Name: Jaini Karthikeya

Usn : 20BTRCD011

In [ ]:

```
```

In [2]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
```

In [122]:

```
df_train=pd.read_csv("file:///C:/Users/Jaini/Downloads/train.csv")
df_train
```

Out[122]:

	Id	MSSubClass	MSZoning	LotFrontage	LotArea	Street	Alley	LotShape	LandCon
0	1	60	RL	65.0	8450	Pave	NaN	Reg	
1	2	20	RL	80.0	9600	Pave	NaN	Reg	
2	3	60	RL	68.0	11250	Pave	NaN	IR1	
3	4	70	RL	60.0	9550	Pave	NaN	IR1	
4	5	60	RL	84.0	14260	Pave	NaN	IR1	
...	...	...	...	...	...	...	...	...	
1455	1456	60	RL	62.0	7917	Pave	NaN	Reg	
1456	1457	20	RL	85.0	13175	Pave	NaN	Reg	
1457	1458	70	RL	66.0	9042	Pave	NaN	Reg	
1458	1459	20	RL	68.0	9717	Pave	NaN	Reg	
1459	1460	20	RL	75.0	9937	Pave	NaN	Reg	

1460 rows × 81 columns

In [4]:

```
df_train.head()
```

Out[4]:

	Id	MSSubClass	MSZoning	LotFrontage	LotArea	Street	Alley	LotShape	LandContour
0	1	60	RL	65.0	8450	Pave	NaN	Reg	Lvl
1	2	20	RL	80.0	9600	Pave	NaN	Reg	Lvl
2	3	60	RL	68.0	11250	Pave	NaN	IR1	Lvl
3	4	70	RL	60.0	9550	Pave	NaN	IR1	Lvl
4	5	60	RL	84.0	14260	Pave	NaN	IR1	Lvl

5 rows × 81 columns

In [6]:

```
df_train.shape
```

Out[6]:

```
(1460, 81)
```

In [7]:

```
df_train.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 [8]:

```
df_train.columns
```

Out[8]:

```
Index(['Id', 'MSSubClass', 'MSZoning', 'LotFrontage', 'LotArea', 'Street',
       'Alley', 'LotShape', 'LandContour', 'Utilities', 'LotConfig',
       'LandSlope', 'Neighborhood', 'Condition1', 'Condition2', 'BldgTyp
e',
       'HouseStyle', 'OverallQual', 'OverallCond', 'YearBuilt', 'YearRemod
Add',
       'RoofStyle', 'RoofMatl', 'Exterior1st', 'Exterior2nd', 'MasVnrTyp
e',
       'MasVnrArea', 'ExterQual', 'ExterCond', 'Foundation', 'BsmtQual',
       'BsmtCond', 'BsmtExposure', 'BsmtFinType1', 'BsmtFinSF1',
       'BsmtFinType2', 'BsmtFinSF2', 'BsmtUnfSF', 'TotalBsmtSF', 'Heatin
g',
       'HeatingQC', 'CentralAir', 'Electrical', '1stFlrSF', '2ndFlrSF',
       'LowQualFinSF', 'GrLivArea', 'BsmtFullBath', 'BsmtHalfBath', 'FullB
ath',
       'HalfBath', 'BedroomAbvGr', 'KitchenAbvGr', 'KitchenQual',
       'TotRmsAbvGrd', 'Functional', 'Fireplaces', 'FireplaceQu', 'GarageT
ype',
       'GarageYrBlt', 'GarageFinish', 'GarageCars', 'GarageArea', 'GarageQ
ual',
       'GarageCond', 'PavedDrive', 'WoodDeckSF', 'OpenPorchSF',
       'EnclosedPorch', '3SsnPorch', 'ScreenPorch', 'PoolArea', 'PoolQC',
       'Fence', 'MiscFeature', 'MiscVal', 'MoSold', 'YrSold', 'SaleType',
       'SaleCondition', 'SalePrice'],
      dtype='object')
```

In [9]:

```
df_train.describe()
```

Out[9]:

	Id	MSSubClass	LotFrontage	LotArea	OverallQual	OverallCond	
count	1460.000000	1460.000000	1201.000000	1460.000000	1460.000000	1460.000000	146
mean	730.500000	56.897260	70.049958	10516.828082	6.099315	5.575342	197
std	421.610009	42.300571	24.284752	9981.264932	1.382997	1.112799	3
min	1.000000	20.000000	21.000000	1300.000000	1.000000	1.000000	187
25%	365.750000	20.000000	59.000000	7553.500000	5.000000	5.000000	195
50%	730.500000	50.000000	69.000000	9478.500000	6.000000	5.000000	197
75%	1095.250000	70.000000	80.000000	11601.500000	7.000000	6.000000	200
max	1460.000000	190.000000	313.000000	215245.000000	10.000000	9.000000	201

