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

In [2]:
```
df=pd.read_csv('train.csv')
pd.set_option('display.max_columns',None)
```

In [27]:
```
df.head()
```

Out[27]:

	Id	MSSubClass	MSZoning	LotFrontage	LotArea	Street	Alley	LotShape	LandContour	Utilities	LotConfig	La
0	1	60	RL	65.0	8450	Pave	NaN	Reg	Lvl	AllPub	Inside	
1	2	20	RL	80.0	9600	Pave	NaN	Reg	Lvl	AllPub	FR2	
2	3	60	RL	68.0	11250	Pave	NaN	IR1	Lvl	AllPub	Inside	
3	4	70	RL	60.0	9550	Pave	NaN	IR1	Lvl	AllPub	Corner	
4	5	60	RL	84.0	14260	Pave	NaN	IR1	Lvl	AllPub	FR2	

In [4]:
```
feature_with_na=[feature for feature in df.columns if df[feature].isnull().sum()>=1]
```

In [5]:
```
feature_with_na
```

Out[5]:
```
['LotFrontage',
 'Alley',
 'MasVnrType',
 'MasVnrArea',
 'BsmtQual',
 'BsmtCond',
 'BsmtExposure',
 'BsmtFinType1',
 'BsmtFinType2',
 'Electrical',
 'FireplaceQu',
 'GarageType',
 'GarageYrBlt',
 'GarageFinish',
 'GarageQual',
 'GarageCond',
 'PoolQC',
 'Fence',
 'MiscFeature']
```

Here we find missing Percentage with respect to nan_feature

In [6]:
```
for feature in feature_with_na:
    print(feature,np.round(df[feature].isnull().mean(),4)*100,'% missing')
```

```
LotFrontage 17.740000000000002 % missing
Alley 93.77 % missing
MasVnrType 0.5499999999999999 % missing
MasVnrArea 0.5499999999999999 % missing
BsmtQual 2.53 % missing
BsmtCond 2.53 % missing
```

```
BsmtExposure 2.6 % missing
BsmtFinType1 2.53 % missing
BsmtFinType2 2.6 % missing
Electrical 0.06999999999999999 % missing
FireplaceQu 47.260000000000005 % missing
GarageType 5.55 % missing
GarageYrBlt 5.55 % missing
GarageFinish 5.55 % missing
GarageQual 5.55 % missing
GarageCond 5.55 % missing
PoolQC 99.52 % missing
Fence 80.75 % missing
MiscFeature 96.3 % missing
```

Relationship between Nan_feature and salesprice

In [13]:

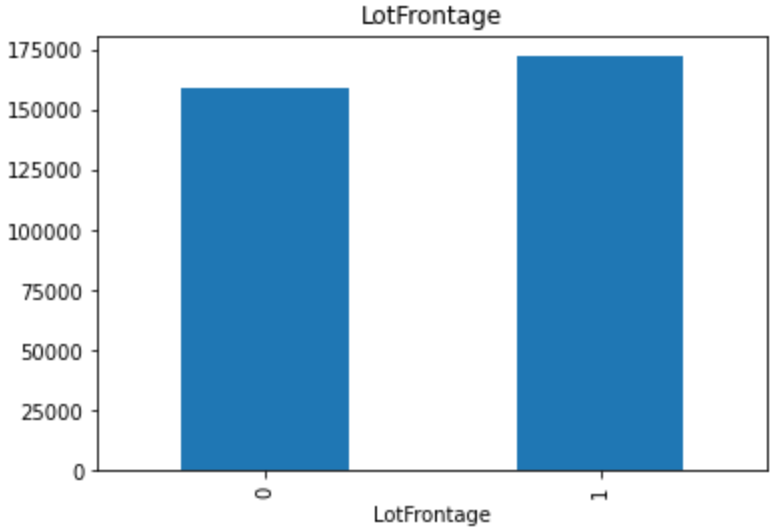
```
# for i in range(len(feature_with_na)):
#     plt.figure(figsize=(20,40))
#     plt.subplot(10,2,i+1)
#     sns.barplot(y=df['SalePrice'],x=df[feature_with_na[i]])
```

In [15]:

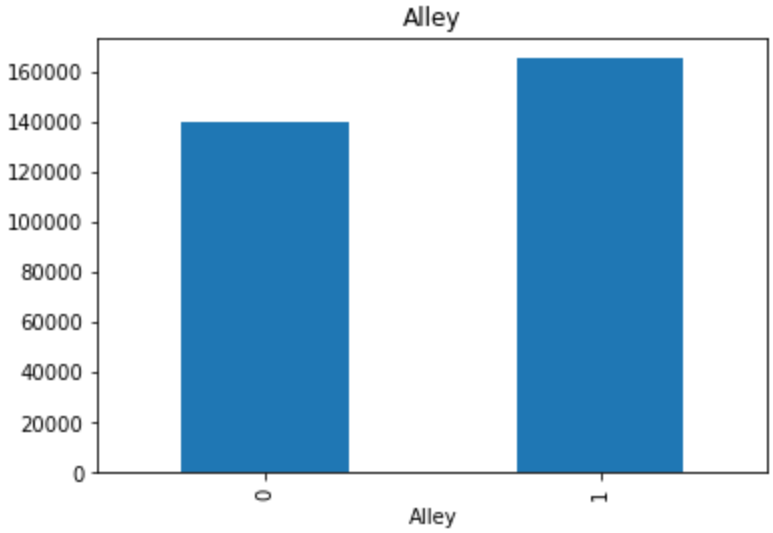
```
for feature in feature_with_na:
    data=df.copy()
    print(feature)
    data[feature]=np.where(data[feature].isnull(),1,0)

    data.groupby(feature)['SalePrice'].median().plot.bar()
    plt.title(feature)
    plt.show()
```

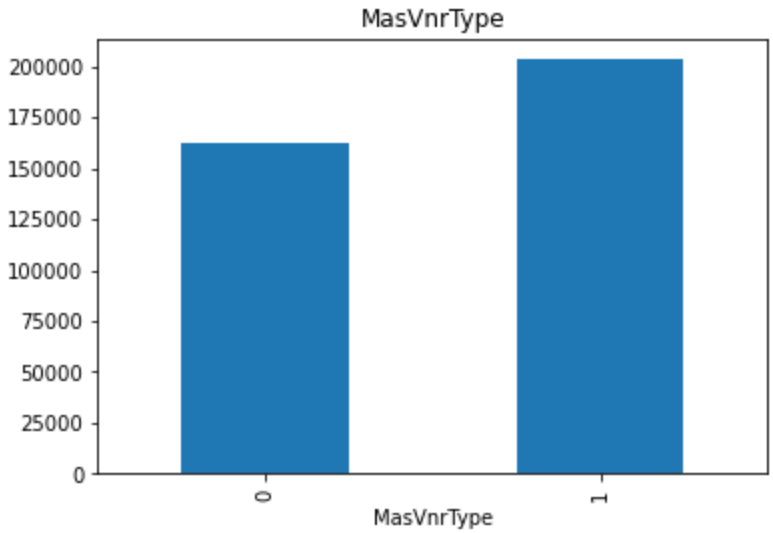
```
LotFrontage
```



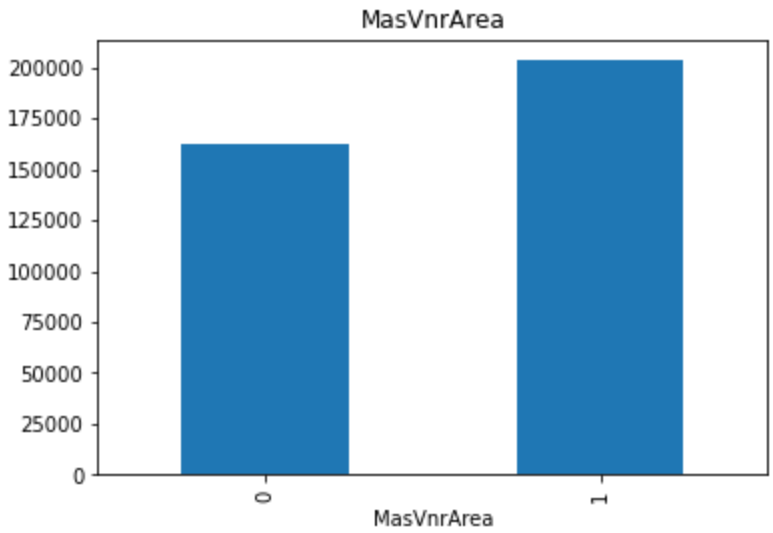

```
Alley
```



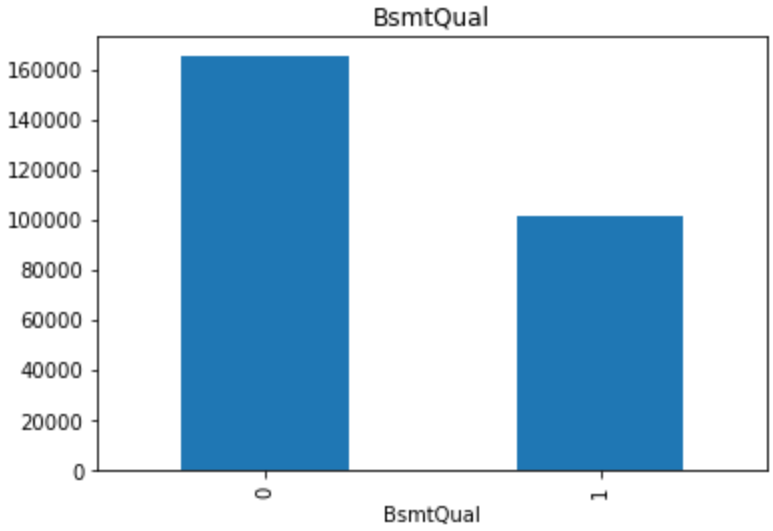
MasVnrType



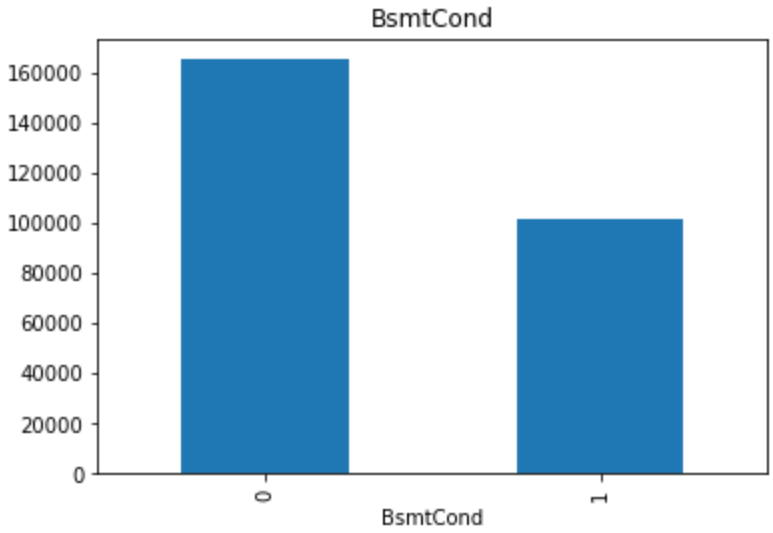
MasVnrArea



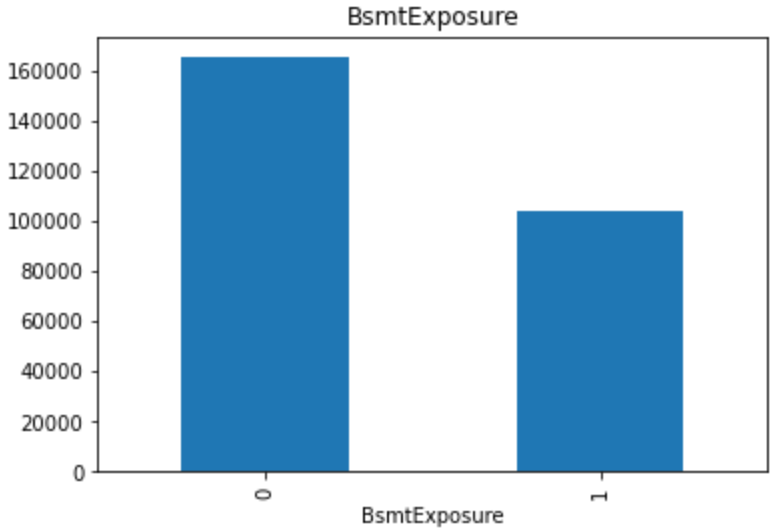
BsmtQual



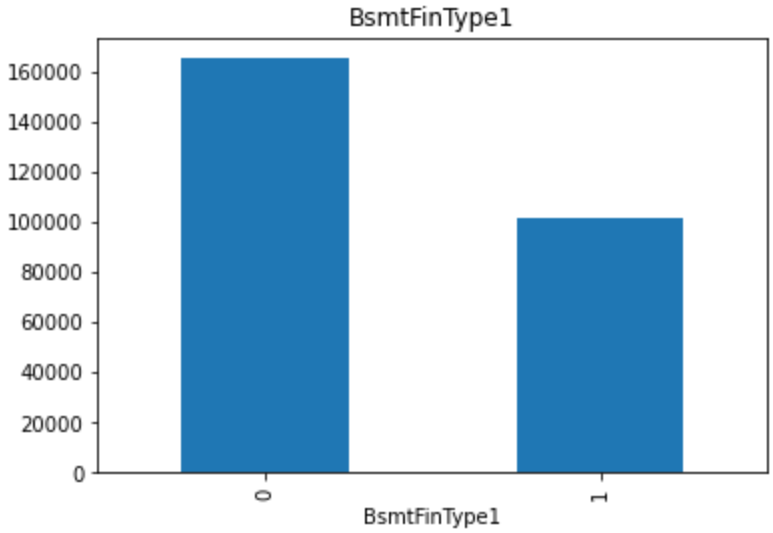
BsmtCond



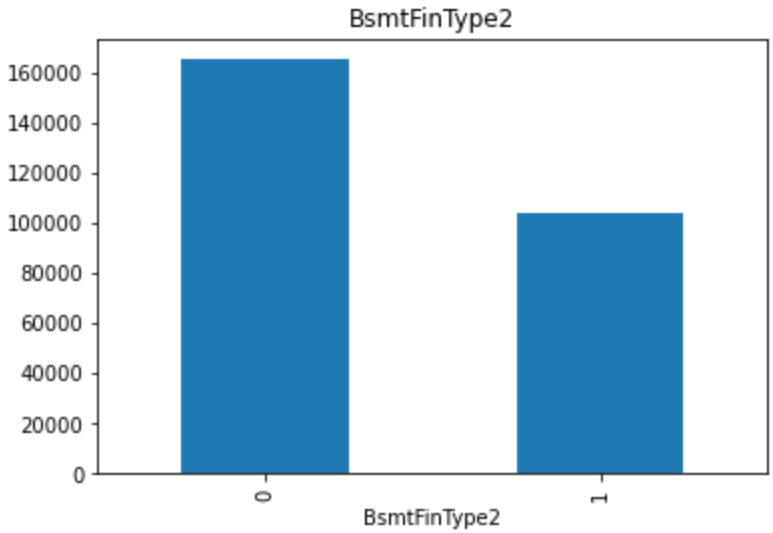
BsmtExposure



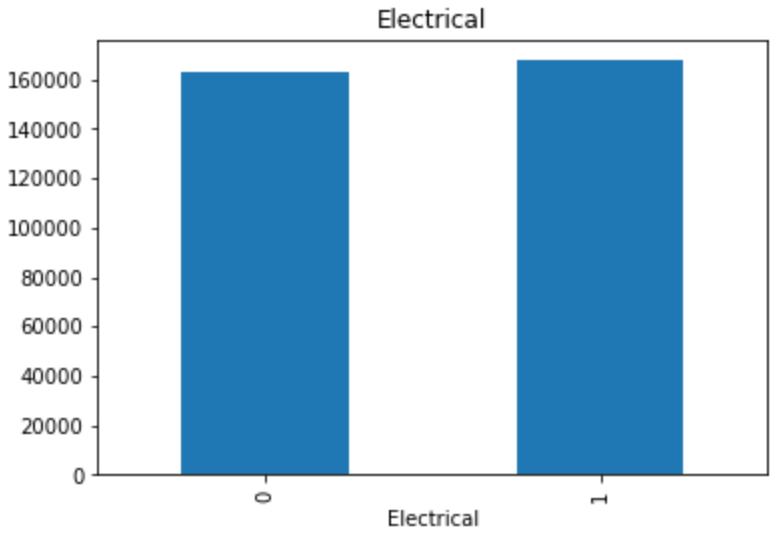
BsmtFinType1



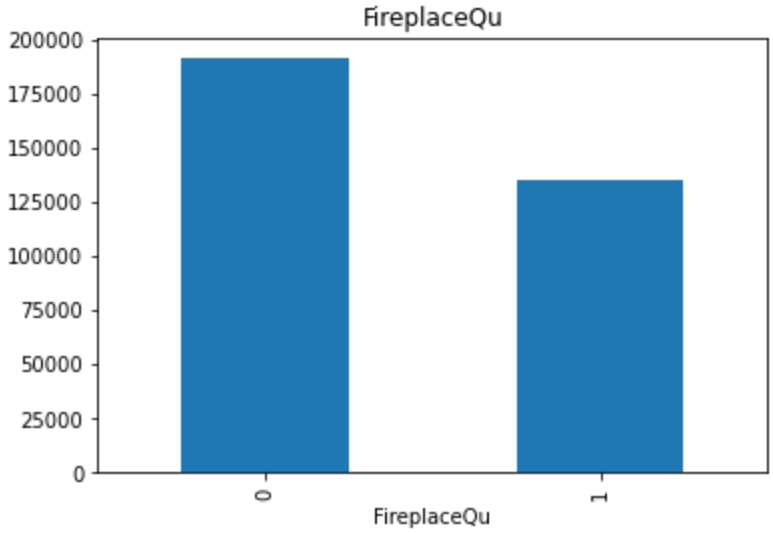
BsmtFinType2



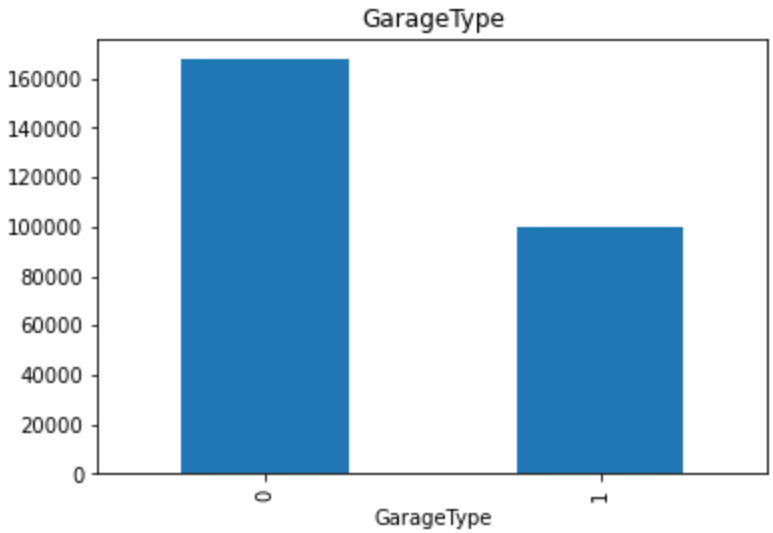
Electrical



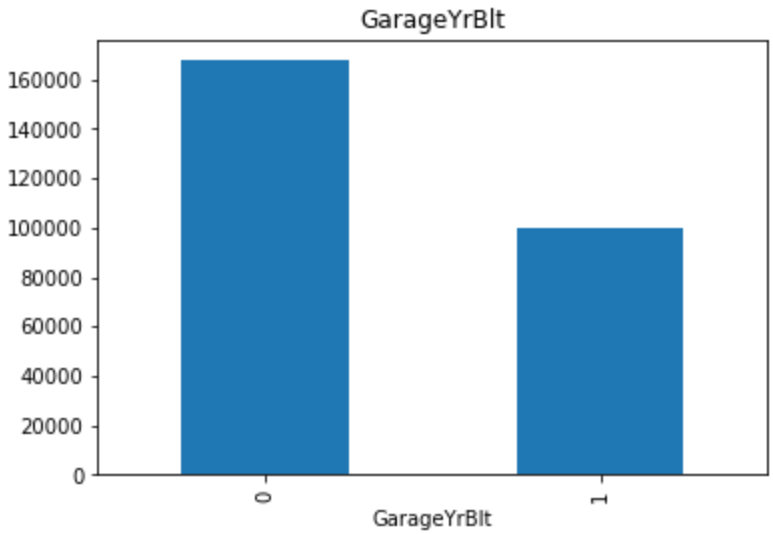
FireplaceQu



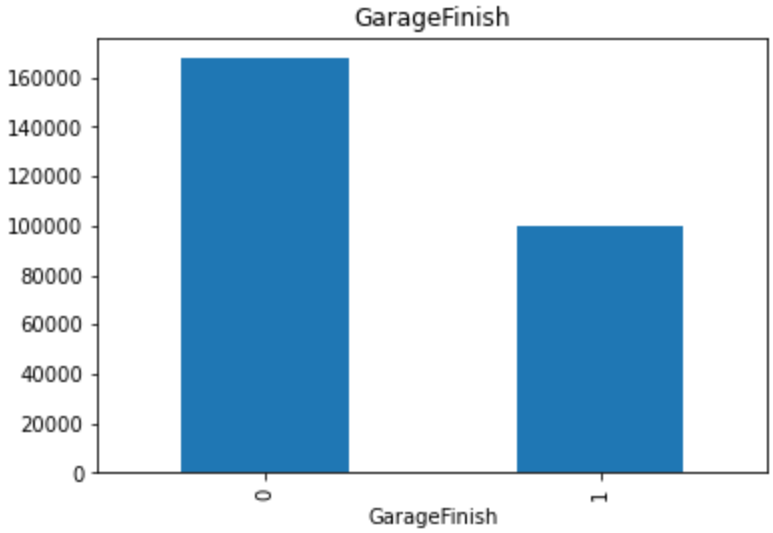
GarageType



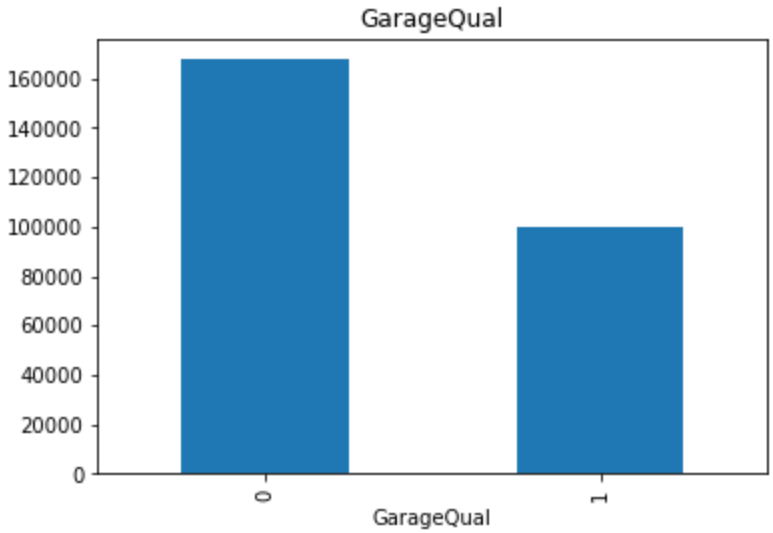
GarageYrBlt



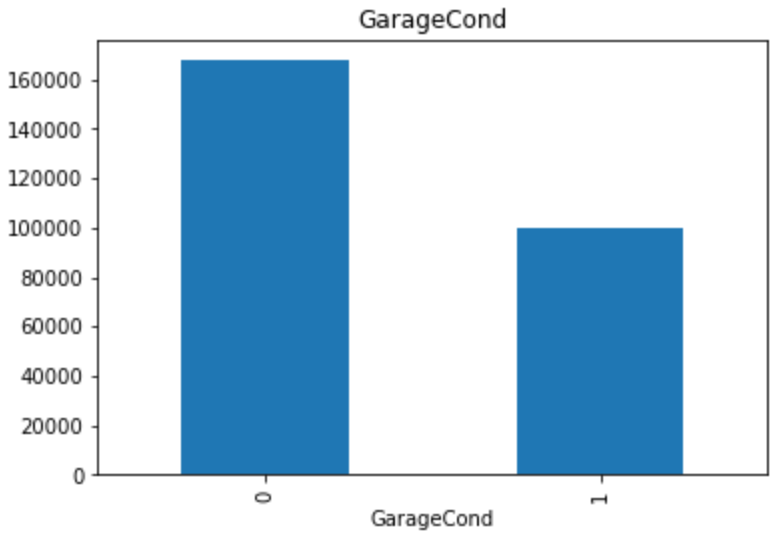
GarageFinish



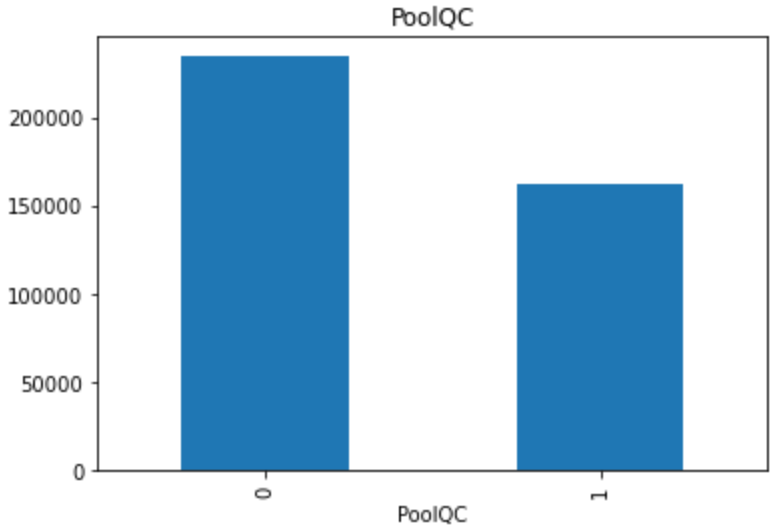
GarageQual



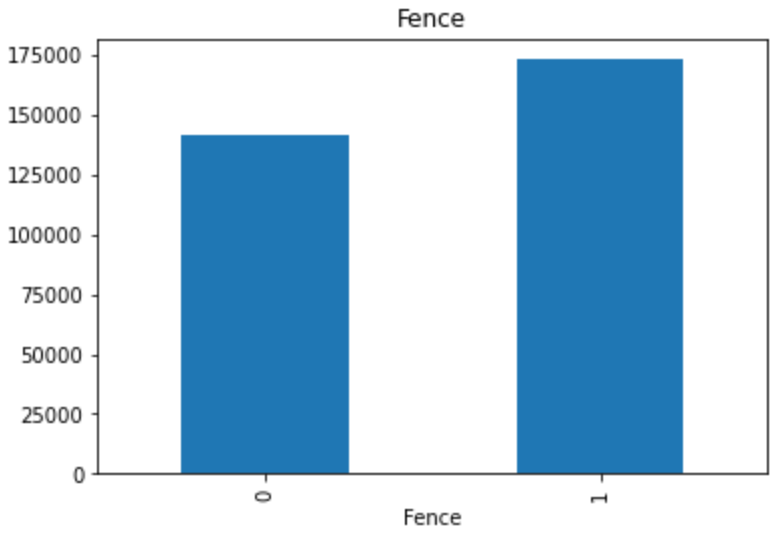
GarageCond



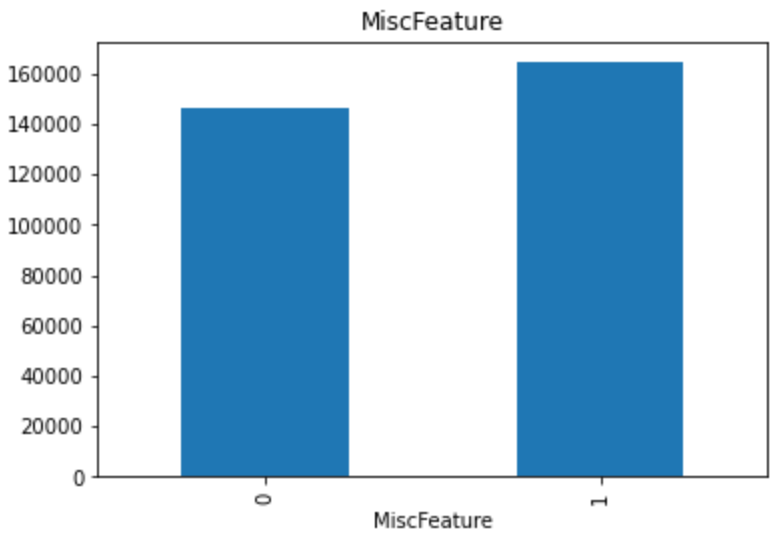
PoolQC



Fence



MiscFeature



conclusion: there are many features which have more null values as compare to not null values

In [16]:
```
numerical_feature=[feature for feature in df.columns if df[feature].dtypes!='O']
categorical_feature=[feature for feature in df.columns if df[feature].dtypes=='O']
```

In [17]:
```
df[numerical_feature].head()
```

Out[17]:

	Id	MSSubClass	LotFrontage	LotArea	OverallQual	OverallCond	YearBuilt	YearRemodAdd	MasVnrArea	BsmtFi

	Id	MSSubClass	LotFrontage	LotArea	OverallQual	OverallCond	YearBuilt	YearRemodAdd	MasVnrArea	BsmtFi
0	1	60	65.0	8450	7	5	2003	2003	196.0	
1	2	20	80.0	9600	6	8	1976	1976	0.0	
2	3	60	68.0	11250	7	5	2001	2002	162.0	
3	4	70	60.0	9550	7	5	1915	1970	0.0	
4	5	60	84.0	14260	8	5	2000	2000	350.0	

From numerical features taking year feature separately

Year Features

In [28]:
```
temporal_variable=[feature for feature in numerical_feature if 'Yr' in feature or 'Year' i
```

In [29]:
```
temporal_variable
```

Out[29]:
```
['YearBuilt', 'YearRemodAdd', 'GarageYrBlt', 'YrSold']
```

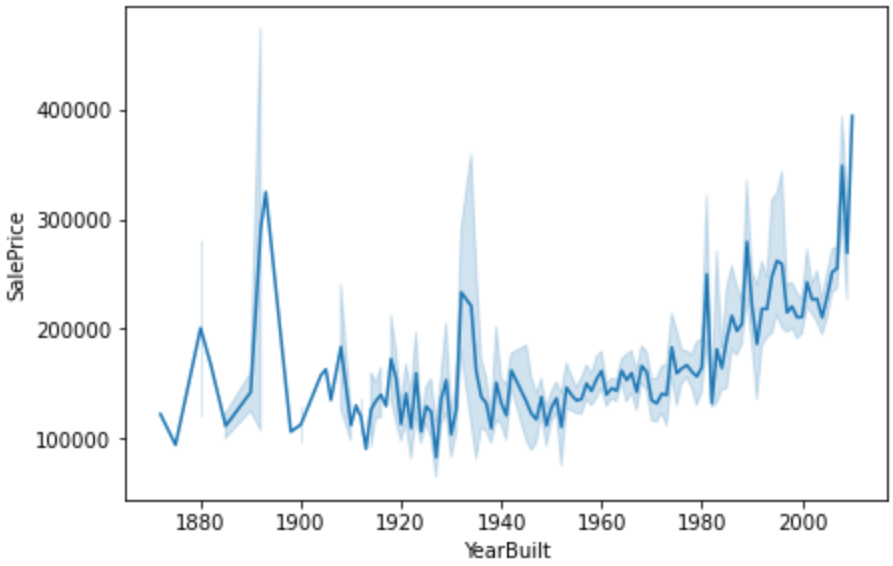
In [30]:
```
df[temporal_variable].head()
```

