=f_com.head(100)
```

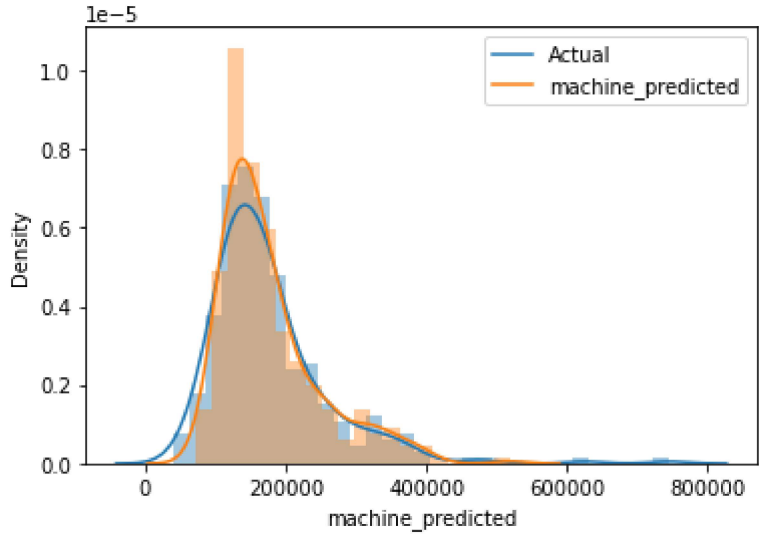
In [97]:
```
1 comp_g.plot(kind='bar',figsize=(25,10));
```



In [98]:
```
1 sns.distplot(f_com['Actual'])
2 sns.distplot(f_com['machine_predicted'])
3 plt.legend(['Actual', 'machine_predicted'])
```

```
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\distributions.py:2557: FutureWarning: `distplot` is a depre
cated function and will be removed in a future version. Please adapt your code to use either `displot` (a figu
re-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\distributions.py:2557: FutureWarning: `distplot` is a depre
cated function and will be removed in a future version. Please adapt your code to use either `displot` (a figu
re-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

Out[98]:
```
<matplotlib.legend.Legend at 0x2013b732640>
```



In [99]:
```
1 # split without compression using PCA
```

In [100]:
```
1 from sklearn.model_selection import train_test_split
```

In [101]:
```
1 x_train1,x_test1,y_train1,y_test1=train_test_split(x_norm,y_dep,test_size=0.2,random_state=1)
```

In [102]:
```
1 from sklearn.ensemble import RandomForestRegressor
```

In [103]:
```
1 model_Rf=RandomForestRegressor()
```

In [105]:
```
1 model_Rf.fit(x_train1,y_train1)
```

Out[105]:
```
RandomForestRegressor()
```

In [106]:
```
1 y_pred_Rf=model_Rf.predict(x_test1)
```

In [107]:
```
1 model_Rf.score(x_test1,y_test1)
```

Out[107]:
```
0.8226193858463314
```

In [108]:
```
1 from sklearn.metrics import mean_squared_error
```

In [109]:
```
1 MSE1=mean_squared_error(y_test1,y_pred_Rf)
```

In [110]:
```
1 MSE1
```

Out[110]:
```
1265068496.2371755
```

In [111]:
```
1 root_mean1=np.sqrt(MSE1)
```

In [112]:
```
1 root_mean1
```

Out[112]:
```
35567.80139729156
```

In [114]:
```
1 f_comp1=pd.DataFrame({'Actual':y_test,'Machine_predicted':y_pred_Rf})
```

In [115]:
```
1 f_comp1
```

Out[115]:

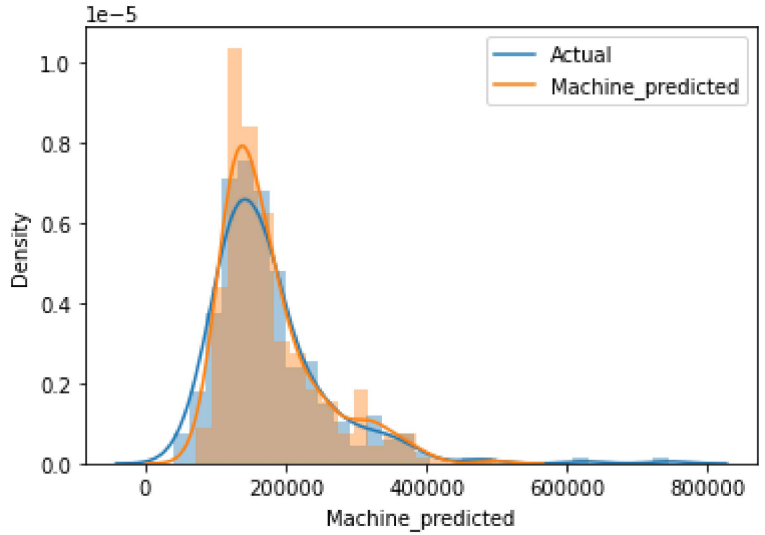
	Actual	Machine_predicted
258	231500	209011.25
267	179500	165412.53
288	122000	119300.08
649	84500	83725.11
1233	142000	146874.50
...	...	...
163	103200	105784.00
47	249700	249094.40
1432	64500	119594.37
98	83000	106316.37
409	339750	313944.67

292 rows × 2 columns

In [116]:
```
1 sns.distplot(f_comp1['Actual'])
2 sns.distplot(f_comp1['Machine_predicted'])
3 plt.legend(['Actual','Machine_predicted'])
```

```
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\distributions.py:2557: FutureWarning: `distplot` is a depre
cated function and will be removed in a future version. Please adapt your code to use either `displot` (a figu
re-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\distributions.py:2557: FutureWarning: `distplot` is a depre
cated function and will be removed in a future version. Please adapt your code to use either `displot` (a figu
re-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

Out[116]: <matplotlib.legend.Legend at 0x2013b8fde80>



accuracy with PCA 83%

accuracy without PCA 82%

In [ ]:
```
1
```