8 rows × 38 columns

In [10]:

```
df_train.isnull().sum()
```

Out[10]:

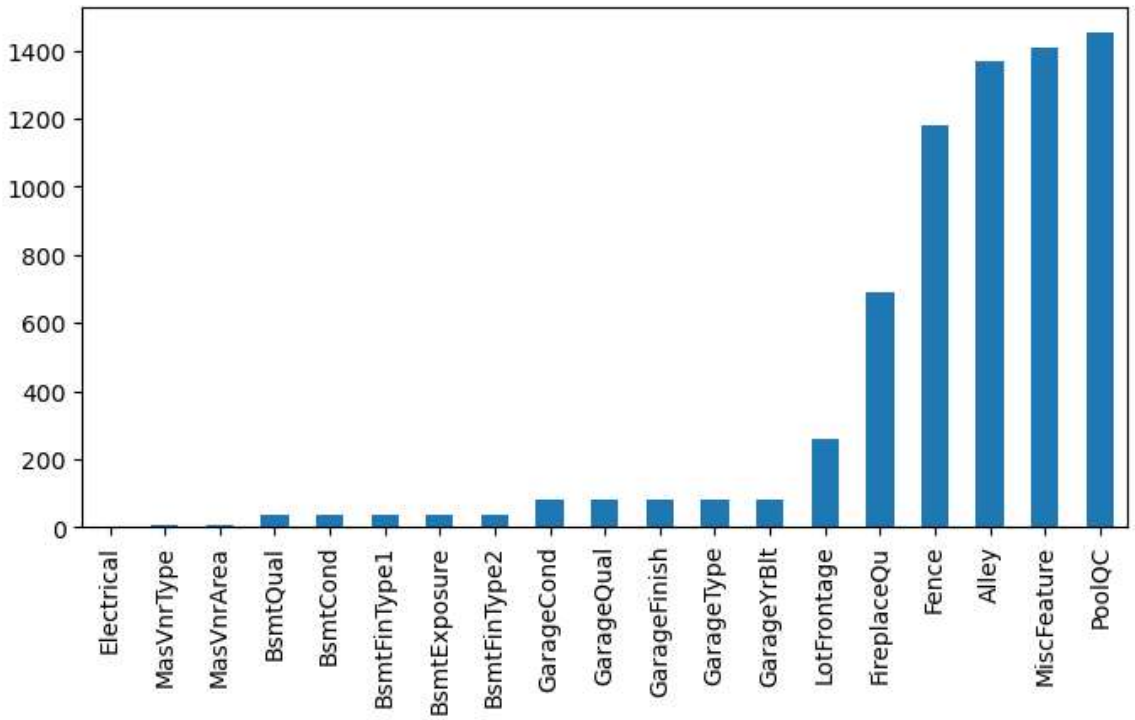
```
Id                0
MSSubClass        0
MSZoning          0
LotFrontage     259
LotArea           0
                 ...
MoSold            0
YrSold            0
SaleType          0
SaleCondition     0
SalePrice         0
Length: 81, dtype: int64
```

In [11]:

```
missing = df_train.isnull().sum()
missing = missing[missing > 0]
missing.sort_values(inplace=True)
plt.figure(figsize=(8,4))
missing.plot.bar()
```

Out[11]:

<AxesSubplot:>



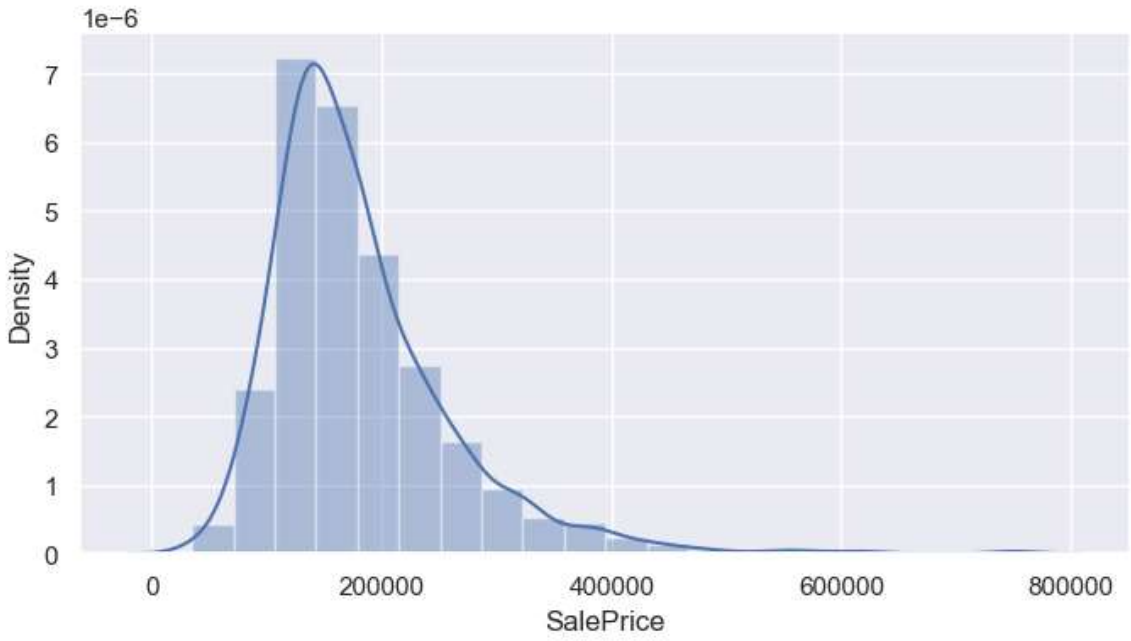
In [12]:

```
sns.set(rc={'figure.figsize' :(8,4)})
sns.distplot(df_train['SalePrice'],bins=20)
```

```
C:\ProgramData\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 fun
ction for histograms).
  warnings.warn(msg, FutureWarning)
```