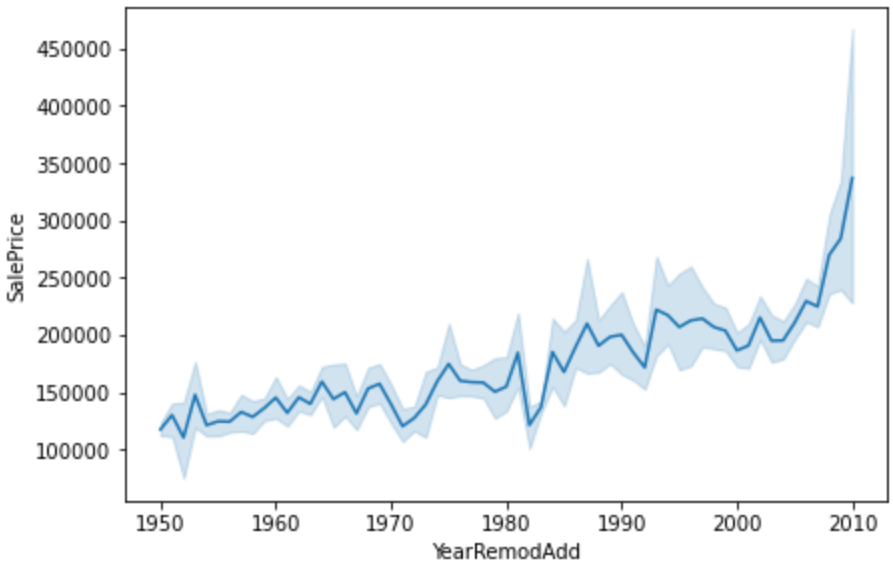
Out[30]:

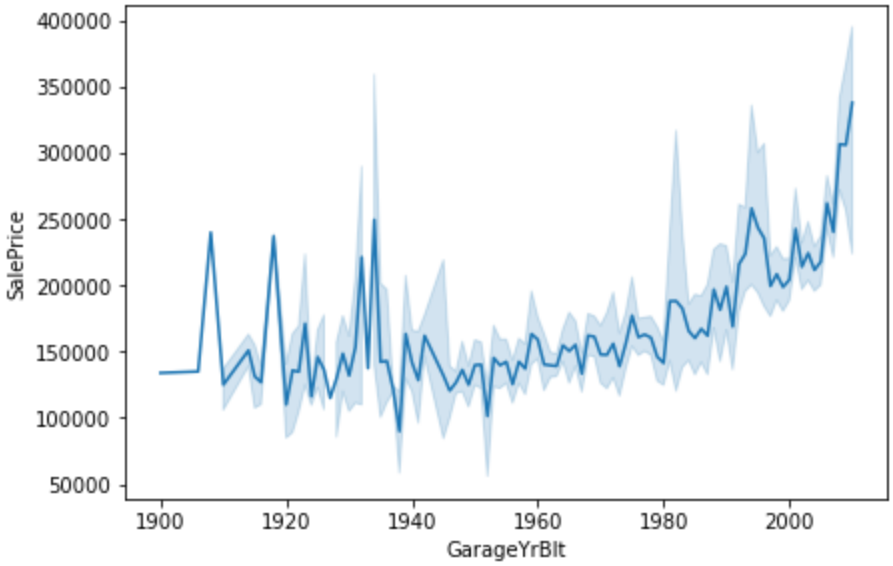
	YearBuilt	YearRemodAdd	GarageYrBlt	YrSold
0	2003	2003	2003.0	2008
1	1976	1976	1976.0	2007
2	2001	2002	2001.0	2008
3	1915	1970	1998.0	2006
4	2000	2000	2000.0	2008

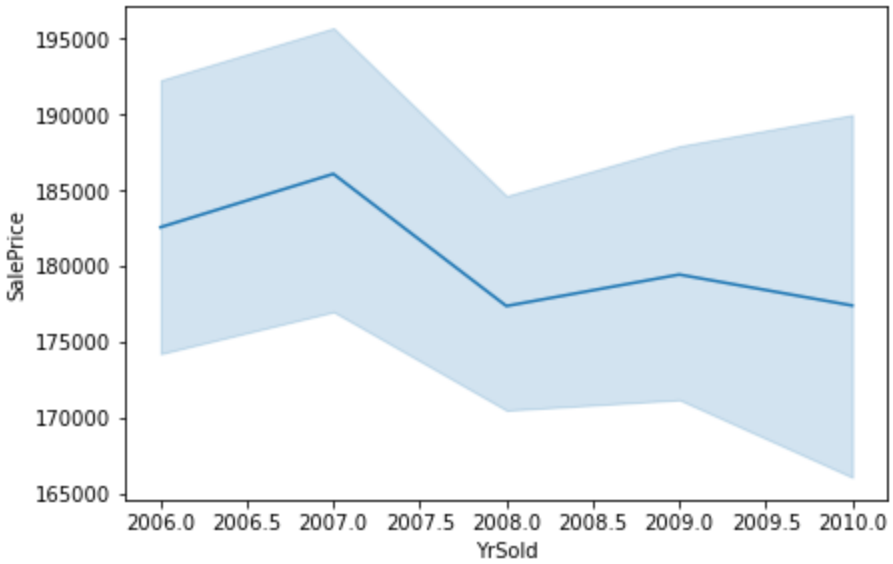
Drawn year features line chart diagram with respect to sales price

In [31]:
```
for i in range(len(temporal_variable)):
    plt.figure(figsize=(15,10))
    plt.subplot(2,2,i+1)
    sns.lineplot(data=df,x=temporal_variable[i],y='SalePrice')
```


SalePrice
400000
300000
200000
100000
1880
1900
1920
1940
1960
1980
2000
YearBuilt


SalePrice
450000
400000
350000
300000
250000
200000
150000
100000
1950
1960
1970
1980
1990
2000
2010
YearRemodAdd

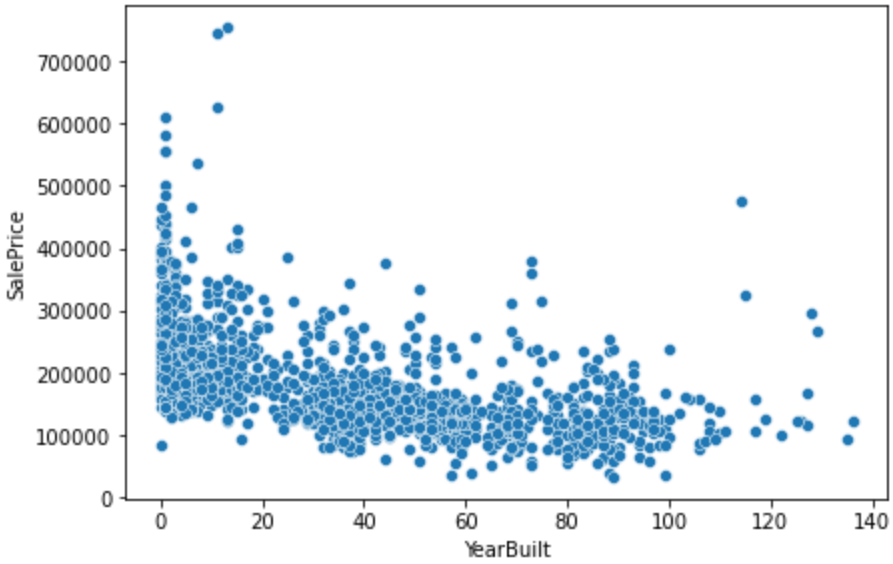

SalePrice
400000
350000
300000
250000
200000
150000
100000
50000
1900
1920
1940
1960
1980
2000
GarageYrBlt

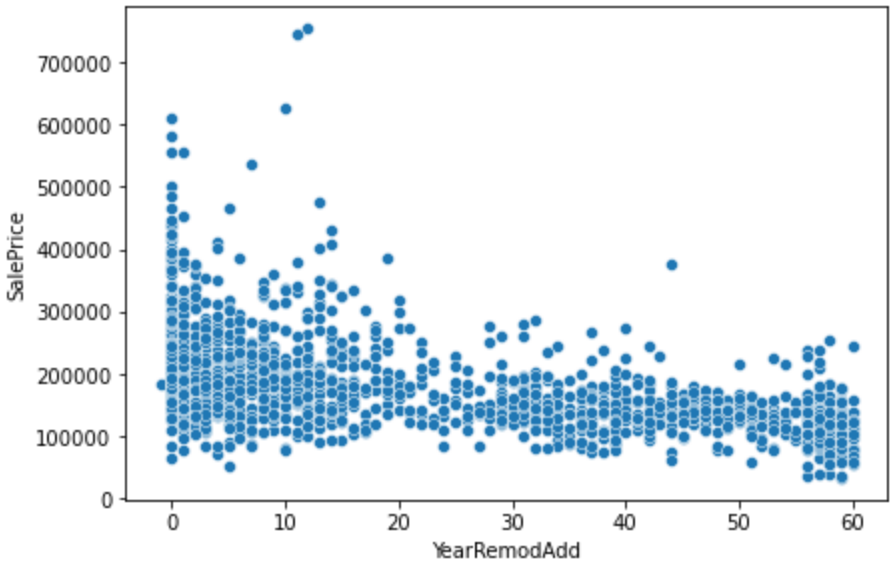


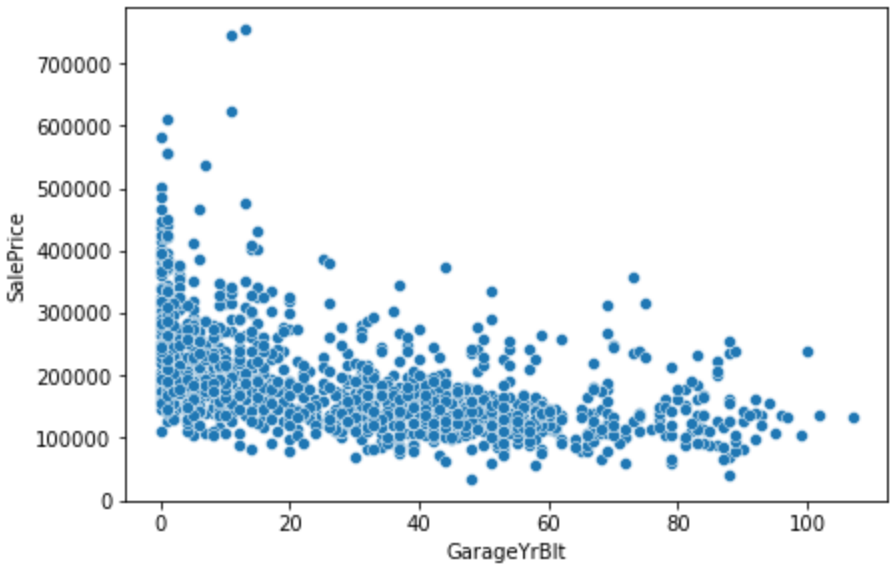
Here below we are subtracting YrSold feature with other year feature, so we get year values with respect to yearsold, after that Drawn a scatterplot

In [34]:

```
for i in range(len(temporal_variable)):
    dataset=df.copy()
    if temporal_variable[i]!='YrSold':
#         print(temporal_variable[i])
        dataset[temporal_variable[i]]=dataset['YrSold']-dataset[temporal_variable[i]]
        plt.figure(figsize=(15,10))
        plt.subplot(2,2,i+1)
        sns.scatterplot(data=df,x=dataset[temporal_variable[i]],y='SalePrice')
```







Conclusion: if YearBuild is between 0-20 year sales price is high , if YearRemodAdd is between 0-10 sales price is high

Discreate features

In [36]:
```
discreate_feature=[feature for feature in numerical_feature if len(df[feature].unique())<2
```

In [37]:
```
discreate_feature
```

Out[37]:
```
['MSSubClass',
 'OverallQual',
 'OverallCond',
 'LowQualFinSF',
 'BsmtFullBath',
 'BsmtHalfBath',
 'FullBath',
 'HalfBath',
 'BedroomAbvGr',
 'KitchenAbvGr',
 'TotRmsAbvGrd',
 'Fireplaces',
 'GarageCars',
 '3SsnPorch',
 'PoolArea',
```

```
'MiscVal',
'MoSold']
```

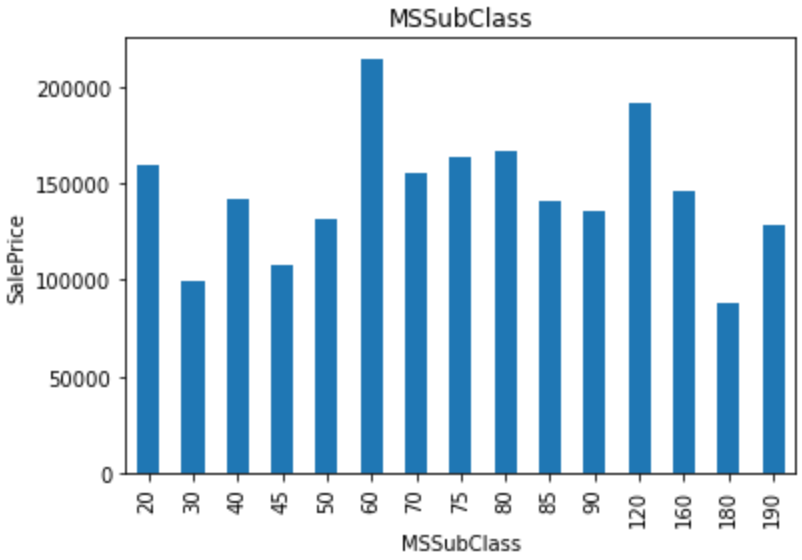
In [25]:

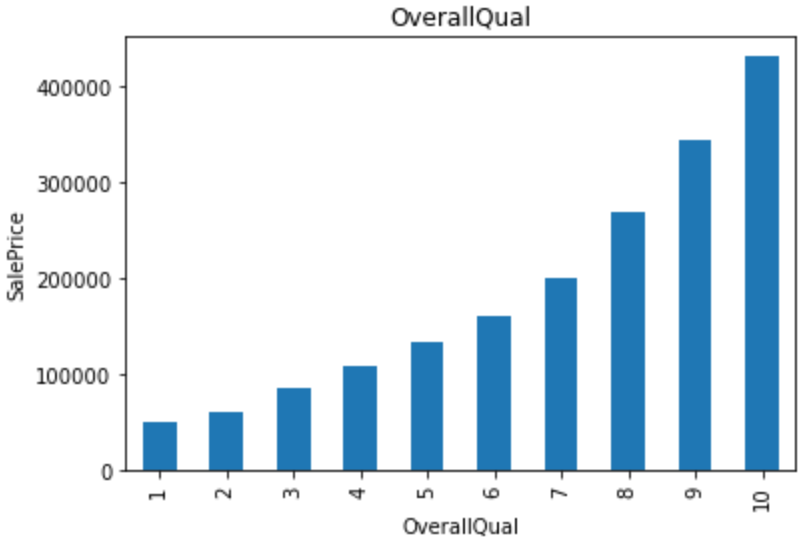
```
# for feature in discreate_feature:
#     data=df.copy()
#     data.groupby(feature)['SalePrice'].count().plot.bar()
#     plt.xlabel(feature)
#     plt.ylabel('SalePrice')
#     plt.title(feature)
#     plt.show()
```

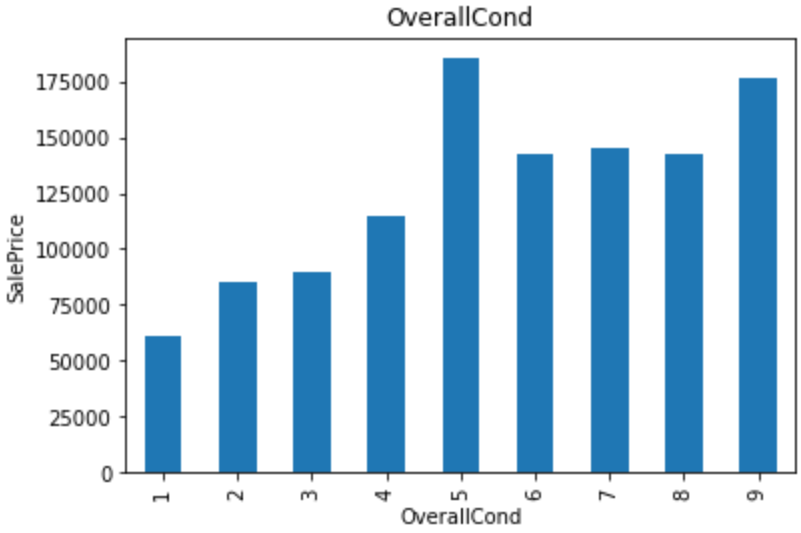
With respect to discreate features drawn a bar plot

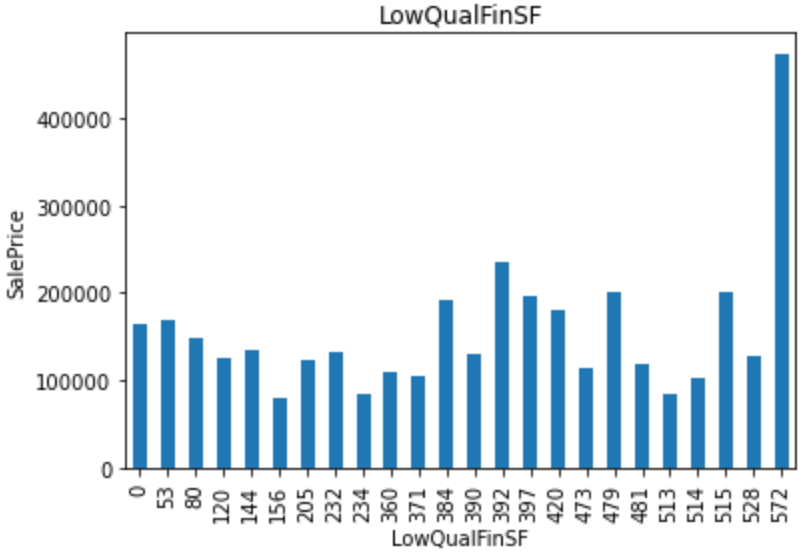
In [38]:

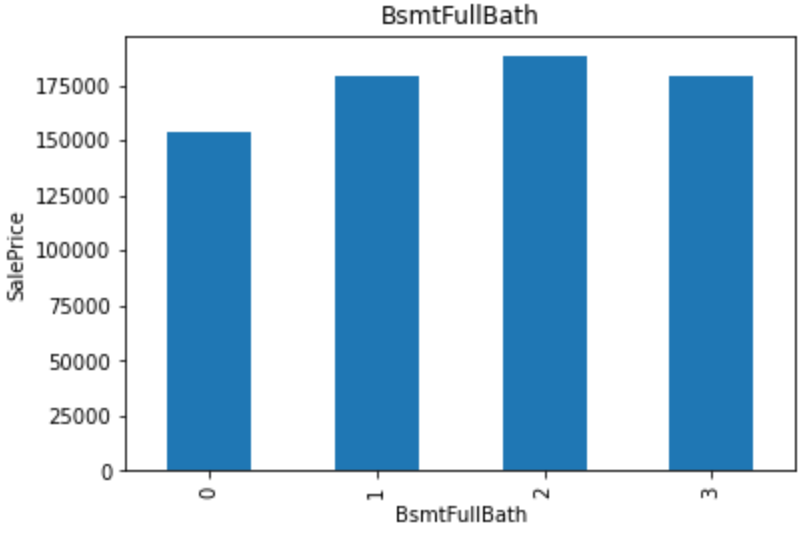
```
for feature in discreate_feature:
    data=df.copy()
    data.groupby(feature)['SalePrice'].median().plot.bar()
    plt.xlabel(feature)
    plt.ylabel('SalePrice')
    plt.title(feature)
    plt.show()
```

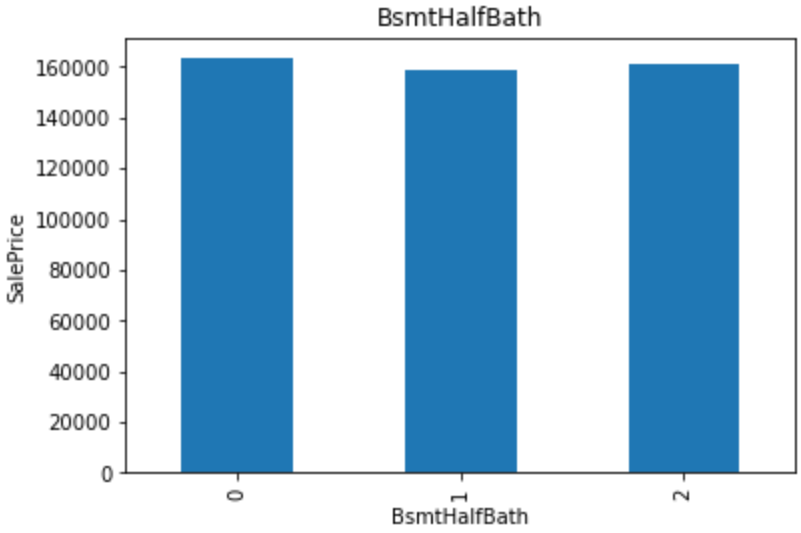


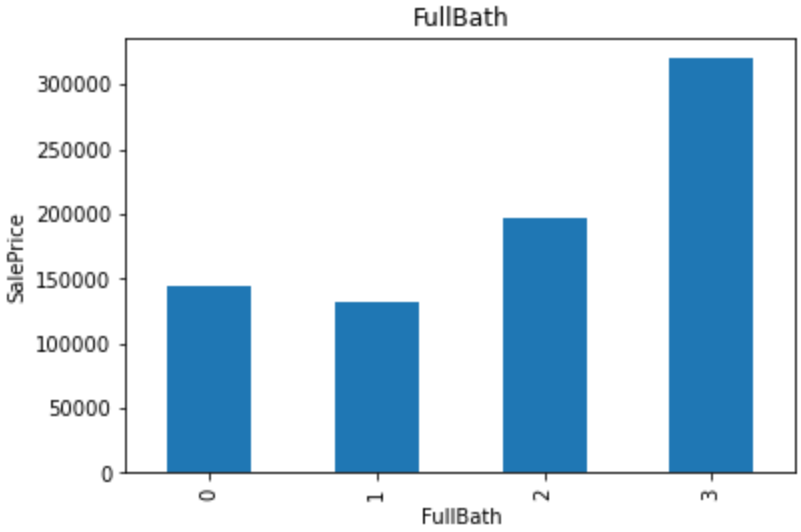


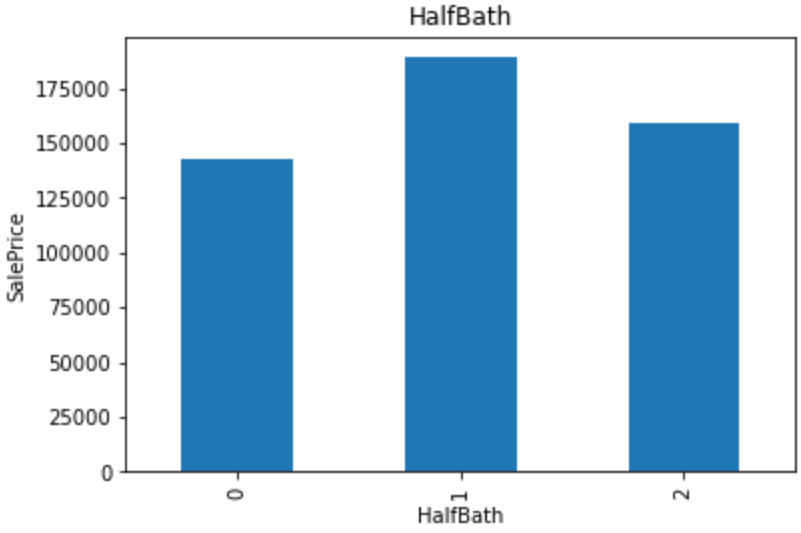

OverallCond
SalePrice
0
25000
50000
75000
100000
125000
150000
175000
1
2
3
4
5
6
7
8
9
OverallCond

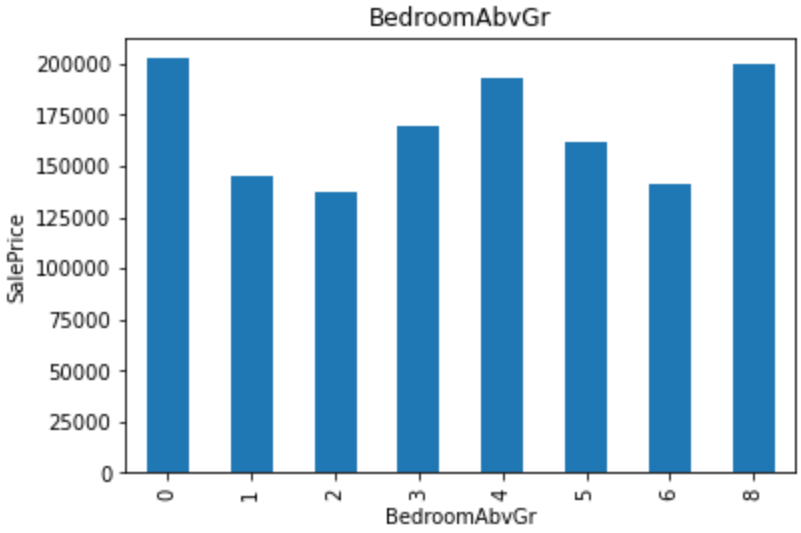

LowQualFinSF
SalePrice
0
100000
200000
300000
400000
0
53
80
120
144
156
205
232
234
360
371
384
390
392
397
420
473
479
481
513
514
515
528
572
LowQualFinSF

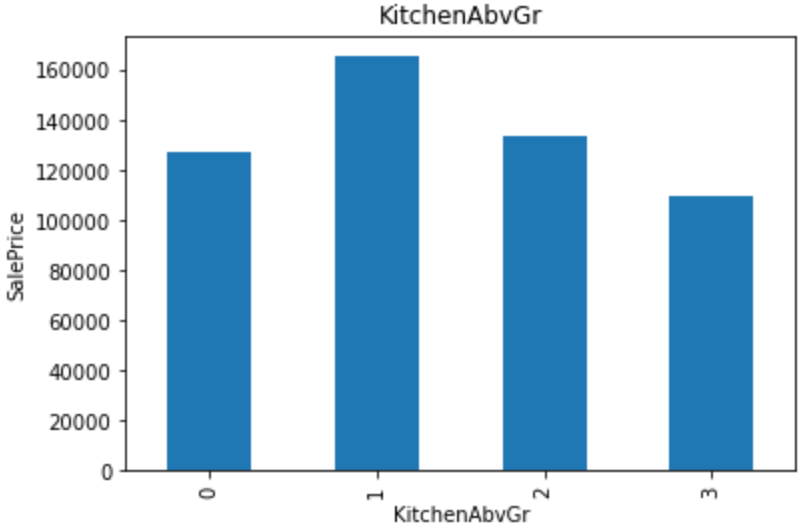

BsmtFullBath
SalePrice
0
25000
50000
75000
100000
125000
150000
175000
0
1
2
3
BsmtFullBath

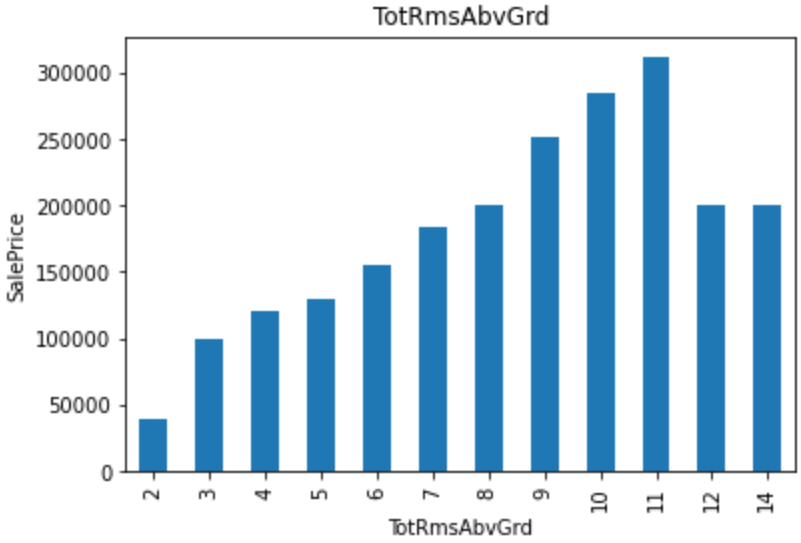

BsmtHalfBath
160000
140000
120000
100000
80000
60000
40000
20000
0
SalePrice
0
1
2
BsmtHalfBath

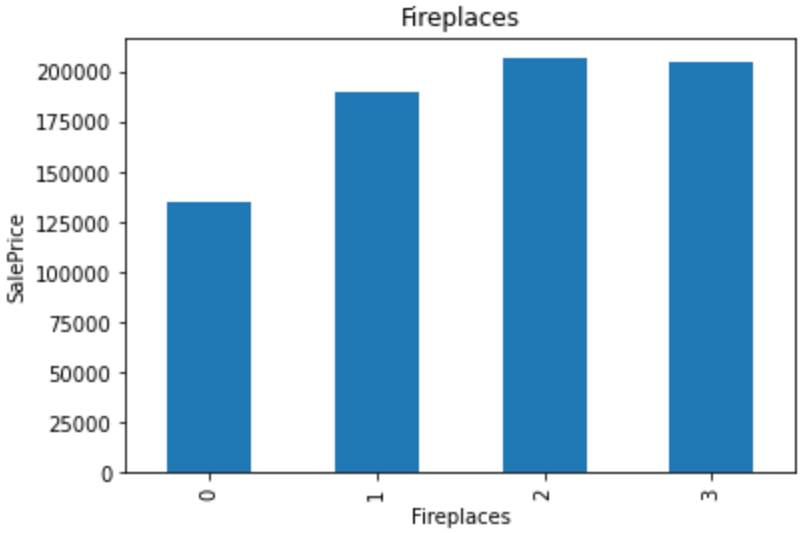

FullBath
300000
250000
200000
150000
100000
50000
0
SalePrice
0
1
2
3
FullBath

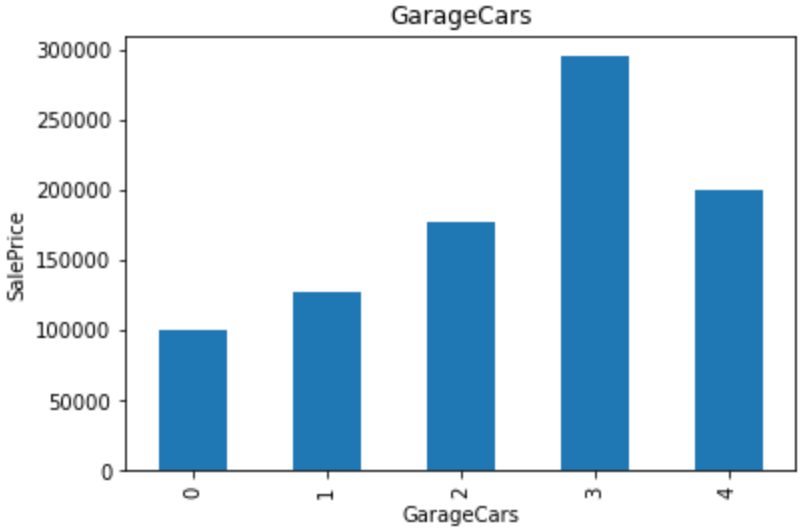

HalfBath
175000
150000
125000
100000
75000
50000
25000
0
SalePrice
0
1
2
HalfBath