Out[12]:

```
<AxesSubplot:xlabel='SalePrice', ylabel='Density'>
```



In [ ]:

Correlation

In [116]:

```
numeric = df_train.select_dtypes(include=[np.number])
```

In [114]:

```
categorical = df_train.select_dtypes(include=[np.object])
```

```
C:\Users\Tejavarma\AppData\Local\Temp\ipykernel_7580\1673716491.py:1: Depr
ecationWarning: `np.object` is a deprecated alias for the builtin `object
`. To silence this warning, use `object` by itself. Doing this will not mo
dify any behavior and is safe.
Deprecated in NumPy 1.20; for more details and guidance: https://numpy.or
g/devdocs/release/1.20.0-notes.html#deprecations (https://numpy.org/devdoc
s/release/1.20.0-notes.html#deprecations)
  categorical = df_train.select_dtypes(include=[np.object])
```

In [67]:

```
from sklearn.preprocessing import OrdinalEncoder
OE = OrdinalEncoder()
for i in categorical.columns:
    df_train[[i]]=OE.fit_transform(df_train[[i]])
```

In [15]:

```
correlation = numeric.corr()
print(correlation['SalePrice'].sort_values(ascending=False),'\n')
```

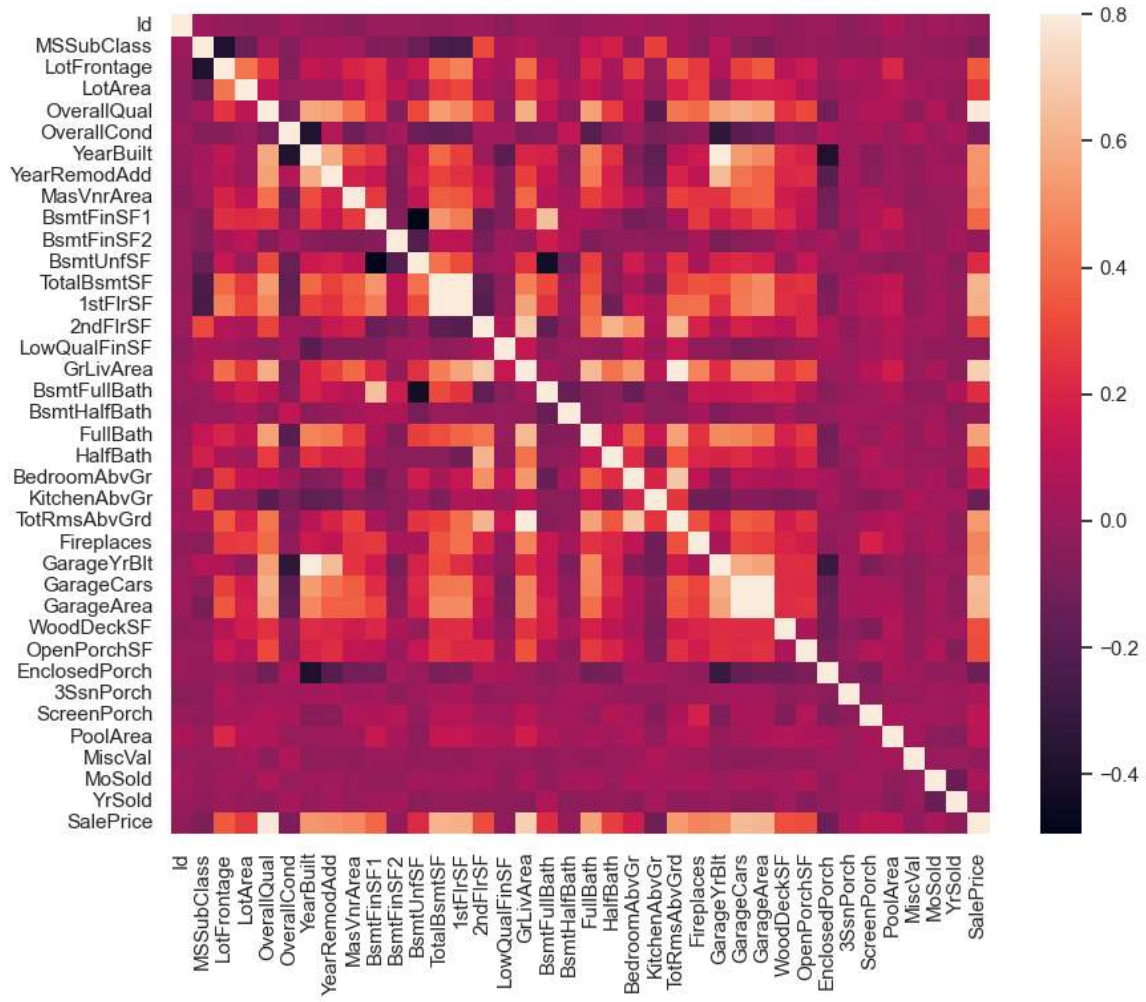
```
SalePrice        1.000000
OverallQual      0.790982
GrLivArea        0.708624
GarageCars       0.640409
GarageArea       0.623431
TotalBsmtSF      0.613581
1stFlrSF         0.605852
FullBath         0.560664
TotRmsAbvGrd     0.533723
YearBuilt        0.522897
YearRemodAdd     0.507101
GarageYrBlt      0.486362
MasVnrArea       0.477493
Fireplaces       0.466929
BsmtFinSF1       0.386420
LotFrontage      0.351799
WoodDeckSF       0.324413
2ndFlrSF         0.319334
OpenPorchSF      0.315856
HalfBath         0.284108
LotArea          0.263843
BsmtFullBath     0.227122
BsmtUnfSF        0.214479
BedroomAbvGr     0.168213
ScreenPorch      0.111447
PoolArea         0.092404
MoSold           0.046432
3SsnPorch        0.044584
BsmtFinSF2      -0.011378
BsmtHalfBath    -0.016844
MiscVal         -0.021190
Id              -0.021917
LowQualFinSF    -0.025606
YrSold          -0.028923
OverallCond     -0.077856
MSSubClass      -0.084284
EnclosedPorch   -0.128578
KitchenAbvGr    -0.135907
Name: SalePrice, dtype: float64
```