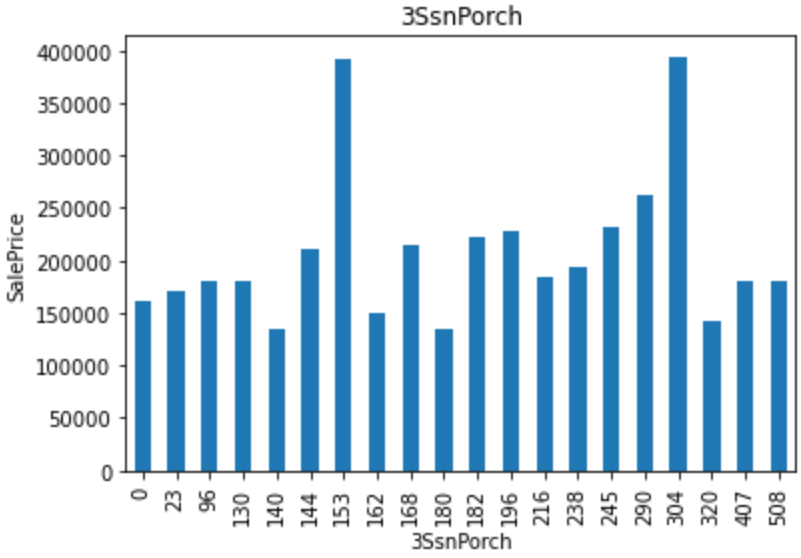

BedroomAbvGr
SalePrice
200000
175000
150000
125000
100000
75000
50000
25000
0
0
1
2
3
4
5
6
8
BedroomAbvGr

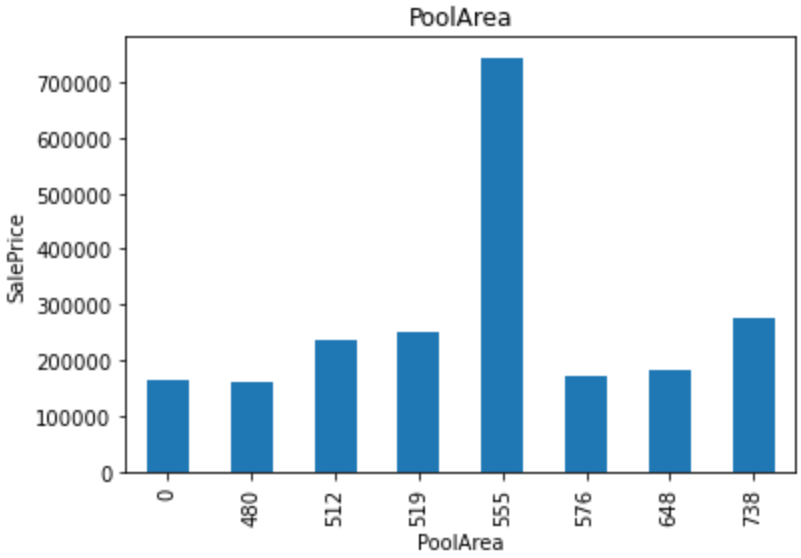

KitchenAbvGr
SalePrice
160000
140000
120000
100000
80000
60000
40000
20000
0
0
1
2
3
KitchenAbvGr

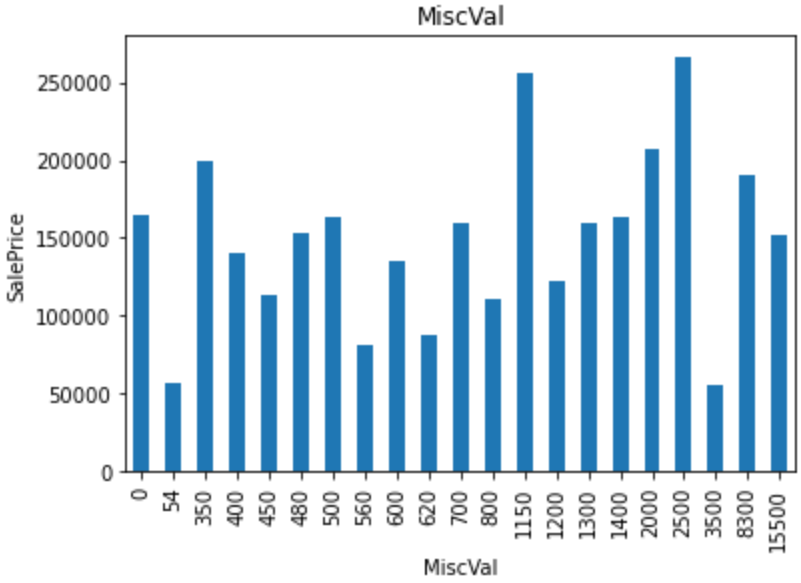

TotRmsAbvGrd
SalePrice
300000
250000
200000
150000
100000
50000
0
2
3
4
5
6
7
8
9
10
11
12
14
TotRmsAbvGrd

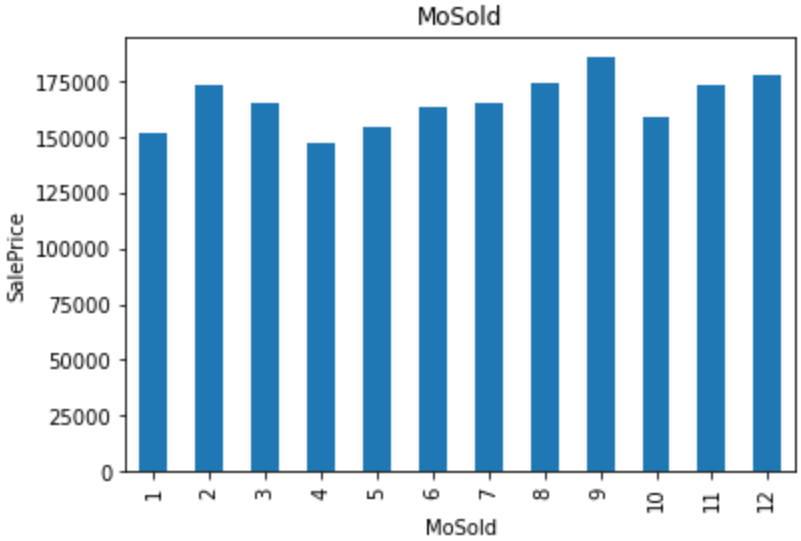

Fireplaces
SalePrice
0
25000
50000
75000
100000
125000
150000
175000
200000
0
1
2
3
Fireplaces


GarageCars
SalePrice
0
50000
100000
150000
200000
250000
300000
0
1
2
3
4
GarageCars


3SsnPorch
SalePrice
0
50000
100000
150000
200000
250000
300000
350000
400000
0
23
96
130
140
144
153
162
168
180
182
196
216
238
245
290
304
320
407
508
3SsnPorch







Conclusion:For some feature we are getting monotonic relationship

continuous features

In [40]:
```
continuous_feature=[feature for feature in numerical_feature if feature not in discreate_f
```

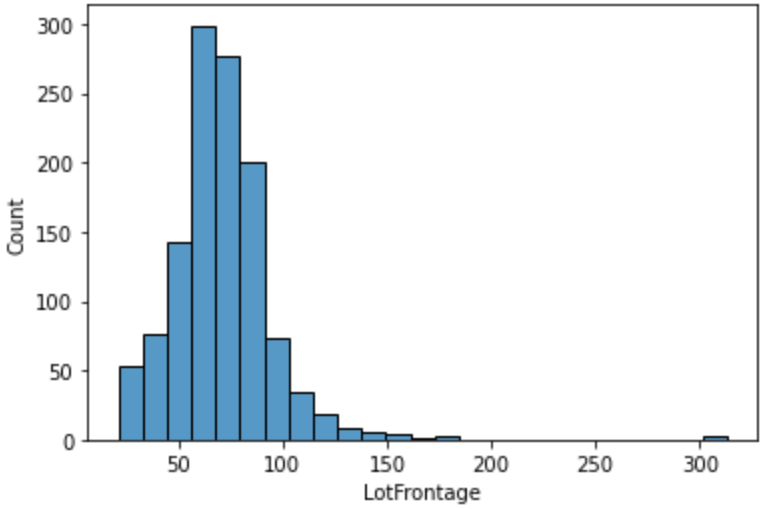
In [41]:

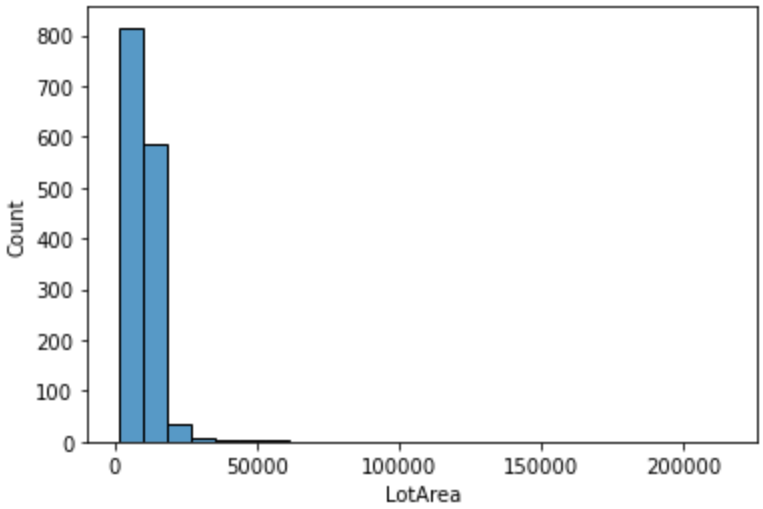
```
continuous_feature
```

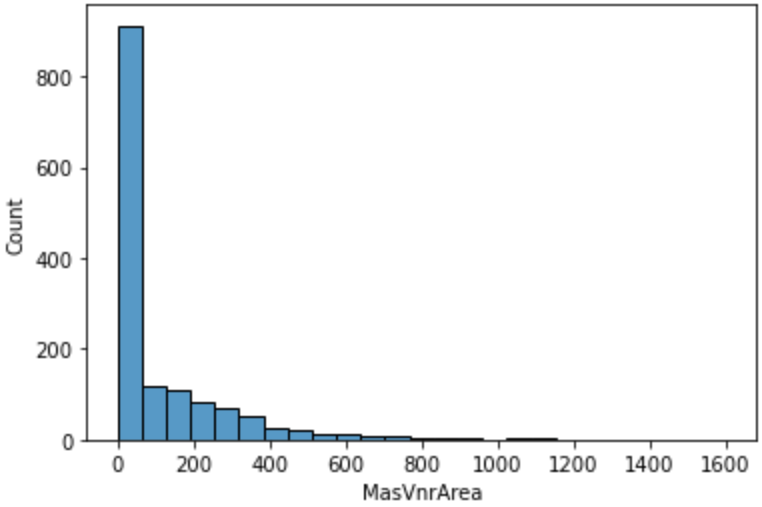
Out[41]:
```
['LotFrontage',
 'LotArea',
 'MasVnrArea',
 'BsmtFinSF1',
 'BsmtFinSF2',
 'BsmtUnfSF',
 'TotalBsmtSF',
 '1stFlrSF',
 '2ndFlrSF',
 'GrLivArea',
 'GarageArea',
 'WoodDeckSF',
 'OpenPorchSF',
 'EnclosedPorch',
 'ScreenPorch',
 'SalePrice']
```

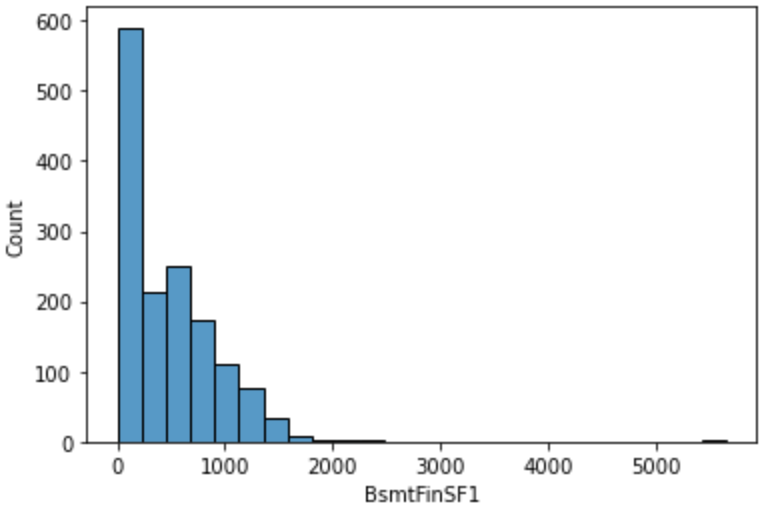
Drawn histplot with respect to continuous feature

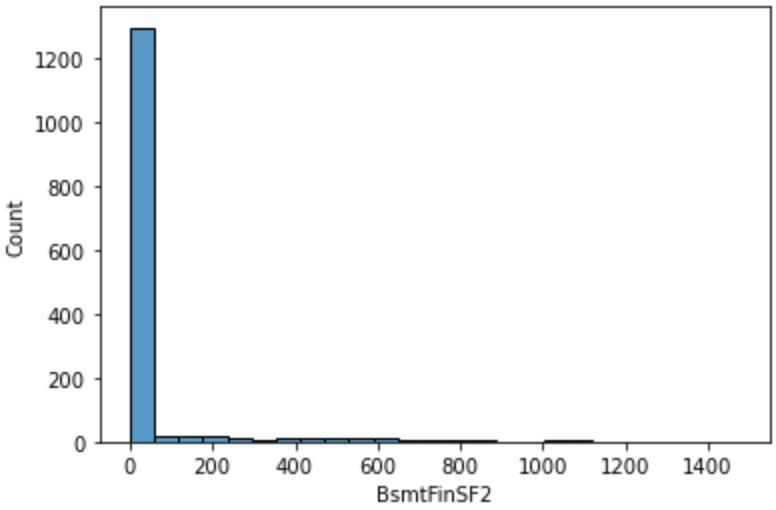
In [42]:
```
for feature in continuous_feature:
    sns.histplot(data=df,x=feature,bins=25)
    plt.show()
```

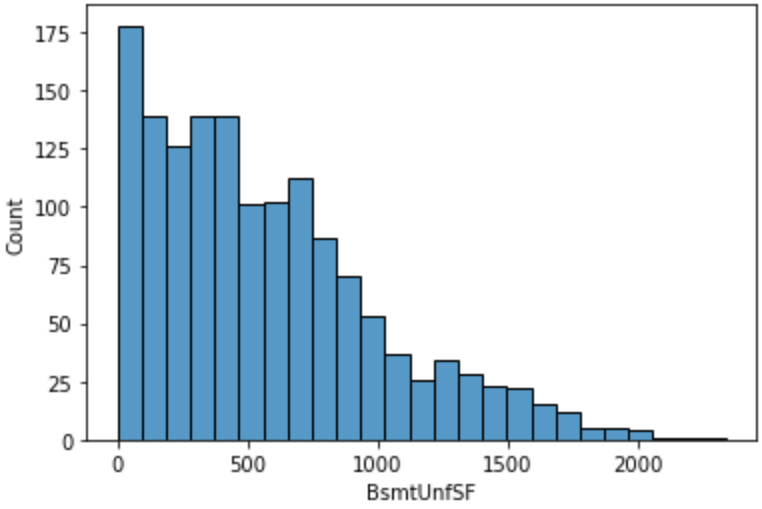


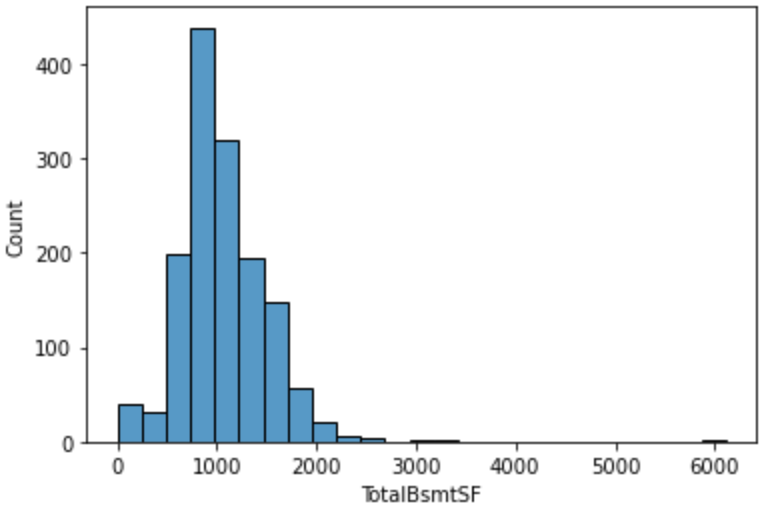


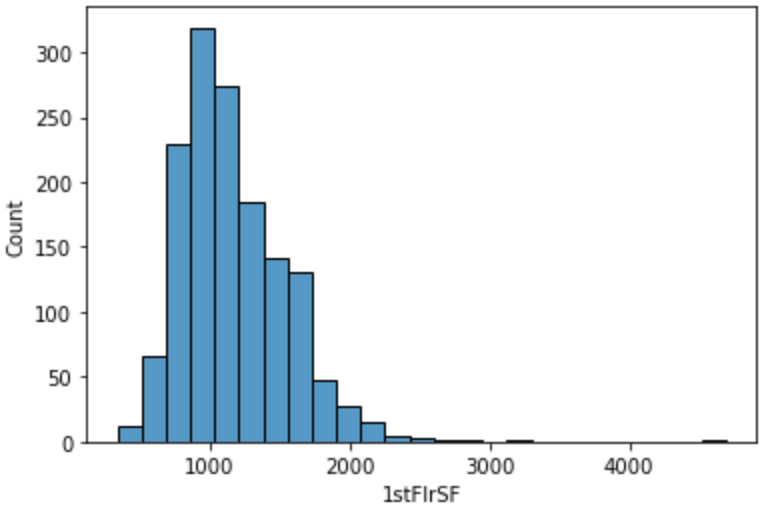

Count
MasVnrArea
0
200
400
600
800
1000
1200
1400
1600

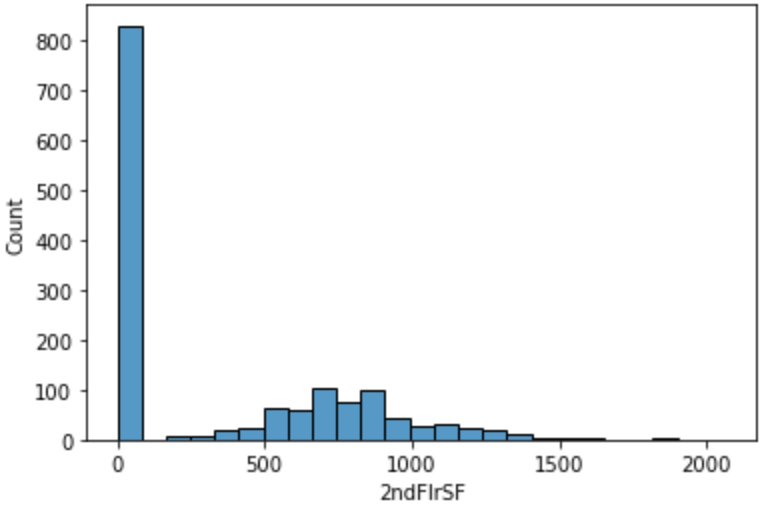

Count
BsmtFinSF1
0
100
200
300
400
500
600
1000
2000
3000
4000
5000

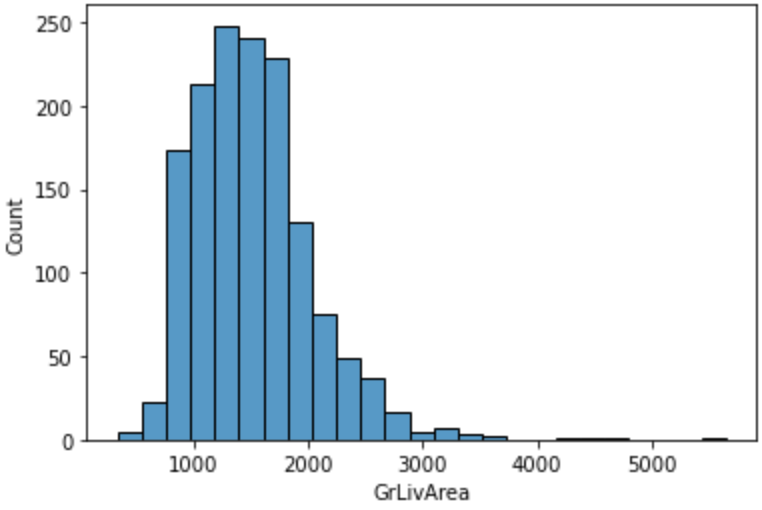

Count
BsmtFinSF2
0
200
400
600
800
1000
1200
1400

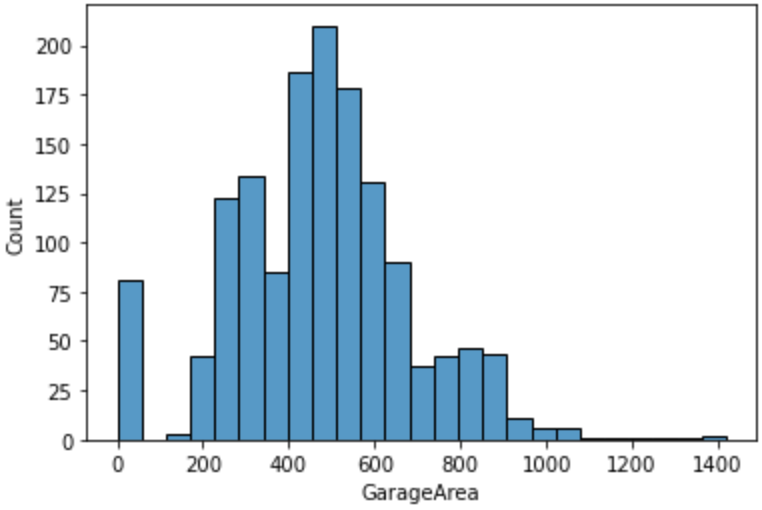

Count
BsmtUnfSF
0
25
50
75
100
125
150
175
0
500
1000
1500
2000

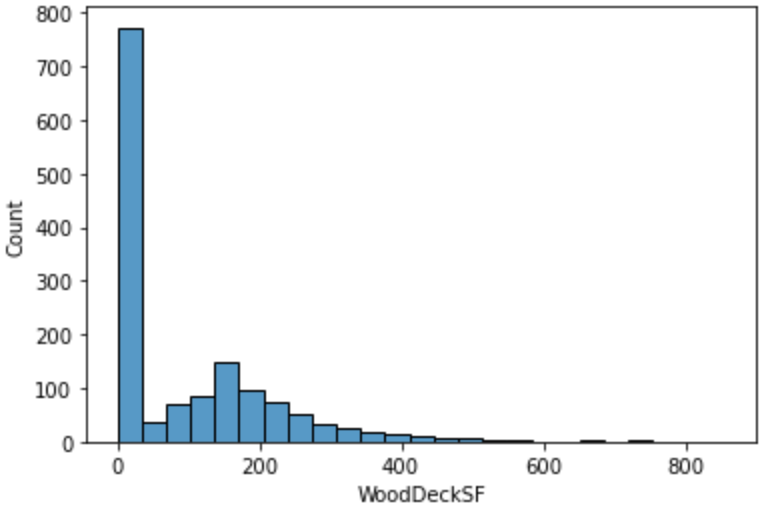

Count
TotalBsmtSF
0
100
200
300
400
0
1000
2000
3000
4000
5000
6000

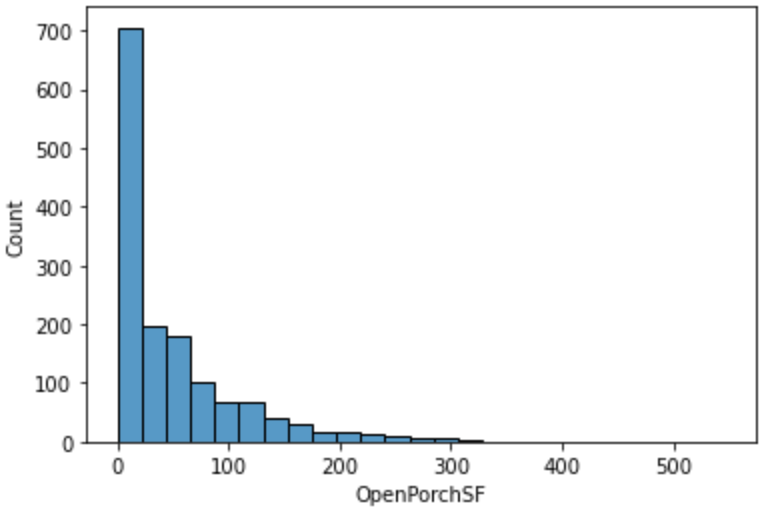

Count
1stFlrSF
0
50
100
150
200
250
300
1000
2000
3000
4000

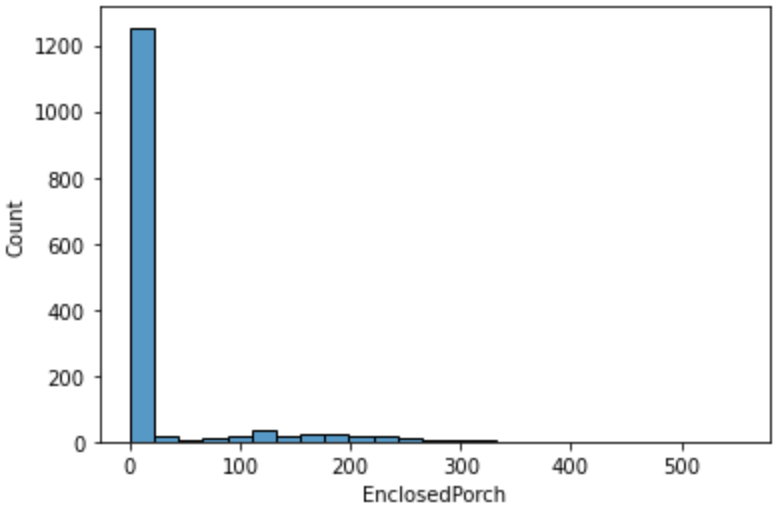

Count
2ndFlrSF
0
100
200
300
400
500
600
700
800
0
500
1000
1500
2000

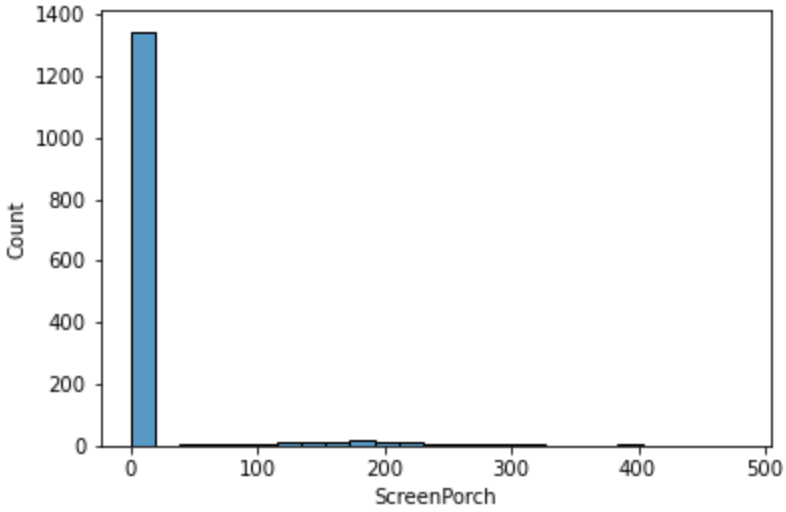

Count
GrLivArea
0
50
100
150
200
250
1000
2000
3000
4000
5000

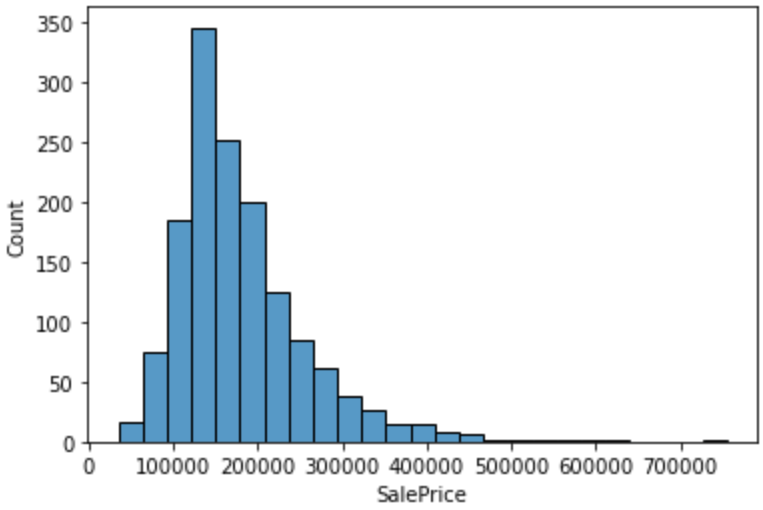

Count
GarageArea
0
25
50
75
100
125
150
175
200
0
200
400
600
800
1000
1200
1400


Count
WoodDeckSF


Count
OpenPorchSF

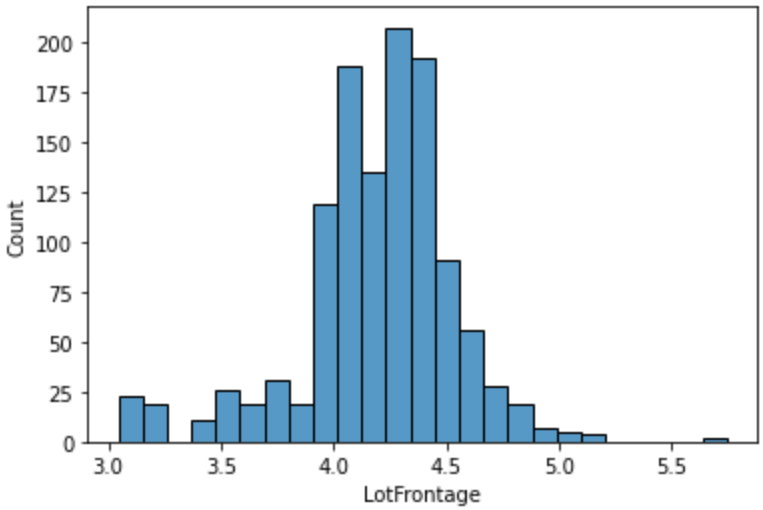

Count
EnclosedPorch

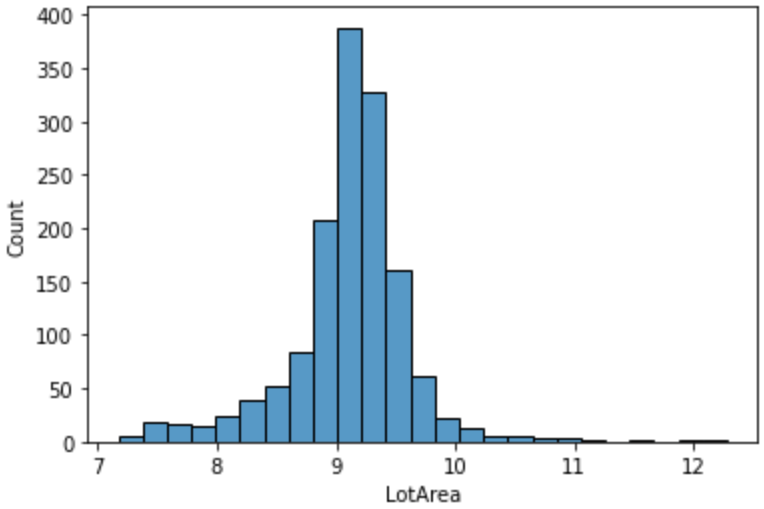




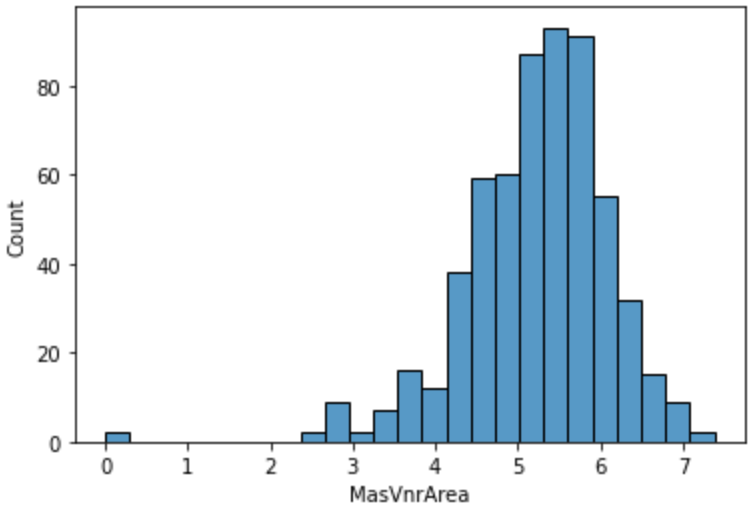
logarithmic tranformation of continuous_feature

In [45]:
```
for feature in continuous_feature:
    data=df.copy()
    data[feature]=np.log(data[feature])
    sns.histplot(data=data,x=data[feature],bins=25)
    plt.show()
```

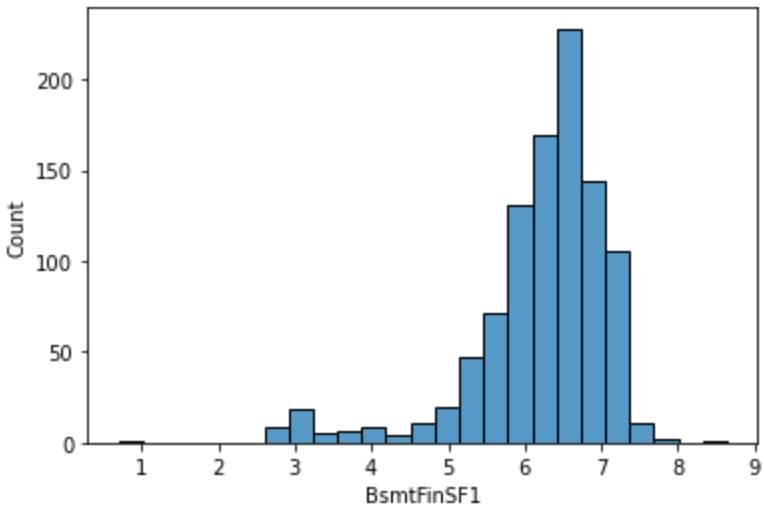




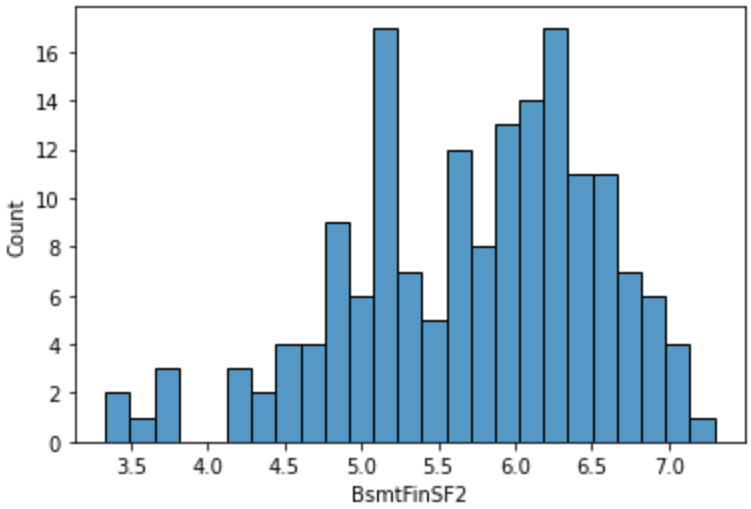

```
C:\Users\40000433\Anaconda3\lib\site-packages\pandas\core\arraylike.py:364: RuntimeWarnin
g: divide by zero encountered in log
  result = getattr(ufunc, method)(*inputs, **kwargs)
```



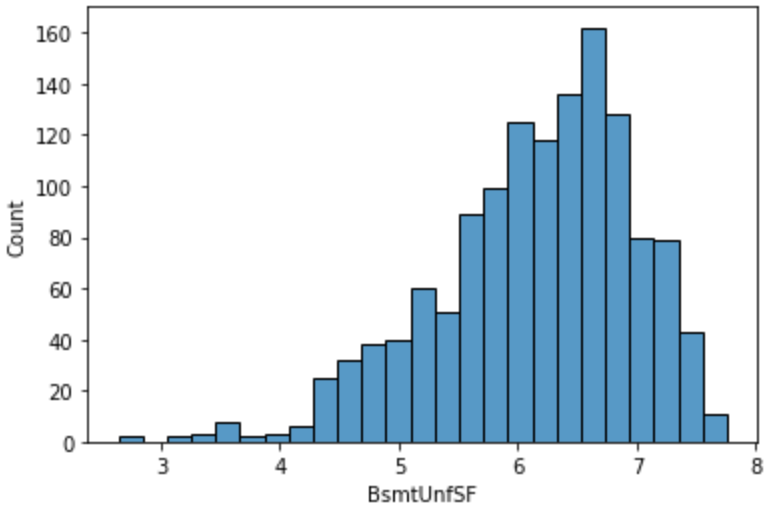

```
C:\Users\40000433\Anaconda3\lib\site-packages\pandas\core\arraylike.py:364: RuntimeWarnin
g: divide by zero encountered in log
  result = getattr(ufunc, method)(*inputs, **kwargs)
```



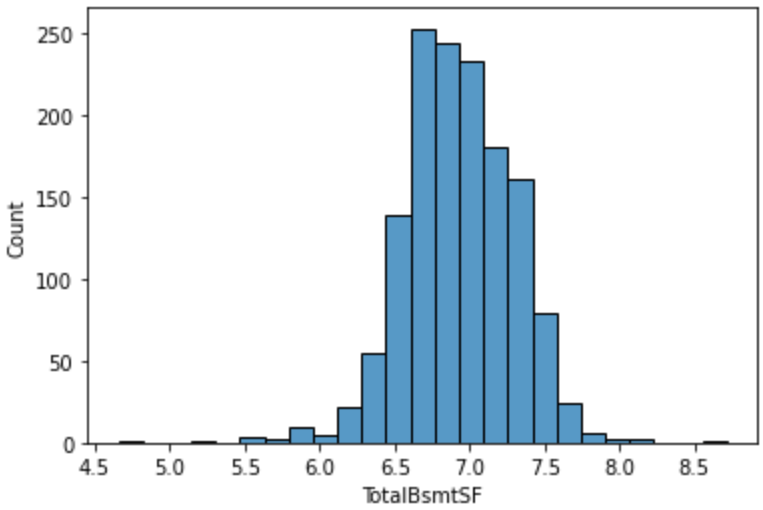

```
C:\Users\40000433\Anaconda3\lib\site-packages\pandas\core\arraylike.py:364: RuntimeWarnin
g: divide by zero encountered in log
  result = getattr(ufunc, method)(*inputs, **kwargs)
```

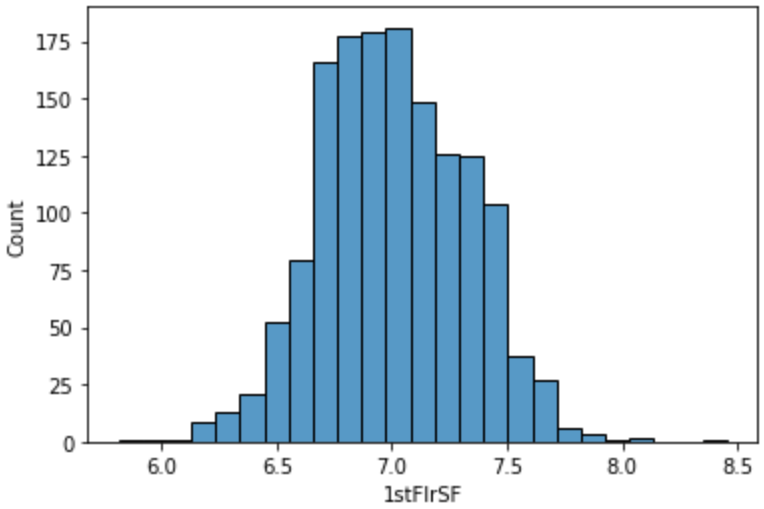



```
C:\Users\40000433\Anaconda3\lib\site-packages\pandas\core\arraylike.py:364: RuntimeWarnin
g: divide by zero encountered in log
  result = getattr(ufunc, method)(*inputs, **kwargs)
```

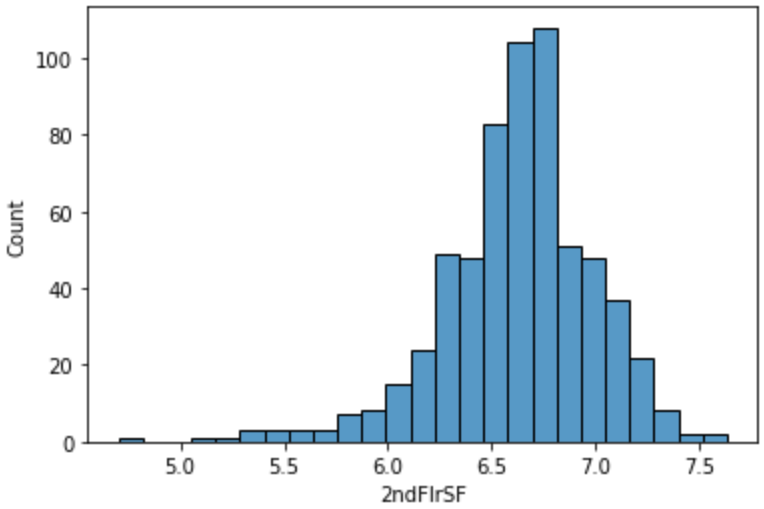


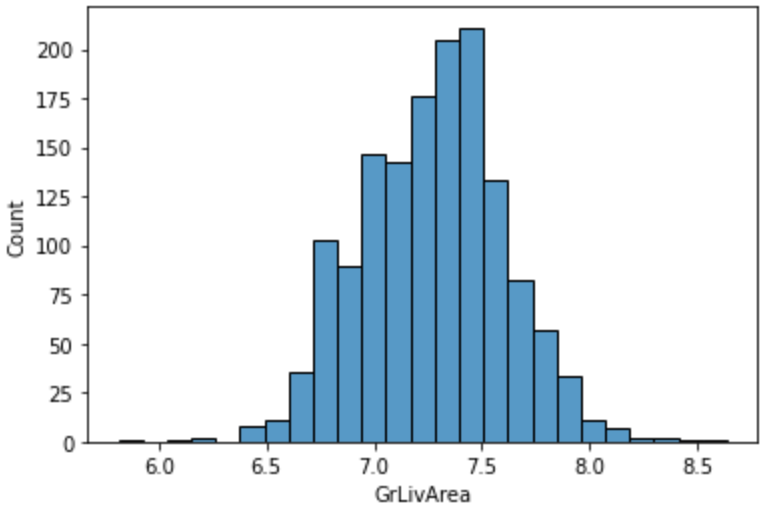

```
C:\Users\40000433\Anaconda3\lib\site-packages\pandas\core\arraylike.py:364: RuntimeWarnin
g: divide by zero encountered in log
  result = getattr(ufunc, method)(*inputs, **kwargs)
```



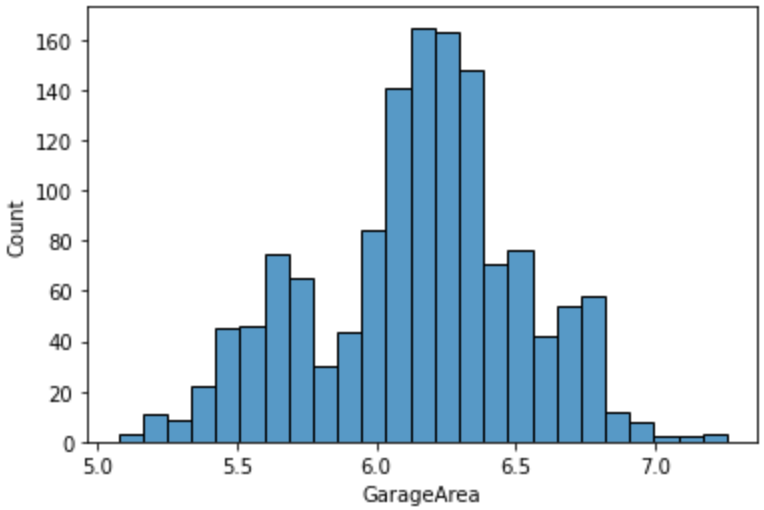



```
C:\Users\40000433\Anaconda3\lib\site-packages\pandas\core\arraylike.py:364: RuntimeWarnin
g: divide by zero encountered in log
  result = getattr(ufunc, method)(*inputs, **kwargs)
```

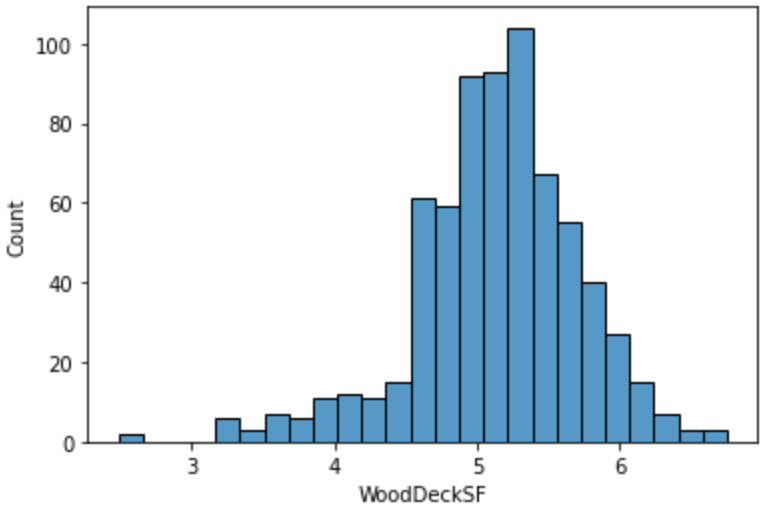




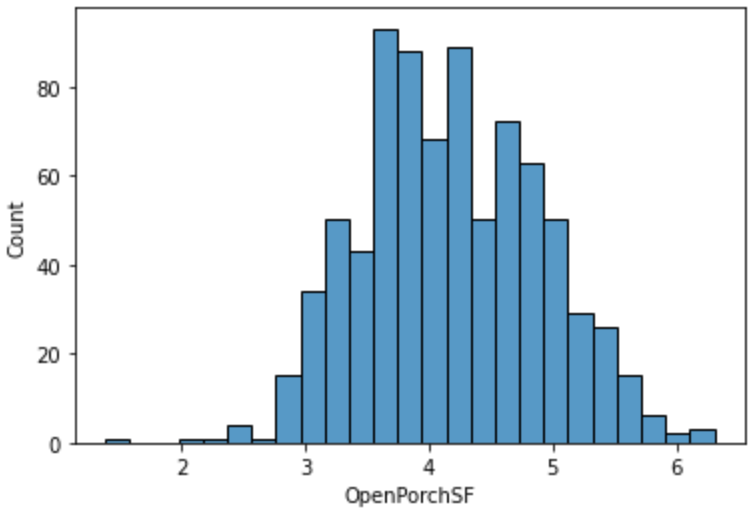

```
C:\Users\40000433\Anaconda3\lib\site-packages\pandas\core\arraylike.py:364: RuntimeWarnin
g: divide by zero encountered in log
  result = getattr(ufunc, method)(*inputs, **kwargs)
```



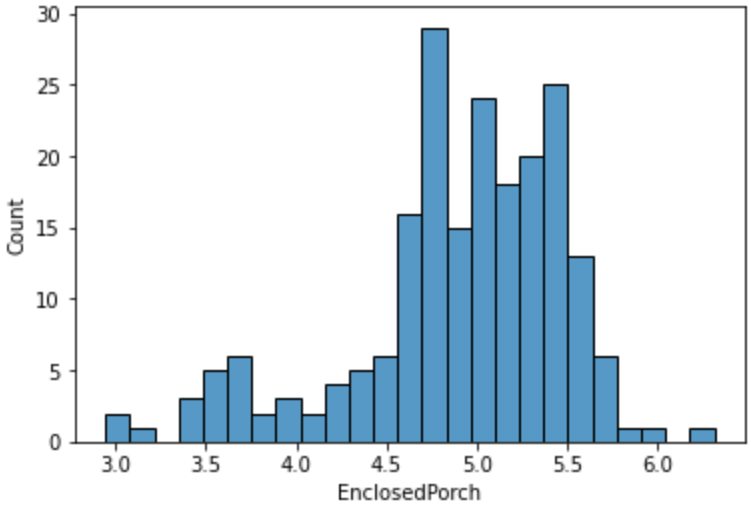

```
C:\Users\40000433\Anaconda3\lib\site-packages\pandas\core\arraylike.py:364: RuntimeWarnin
g: divide by zero encountered in log
  result = getattr(ufunc, method)(*inputs, **kwargs)
```



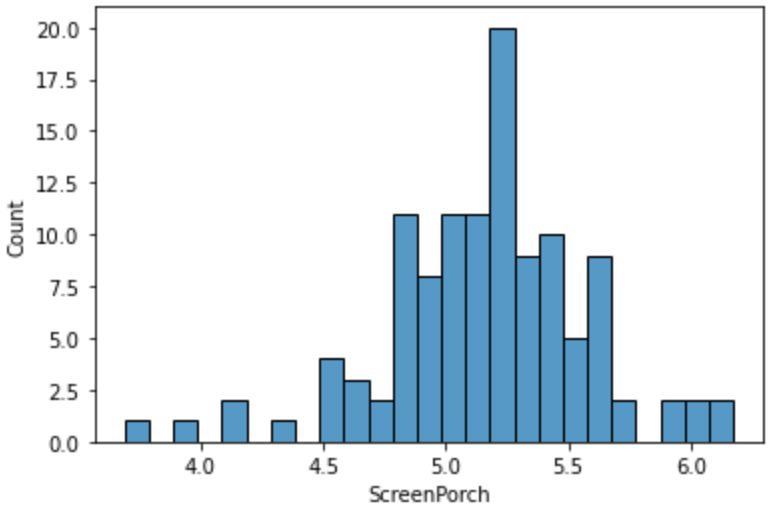

```
C:\Users\40000433\Anaconda3\lib\site-packages\pandas\core\arraylike.py:364: RuntimeWarnin
g: divide by zero encountered in log
  result = getattr(ufunc, method)(*inputs, **kwargs)
```

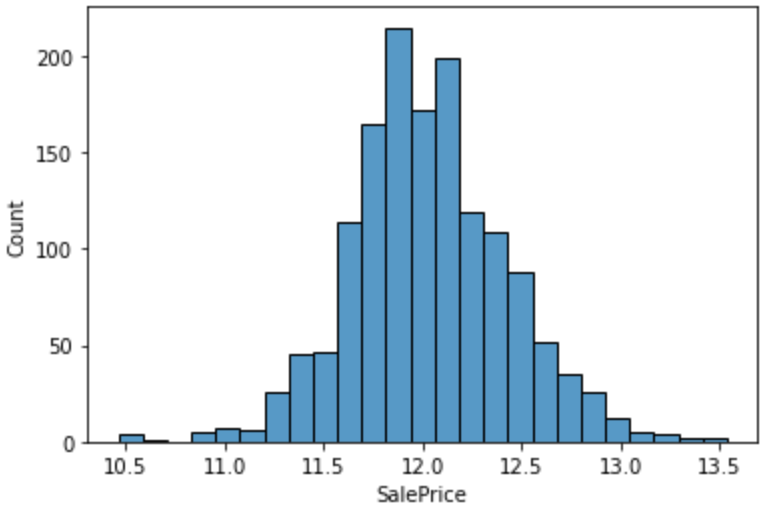



```
C:\Users\40000433\Anaconda3\lib\site-packages\pandas\core\arraylike.py:364: RuntimeWarnin
g: divide by zero encountered in log
  result = getattr(ufunc, method)(*inputs, **kwargs)
```




```
C:\Users\40000433\Anaconda3\lib\site-packages\pandas\core\arraylike.py:364: RuntimeWarnin
g: divide by zero encountered in log
  result = getattr(ufunc, method)(*inputs, **kwargs)
```

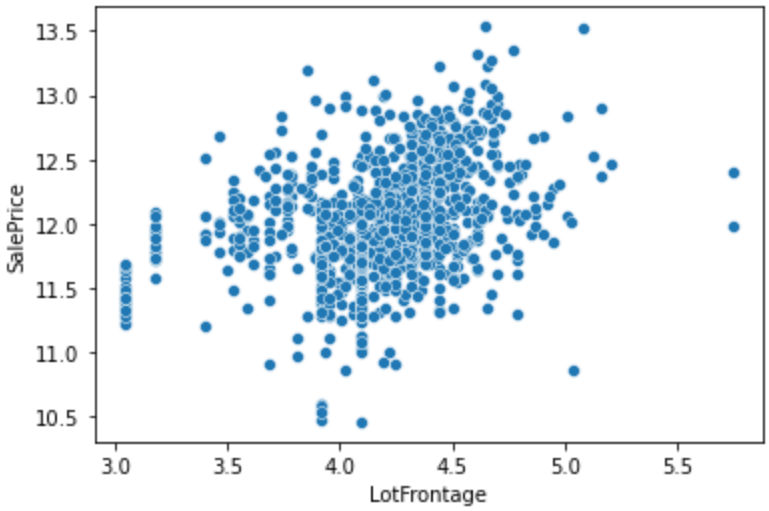


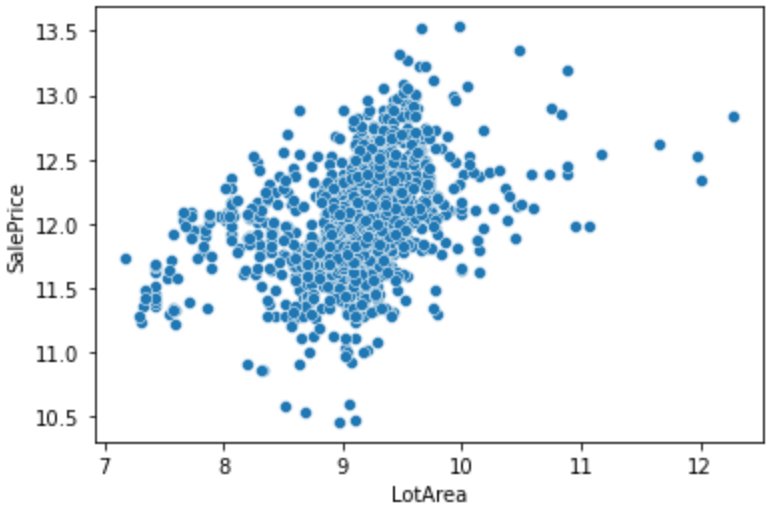


After logarithmic tranformation drawn scatterplot with respect to continuous feature

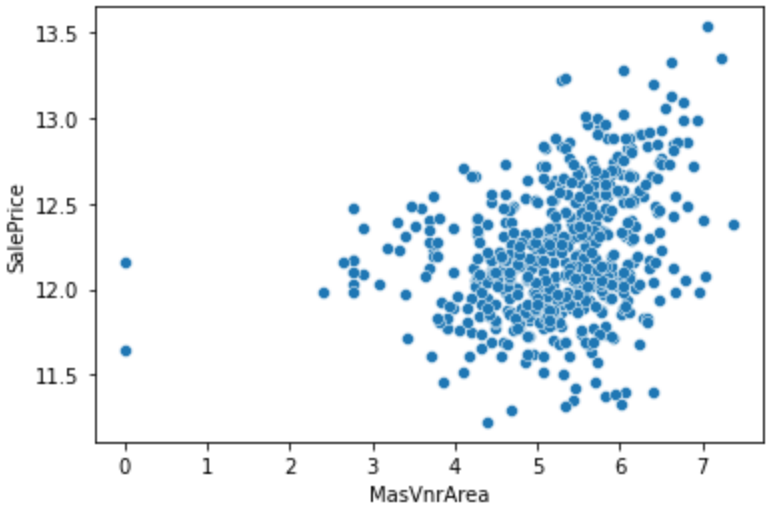
In [47]:

```
data=df.copy()
data['SalePrice']=np.log(data['SalePrice'])
for feature in continuous_feature:
#     data=df.copy()
    data[feature]=np.log(data[feature])
    sns.Drawn(data=data,x=data[feature],y=data['SalePrice'])
    plt.show()
```

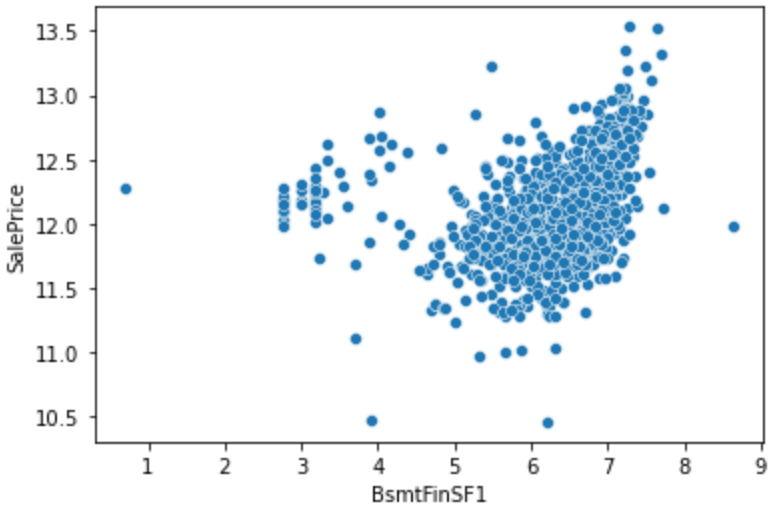




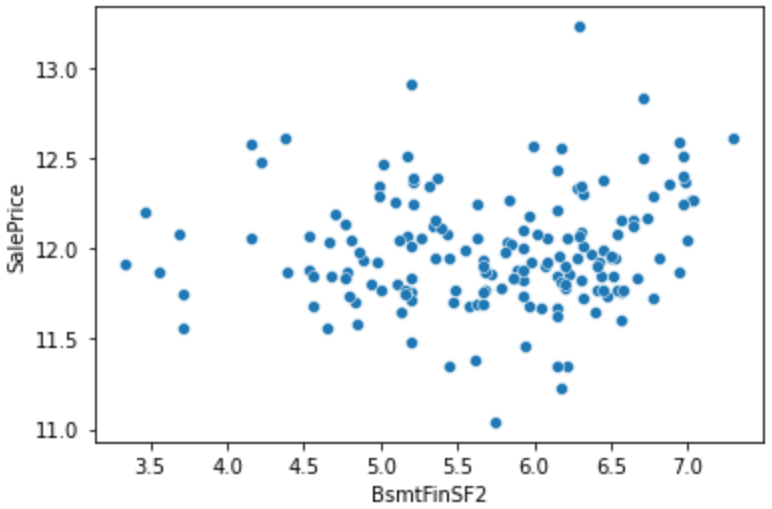

```
C:\Users\40000433\Anaconda3\lib\site-packages\pandas\core\arraylike.py:364: RuntimeWarnin
g: divide by zero encountered in log
  result = getattr(ufunc, method)(*inputs, **kwargs)
```



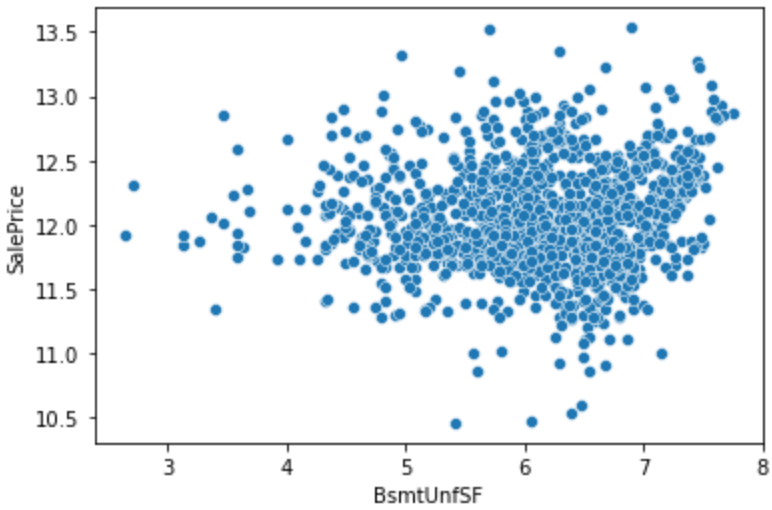

```
C:\Users\40000433\Anaconda3\lib\site-packages\pandas\core\arraylike.py:364: RuntimeWarnin
g: divide by zero encountered in log
  result = getattr(ufunc, method)(*inputs, **kwargs)
```



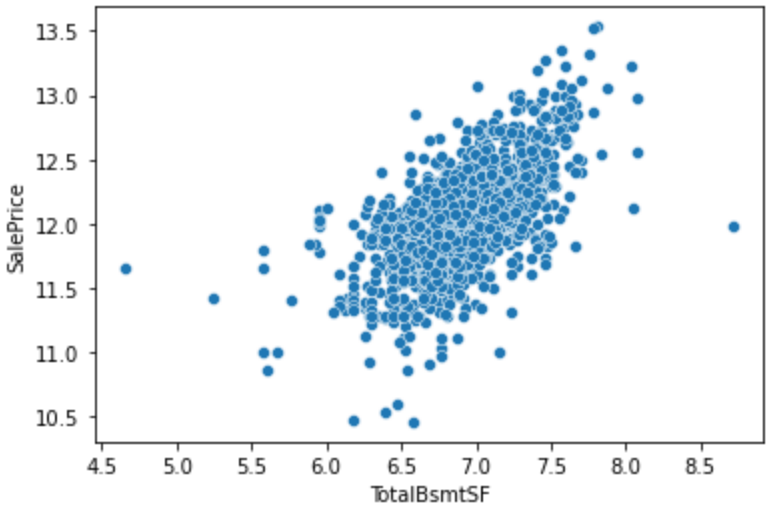

```
C:\Users\40000433\Anaconda3\lib\site-packages\pandas\core\arraylike.py:364: RuntimeWarnin
g: divide by zero encountered in log
  result = getattr(ufunc, method)(*inputs, **kwargs)
```

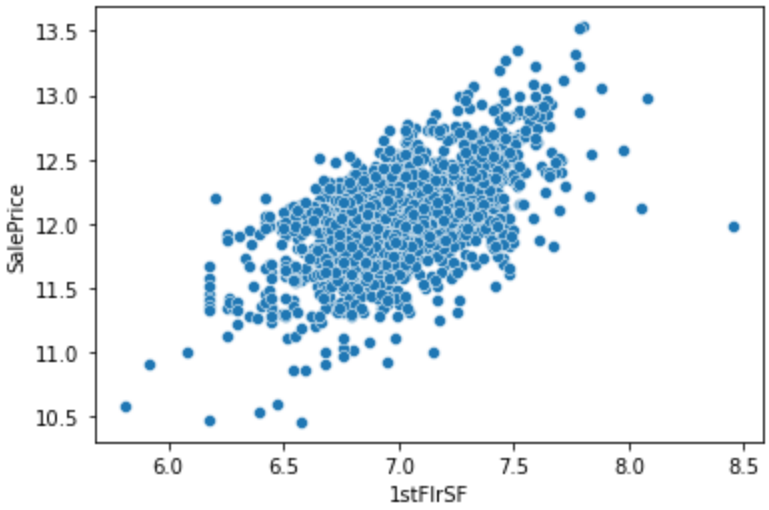



```
C:\Users\40000433\Anaconda3\lib\site-packages\pandas\core\arraylike.py:364: RuntimeWarnin
g: divide by zero encountered in log
  result = getattr(ufunc, method)(*inputs, **kwargs)
```