In [16]:

```
f,ax=plt.subplots(figsize=(10,8))
sns.heatmap(correlation,square=True, vmax=0.8)
```

Out[16]:

<AxesSubplot:>



In [17]:

```
k = 11
cols = correlation.nlargest(k,'SalePrice')['SalePrice'].index
print(cols)
cm = np.corrcoef(df_train[cols].values.T)
f, ax = plt.subplots(figsize=(14,12))
sns.heatmap(cm, vmax=0.8, linewidths=0.01,square=True,annot=True,cmap='viridis',
            linecolor="white",xticklabels= cols.values, annot_kws={'size':12},yticklabels
```

```
Index(['SalePrice', 'OverallQual', 'GrLivArea', 'GarageCars', 'GarageAre
a',
       'TotalBsmtSF', '1stFlrSF', 'FullBath', 'TotRmsAbvGrd', 'YearBuilt',
       'YearRemodAdd'],
      dtype='object')
```

Out[17]:

<AxesSubplot:>

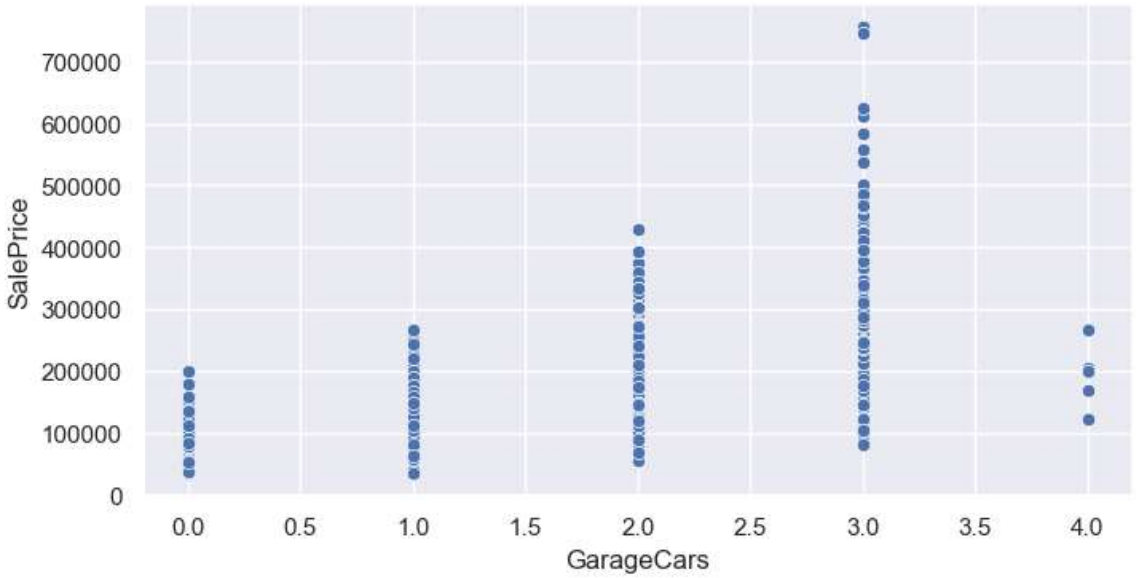
Scatterplot

In [18]:

```
sns.scatterplot(x=df_train['GarageCars'],y=df_train['SalePrice'])
```

Out[18]:

```
<AxesSubplot:xlabel='GarageCars', ylabel='SalePrice'>
```

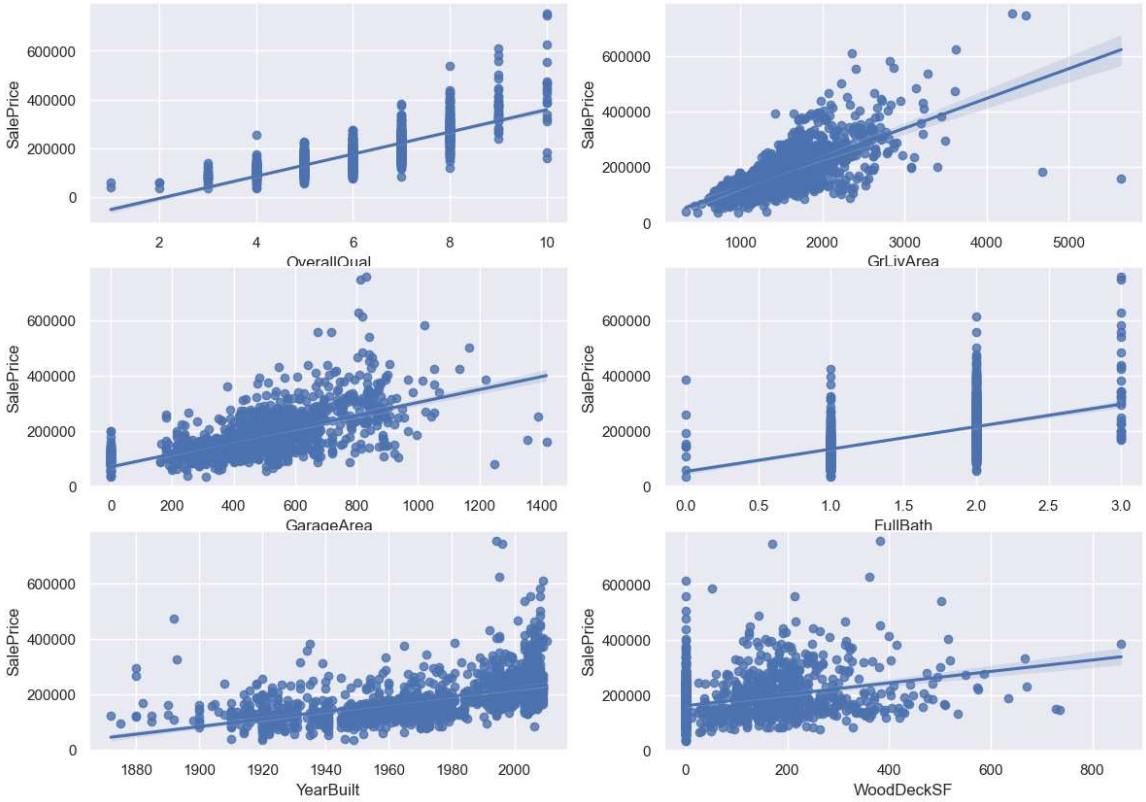


In [19]:

```
fig, ((ax1,ax2),(ax3,ax4),(ax5,ax6)) = plt.subplots(nrows=3,ncols=2,figsize=(14,10))

sns.regplot(x=df_train['OverallQual'],y=df_train['SalePrice'], scatter = True, fit_reg= T

sns.regplot(x=df_train['GrLivArea'],y=df_train['SalePrice'], scatter = True, fit_reg= Tru

sns.regplot(x=df_train['GarageArea'],y=df_train['SalePrice'], scatter = True, fit_reg= Tr

sns.regplot(x=df_train['FullBath'],y=df_train['SalePrice'], scatter = True, fit_reg= True

sns.regplot(x=df_train['YearBuilt'],y=df_train['SalePrice'], scatter = True, fit_reg= Tru

sns.regplot(x=df_train['WoodDeckSF'],y=df_train['SalePrice'], scatter = True, fit_reg= Tr
```