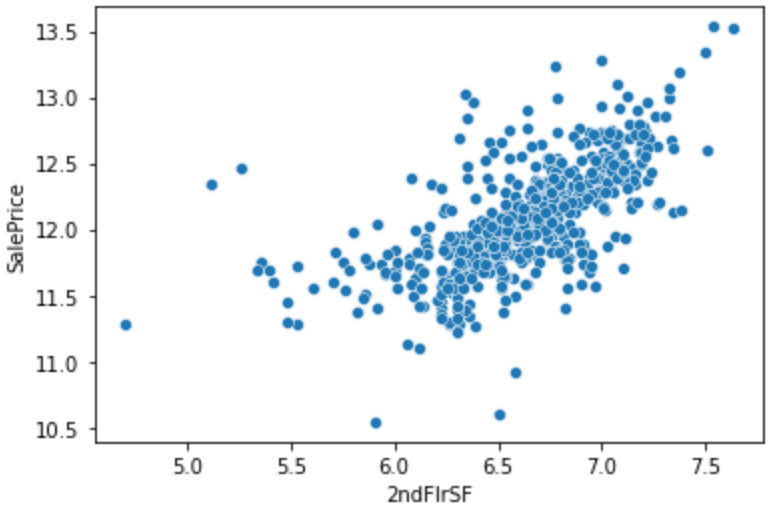


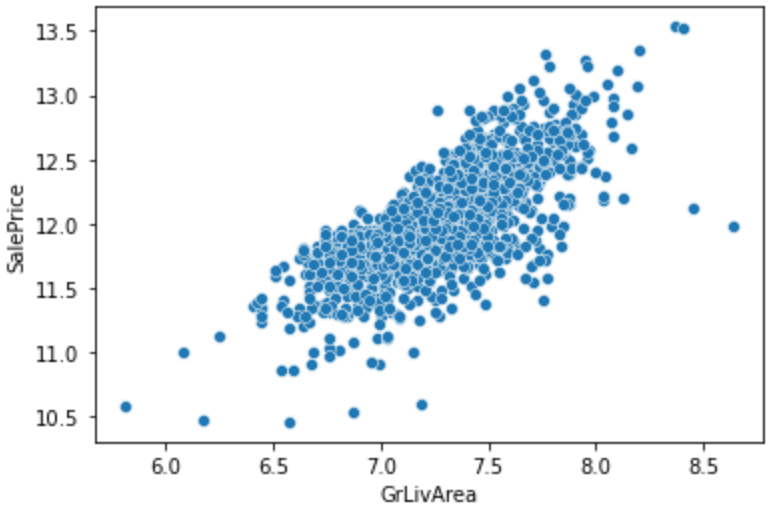

```
C:\Users\40000433\Anaconda3\lib\site-packages\pandas\core\arraylike.py:364: RuntimeWarnin
g: divide by zero encountered in log
  result = getattr(ufunc, method)(*inputs, **kwargs)
```



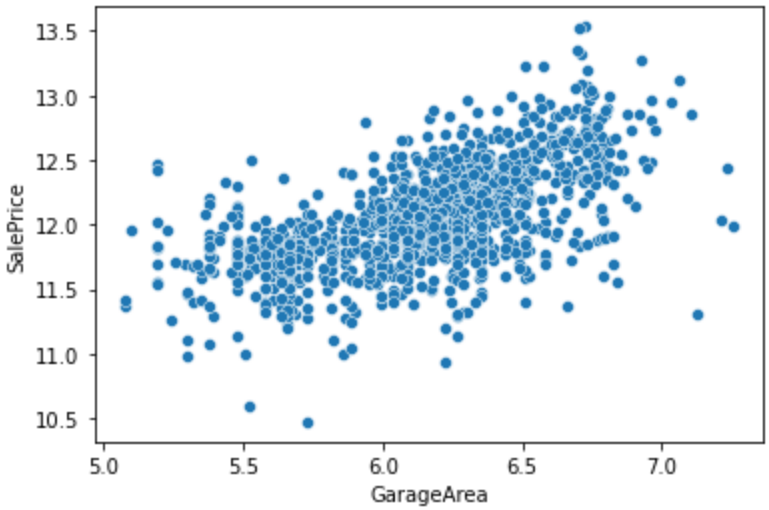



```
C:\Users\40000433\Anaconda3\lib\site-packages\pandas\core\arraylike.py:364: RuntimeWarnin
g: divide by zero encountered in log
  result = getattr(ufunc, method)(*inputs, **kwargs)
```

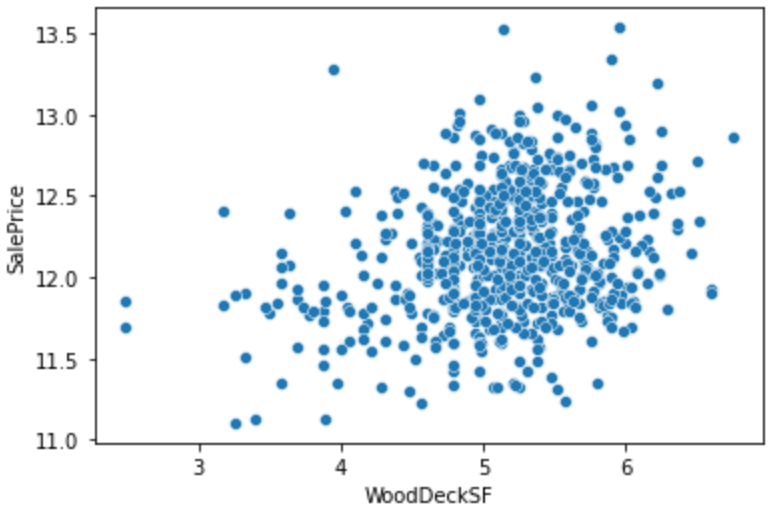




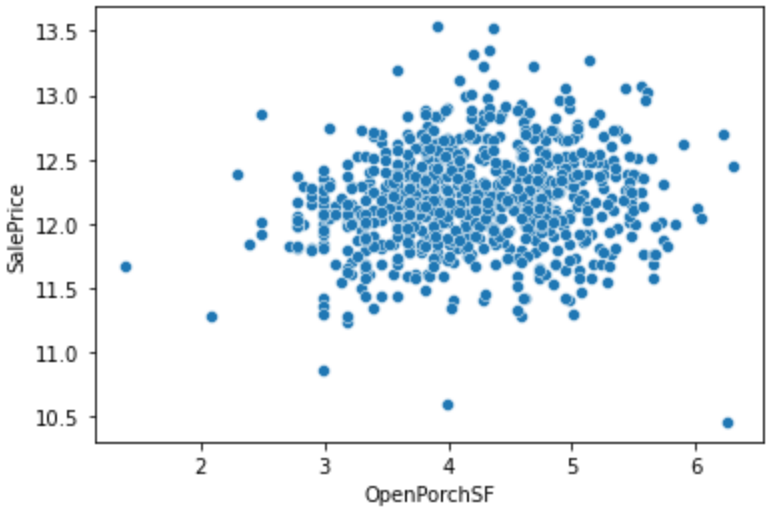

```
C:\Users\40000433\Anaconda3\lib\site-packages\pandas\core\arraylike.py:364: RuntimeWarnin
g: divide by zero encountered in log
  result = getattr(ufunc, method)(*inputs, **kwargs)
```



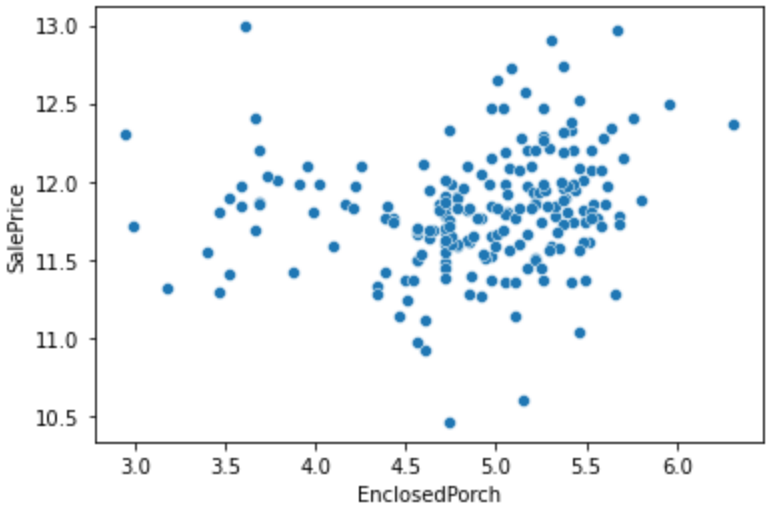

```
C:\Users\40000433\Anaconda3\lib\site-packages\pandas\core\arraylike.py:364: RuntimeWarnin
g: divide by zero encountered in log
  result = getattr(ufunc, method)(*inputs, **kwargs)
```



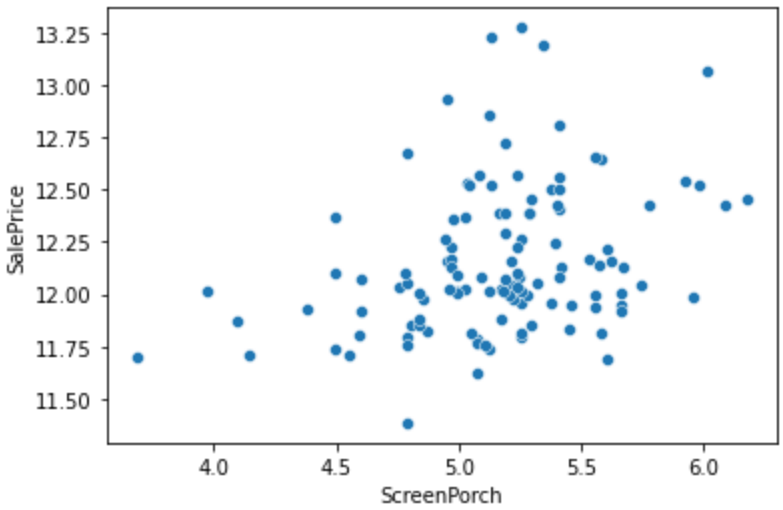

```
C:\Users\40000433\Anaconda3\lib\site-packages\pandas\core\arraylike.py:364: RuntimeWarnin
g: divide by zero encountered in log
  result = getattr(ufunc, method)(*inputs, **kwargs)
```

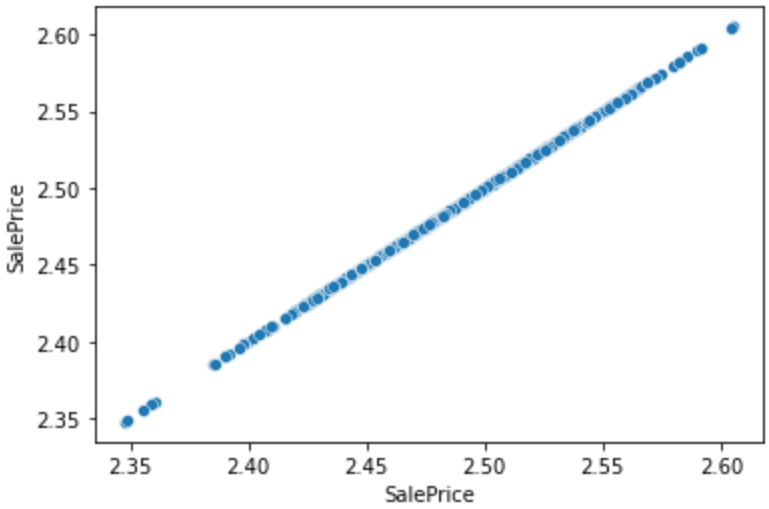



```
C:\Users\40000433\Anaconda3\lib\site-packages\pandas\core\arraylike.py:364: RuntimeWarnin
g: divide by zero encountered in log
  result = getattr(ufunc, method)(*inputs, **kwargs)
```




```
C:\Users\40000433\Anaconda3\lib\site-packages\pandas\core\arraylike.py:364: RuntimeWarnin
g: divide by zero encountered in log
  result = getattr(ufunc, method)(*inputs, **kwargs)
```



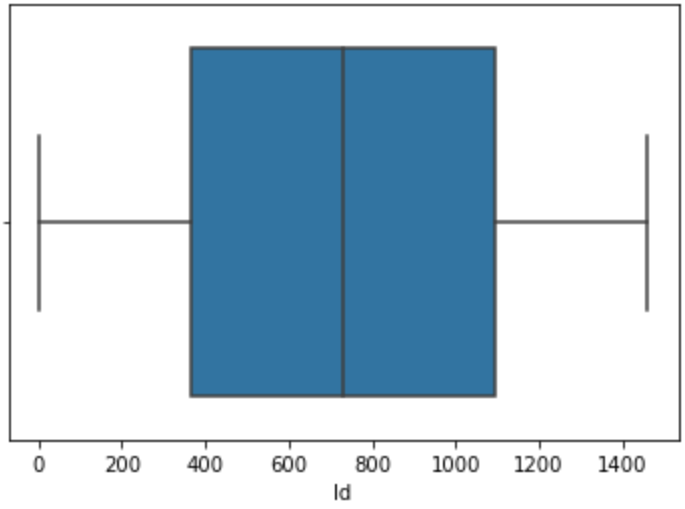


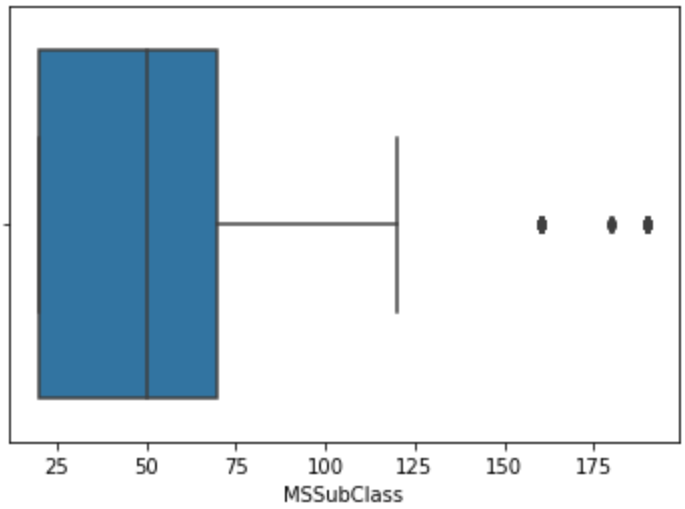
In [48]:

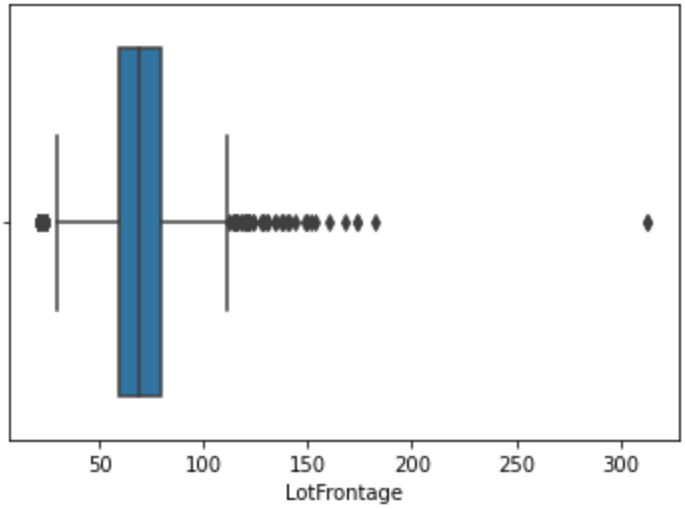
```
# checking outliears with respect to numerical Feature with the help of BoxPlot
```

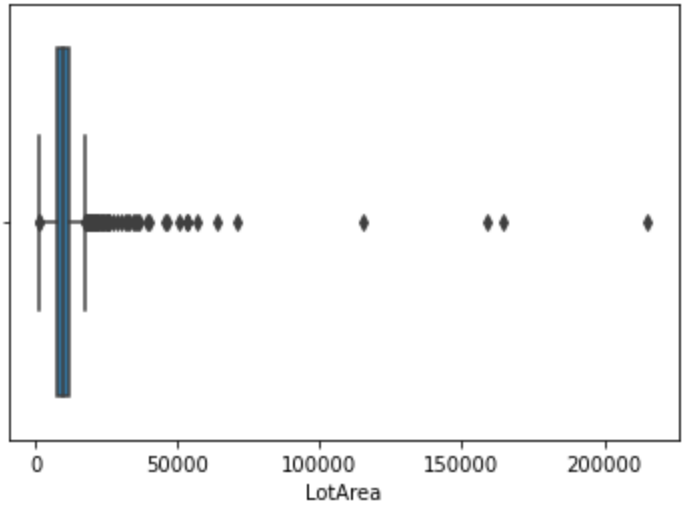
In [49]:

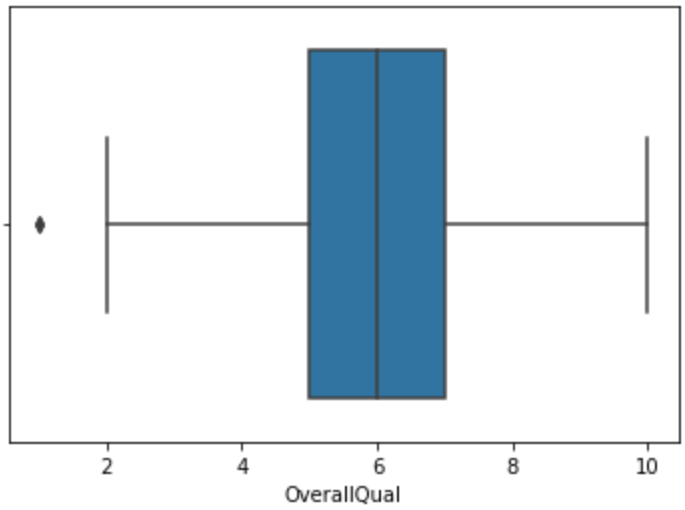
```
for feature in numerical_feature:
    sns.boxplot(data=df,x=feature)
    plt.show()
```