Out[19]:

```
<AxesSubplot:xlabel='WoodDeckSF', ylabel='SalePrice'>
```



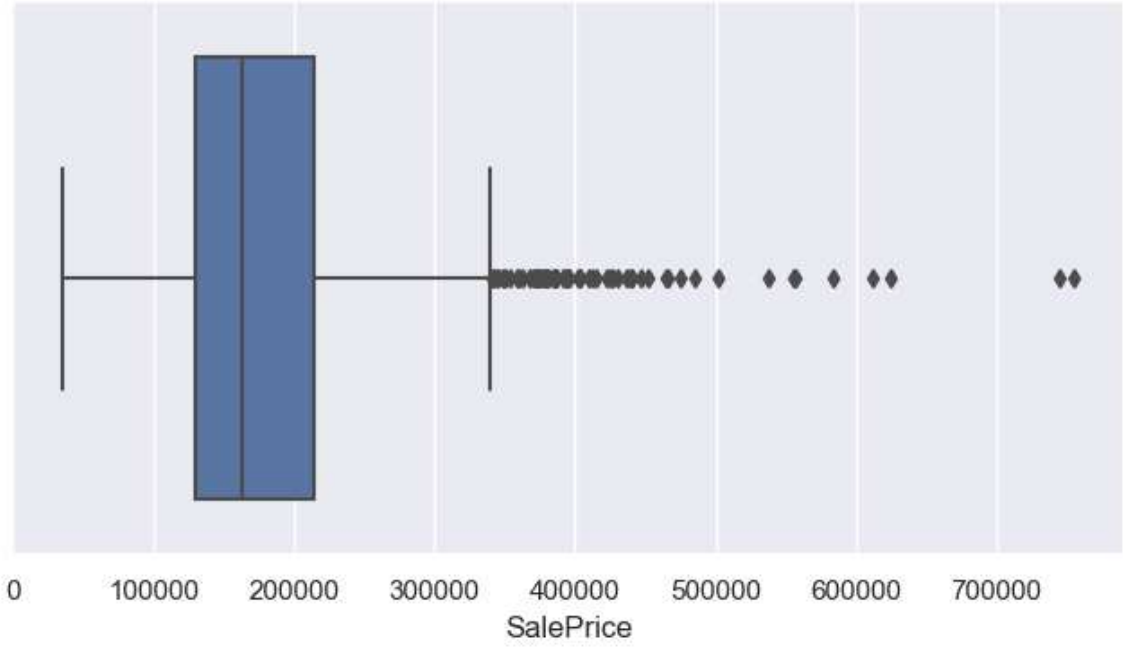
BoxPlot

In [20]:

```
sns.boxplot(x=df_train["SalePrice"])
```

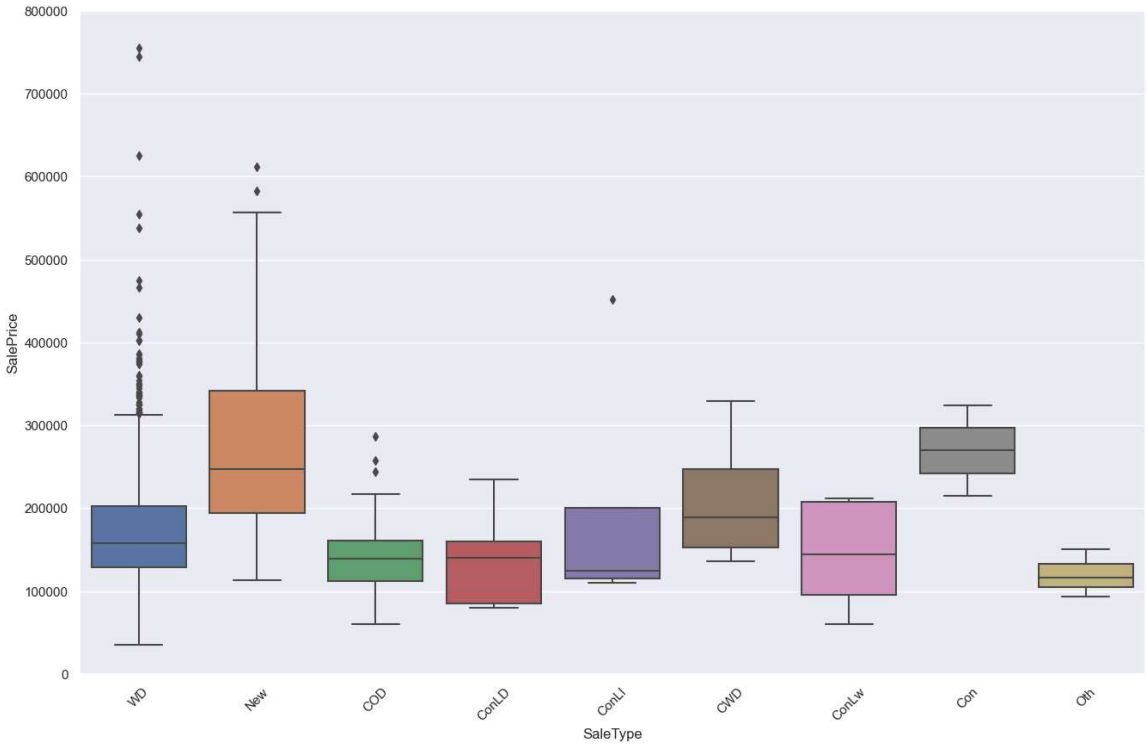
Out[20]:

```
<AxesSubplot:xlabel='SalePrice'>
```



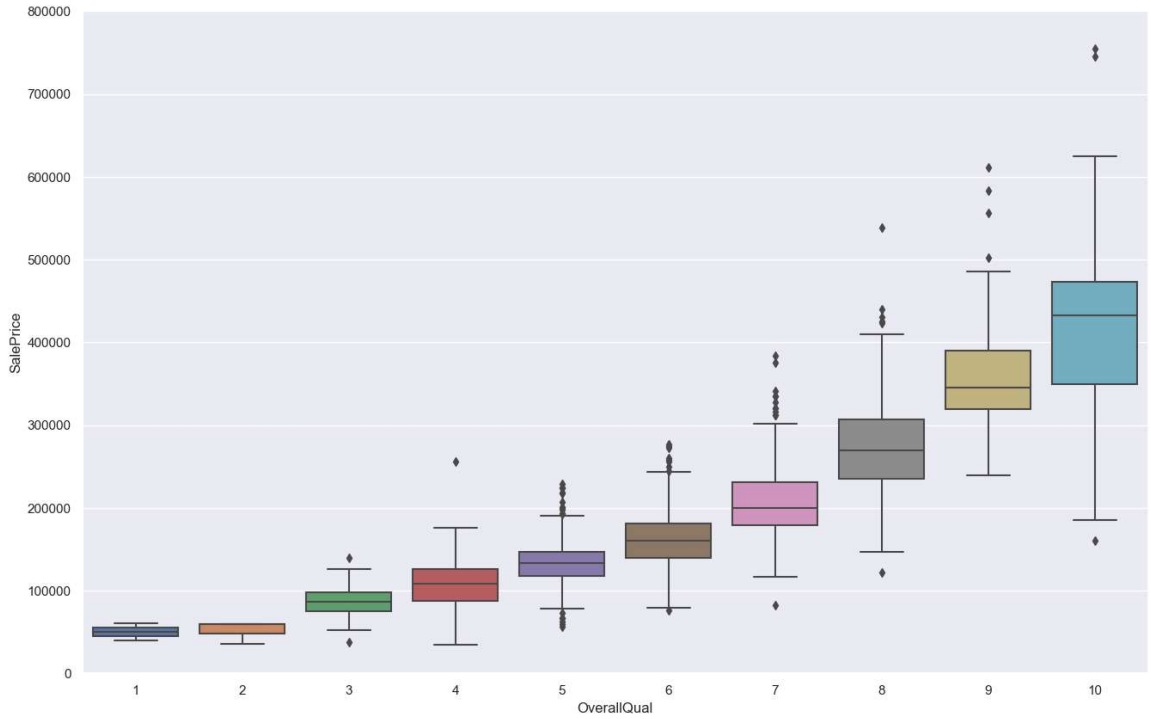
In [21]:

```
f, ax = plt.subplots(figsize=(16,10))
fig = sns.boxplot(x=df_train['SaleType'],y=df_train["SalePrice"])
fig.axis(ymin=0,ymax=800000);
xt = plt.xticks(rotation=45)
```



In [22]:

```
f, ax = plt.subplots(figsize=(16,10))
fig = sns.boxplot(x=df_train['OverallQual'],y=df_train["SalePrice"])
fig.axis(ymin=0,ymax=800000);
```



Removal of Outliers

In [ ]:

```
```

In [23]:

```
df_train['SalePrice'].describe()
```

Out[23]:

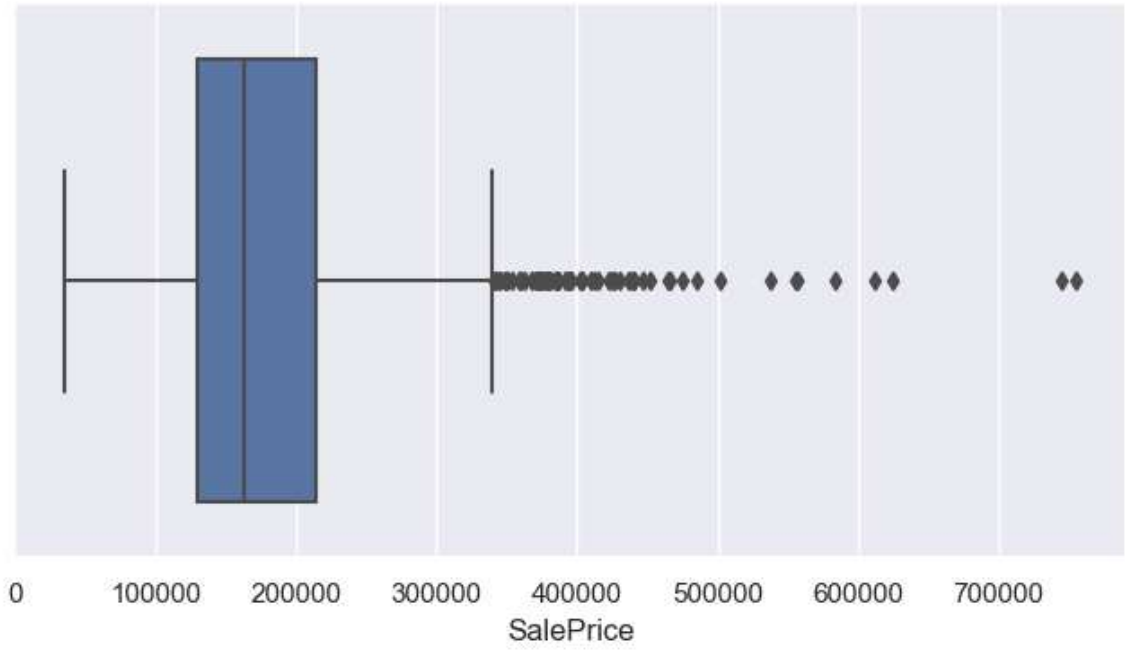
```
count      1460.000000
mean     180921.195890
std       79442.502883
min       34900.000000
25%      129975.000000
50%      163000.000000
75%      214000.000000
max      755000.000000
Name: SalePrice, dtype: float64
```

In [24]:

```
sns.boxplot(x=df_train["SalePrice"])
```

Out[24]:

```
<AxesSubplot:xlabel='SalePrice'>
```



In [25]:

```
df_train.shape
```

Out[25]:

```
(1460, 81)
```

In [26]:

```
first_quartile = df_train['SalePrice'].quantile(.25)
third_quartile = df_train['SalePrice'].quantile(.75)
IQR=third_quartile-first_quartile
```

In [27]:

```
new_boundary = third_quartile + 3*IQR
```

In [28]:

```
df_train.drop(df_train[df_train['SalePrice']>new_boundary].index,axis=0,inplace=True)
```

In [29]:

```
df_train.shape
```

Out[29]:

```
(1448, 81)
```

In [80]:

```
df_train.fillna(method="bfill",inplace =True)
```

Removing Unnecessary Features

In [43]:

```
columns_to_remove=['BsmtFinSF1','LotFrontage','WoodDeckSF','2ndFlrSF','OpenPorchSF','Half
                   'BsmtUnfSF','BedroomAbvGr','ScreenPorch','PoolArea','MoSold','3SsnPorc
                   'MiscVal','Id','LowQualFinSF','YrSold', 'OverallCond','MSSubClass','En
```

In [105]:

```
df_train.shape
```

Out[105]:

```
(1412, 57)
```

In [110]:

```
x = df_train.drop('SalePrice',axis = 1)
y = df_train.SalePrice
```

In [107]:

```
from sklearn.model_selection import train_test_split
trainx,testx,trainy,testy = train_test_split(x,y)
```

In [108]:

```
from sklearn.linear_model import LinearRegression
lr = LinearRegression()
```

In [109]:

```
lr.fit(trainx,trainy)
```

Out[109]:

```
LinearRegression()
```

In [111]:

```
yp = lr.predict(testx)
```

In [112]:

```
from sklearn.metrics import r2_score
r2_score(testy,yp)
```

Out[112]:

```
0.7943658068630342
```

In [ ]:

```
```

In [ ]:

```
```

In [83]:

```
df_train.dropna(inplace = True)
```

In [102]:

```
for i in df_train.columns:
    df_train[i] = df_train[i].astype("int")
```

In [ ]:

```
```

In [117]:

```
from sklearn.ensemble import RandomForestRegressor
rf = RandomForestRegressor()
```

In [118]:

```
rf.fit(trainx,trainy)
```

Out[118]:

```
RandomForestRegressor()
```

In [119]:

```
ypr = rf.predict(testx)
```

In [120]:

```
r2_score(testy,ypr)
```

Out[120]:

```
0.8355315264804883
```