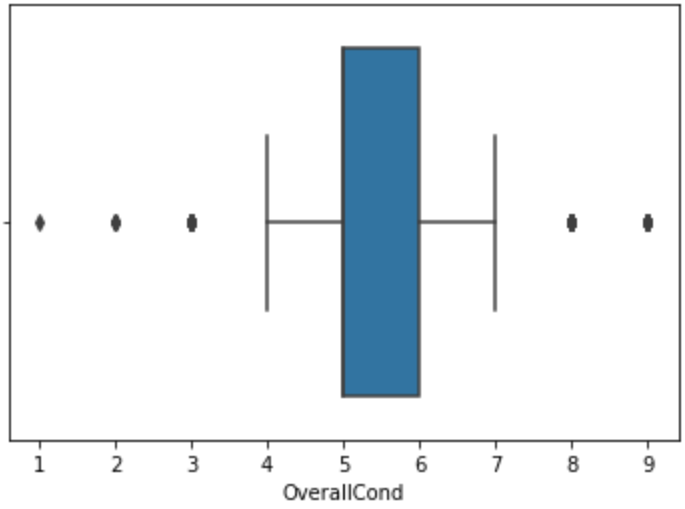


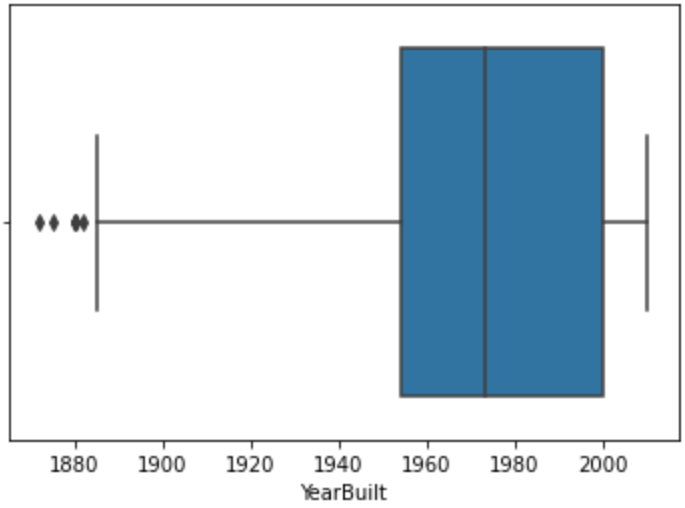


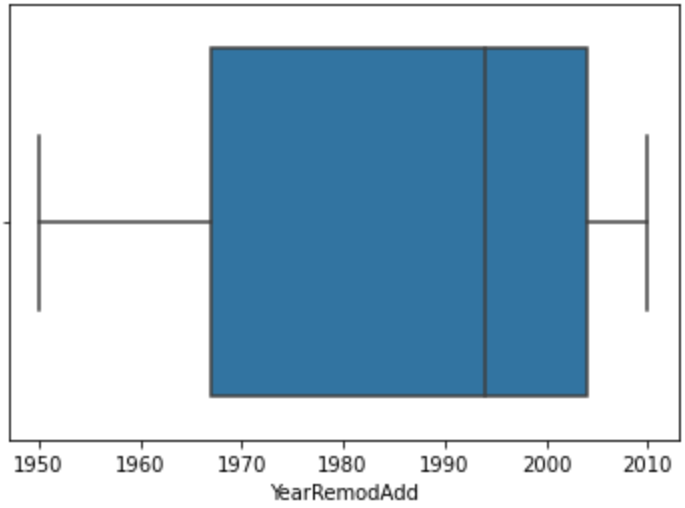

50
100
150
200
250
300
LotFrontage

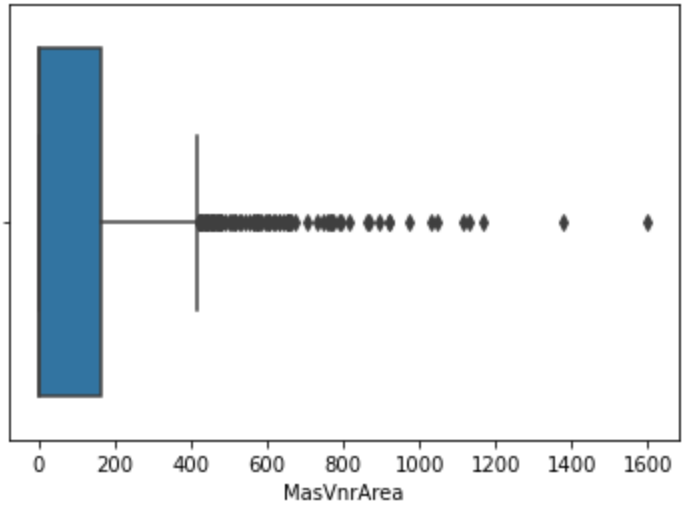

0
50000
100000
150000
200000
LotArea

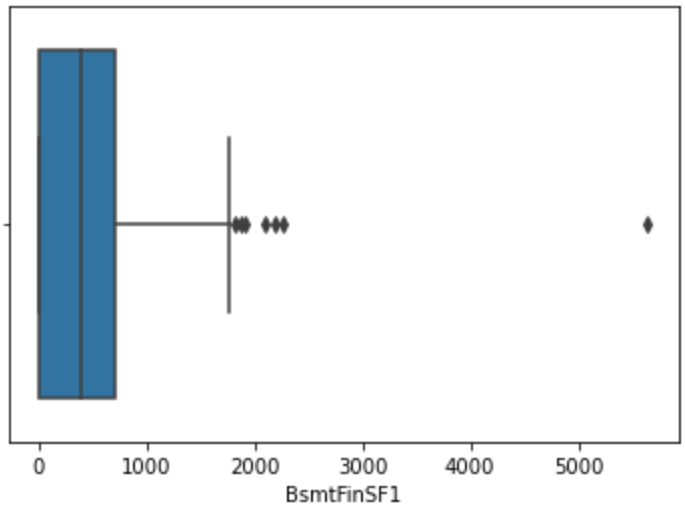

2
4
6
8
10
OverallQual

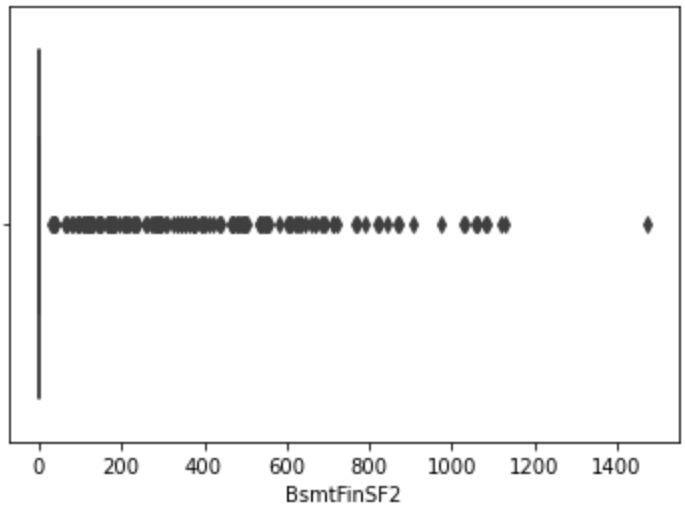

1
2
3
4
5
6
7
8
9
OverallCond

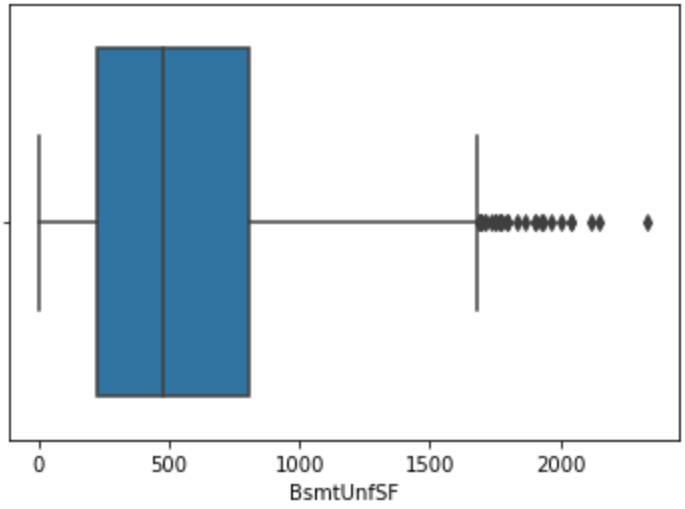

1880
1900
1920
1940
1960
1980
2000
YearBuilt

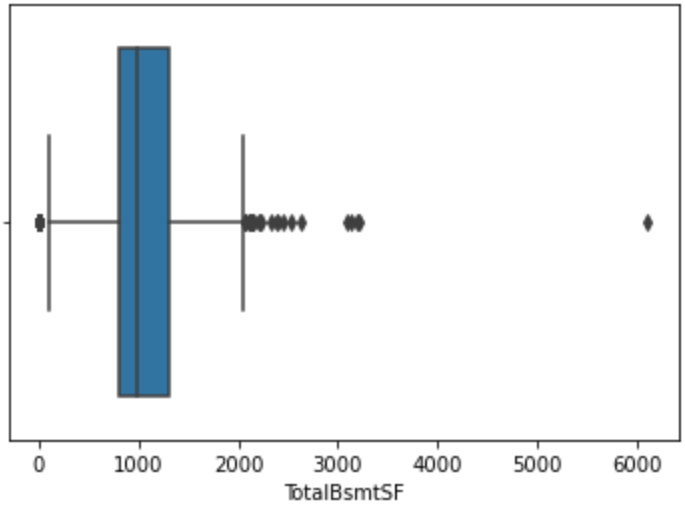

1950
1960
1970
1980
1990
2000
2010
YearRemodAdd

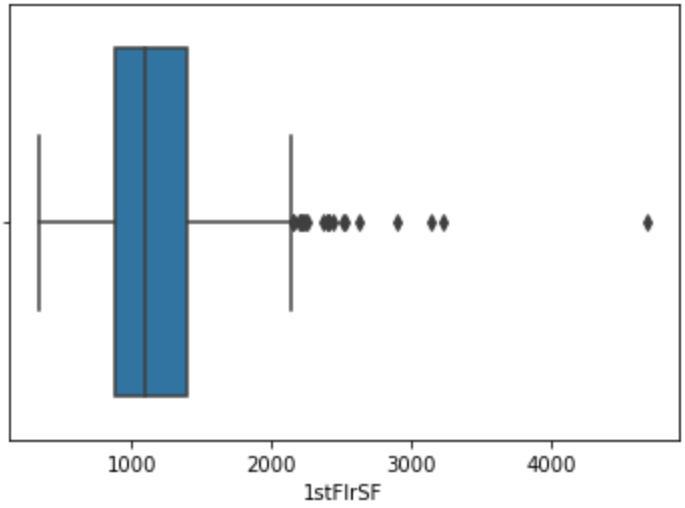

0
200
400
600
800
1000
1200
1400
1600
MasVnrArea

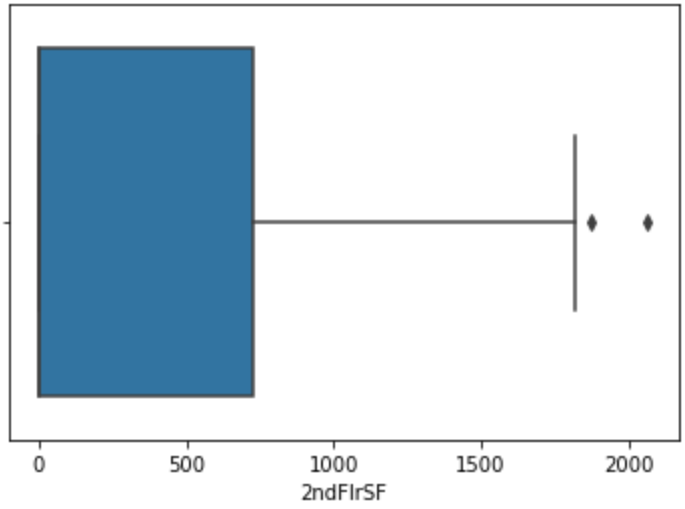

0
1000
2000
3000
4000
5000
BsmtFinSF1

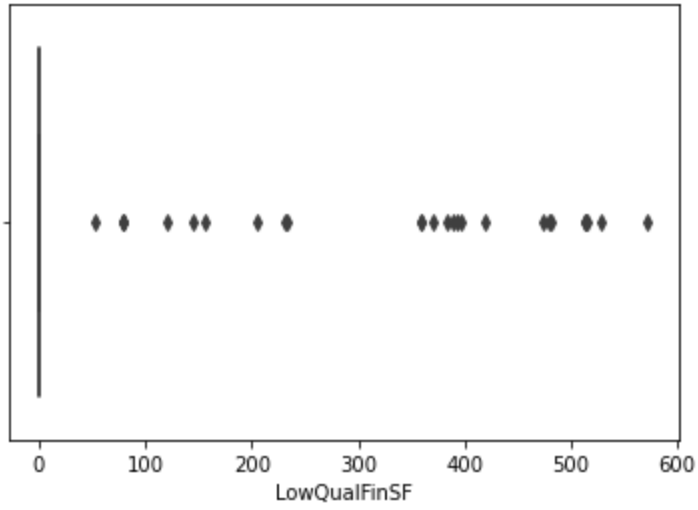

0
200
400
600
800
1000
1200
1400
BsmtFinSF2

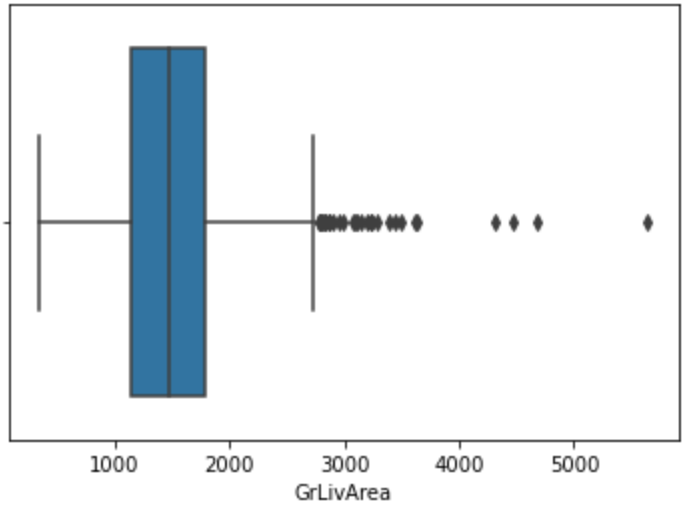

0
500
1000
1500
2000
BsmtUnfSF

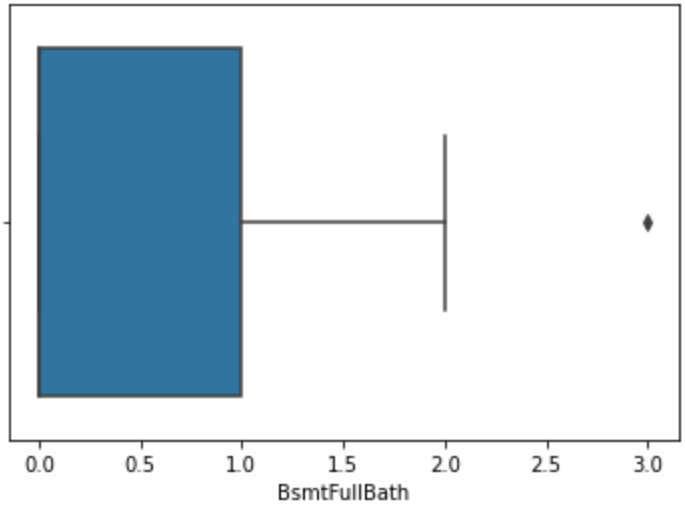

0
1000
2000
3000
4000
5000
6000
TotalBsmtSF

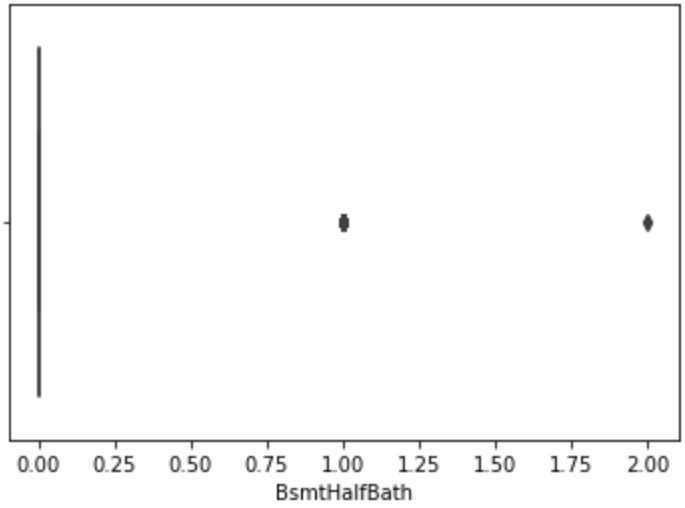

1000
2000
3000
4000
1stFlrSF

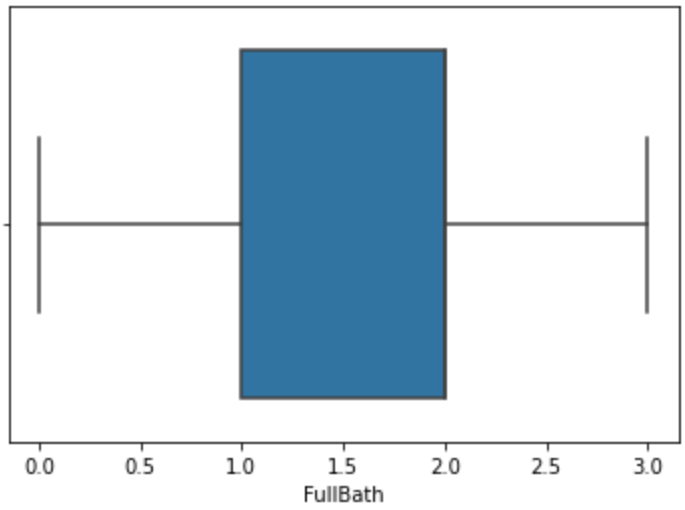

0
500
1000
1500
2000
2ndFlrSF

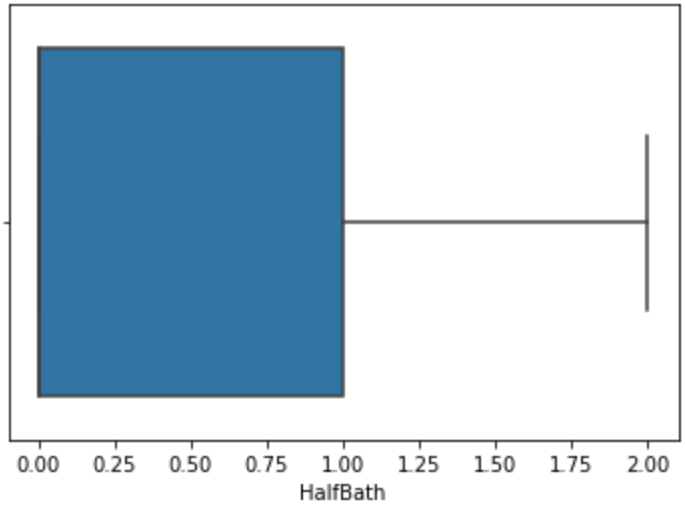

0
100
200
300
400
500
600
LowQualFinSF

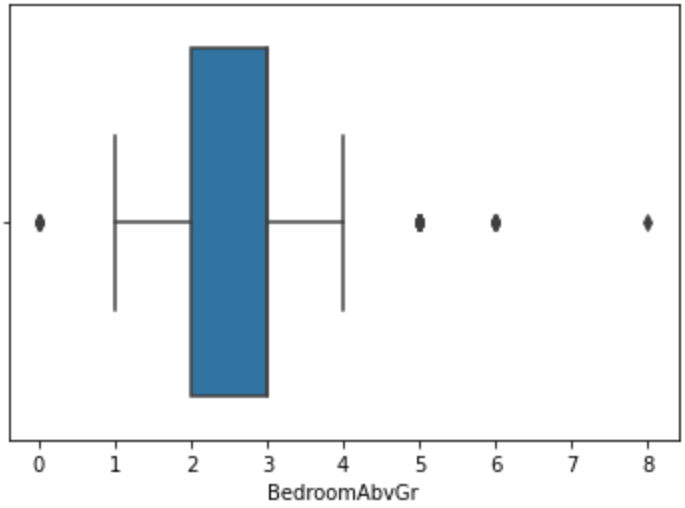

1000
2000
3000
4000
5000
GrLivArea

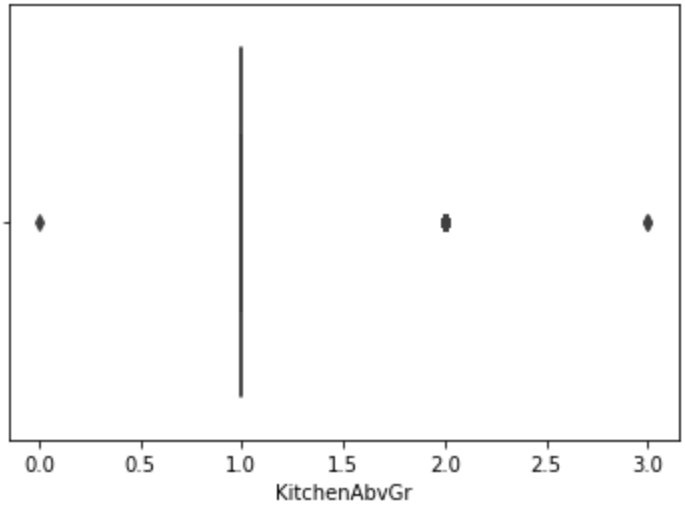

0.0
0.5
1.0
1.5
2.0
2.5
3.0
BsmtFullBath

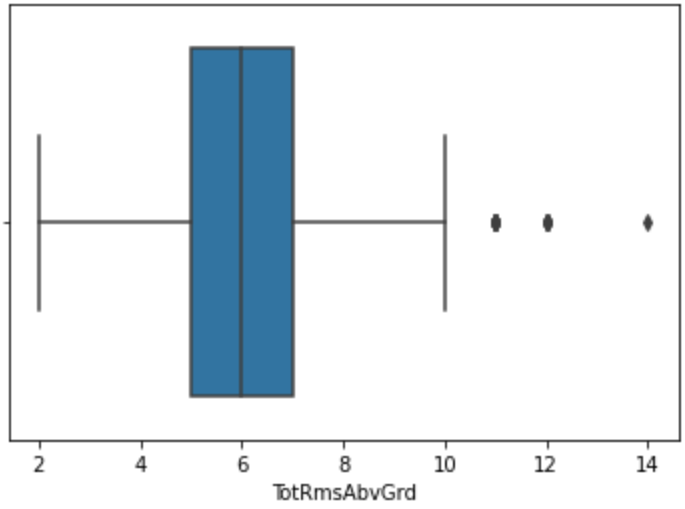

0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
2.00
BsmtHalfBath

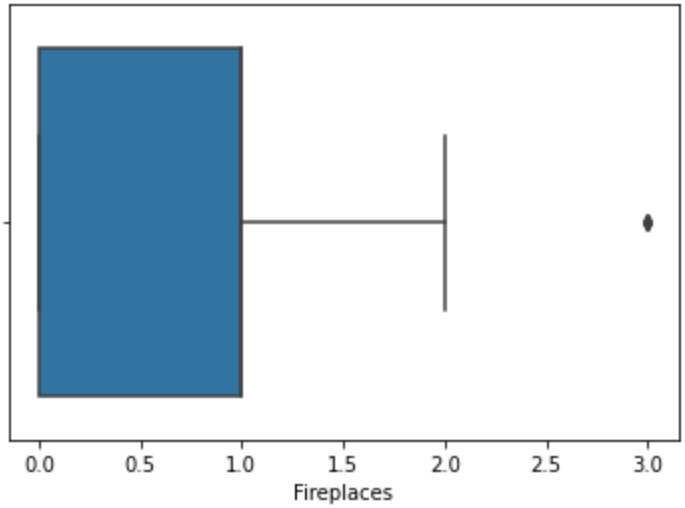

0.0
0.5
1.0
1.5
2.0
2.5
3.0
FullBath

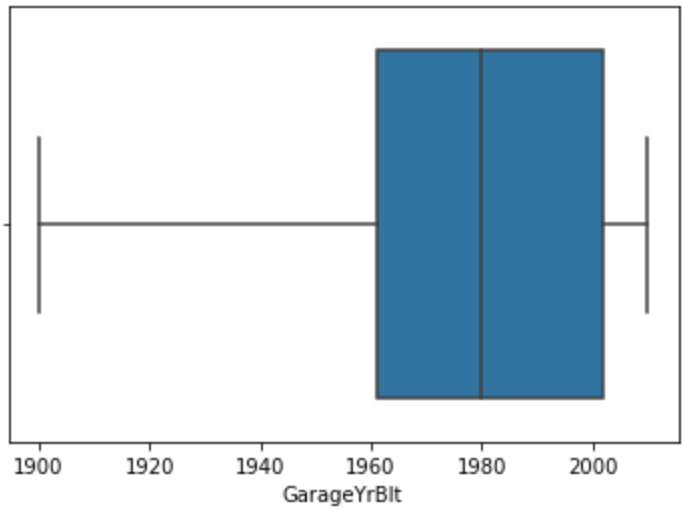

0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
2.00
HalfBath

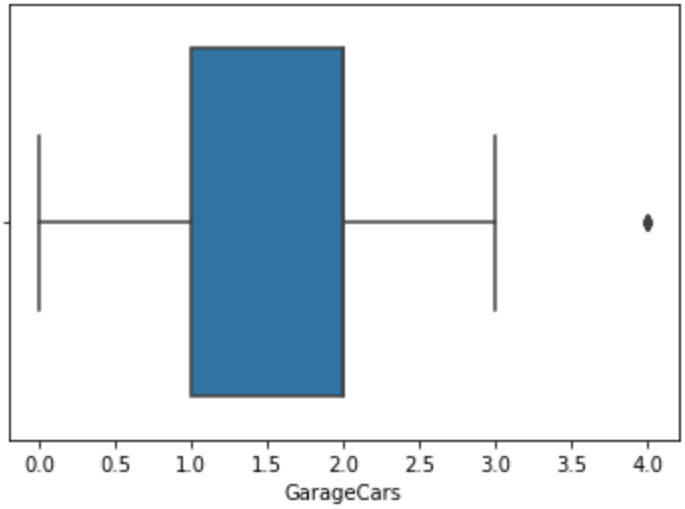

0
1
2
3
4
5
6
7
8
BedroomAbvGr

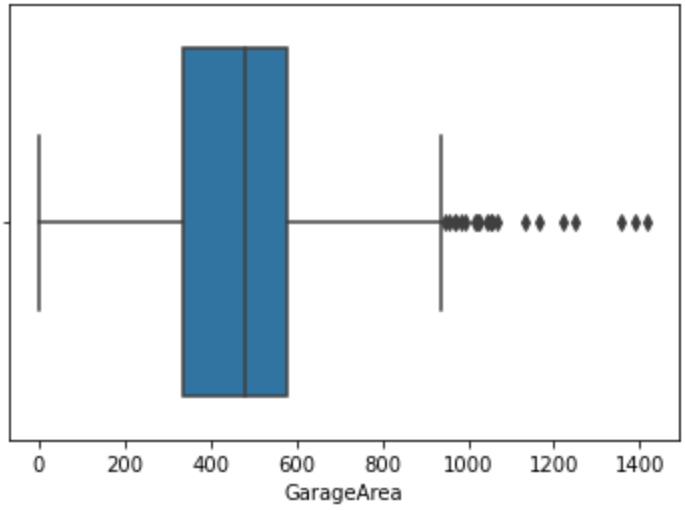

0.0
0.5
1.0
1.5
2.0
2.5
3.0
KitchenAbvGr

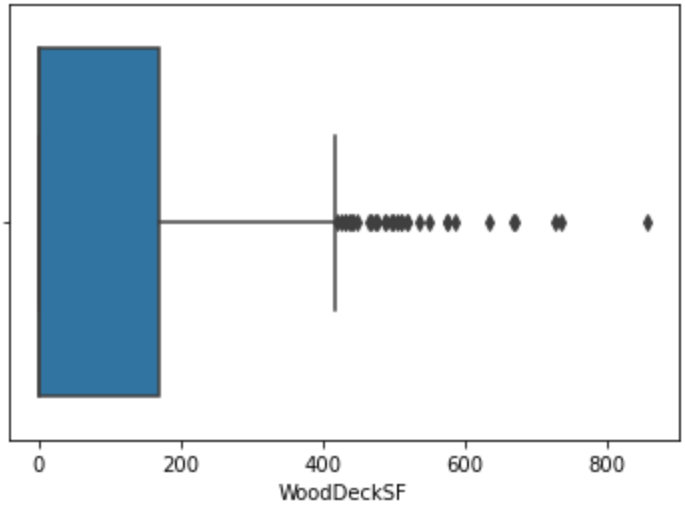

2
4
6
8
10
12
14
TotRmsAbvGrd

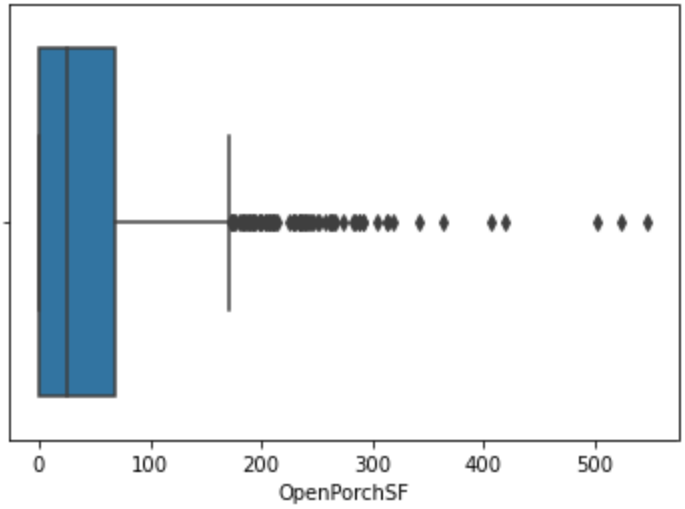

0.0
0.5
1.0
1.5
2.0
2.5
3.0
Fireplaces

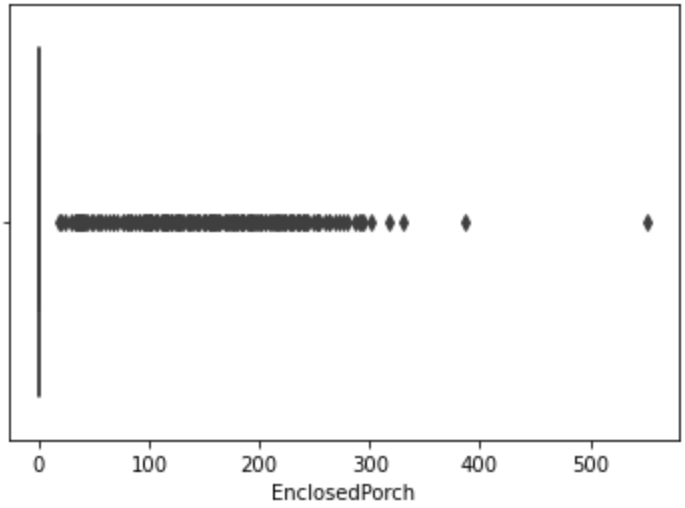

1900
1920
1940
1960
1980
2000
GarageYrBlt

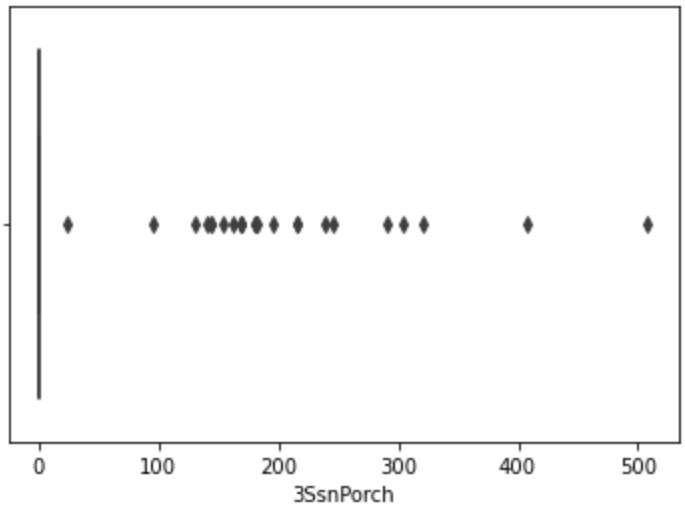

0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
GarageCars

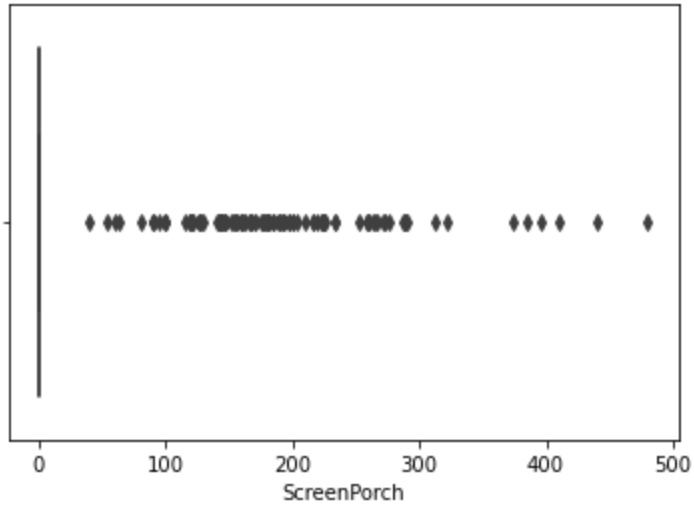

0
200
400
600
800
1000
1200
1400
GarageArea

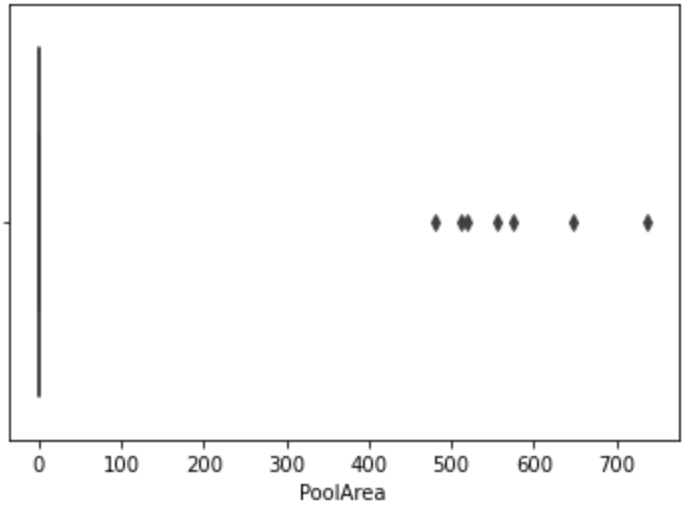

0
200
400
600
800
WoodDeckSF

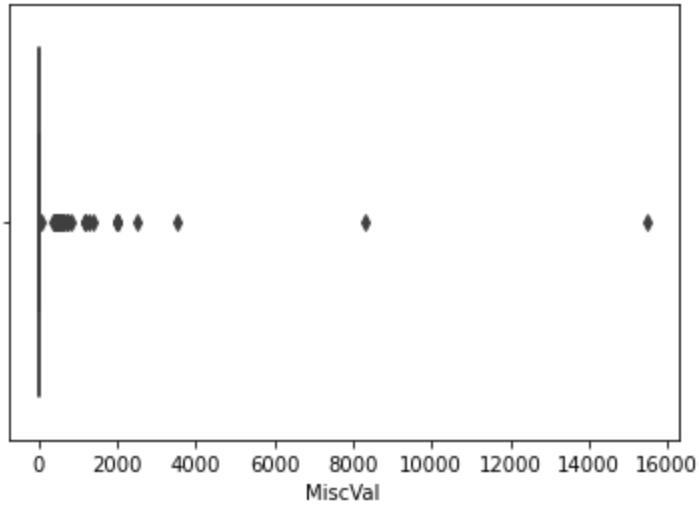

0
100
200
300
400
500
OpenPorchSF

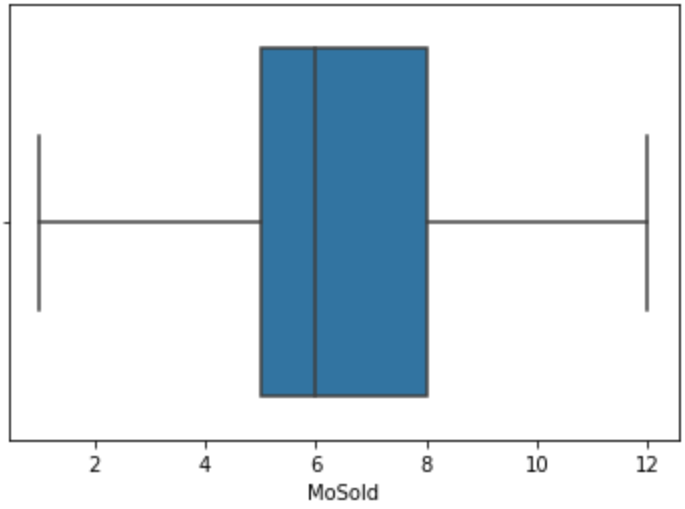

0
100
200
300
400
500
EnclosedPorch

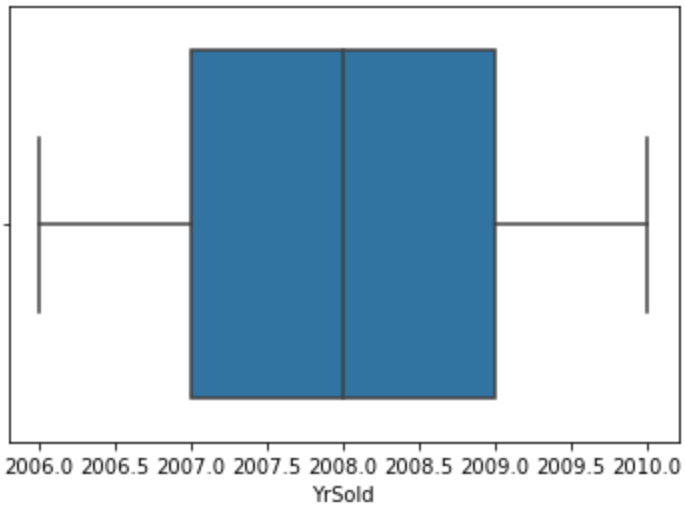

0
100
200
300
400
500
3SsnPorch

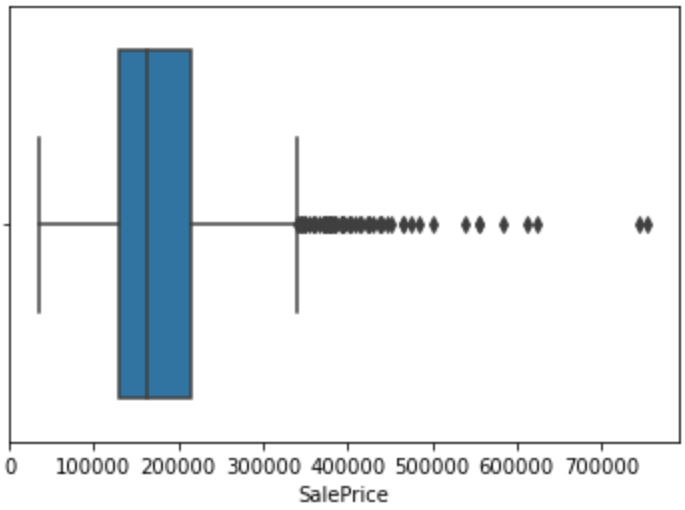

0
100
200
300
400
500
ScreenPorch


0
100
200
300
400
500
600
700
PoolArea


0
2000
4000
6000
8000
10000
12000
14000
16000
MiscVal







Categorical Feature

In [36]:
```
df[categorical_feature].head()
```

Out[36]:

	MSZoning	Street	Alley	LotShape	LandContour	Utilities	LotConfig	LandSlope	Neighborhood	Condition1	Co
0	RL	Pave	NaN	Reg	Lvl	AllPub	Inside	Gtl	CollgCr	Norm	
1	RL	Pave	NaN	Reg	Lvl	AllPub	FR2	Gtl	Veenker	Feedr	
2	RL	Pave	NaN	IR1	Lvl	AllPub	Inside	Gtl	CollgCr	Norm	

	MSZoning	Street	Alley	LotShape	LandContour	Utilities	LotConfig	LandSlope	Neighborhood	Condition1	Co
3	RL	Pave	NaN	IR1	Lvl	AllPub	Corner	Gtl	Crawfor	Norm	
4	RL	Pave	NaN	IR1	Lvl	AllPub	FR2	Gtl	NoRidge	Norm	

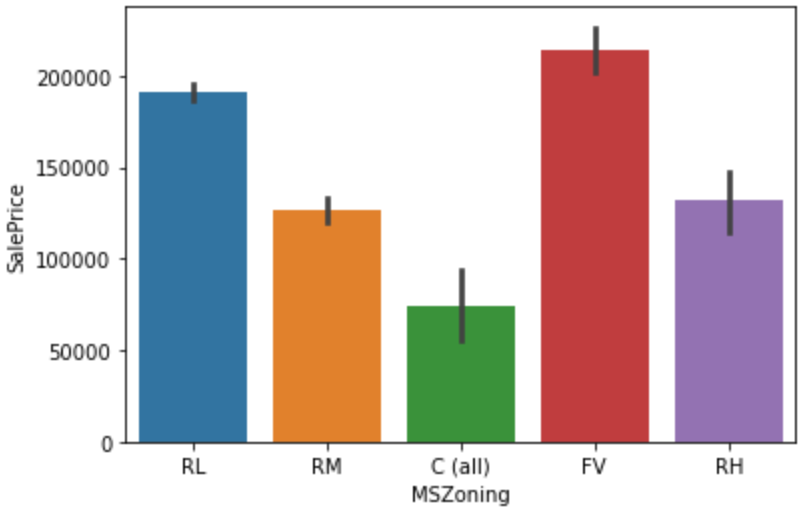
Finding unique categories with respect to each categorical features

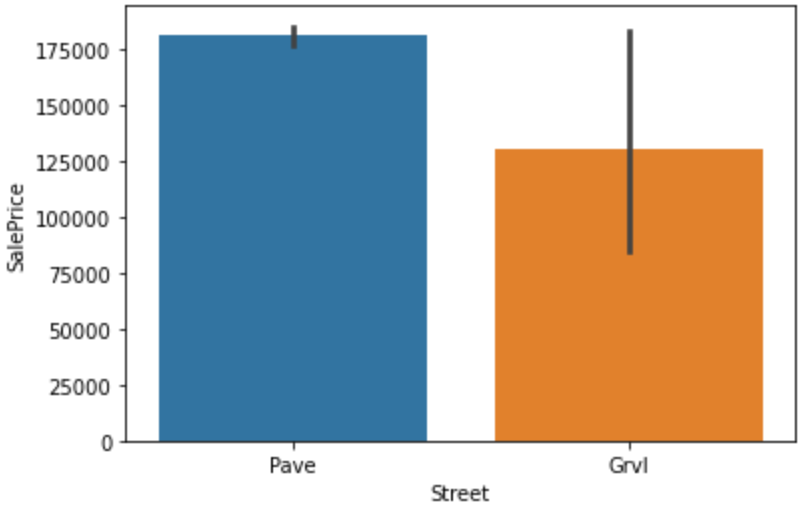
In [50]:
```
for feature in categorical_feature:
    data=df.copy()
    print(feature,df[feature].unique(),len(df[feature].unique()))
```

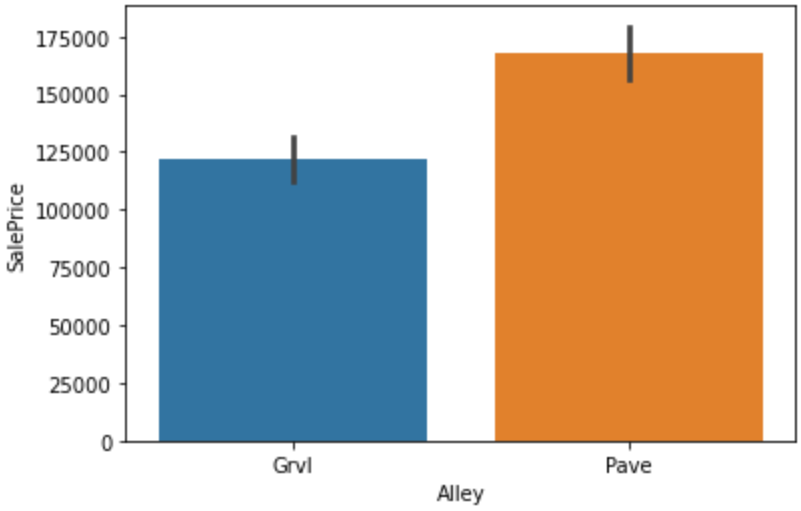
```
MSZoning ['RL' 'RM' 'C (all)' 'FV' 'RH'] 5
Street ['Pave' 'Grvl'] 2
Alley [nan 'Grvl' 'Pave'] 3
LotShape ['Reg' 'IR1' 'IR2' 'IR3'] 4
LandContour ['Lvl' 'Bnk' 'Low' 'HLS'] 4
Utilities ['AllPub' 'NoSeWa'] 2
LotConfig ['Inside' 'FR2' 'Corner' 'CulDSac' 'FR3'] 5
LandSlope ['Gtl' 'Mod' 'Sev'] 3
Neighborhood ['CollgCr' 'Veenker' 'Crawfor' 'NoRidge' 'Mitchel' 'Somerst' 'NWAmes'
 'OldTown' 'BrkSide' 'Sawyer' 'NridgHt' 'NAmes' 'SawyerW' 'IDOTRR'
 'MeadowV' 'Edwards' 'Timber' 'Gilbert' 'StoneBr' 'ClearCr' 'NPkVill'
 'Blmngtn' 'BrDale' 'SWISU' 'Blueste'] 25
Condition1 ['Norm' 'Feedr' 'PosN' 'Artery' 'RRAe' 'RRNn' 'RRAn' 'PosA' 'RRNe'] 9
Condition2 ['Norm' 'Artery' 'RRNn' 'Feedr' 'PosN' 'PosA' 'RRAn' 'RRAe'] 8
BldgType ['1Fam' '2fmCon' 'Duplex' 'TwnhsE' 'Twnhs'] 5
HouseStyle ['2Story' '1Story' '1.5Fin' '1.5Unf' 'SFoyer' 'SLvl' '2.5Unf' '2.5Fin'] 8
RoofStyle ['Gable' 'Hip' 'Gambrel' 'Mansard' 'Flat' 'Shed'] 6
RoofMatl ['CompShg' 'WdShngl' 'Metal' 'WdShake' 'Membran' 'Tar&Grv' 'Roll'
 'ClyTile'] 8
Exterior1st ['VinylSd' 'MetalSd' 'Wd Sdng' 'HdBoard' 'BrkFace' 'WdShing' 'CemntBd'
 'Plywood' 'AsbShng' 'Stucco' 'BrkComm' 'AsphShn' 'Stone' 'ImStucc'
 'CBlock'] 15
Exterior2nd ['VinylSd' 'MetalSd' 'Wd Shng' 'HdBoard' 'Plywood' 'Wd Sdng' 'CmentBd'
 'BrkFace' 'Stucco' 'AsbShng' 'Brk Cmn' 'ImStucc' 'AsphShn' 'Stone'
 'Other' 'CBlock'] 16
MasVnrType ['BrkFace' 'None' 'Stone' 'BrkCmn' nan] 5
ExterQual ['Gd' 'TA' 'Ex' 'Fa'] 4
ExterCond ['TA' 'Gd' 'Fa' 'Po' 'Ex'] 5
Foundation ['PConc' 'CBlock' 'BrkTil' 'Wood' 'Slab' 'Stone'] 6
BsmtQual ['Gd' 'TA' 'Ex' nan 'Fa'] 5
BsmtCond ['TA' 'Gd' nan 'Fa' 'Po'] 5
BsmtExposure ['No' 'Gd' 'Mn' 'Av' nan] 5
BsmtFinType1 ['GLQ' 'ALQ' 'Unf' 'Rec' 'BLQ' nan 'LwQ'] 7
BsmtFinType2 ['Unf' 'BLQ' nan 'ALQ' 'Rec' 'LwQ' 'GLQ'] 7
Heating ['GasA' 'GasW' 'Grav' 'Wall' 'OthW' 'Floor'] 6
HeatingQC ['Ex' 'Gd' 'TA' 'Fa' 'Po'] 5
CentralAir ['Y' 'N'] 2
Electrical ['SBrkr' 'FuseF' 'FuseA' 'FuseP' 'Mix' nan] 6
KitchenQual ['Gd' 'TA' 'Ex' 'Fa'] 4
Functional ['Typ' 'Min1' 'Maj1' 'Min2' 'Mod' 'Maj2' 'Sev'] 7
FireplaceQu [nan 'TA' 'Gd' 'Fa' 'Ex' 'Po'] 6
GarageType ['Attchd' 'Detchd' 'BuiltIn' 'CarPort' nan 'Basment' '2Types'] 7
GarageFinish ['RFn' 'Unf' 'Fin' nan] 4
GarageQual ['TA' 'Fa' 'Gd' nan 'Ex' 'Po'] 6
GarageCond ['TA' 'Fa' nan 'Gd' 'Po' 'Ex'] 6
PavedDrive ['Y' 'N' 'P'] 3
PoolQC [nan 'Ex' 'Fa' 'Gd'] 4
Fence [nan 'MnPrv' 'GdWo' 'GdPrv' 'MnWw'] 5
MiscFeature [nan 'Shed' 'Gar2' 'Othr' 'TenC'] 5
SaleType ['WD' 'New' 'COD' 'ConLD' 'ConLI' 'CWD' 'ConLw' 'Con' 'Oth'] 9
SaleCondition ['Normal' 'Abnorml' 'Partial' 'AdjLand' 'Alloca' 'Family'] 6
```

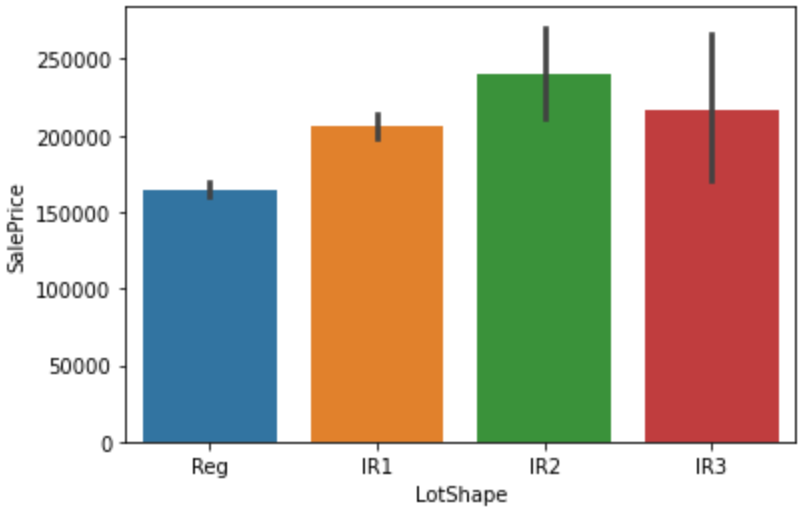
In [51]:

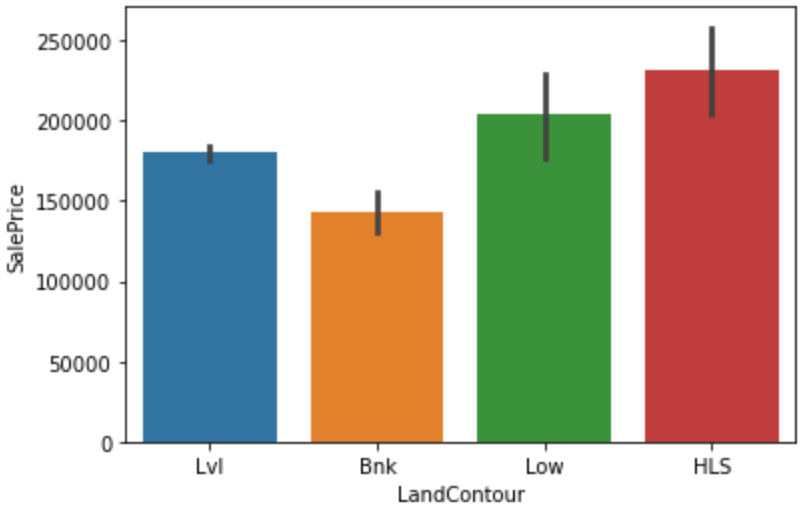
```
for feature in categorical_feature:
    data=df.copy()
    sns.barplot(data=data,y=df['SalePrice'],x=df[feature])
    plt.show()
```

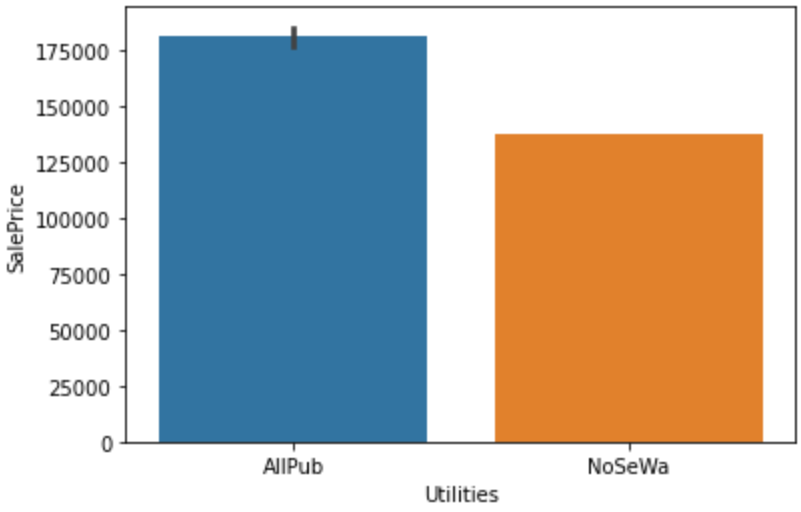


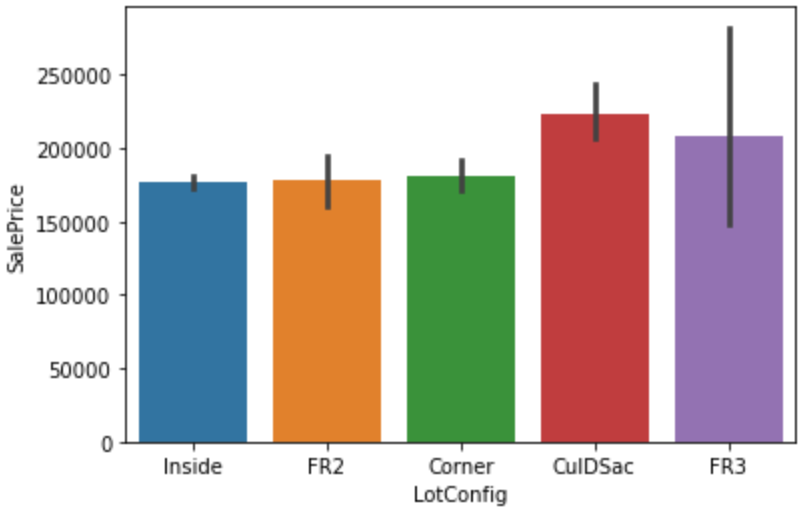


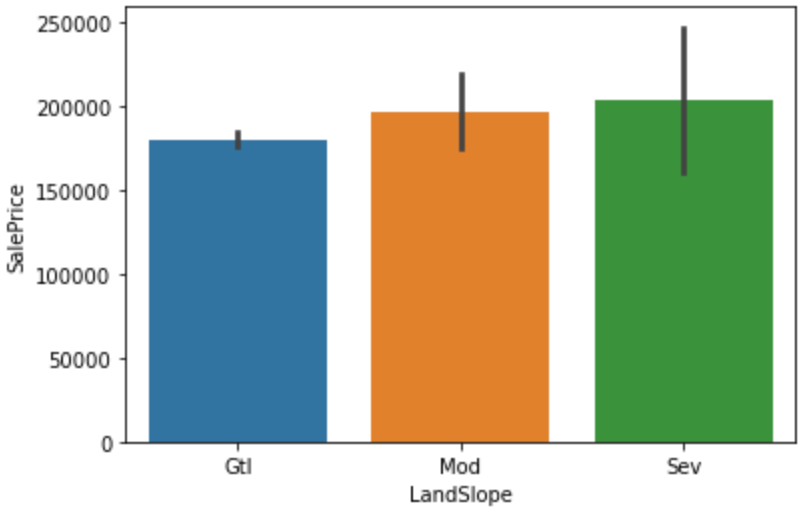


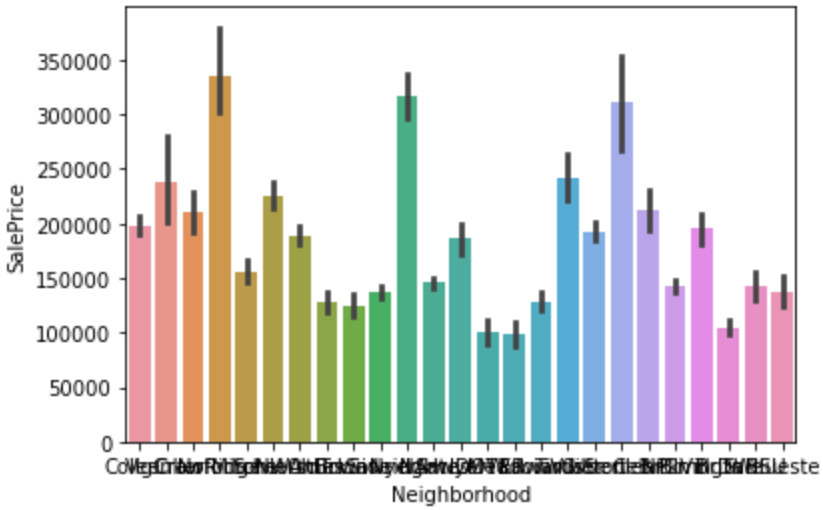

250000
200000
150000
100000
50000
0
SalePrice
Reg
IR1
IR2
IR3
LotShape

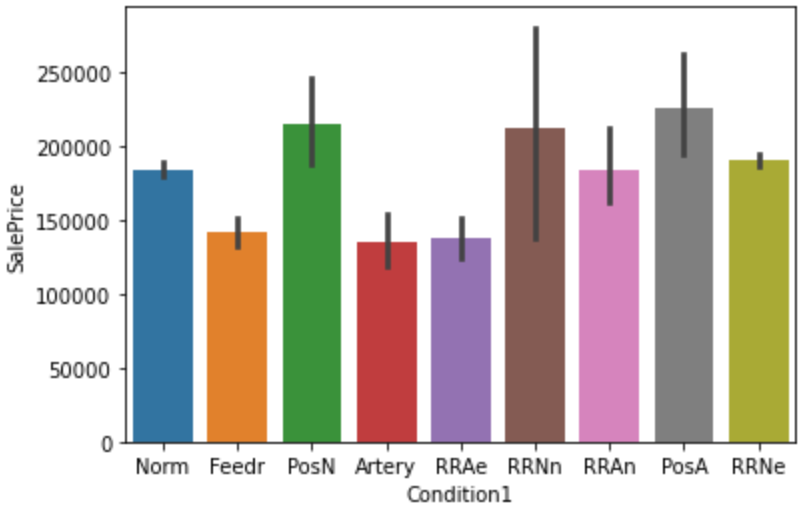

250000
200000
150000
100000
50000
0
SalePrice
Lvl
Bnk
Low
HLS
LandContour

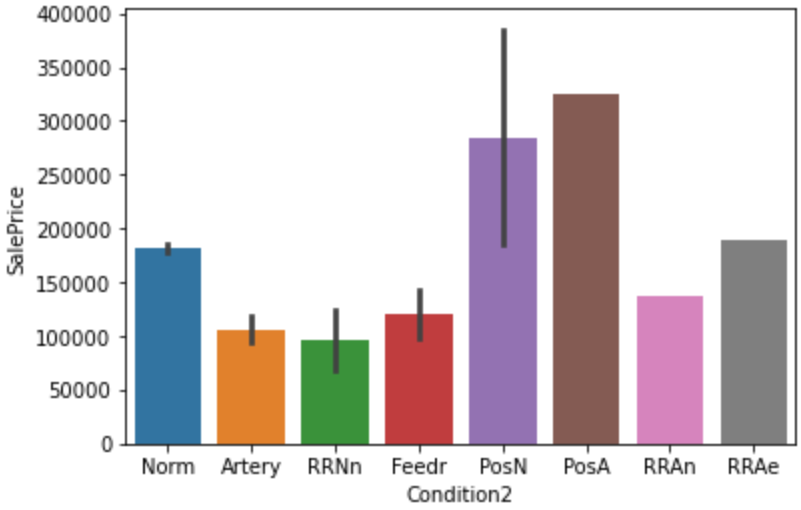

175000
150000
125000
100000
75000
50000
25000
0
SalePrice
AllPub
NoSeWa
Utilities

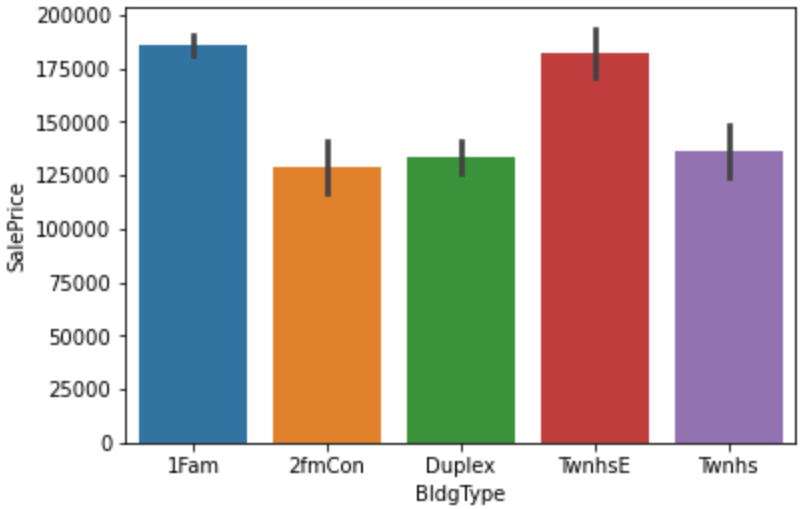

SalePrice
250000
200000
150000
100000
50000
0
Inside
FR2
Corner
CulDSac
FR3
LotConfig

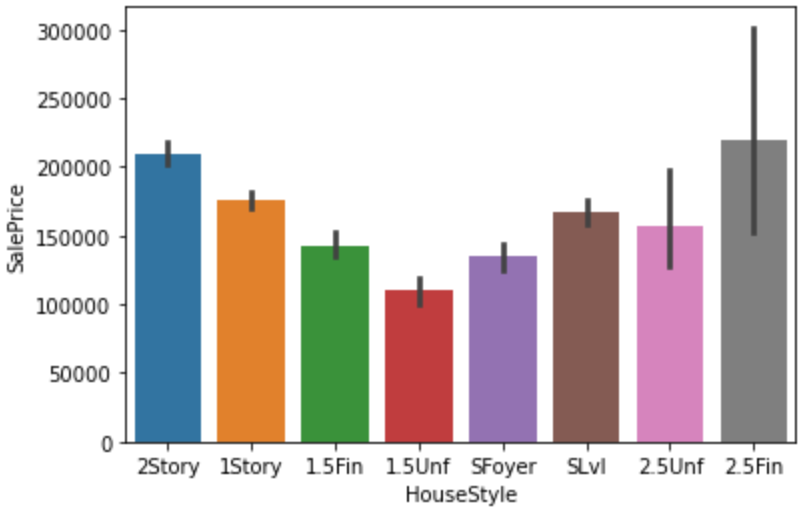

SalePrice
250000
200000
150000
100000
50000
0
Gtl
Mod
Sev
LandSlope

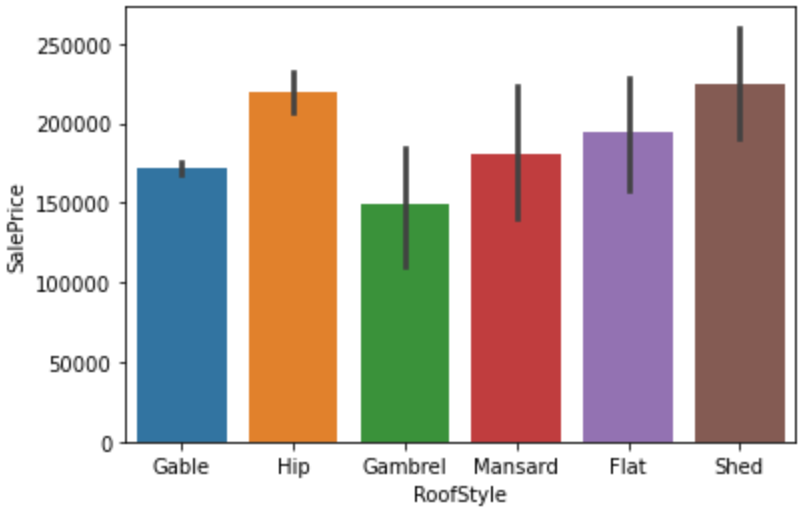

SalePrice
350000
300000
250000
200000
150000
100000
50000
0
Neighborhood

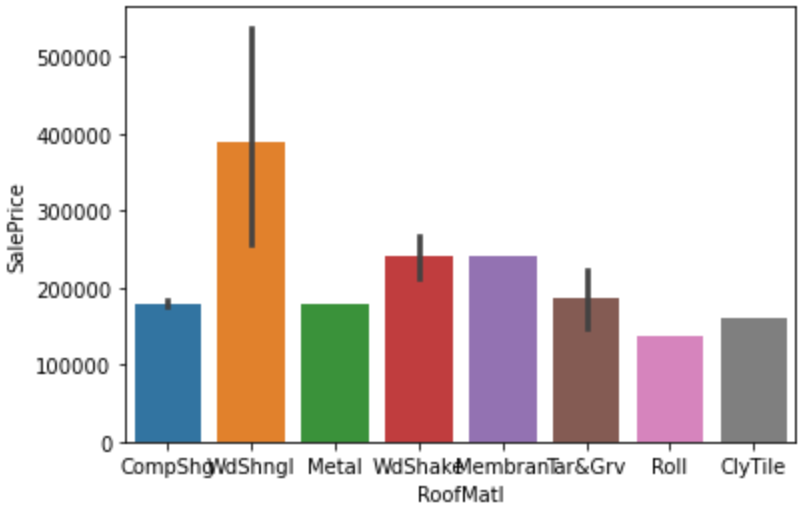

SalePrice
250000
200000
150000
100000
50000
0
Norm
Feedr
PosN
Artery
RRAe
RRNn
RRAn
PosA
RRNe
Condition1

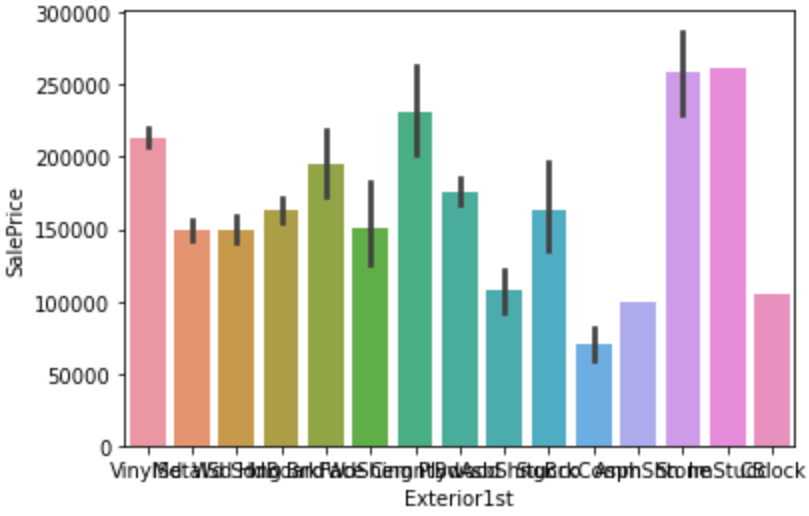

SalePrice
400000
350000
300000
250000
200000
150000
100000
50000
0
Norm
Artery
RRNn
Feedr
PosN
PosA
RRAn
RRAe
Condition2

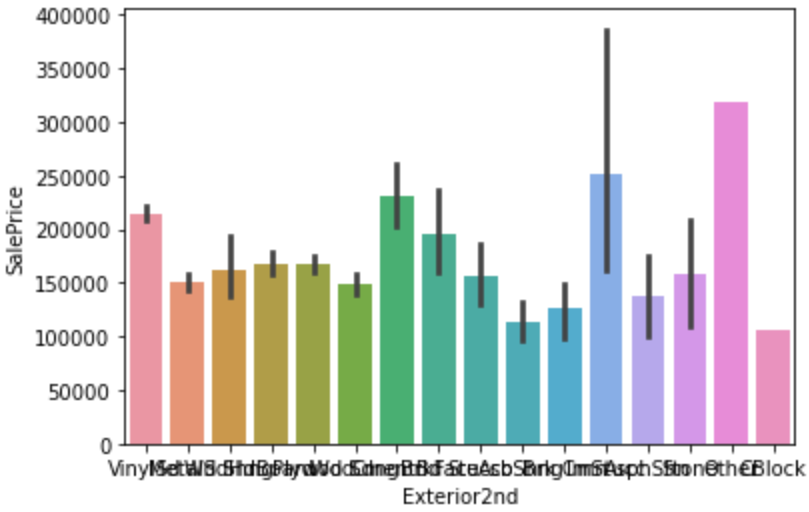

SalePrice
200000
175000
150000
125000
100000
75000
50000
25000
0
1Fam
2fmCon
Duplex
TwnhsE
Twnhs
BldgType

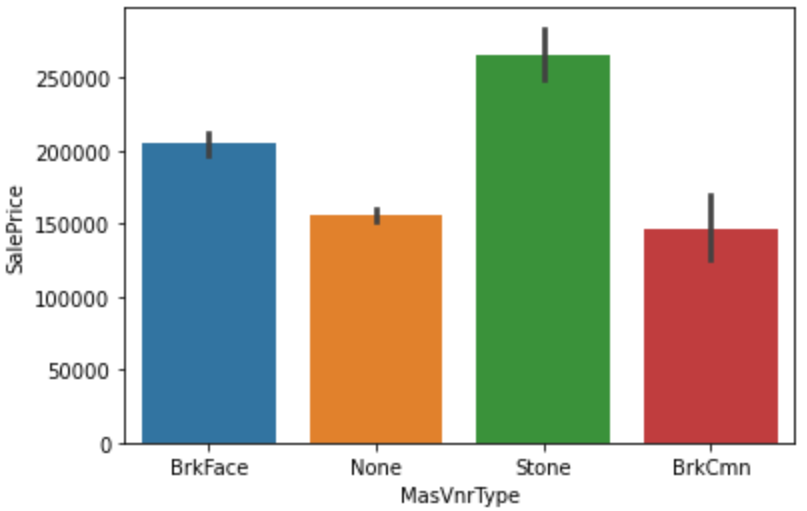

SalePrice
300000
250000
200000
150000
100000
50000
0
2Story
1Story
1.5Fin
1.5Unf
SFoyer
SLvl
2.5Unf
2.5Fin
HouseStyle

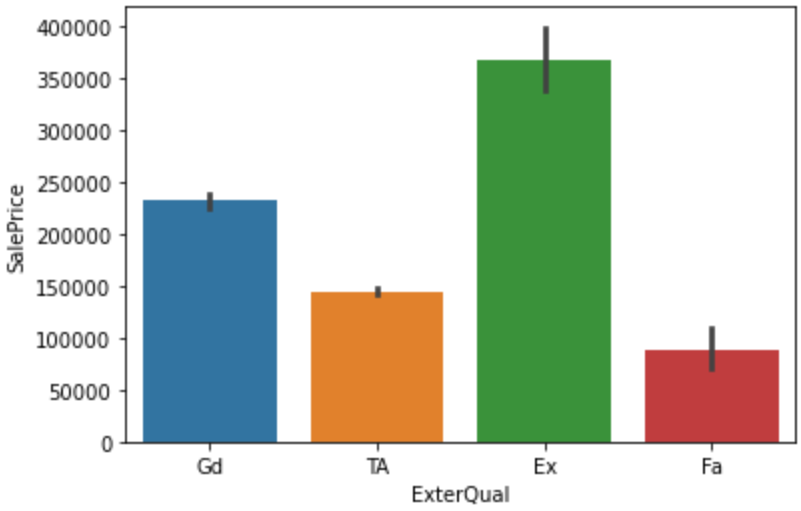

SalePrice
250000
200000
150000
100000
50000
0
Gable
Hip
Gambrel
Mansard
Flat
Shed
RoofStyle

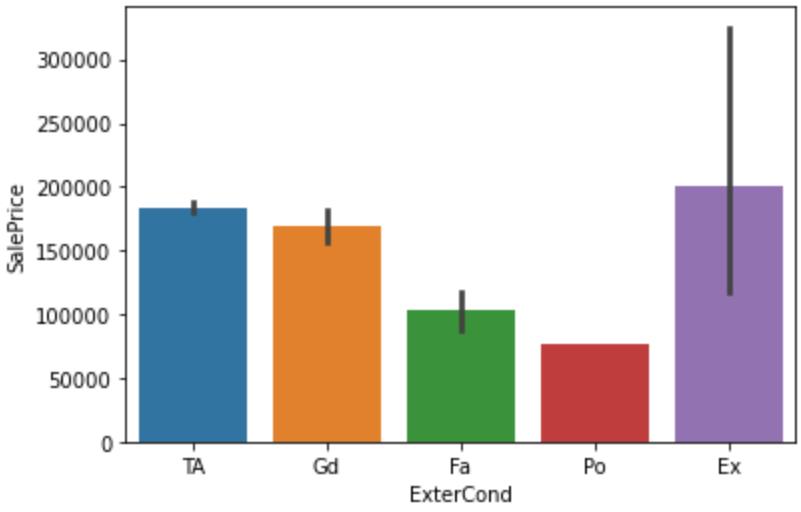

SalePrice
500000
400000
300000
200000
100000
0
CompShg
WdShngl
Metal
WdShake
Membran
Tar&Grv
Roll
ClyTile
RoofMatl

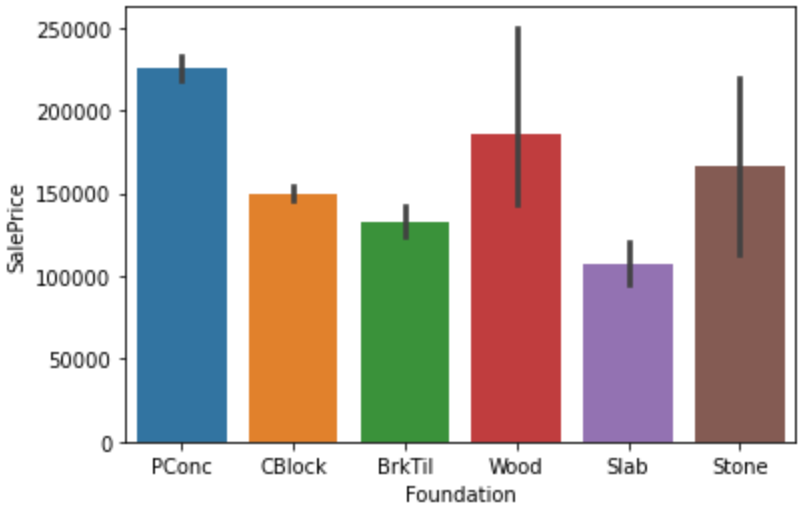

SalePrice
300000
250000
200000
150000
100000
50000
0
Exterior1st

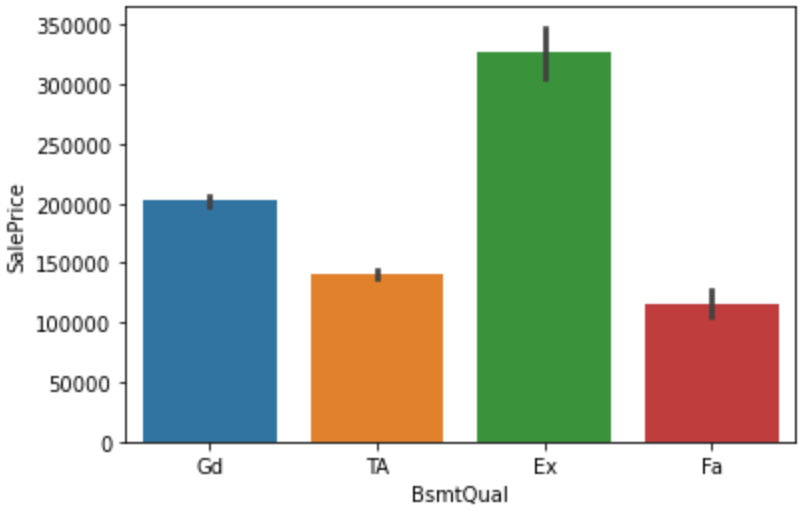

SalePrice
400000
350000
300000
250000
200000
150000
100000
50000
0
Exterior2nd

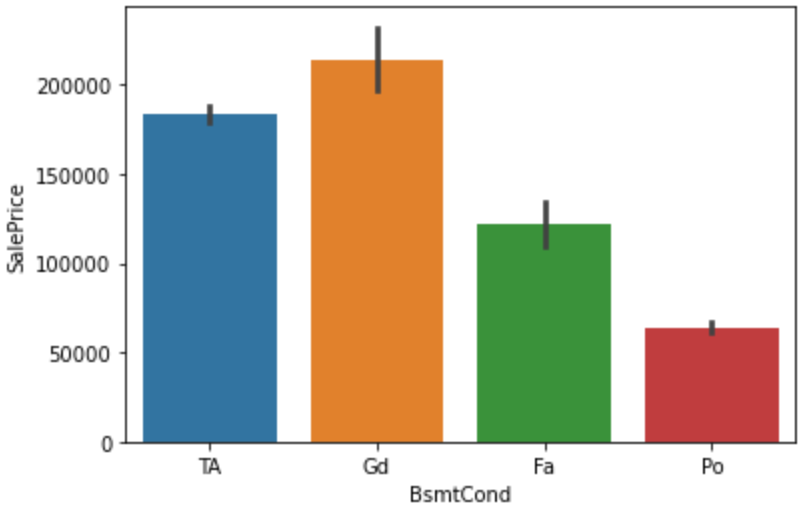

SalePrice
250000
200000
150000
100000
50000
0
BrkFace
None
Stone
BrkCmn
MasVnrType

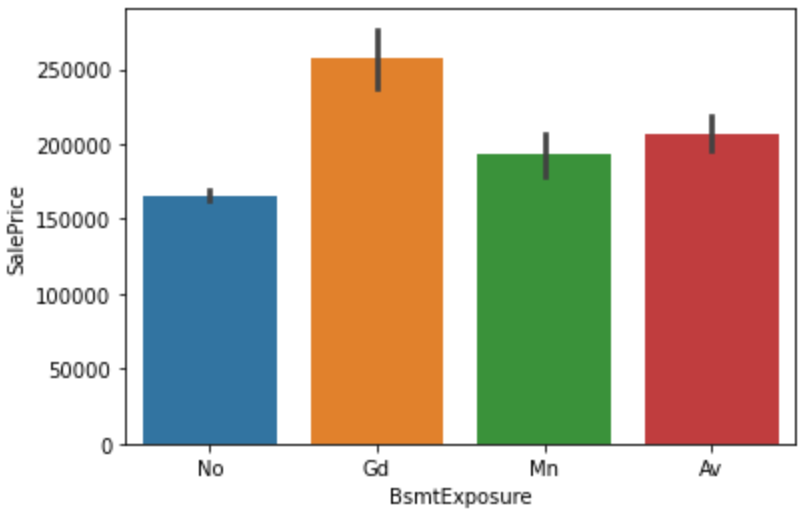

400000
350000
300000
250000
200000
150000
100000
50000
0
SalePrice
Gd
TA
Ex
Fa
ExterQual

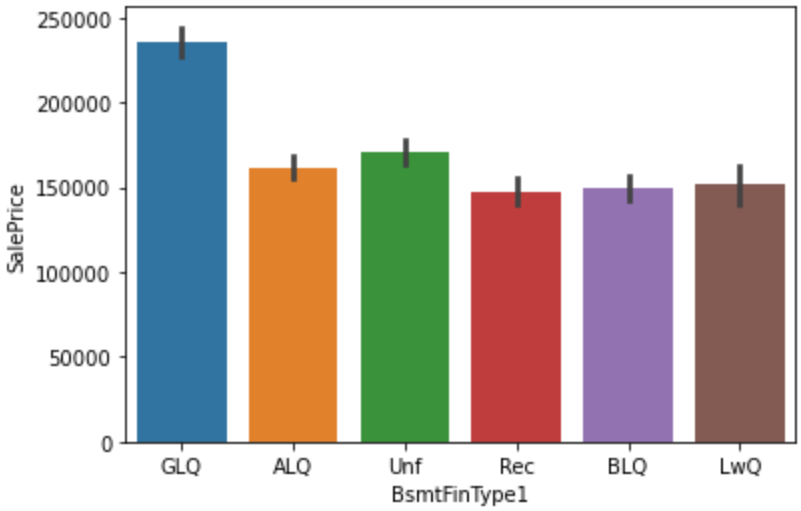

300000
250000
200000
150000
100000
50000
0
SalePrice
TA
Gd
Fa
Po
Ex
ExterCond

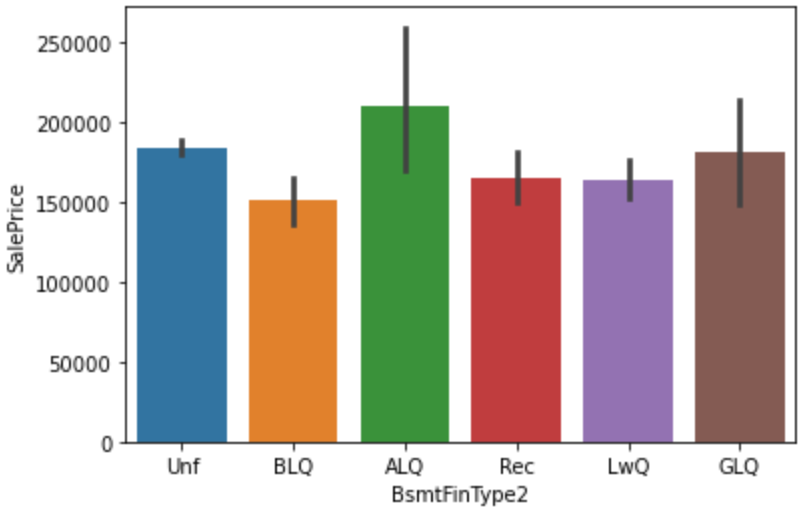

250000
200000
150000
100000
50000
0
SalePrice
PConc
CBlock
BrkTil
Wood
Slab
Stone
Foundation

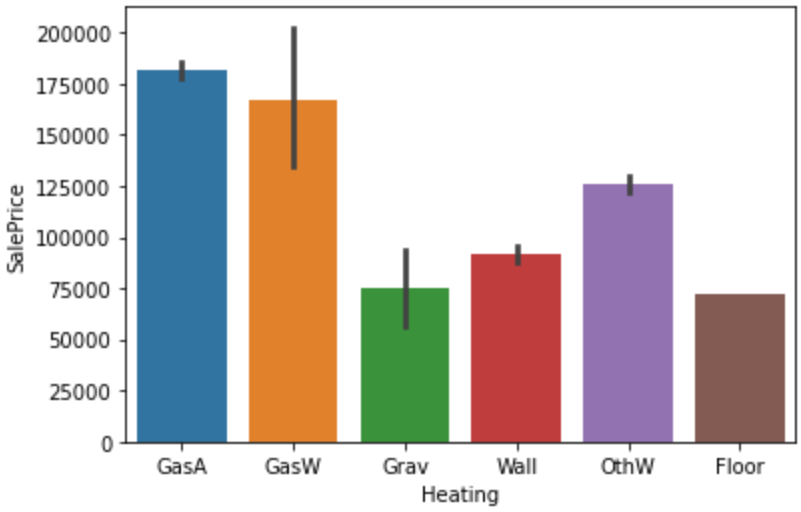

350000
300000
250000
200000
150000
100000
50000
0
SalePrice
Gd
TA
Ex
Fa
BsmtQual

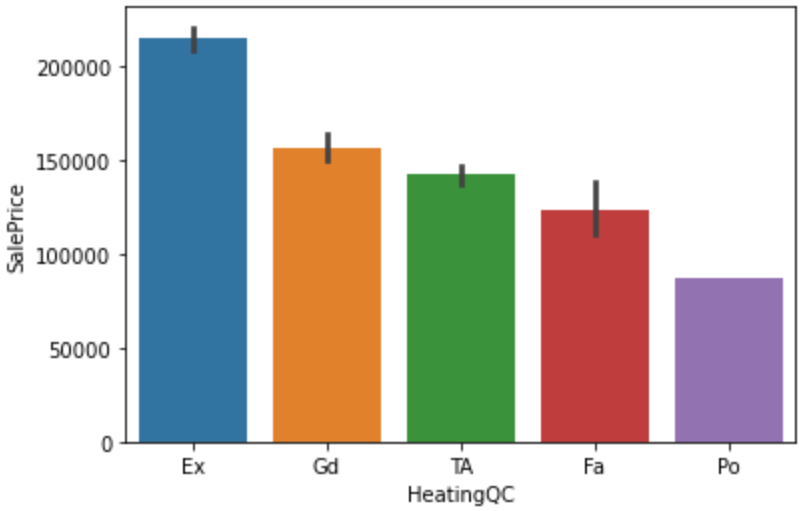

200000
150000
100000
50000
0
SalePrice
TA
Gd
Fa
Po
BsmtCond

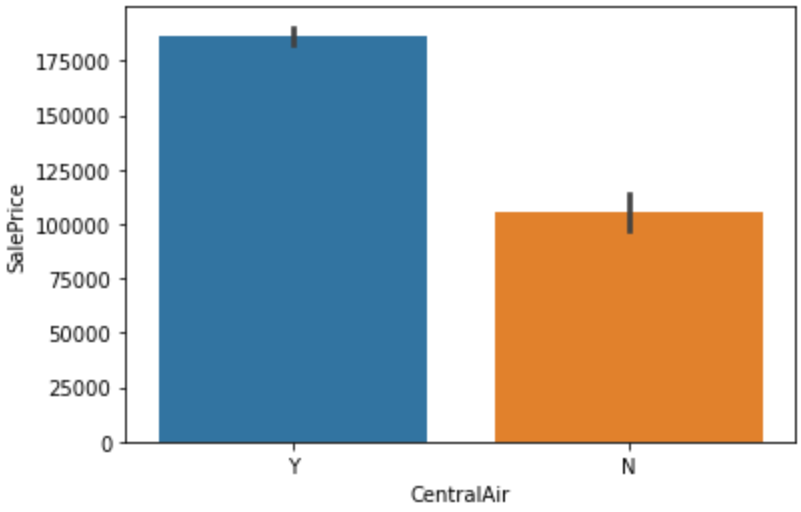

250000
200000
150000
100000
50000
0
SalePrice
No
Gd
Mn
Av
BsmtExposure

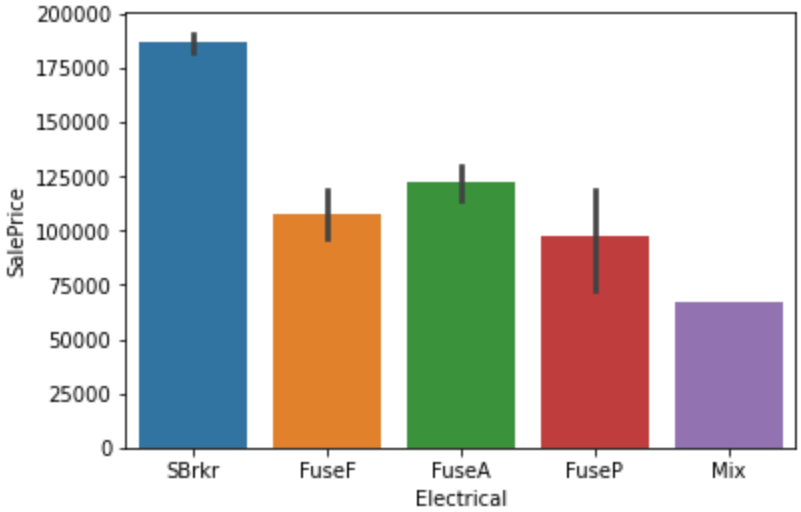

250000
200000
150000
100000
50000
0
SalePrice
GLQ
ALQ
Unf
Rec
BLQ
LwQ
BsmtFinType1

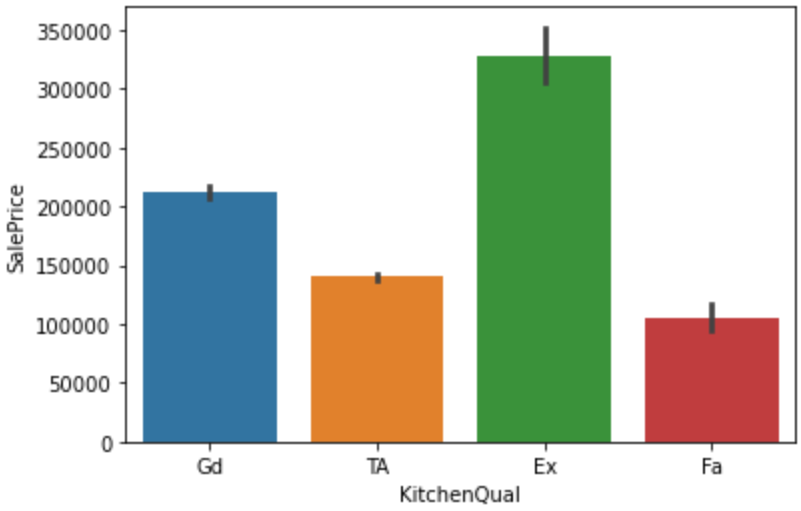

250000
200000
150000
100000
50000
0
SalePrice
Unf
BLQ
ALQ
Rec
LwQ
GLQ
BsmtFinType2

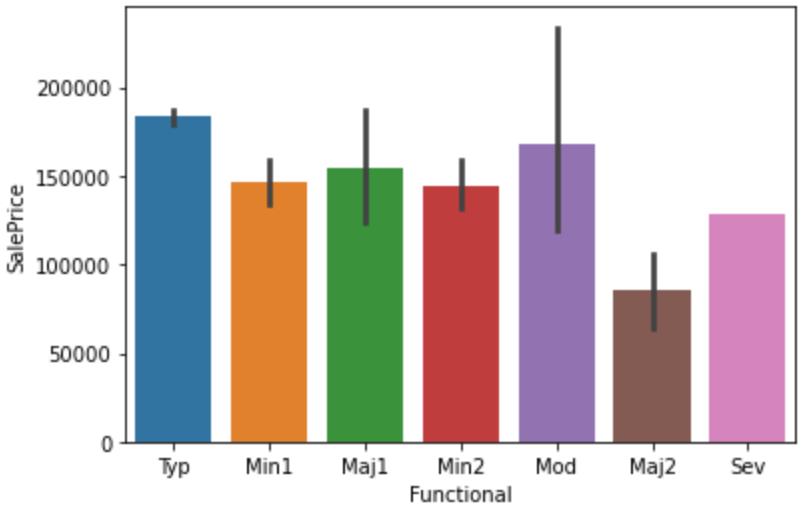

200000
175000
150000
125000
100000
75000
50000
25000
0
SalePrice
GasA
GasW
Grav
Wall
OthW
Floor
Heating

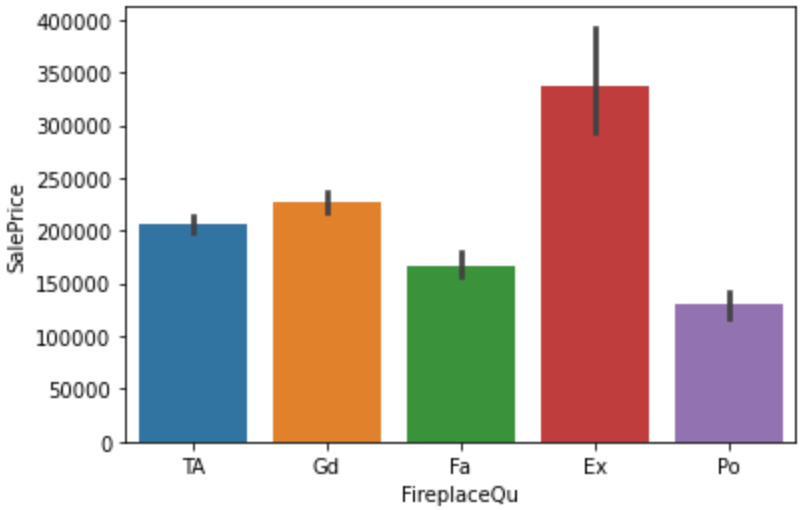

200000
150000
100000
50000
0
SalePrice
Ex
Gd
TA
Fa
Po
HeatingQC

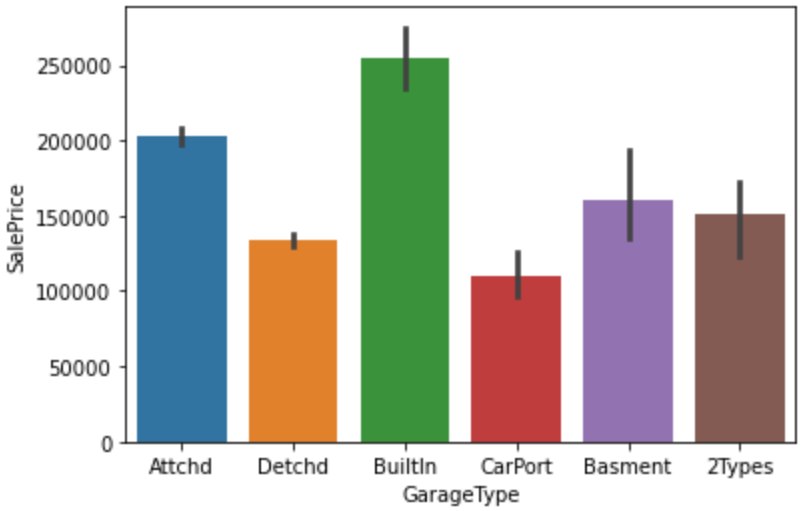

175000
150000
125000
100000
75000
50000
25000
0
SalePrice
Y
N
CentralAir

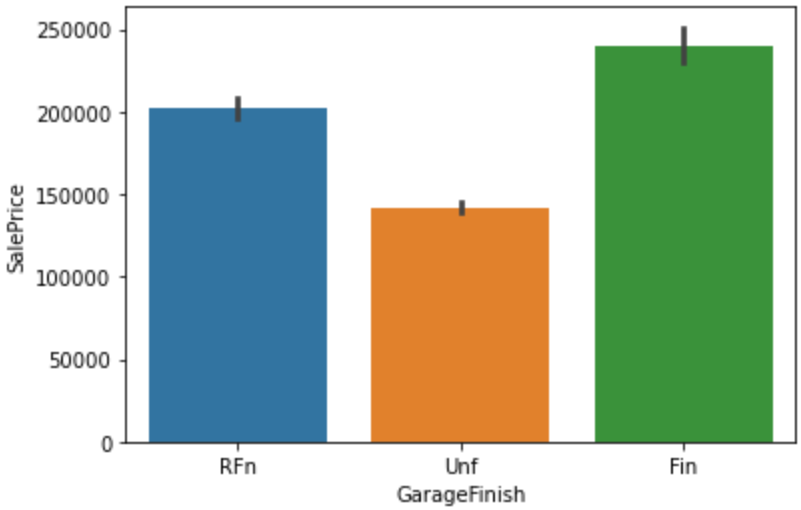

200000
175000
150000
125000
100000
75000
50000
25000
0
SalePrice
SBrkr
FuseF
FuseA
FuseP
Mix
Electrical

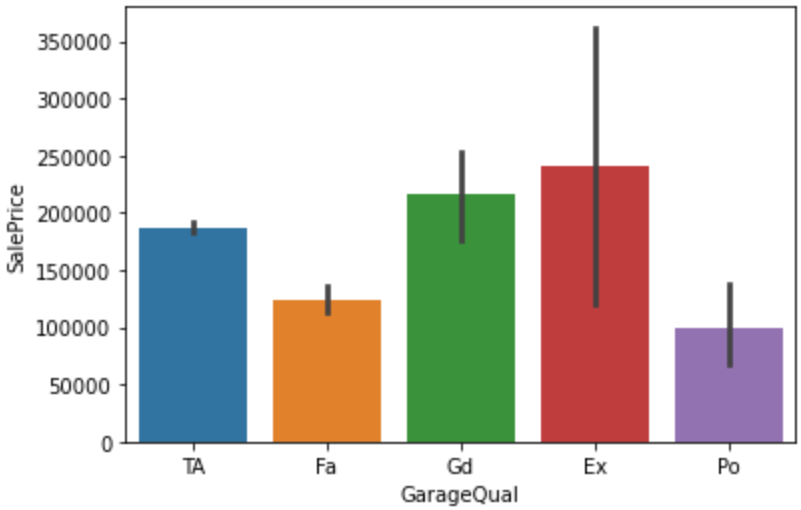

350000
300000
250000
200000
150000
100000
50000
0
SalePrice
Gd
TA
Ex
Fa
KitchenQual

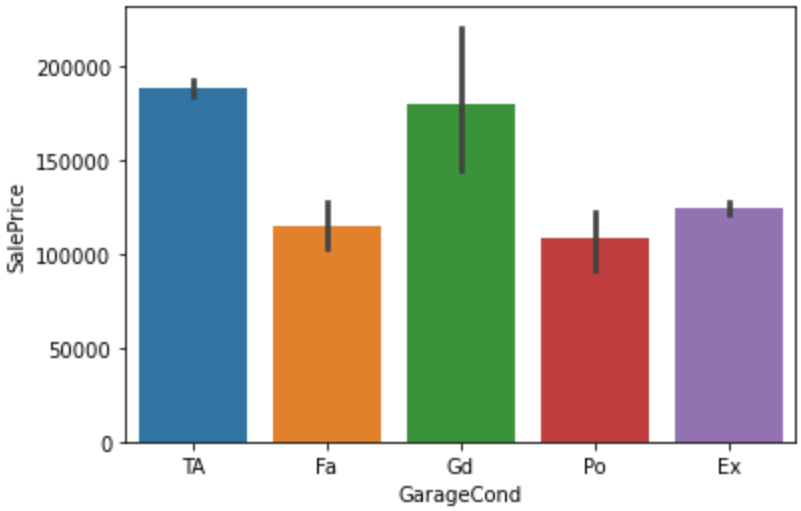

200000
150000
100000
50000
0
SalePrice
Typ
Min1
Maj1
Min2
Mod
Maj2
Sev
Functional

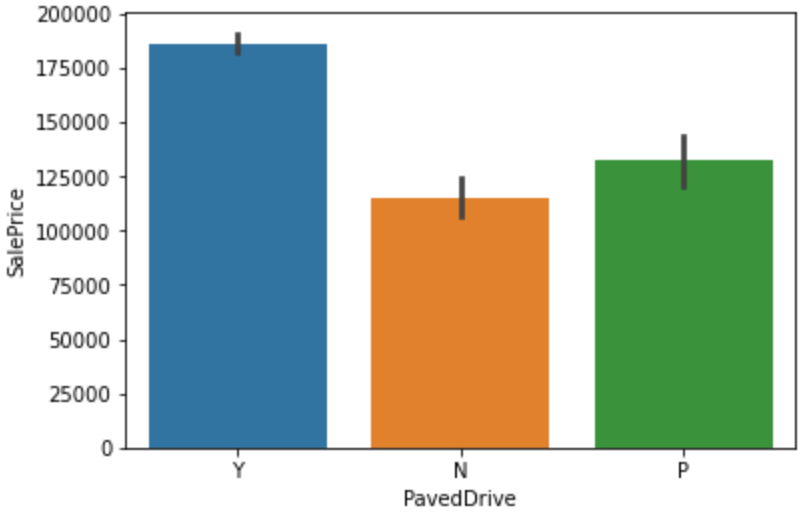

400000
350000
300000
250000
200000
150000
100000
50000
0
SalePrice
TA
Gd
Fa
Ex
Po
FireplaceQu

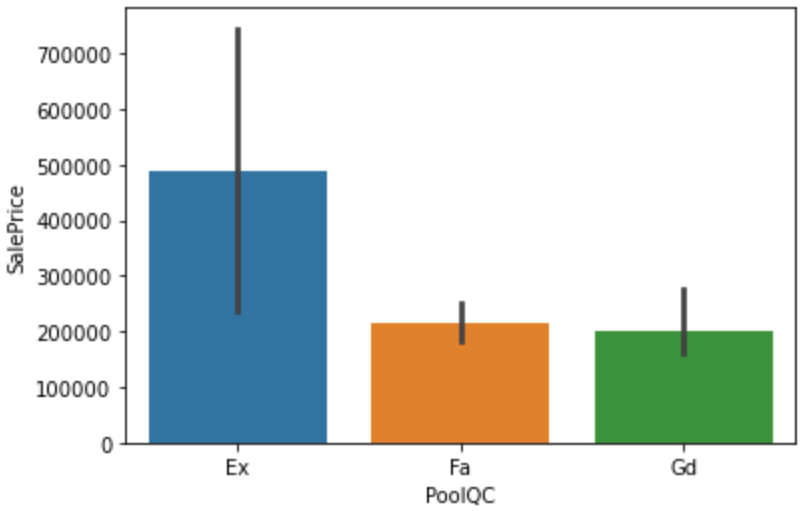

SalePrice
250000
200000
150000
100000
50000
0
Attchd
Detchd
BuiltIn
CarPort
Basment
2Types
GarageType

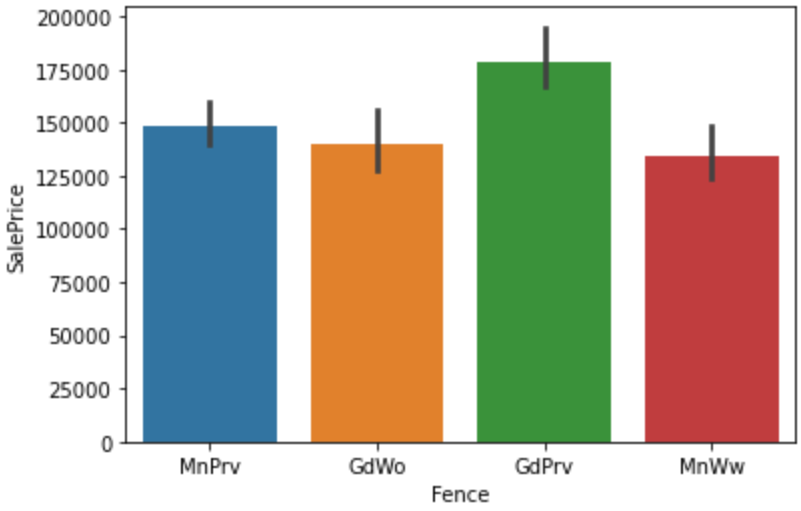

SalePrice
250000
200000
150000
100000
50000
0
RFn
Unf
Fin
GarageFinish

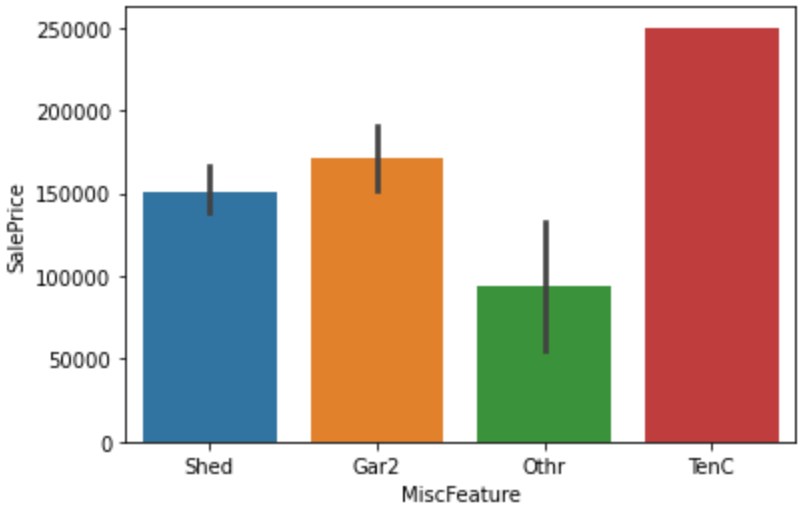

SalePrice
350000
300000
250000
200000
150000
100000
50000
0
TA
Fa
Gd
Ex
Po
GarageQual

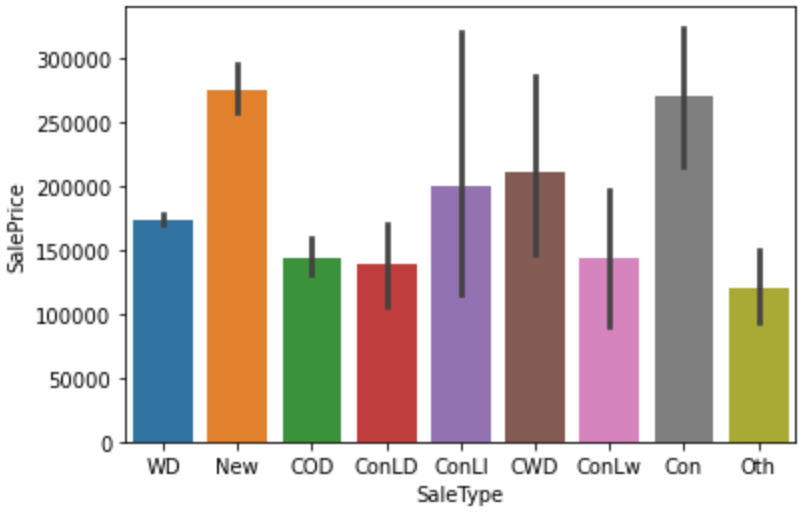

200000
150000
100000
50000
0
SalePrice
TA
Fa
Gd
Po
Ex
GarageCond

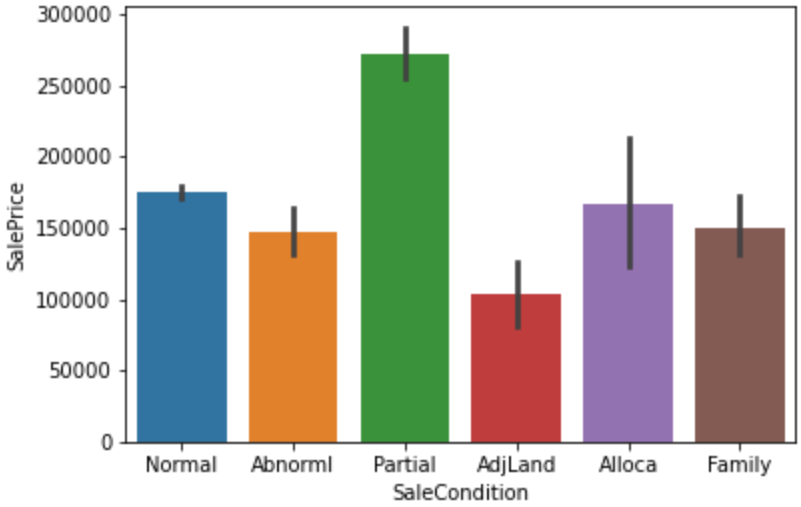

200000
175000
150000
125000
100000
75000
50000
25000
0
SalePrice
Y
N
P
PavedDrive


700000
600000
500000
400000
300000
200000
100000
0
SalePrice
Ex
Fa
Gd
PoolQC


SalePrice
200000
175000
150000
125000
100000
75000
50000
25000
0
MnPrv
GdWo
GdPrv
MnWw
Fence


SalePrice
250000
200000
150000
100000
50000
0
Shed
Gar2
Othr
TenC
MiscFeature


SalePrice
300000
250000
200000
150000
100000
50000
0
WD
New
COD
ConLD
ConLI
CWD
ConLw
Con
Oth
SaleType



In [39]:

```
# for feature in categorical_feature:
#     data=df.copy()
#     data.groupby(feature)['SalePrice'].count().plot.bar()
#     plt.show()
```

In [ ]: