HOUSE PRICE PREDICTION MODEL

In [1]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
import klib
import warnings
warnings.filterwarnings("ignore")


#To show all the columns of datasets
pd.set_option('display.max_columns', None)
```

Working on Train data

In [2]:

```
df1=pd.read_csv("train1.csv")
```

In [3]:

```
df1.head()
```

Out[3]:

	Id	MSSubClass	MSZoning	LotFrontage	LotArea	Street	Alley	LotShape	LandContour	Utilities	LotConfig	LandSlope	Neighborhood	Condition1
0	1	60	RL	65.0	8450	Pave	NaN	Reg	Lvl	AllPub	Inside	Gtl	CollgCr	Norm
1	2	20	RL	80.0	9600	Pave	NaN	Reg	Lvl	AllPub	FR2	Gtl	Veenker	Feedr
2	3	60	RL	68.0	11250	Pave	NaN	IR1	Lvl	AllPub	Inside	Gtl	CollgCr	Norm
3	4	70	RL	60.0	9550	Pave	NaN	IR1	Lvl	AllPub	Corner	Gtl	Crawfor	Norm
4	5	60	RL	84.0	14260	Pave	NaN	IR1	Lvl	AllPub	FR2	Gtl	NoRidge	Norm

In [4]:

```
df1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1460 entries, 0 to 1459
Data columns (total 81 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   Id             1460 non-null   int64
 1   MSSubClass     1460 non-null   int64
 2   MSZoning       1460 non-null   object
 3   LotFrontage    1201 non-null   float64
 4   LotArea        1460 non-null   int64
 5   Street         1460 non-null   object
 6   Alley          91 non-null     object
 7   LotShape       1460 non-null   object
 8   LandContour    1460 non-null   object
 9   Utilities      1460 non-null   object
 10  LotConfig      1460 non-null   object
 11  LandSlope      1460 non-null   object
 12  Neighborhood   1460 non-null   object
 13  Condition1     1460 non-null   object
 14  Condition2     1460 non-null   object
 15  BldgType       1460 non-null   object
 16  HouseStyle     1460 non-null   object
 17  OverallQual    1460 non-null   int64
 18  OverallCond    1460 non-null   int64
 19  YearBuilt      1460 non-null   int64
 20  YearRemodAdd   1460 non-null   int64
 21  RoofStyle      1460 non-null   object
 22  RoofMatl       1460 non-null   object
 23  Exterior1st    1460 non-null   object
 24  Exterior2nd    1460 non-null   object
 25  MasVnrType     1452 non-null   object
 26  MasVnrArea     1452 non-null   float64
 27  ExterQual      1460 non-null   object
 28  ExterCond      1460 non-null   object
 29  Foundation     1460 non-null   object
 30  BsmtQual       1423 non-null   object
 31  BsmtCond       1423 non-null   object
 32  BsmtExposure   1422 non-null   object
 33  BsmtFinType1   1423 non-null   object
 34  BsmtFinSF1     1460 non-null   int64
 35  BsmtFinType2   1422 non-null   object
 36  BsmtFinSF2     1460 non-null   int64
 37  BsmtUnfSF      1460 non-null   int64
 38  TotalBsmtSF    1460 non-null   int64
 39  Heating        1460 non-null   object
 40  HeatingQC      1460 non-null   object
 41  CentralAir     1460 non-null   object
```

```
 42  Electrical     1459 non-null   object
 43  1stFlrSF       1460 non-null   int64
 44  2ndFlrSF       1460 non-null   int64
 45  LowQualFinSF   1460 non-null   int64
 46  GrLivArea      1460 non-null   int64
 47  BsmtFullBath   1460 non-null   int64
 48  BsmtHalfBath   1460 non-null   int64
 49  FullBath       1460 non-null   int64
 50  HalfBath       1460 non-null   int64
 51  BedroomAbvGr   1460 non-null   int64
 52  KitchenAbvGr   1460 non-null   int64
 53  KitchenQual    1460 non-null   object
 54  TotRmsAbvGrd   1460 non-null   int64
 55  Functional     1460 non-null   object
 56  Fireplaces     1460 non-null   int64
 57  FireplaceQu    770 non-null    object
 58  GarageType     1379 non-null   object
 59  GarageYrBlt    1379 non-null   float64
 60  GarageFinish   1379 non-null   object
 61  GarageCars     1460 non-null   int64
 62  GarageArea     1460 non-null   int64
 63  GarageQual     1379 non-null   object
 64  GarageCond     1379 non-null   object
 65  PavedDrive     1460 non-null   object
 66  WoodDeckSF     1460 non-null   int64
 67  OpenPorchSF    1460 non-null   int64
 68  EnclosedPorch  1460 non-null   int64
 69  3SsnPorch      1460 non-null   int64
 70  ScreenPorch    1460 non-null   int64
 71  PoolArea       1460 non-null   int64
 72  PoolQC         7 non-null      object
 73  Fence          281 non-null    object
 74  MiscFeature    54 non-null     object
 75  MiscVal        1460 non-null   int64
 76  MoSold         1460 non-null   int64
 77  YrSold         1460 non-null   int64
 78  SaleType       1460 non-null   object
 79  SaleCondition  1460 non-null   object
 80  SalePrice      1460 non-null   int64
dtypes: float64(3), int64(35), object(43)
memory usage: 924.0+ KB
```

In [5]:

```
df1.shape
```

Out[5]:

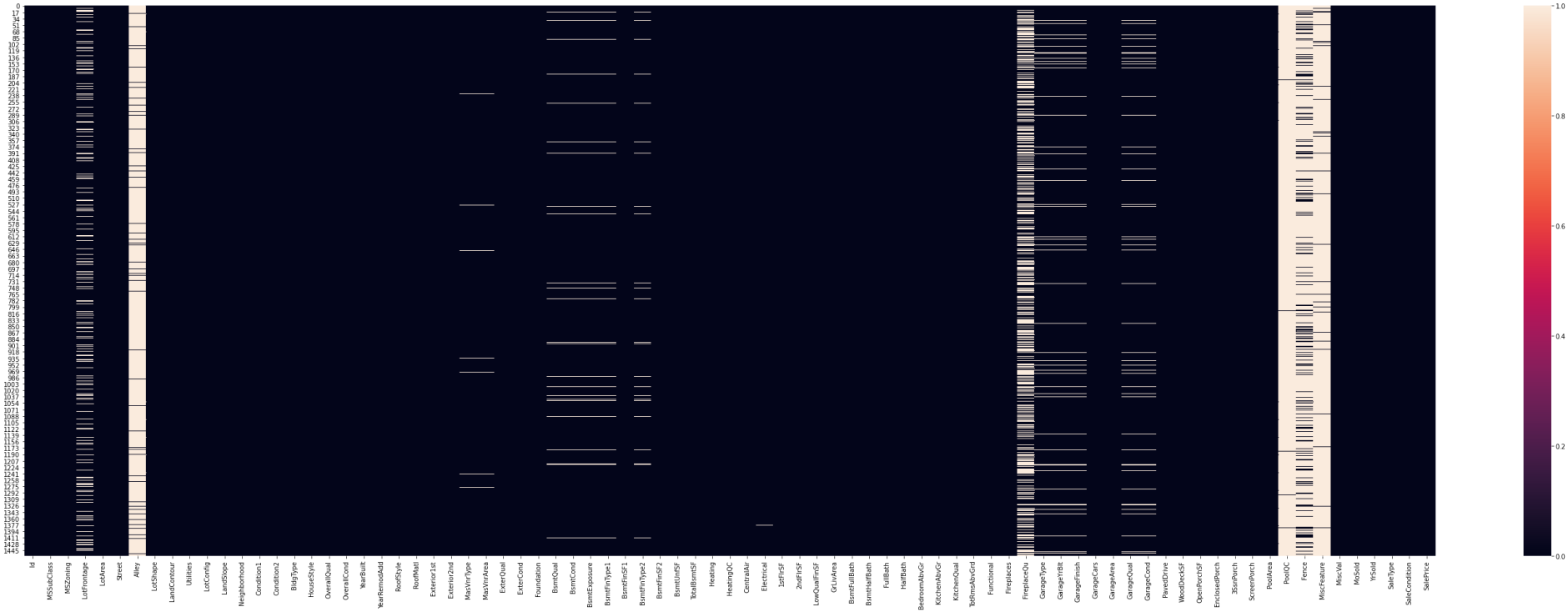
```
(1460, 81)
```

Handling Missing Values in Data

In [6]:

```
plt.figure(figsize=(50,16))
sns.heatmap(df1.isnull())
```

Out[6]: <AxesSubplot:>



In [7]:

```
feature_na=[feature for feature in df1.columns if df1[feature].isnull().sum()>1]

for feature in feature_na:
    print(feature,np.round(df1[feature].isnull().mean(),5),'<--missing value %')
```

```
LotFrontage 0.1774 <--missing value %
Alley 0.93767 <--missing value %
MasVnrType 0.00548 <--missing value %
MasVnrArea 0.00548 <--missing value %
BsmtQual 0.02534 <--missing value %
BsmtCond 0.02534 <--missing value %
BsmtExposure 0.02603 <--missing value %
```

```
BsmtFinType1 0.02534 <--missing value %
BsmtFinType2 0.02603 <--missing value %
FireplaceQu 0.4726 <--missing value %
GarageType 0.05548 <--missing value %
GarageYrBlt 0.05548 <--missing value %
GarageFinish 0.05548 <--missing value %
GarageQual 0.05548 <--missing value %
GarageCond 0.05548 <--missing value %
PoolQC 0.99521 <--missing value %
Fence 0.80753 <--missing value %
MiscFeature 0.96301 <--missing value %
```

In [8]:

```
df1.isnull().sum().sort_values(ascending=False).head(20)
```

Out[8]:

```
PoolQC          1453
MiscFeature     1406
Alley           1369
Fence           1179
FireplaceQu      690
LotFrontage      259
GarageYrBlt       81
GarageCond        81
GarageType        81
GarageFinish      81
GarageQual        81
BsmtFinType2      38
BsmtExposure      38
BsmtQual          37
BsmtCond          37
BsmtFinType1      37
MasVnrArea         8
MasVnrType         8
Electrical         1
Id                 0
dtype: int64
```

In [9]:

```
df1.drop(["PoolQC","MiscFeature","Alley","Fence"],inplace=True,axis=1)
```

In [10]:

```
numeric_data = df1.select_dtypes(include=[np.number])
categorical_data = df1.select_dtypes(exclude=[np.number])
```

In [11]:

```
for c in categorical_data.columns:
```

```
    df1[c].fillna('NA', inplace=True)

df1['MasVnrType'].fillna('None', inplace=True)
```

In [12]:

```
for c in numeric_data.columns:
    df1[c].fillna(0, inplace=True)
```

In [13]:

```
df1.isnull().sum().sort_values(ascending=False).head(20)
```

Out[13]:

```
Id              0
HalfBath        0
FireplaceQu     0
Fireplaces      0
Functional      0
TotRmsAbvGrd    0
KitchenQual     0
KitchenAbvGr    0
BedroomAbvGr    0
FullBath        0
HeatingQC       0
BsmtHalfBath    0
BsmtFullBath    0
GrLivArea       0
LowQualFinSF    0
2ndFlrSF        0
1stFlrSF        0
Electrical      0
GarageType      0
GarageYrBlt     0
dtype: int64
```

Visualizing the train dataset

In [14]:

```
klib.corr_mat(df1)
```

Out[14]:

	Id	MSSubClass	LotFrontage	LotArea	OverallQual	OverallCond	YearBuilt	YearRemodAdd	MasVnrArea	BsmtFinSF1	BsmtFinSF2	Bs
Id	1.00	0.01	-0.02	-0.03	-0.03	0.01	-0.01	-0.02	-0.05	-0.01	-0.01	
MSSubClass	0.01	1.00	-0.22	-0.14	0.03	-0.06	0.03	0.04	0.02	-0.07	-0.07	

	Id	MSSubClass	LotFrontage	LotArea	OverallQual	OverallCond	YearBuilt	YearRemodAdd	MasVnrArea	BsmtFinSF1	BsmtFinSF2	Bs
LotFrontage	-0.02	-0.22	1.00	0.10	0.18	-0.05	0.04	0.08	0.11	0.08	-0.01	
LotArea	-0.03	-0.14	0.10	1.00	0.11	-0.01	0.01	0.01	0.10	0.21	0.11	
OverallQual	-0.03	0.03	0.18	0.11	1.00	-0.09	0.57	0.55	0.41	0.24	-0.06	
OverallCond	0.01	-0.06	-0.05	-0.01	-0.09	1.00	-0.38	0.07	-0.13	-0.05	0.04	
YearBuilt	-0.01	0.03	0.04	0.01	0.57	-0.38	1.00	0.59	0.31	0.25	-0.05	
YearRemodAdd	-0.02	0.04	0.08	0.01	0.55	0.07	0.59	1.00	0.18	0.13	-0.07	
MasVnrArea	-0.05	0.02	0.11	0.10	0.41	-0.13	0.31	0.18	1.00	0.26	-0.07	
BsmtFinSF1	-0.01	-0.07	0.08	0.21	0.24	-0.05	0.25	0.13	0.26	1.00	-0.05	
BsmtFinSF2	-0.01	-0.07	-0.01	0.11	-0.06	0.04	-0.05	-0.07	-0.07	-0.05	1.00	
BsmtUnfSF	-0.01	-0.14	0.16	-0.00	0.31	-0.14	0.15	0.18	0.11	-0.50	-0.21	
TotalBsmtSF	-0.02	-0.24	0.24	0.26	0.54	-0.17	0.39	0.29	0.36	0.52	0.10	
1stFlrSF	0.01	-0.25	0.25	0.30	0.48	-0.14	0.28	0.24	0.34	0.45	0.10	
2ndFlrSF	0.01	0.31	0.04	0.05	0.30	0.03	0.01	0.14	0.17	-0.14	-0.10	
LowQualFinSF	-0.04	0.05	0.05	0.00	-0.03	0.03	-0.18	-0.06	-0.07	-0.06	0.01	
GrLivArea	0.01	0.07	0.22	0.26	0.59	-0.08	0.20	0.29	0.39	0.21	-0.01	
BsmtFullBath	0.00	0.00	0.01	0.16	0.11	-0.05	0.19	0.12	0.08	0.65	0.16	
BsmtHalfBath	-0.02	-0.00	-0.03	0.05	-0.04	0.12	-0.04	-0.01	0.03	0.07	0.07	
FullBath	0.01	0.13	0.12	0.13	0.55	-0.19	0.47	0.44	0.27	0.06	-0.08	
HalfBath	0.01	0.18	-0.01	0.01	0.27	-0.06	0.24	0.18	0.20	0.00	-0.03	
BedroomAbvGr	0.04	-0.02	0.14	0.12	0.10	0.01	-0.07	-0.04	0.10	-0.11	-0.02	
KitchenAbvGr	0.00	0.28	0.03	-0.02	-0.18	-0.09	-0.17	-0.15	-0.04	-0.08	-0.04	
TotRmsAbvGrd	0.03	0.04	0.22	0.19	0.43	-0.06	0.10	0.19	0.28	0.04	-0.04	
Fireplaces	-0.02	-0.05	0.04	0.27	0.40	-0.02	0.15	0.11	0.25	0.26	0.05	
GarageYrBlt	0.01	-0.08	0.02	0.07	0.29	-0.01	0.27	0.15	0.13	0.12	0.04	

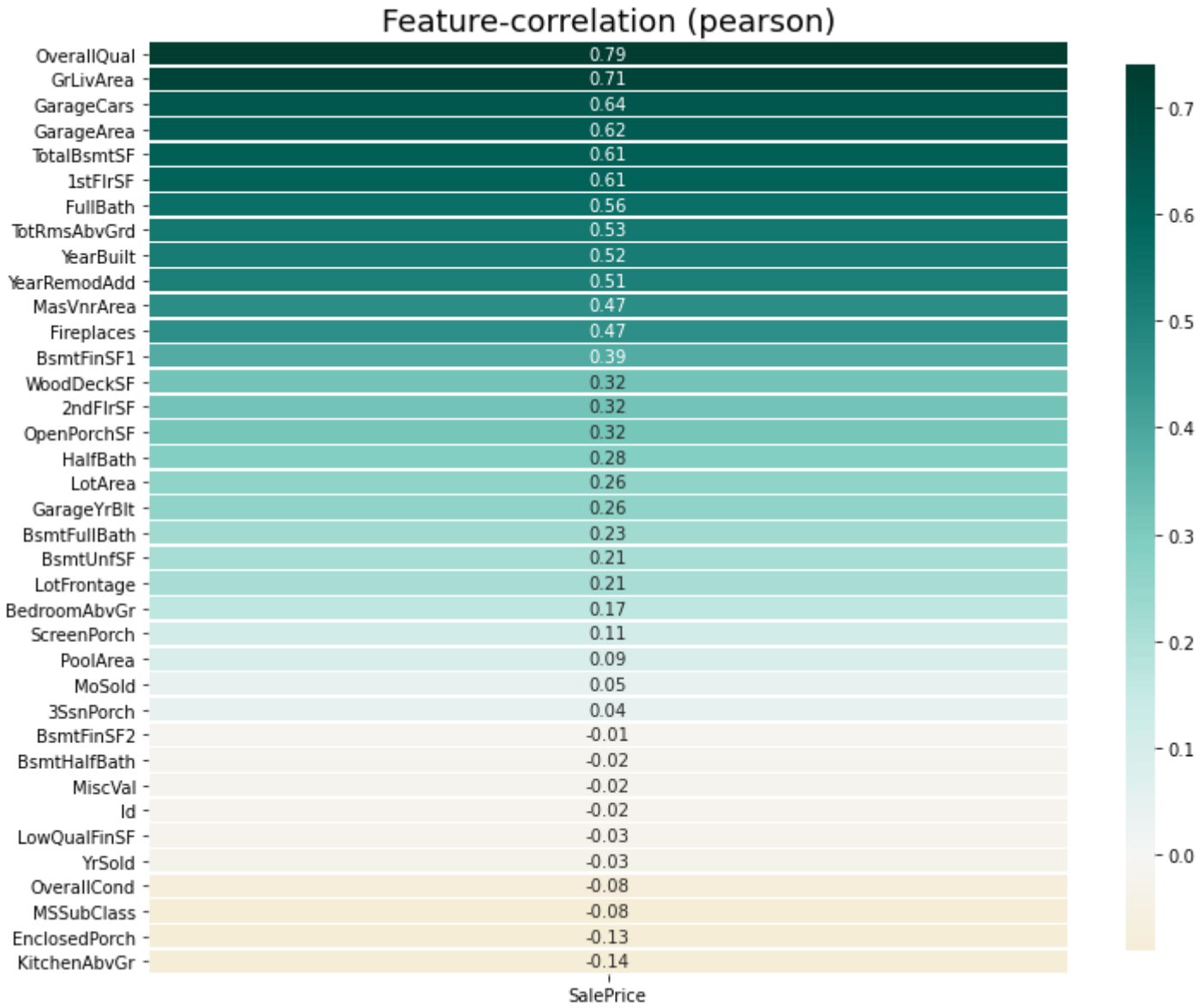
	Id	MSSubClass	LotFrontage	LotArea	OverallQual	OverallCond	YearBuilt	YearRemodAdd	MasVnrArea	BsmtFinSF1	BsmtFinSF2	Bs
GarageCars	0.02	-0.04	0.17	0.15	0.60	-0.19	0.54	0.42	0.36	0.22	-0.04	
GarageArea	0.02	-0.10	0.20	0.18	0.56	-0.15	0.48	0.37	0.37	0.30	-0.02	
WoodDeckSF	-0.03	-0.01	-0.02	0.17	0.24	-0.00	0.22	0.21	0.16	0.20	0.07	
OpenPorchSF	-0.00	-0.01	0.07	0.08	0.31	-0.03	0.19	0.23	0.12	0.11	0.00	
EnclosedPorch	0.00	-0.01	0.03	-0.02	-0.11	0.07	-0.39	-0.19	-0.11	-0.10	0.04	
3SsnPorch	-0.05	-0.04	0.02	0.02	0.03	0.03	0.03	0.05	0.02	0.03	-0.03	
ScreenPorch	0.00	-0.03	0.02	0.04	0.06	0.05	-0.05	-0.04	0.06	0.06	0.09	
PoolArea	0.06	0.01	0.11	0.08	0.07	-0.00	0.00	0.01	0.01	0.14	0.04	
MiscVal	-0.01	-0.01	-0.06	0.04	-0.03	0.07	-0.03	-0.01	-0.03	0.00	0.00	
MoSold	0.02	-0.01	0.02	0.00	0.07	-0.00	0.01	0.02	-0.01	-0.02	-0.02	
YrSold	0.00	-0.02	-0.01	-0.01	-0.03	0.04	-0.01	0.04	-0.01	0.01	0.03	
SalePrice	-0.02	-0.08	0.21	0.26	0.79	-0.08	0.52	0.51	0.47	0.39	-0.01	

In [15]:

```
klib.corr_plot(df1, target='SalePrice')
```

Out[15]:

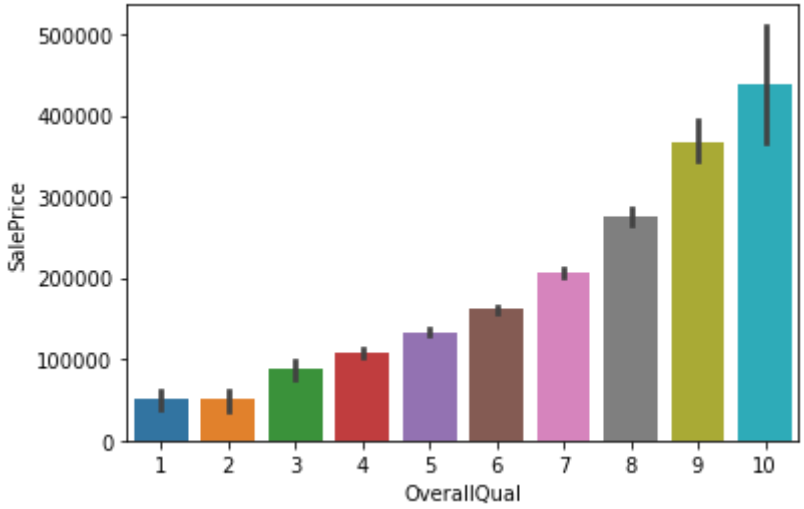
```
<AxesSubplot:title={'center':'Feature-correlation (pearson)'}>
```



In [16]:

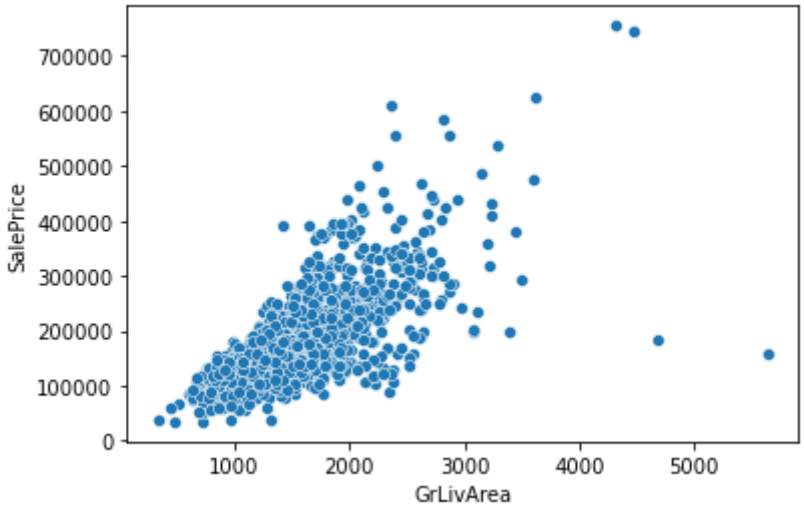
```
sns.barplot(x=df1["OverallQual"],y=df1["SalePrice"])
```

Out[16]: <AxesSubplot:xlabel='OverallQual', ylabel='SalePrice'>



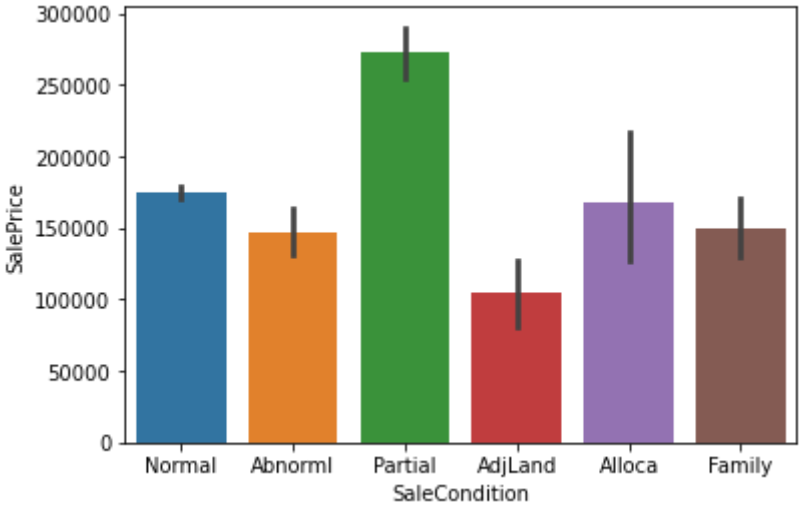
In [17]:
```
sns.scatterplot(x=df1["GrLivArea"],y=df1["SalePrice"])
```

Out[17]: <AxesSubplot:xlabel='GrLivArea', ylabel='SalePrice'>



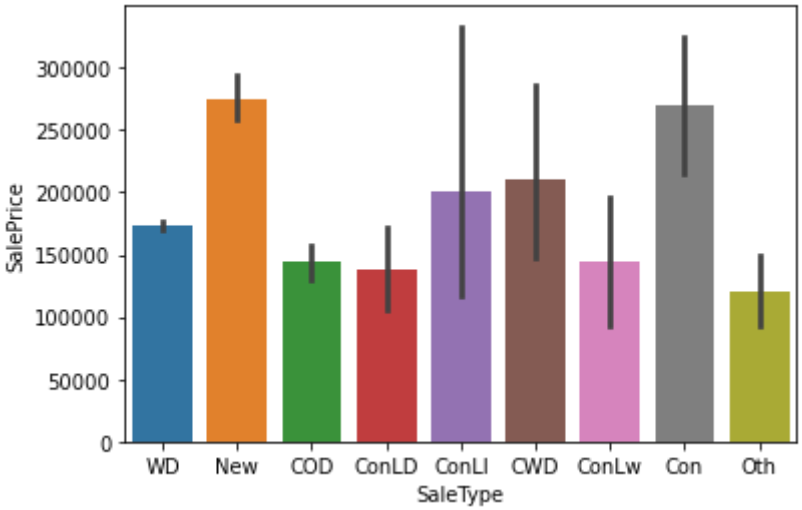
In [18]:
```
sns.barplot(x=df1["SaleCondition"],y=df1["SalePrice"])
```

Out[18]: <AxesSubplot:xlabel='SaleCondition', ylabel='SalePrice'>



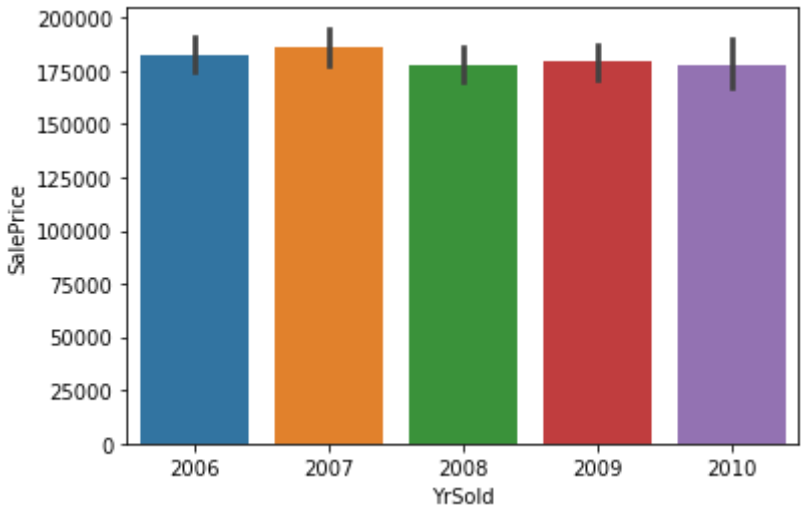
In [19]:
```
sns.barplot(x=df1["SaleType"],y=df1["SalePrice"])
```

Out[19]: <AxesSubplot:xlabel='SaleType', ylabel='SalePrice'>



In [20]:
```
sns.barplot(x=df1["YrSold"],y=df1["SalePrice"])
```

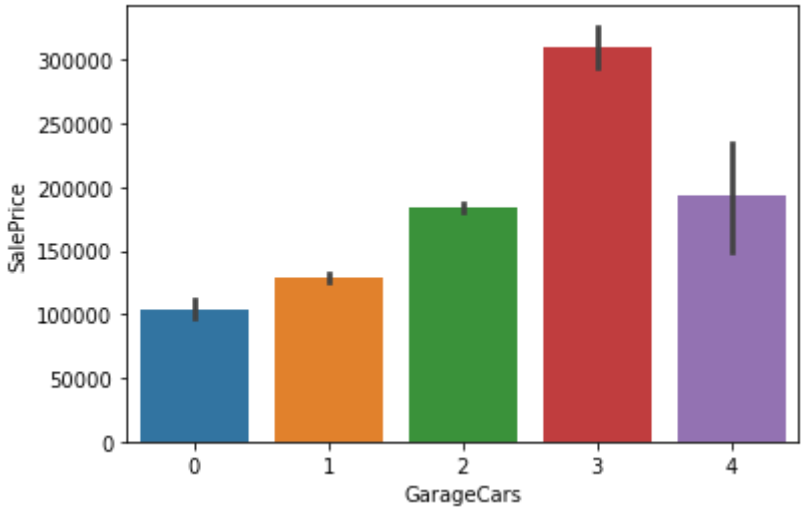
Out[20]: <AxesSubplot:xlabel='YrSold', ylabel='SalePrice'>



In [21]:

```
sns.barplot(x=df1["GarageCars"],y=df1["SalePrice"])
```

Out[21]: <AxesSubplot:xlabel='GarageCars', ylabel='SalePrice'>



In [22]:

```
year_feature = [feature for feature in numeric_data if 'Yr' in feature or 'Year' in feature]
year_feature
```

```
['YearBuilt', 'YearRemodAdd', 'GarageYrBlt', 'YrSold']
```

Out[22]:

In [23]:

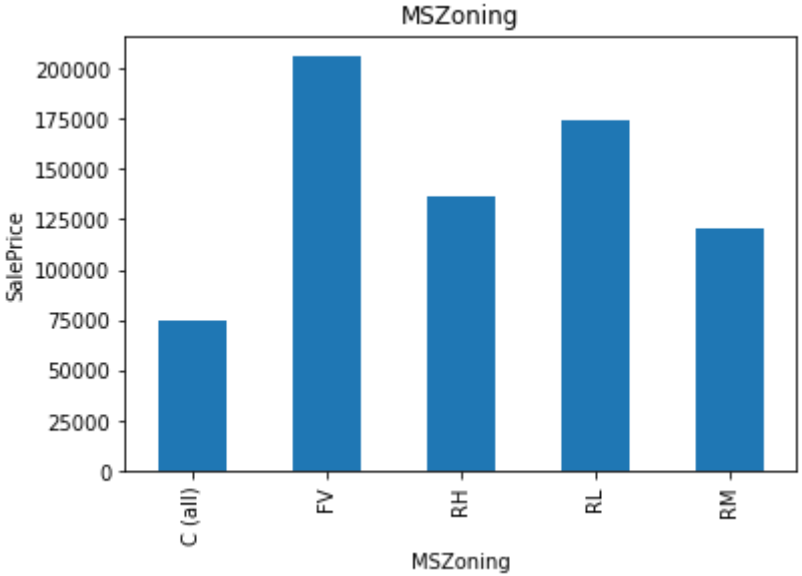
```
categorical_data.head()
```

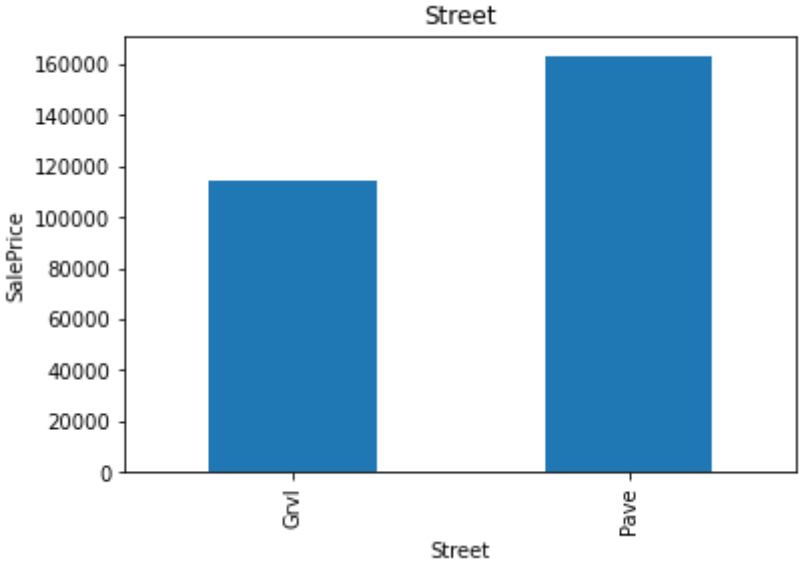
Out[23]:

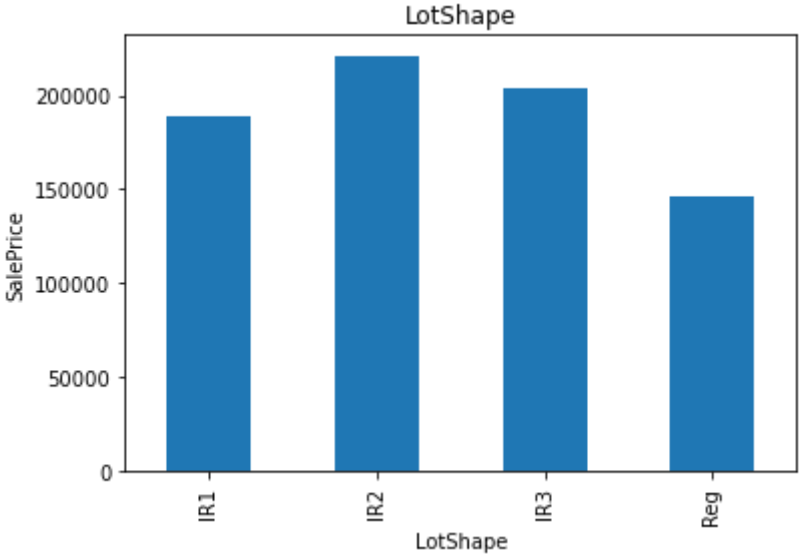
	MSZoning	Street	LotShape	LandContour	Utilities	LotConfig	LandSlope	Neighborhood	Condition1	Condition2	BldgType	HouseStyle	RoofStyle
0	RL	Pave	Reg	Lvl	AllPub	Inside	Gtl	CollgCr	Norm	Norm	1Fam	2Story	Gable
1	RL	Pave	Reg	Lvl	AllPub	FR2	Gtl	Veenker	Feedr	Norm	1Fam	1Story	Gable
2	RL	Pave	IR1	Lvl	AllPub	Inside	Gtl	CollgCr	Norm	Norm	1Fam	2Story	Gable
3	RL	Pave	IR1	Lvl	AllPub	Corner	Gtl	Crawfor	Norm	Norm	1Fam	2Story	Gable
4	RL	Pave	IR1	Lvl	AllPub	FR2	Gtl	NoRidge	Norm	Norm	1Fam	2Story	Gable

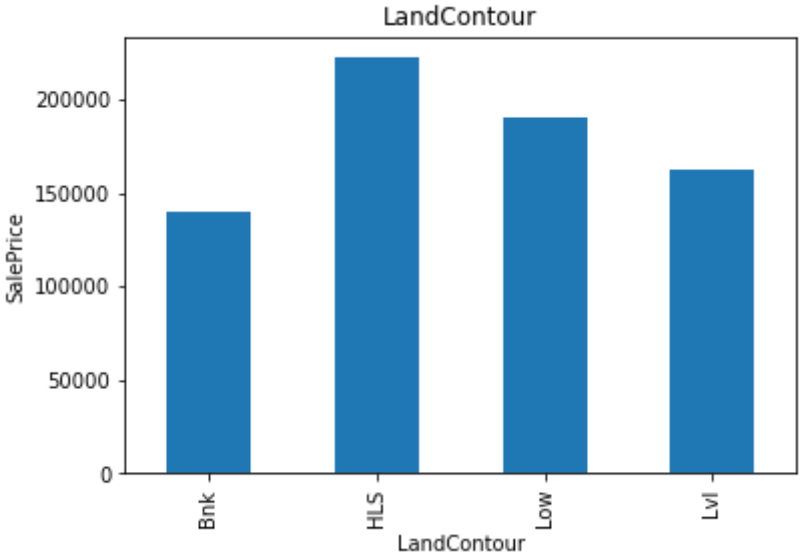
In [24]:

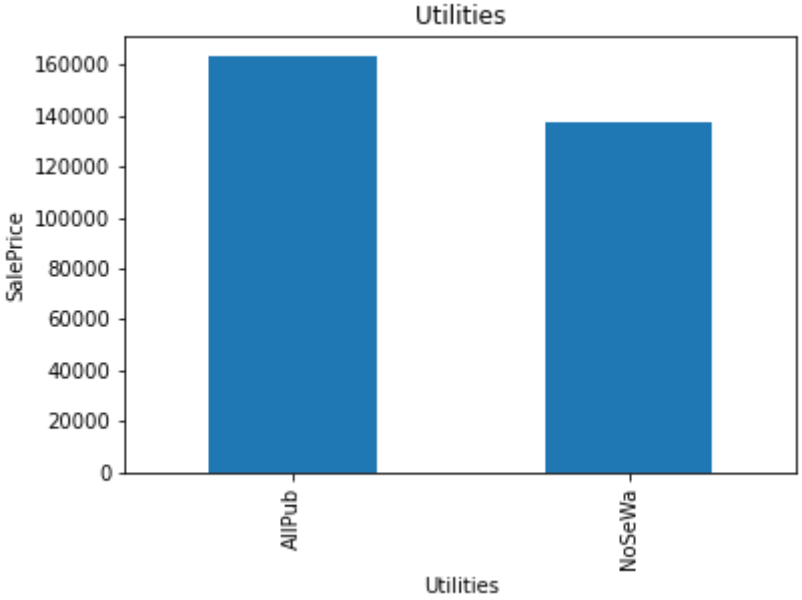
```
for feature in categorical_data:
    data=df1.copy()
    data.groupby(feature)['SalePrice'].median().plot.bar()
    plt.xlabel(feature)
    plt.ylabel('SalePrice')
    plt.title(feature)
    plt.show()
```

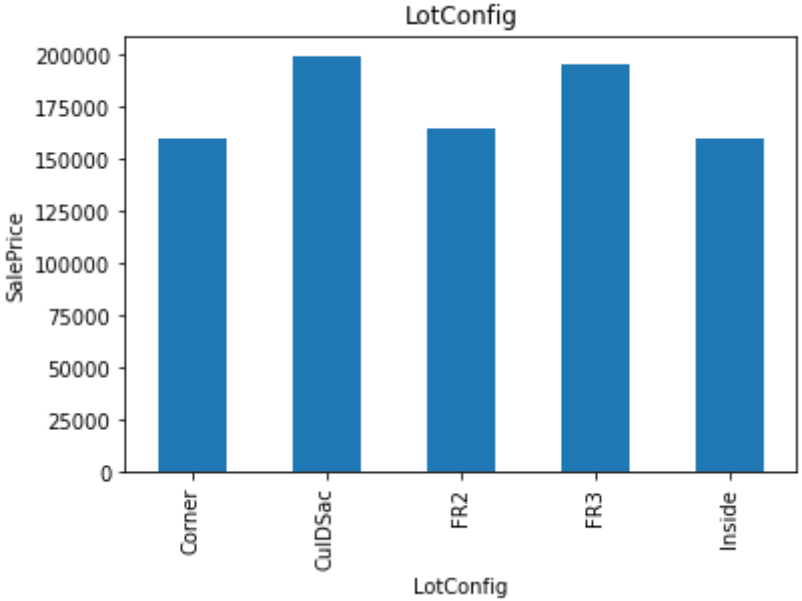

MSZoning
SalePrice
200000
175000
150000
125000
100000
75000
50000
25000
0
C (all)
FV
RH
RL
RM
MSZoning

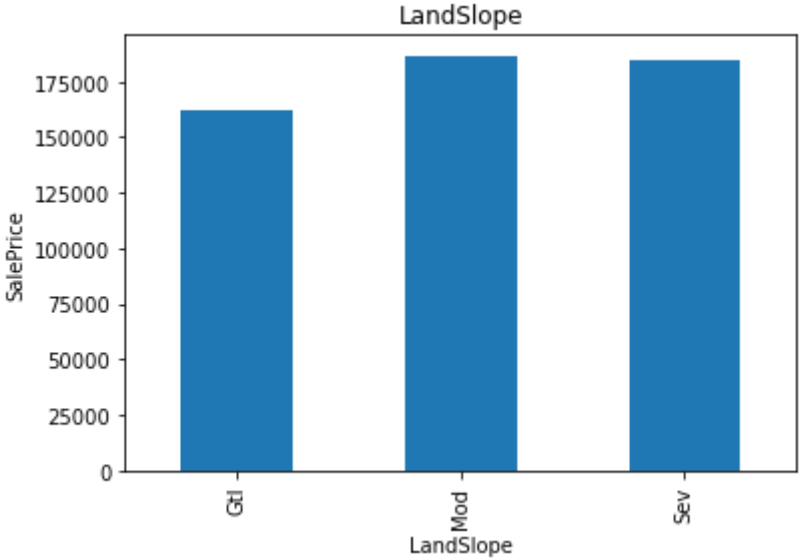

Street
SalePrice
160000
140000
120000
100000
80000
60000
40000
20000
0
Grvl
Pave
Street

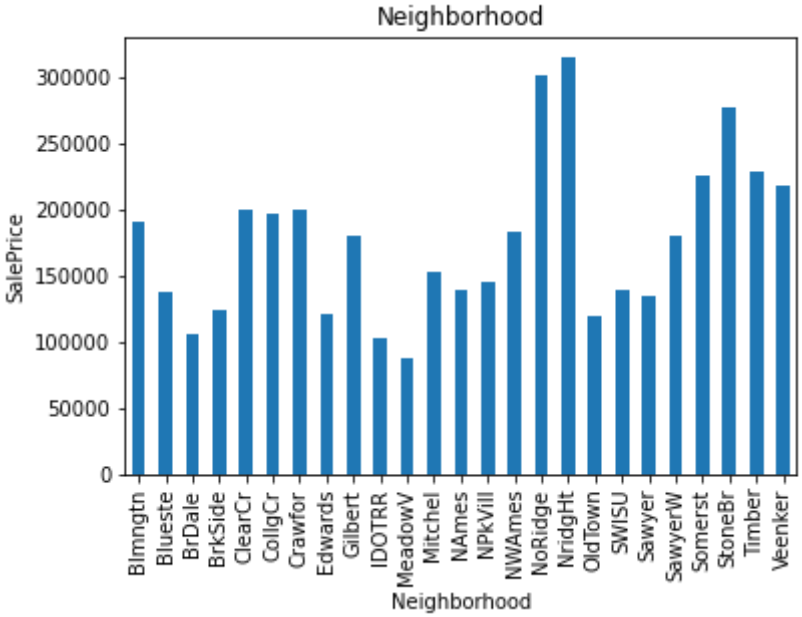

LotShape
SalePrice
200000
150000
100000
50000
0
IR1
IR2
IR3
Reg
LotShape

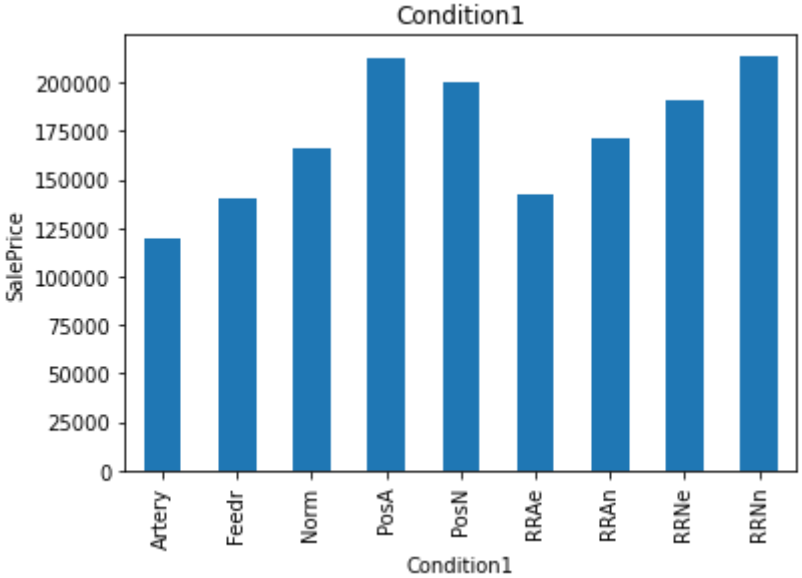

LandContour
SalePrice
200000
150000
100000
50000
0
Bnk
HLS
Low
Lvl
LandContour

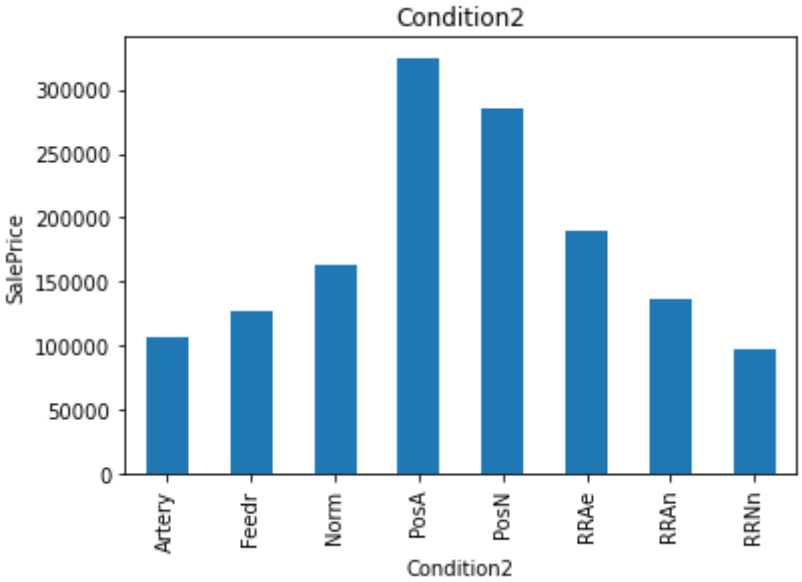

Utilities
SalePrice
160000
140000
120000
100000
80000
60000
40000
20000
0
AllPub
NoSeWa
Utilities

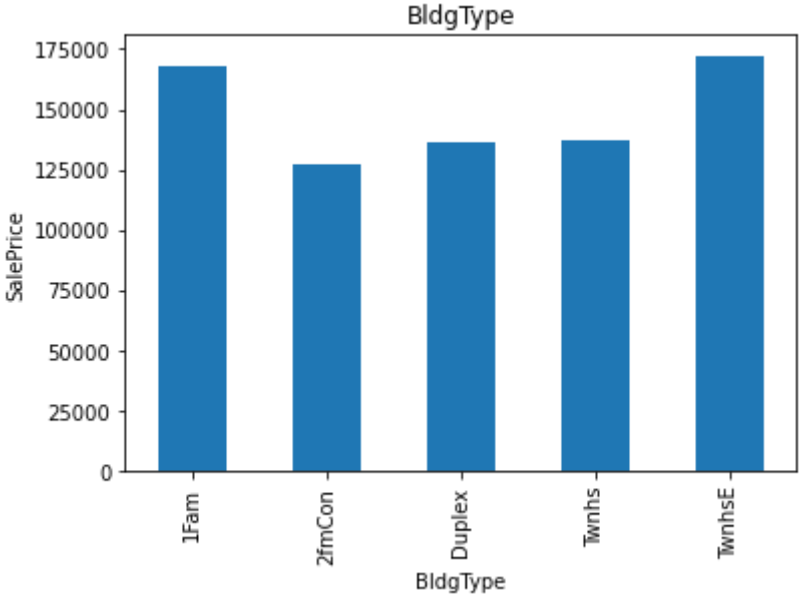

LotConfig
SalePrice
200000
175000
150000
125000
100000
75000
50000
25000
0
Corner
CulDSac
FR2
FR3
Inside
LotConfig

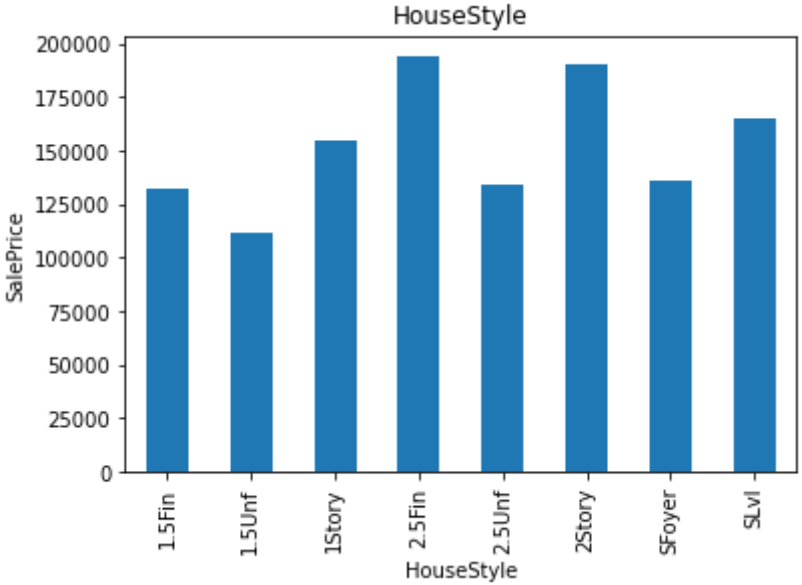

LandSlope
SalePrice
175000
150000
125000
100000
75000
50000
25000
0
Gtl
Mod
Sev
LandSlope

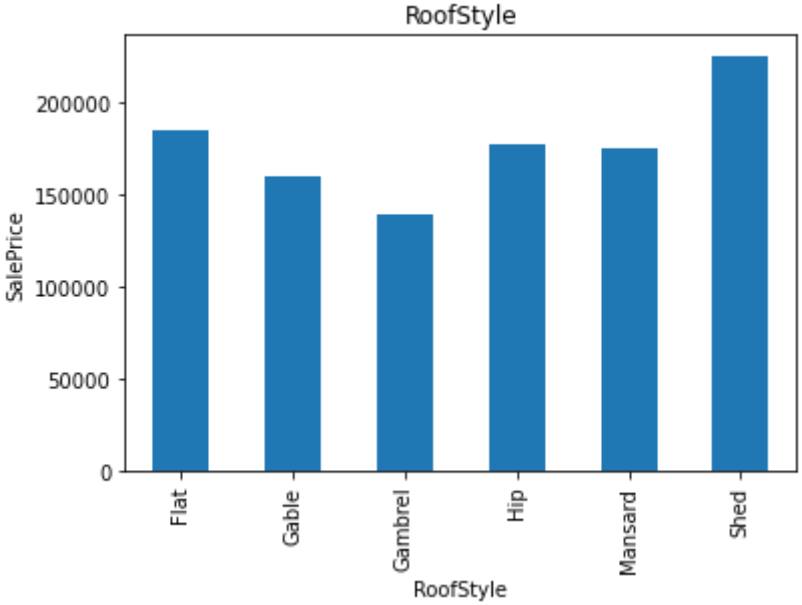

Neighborhood
SalePrice
300000
250000
200000
150000
100000
50000
0
Blmngtn
Blueste
BrDale
BrkSide
ClearCr
CollgCr
Crawfor
Edwards
Gilbert
IDOTRR
MeadowV
Mitchel
NAmes
NPkVill
NWAmes
NoRidge
NridgHt
OldTown
SWISU
Sawyer
SawyerW
Somerst
StoneBr
Timber
Veenker
Neighborhood

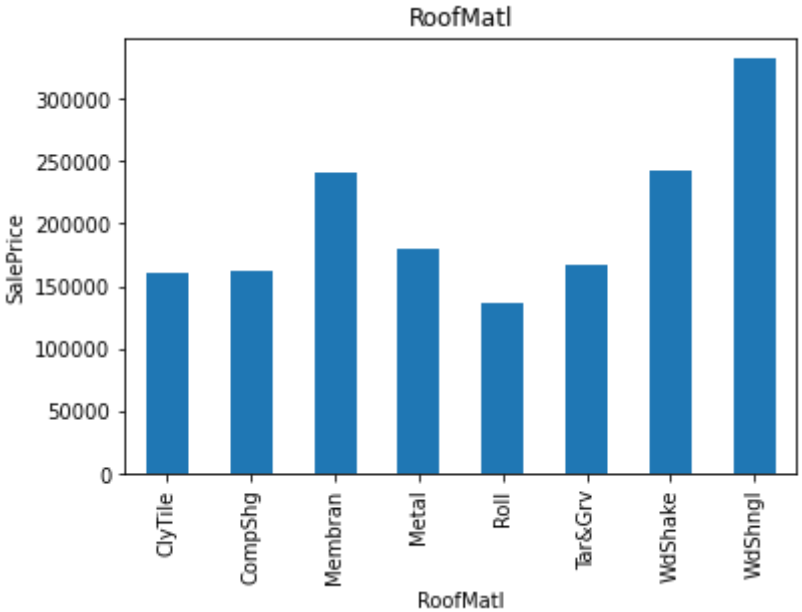

Condition1
SalePrice
0
25000
50000
75000
100000
125000
150000
175000
200000
Artery
Feedr
Norm
PosA
PosN
RRAe
RRAn
RRNe
RRNn
Condition1

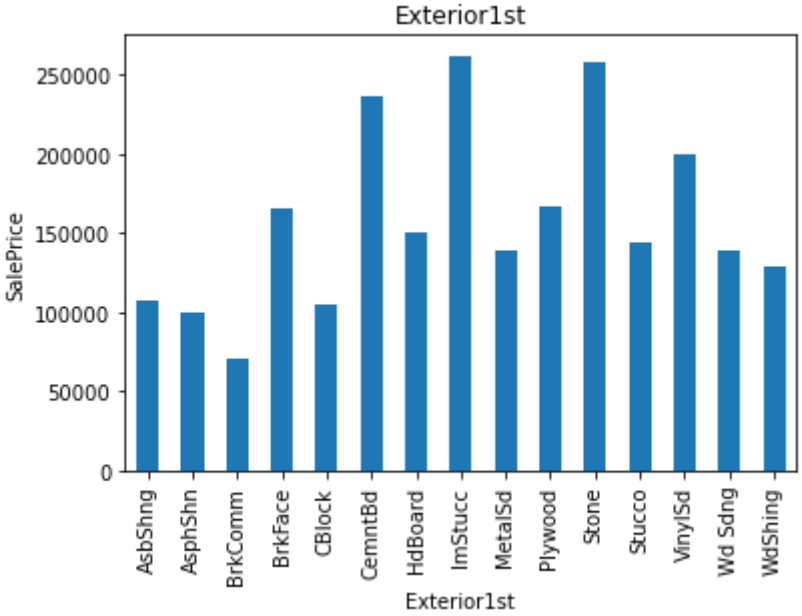

Condition2
SalePrice
0
50000
100000
150000
200000
250000
300000
Artery
Feedr
Norm
PosA
PosN
RRAe
RRAn
RRNn
Condition2

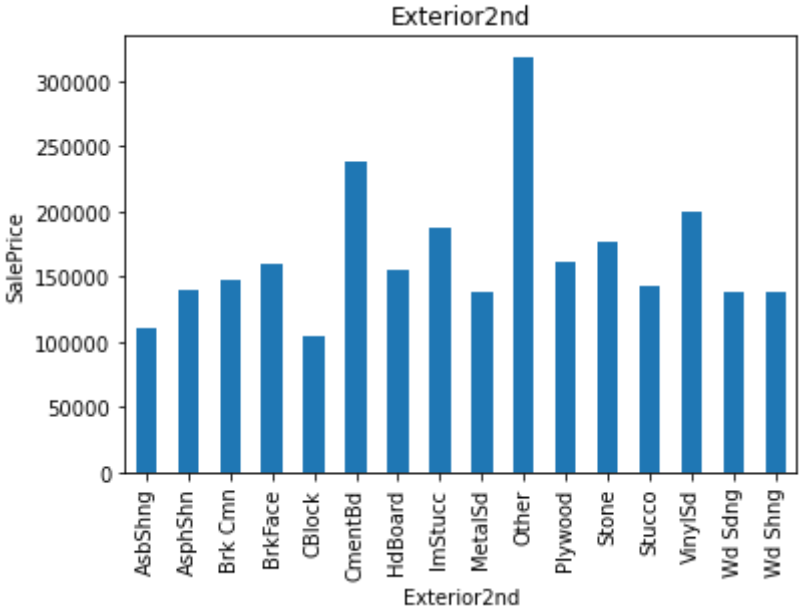

BldgType
175000
150000
125000
100000
75000
50000
25000
0
SalePrice
1Fam
2fmCon
Duplex
Twnhs
TwnhsE
BldgType

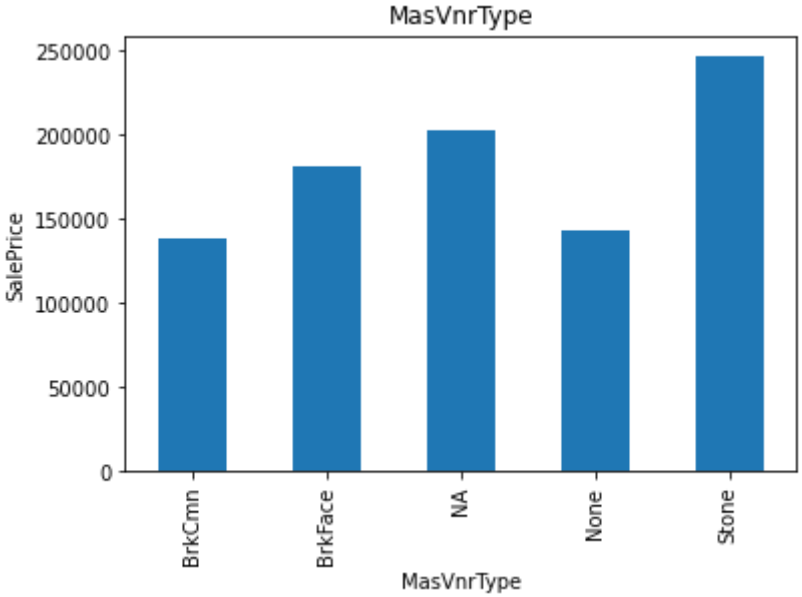

HouseStyle
200000
175000
150000
125000
100000
75000
50000
25000
0
SalePrice
1.5Fin
1.5Unf
1Story
2.5Fin
2.5Unf
2Story
SFoyer
SLvl
HouseStyle

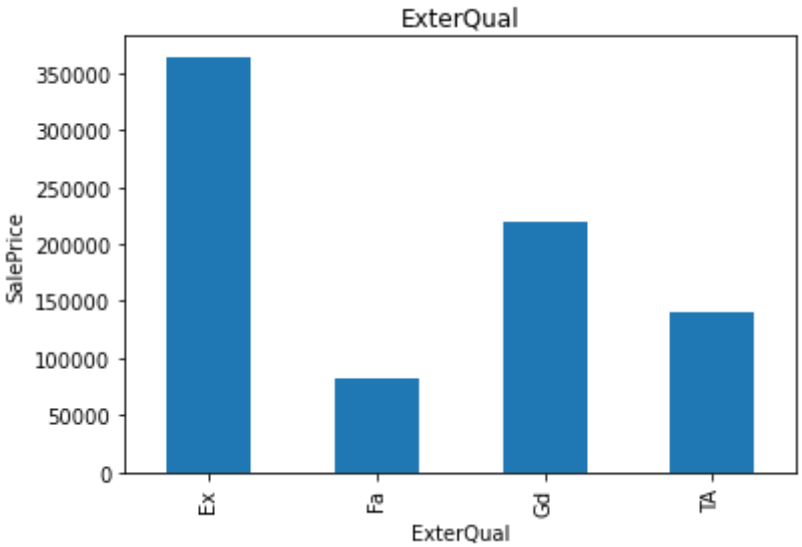

RoofStyle
SalePrice
200000
150000
100000
50000
0
Flat
Gable
Gambrel
Hip
Mansard
Shed
RoofStyle

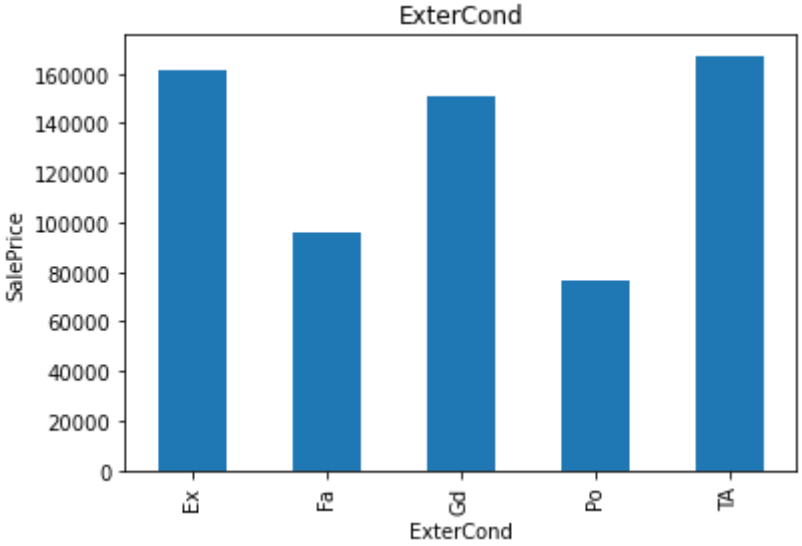

RoofMatl
SalePrice
300000
250000
200000
150000
100000
50000
0
ClyTile
CompShg
Membran
Metal
Roll
Tar&Grv
WdShake
WdShngl
RoofMatl

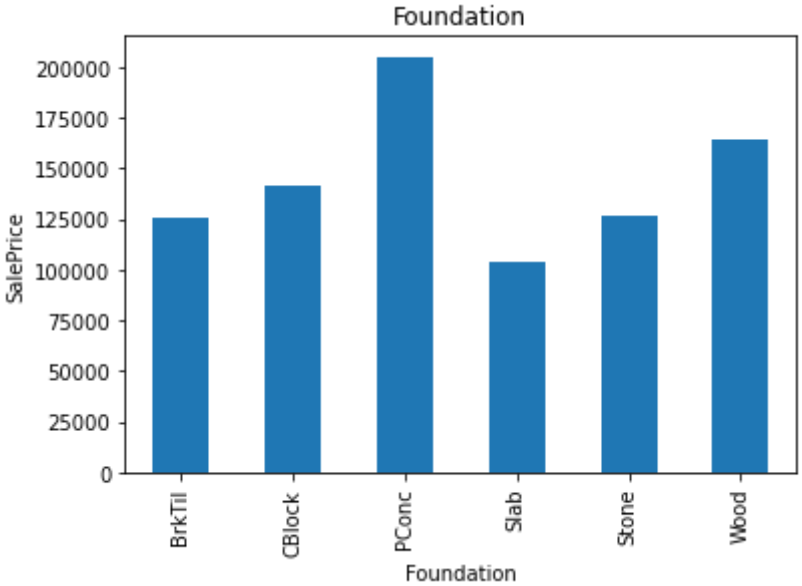

Exterior1st
SalePrice
250000
200000
150000
100000
50000
0
AsbShng
AsphShn
BrkComm
BrkFace
CBlock
CemntBd
HdBoard
ImStucc
MetalSd
Plywood
Stone
Stucco
VinylSd
Wd Sdng
WdShing
Exterior1st

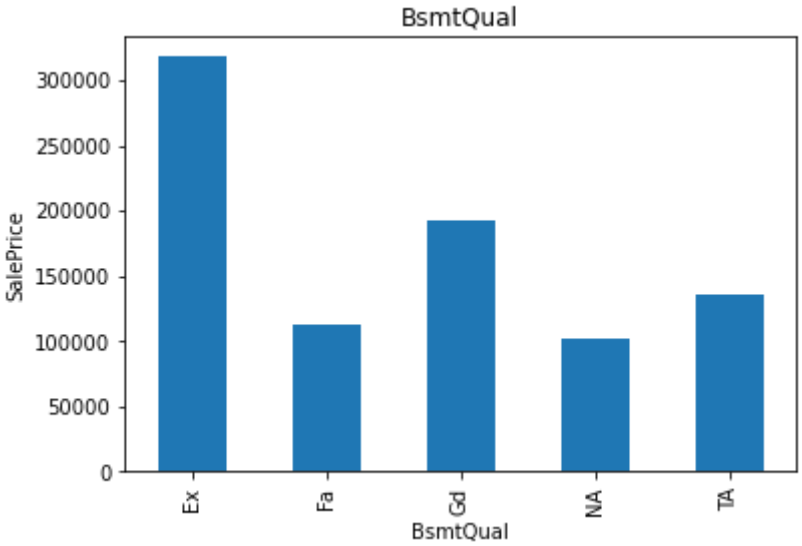

Exterior2nd
SalePrice
300000
250000
200000
150000
100000
50000
0
AsbShng
AsphShn
Brk Cmn
BrkFace
CBlock
CmentBd
HdBoard
ImStucc
MetalSd
Other
Plywood
Stone
Stucco
VinylSd
Wd Sdng
Wd Shng
Exterior2nd

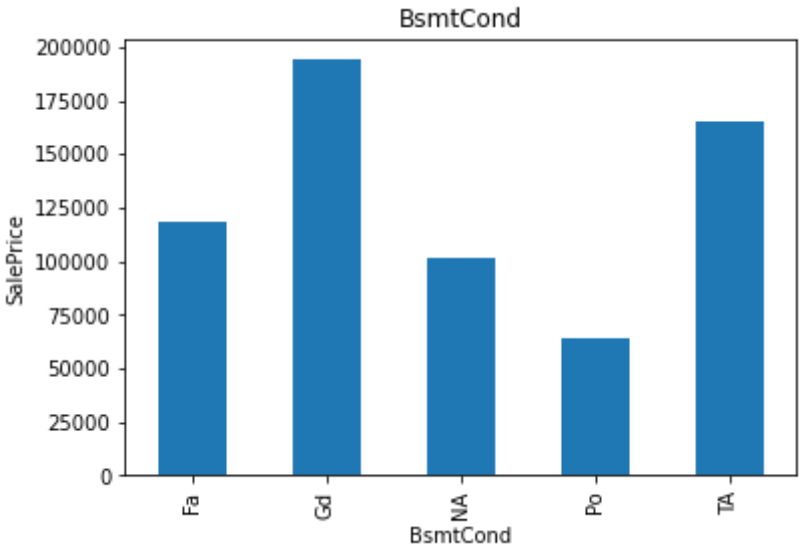

MasVnrType
250000
200000
150000
100000
50000
0
SalePrice
BrkCmn
BrkFace
NA
None
Stone
MasVnrType

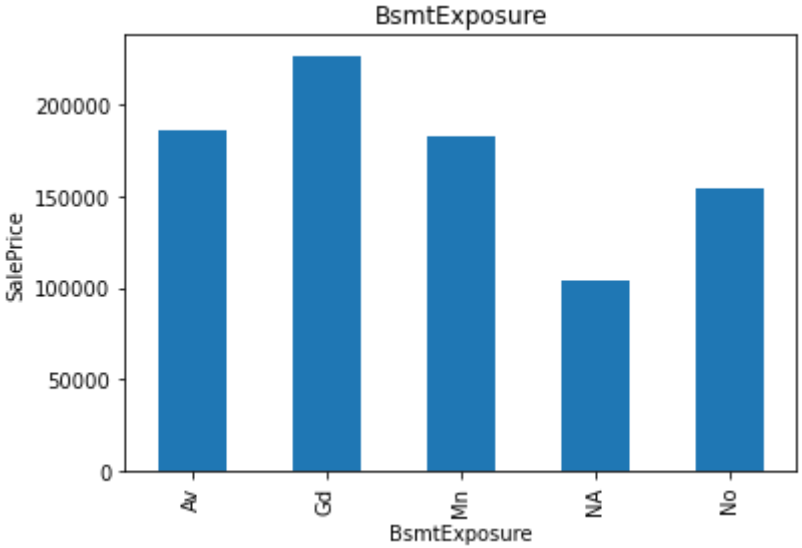

ExterQual
350000
300000
250000
200000
150000
100000
50000
0
SalePrice
Ex
Fa
Gd
TA
ExterQual

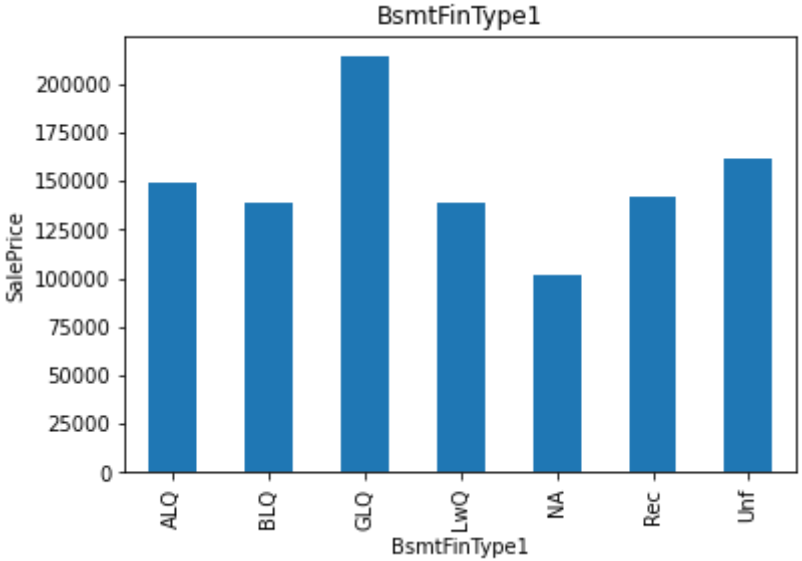

ExterCond
SalePrice
160000
140000
120000
100000
80000
60000
40000
20000
0
Ex
Fa
Gd
Po
TA
ExterCond

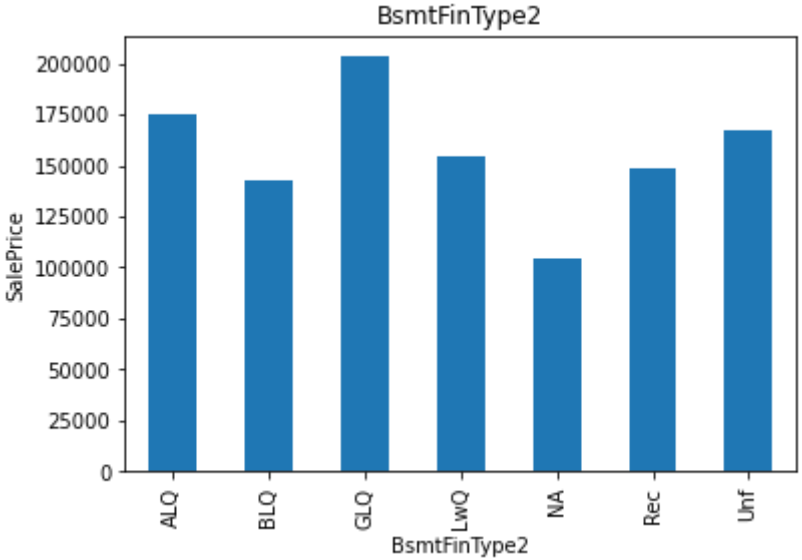

Foundation
SalePrice
200000
175000
150000
125000
100000
75000
50000
25000
0
BrkTil
CBlock
PConc
Slab
Stone
Wood
Foundation

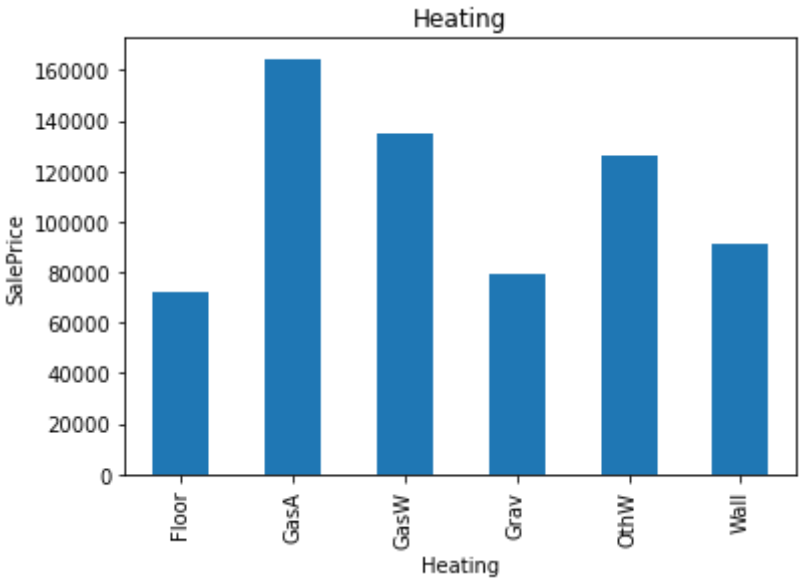

BsmtQual
SalePrice
300000
250000
200000
150000
100000
50000
0
Ex
Fa
Gd
NA
TA
BsmtQual

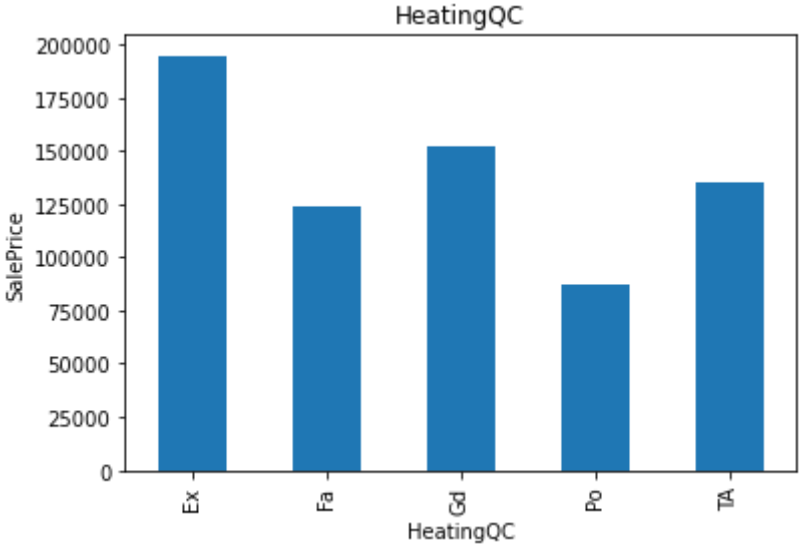

BsmtCond
SalePrice
200000
175000
150000
125000
100000
75000
50000
25000
0
Fa
Gd
NA
Po
TA
BsmtCond

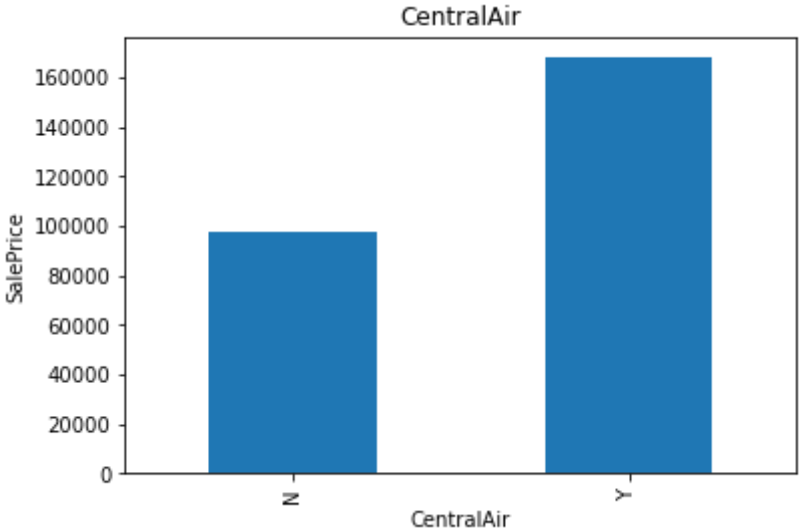

BsmtExposure
SalePrice
0
50000
100000
150000
200000
Av
Gd
Mn
NA
No
BsmtExposure

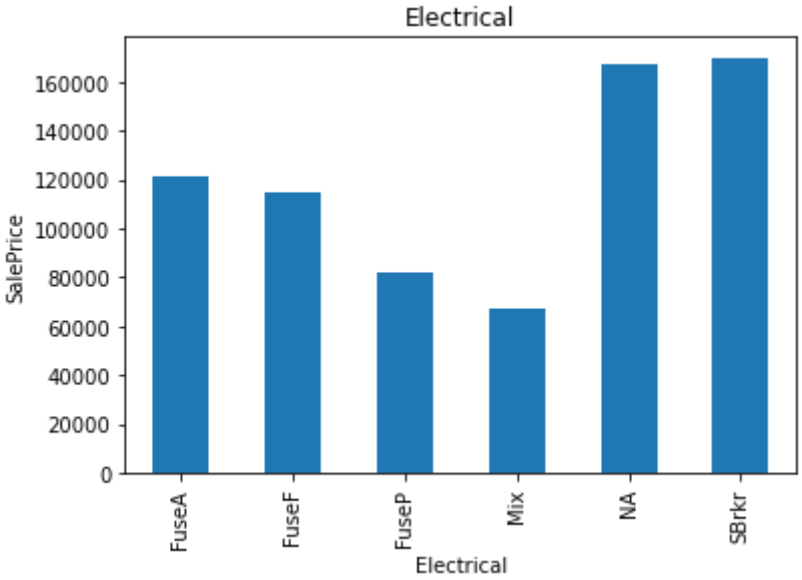

BsmtFinType1
SalePrice
0
25000
50000
75000
100000
125000
150000
175000
200000
ALQ
BLQ
GLQ
LwQ
NA
Rec
Unf
BsmtFinType1

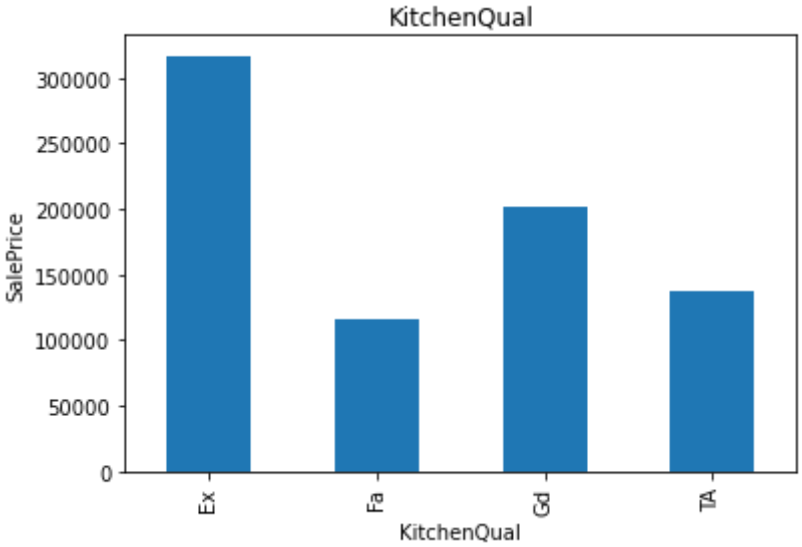

BsmtFinType2
SalePrice
200000
175000
150000
125000
100000
75000
50000
25000
0
ALQ
BLQ
GLQ
LwQ
NA
Rec
Unf
BsmtFinType2

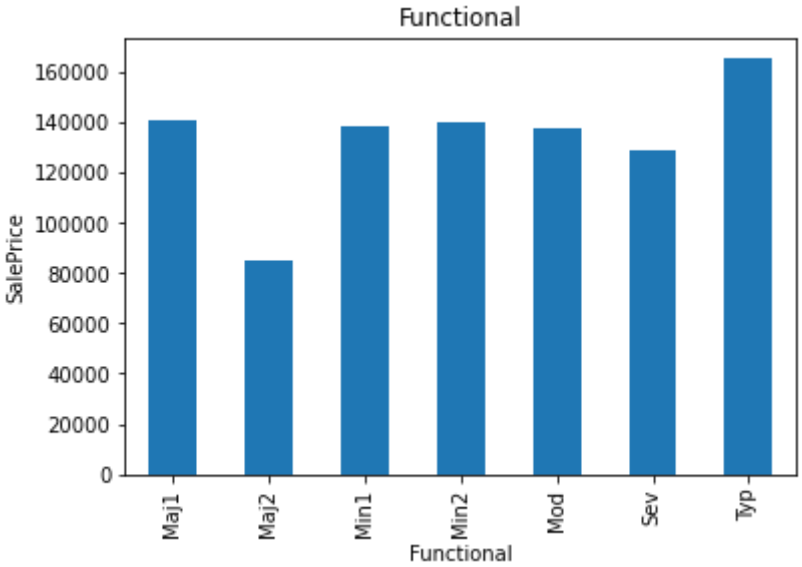

Heating
SalePrice
160000
140000
120000
100000
80000
60000
40000
20000
0
Floor
GasA
GasW
Grav
OthW
Wall
Heating

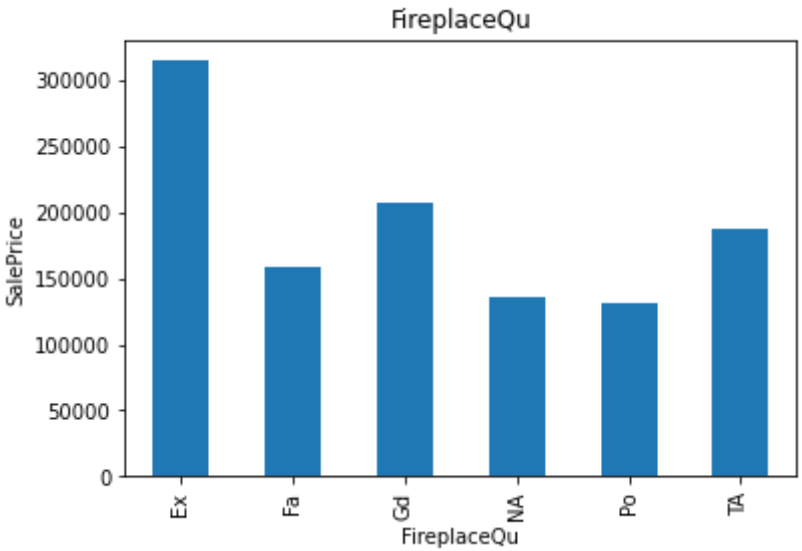

HeatingQC
200000
175000
150000
125000
100000
75000
50000
25000
0
SalePrice
Ex
Fa
Gd
Po
TA
HeatingQC

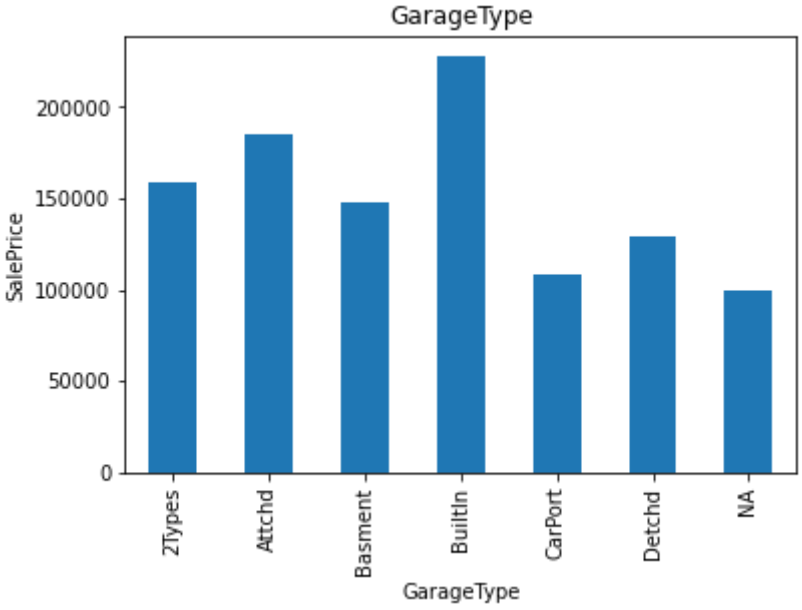

CentralAir
160000
140000
120000
100000
80000
60000
40000
20000
0
SalePrice
N
Y
CentralAir

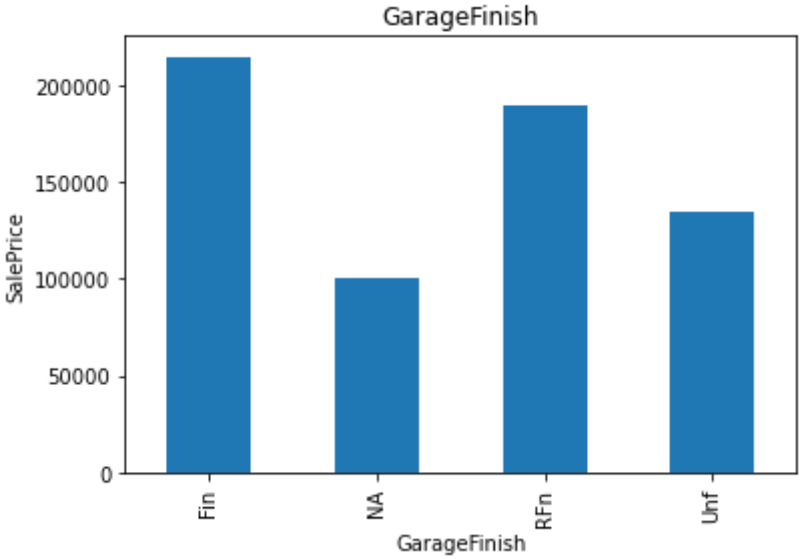

Electrical
SalePrice
160000
140000
120000
100000
80000
60000
40000
20000
0
FuseA
FuseF
FuseP
Mix
NA
SBrkr
Electrical

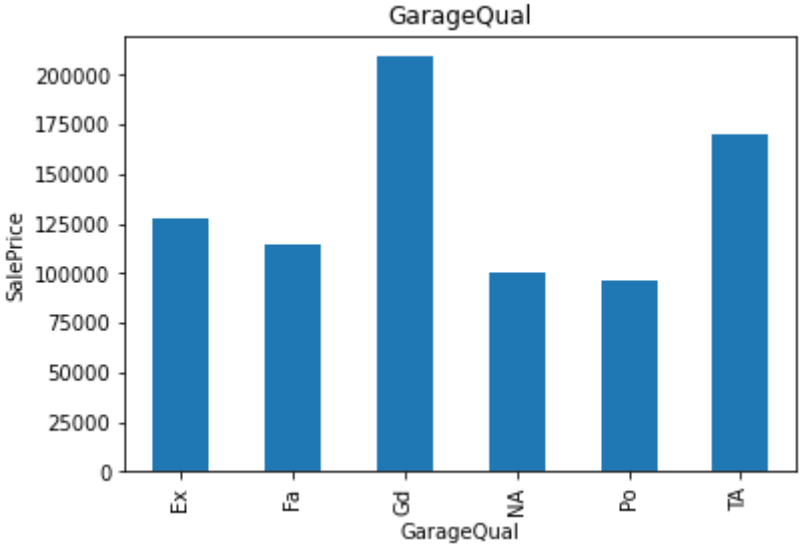

KitchenQual
SalePrice
300000
250000
200000
150000
100000
50000
0
Ex
Fa
Gd
TA
KitchenQual

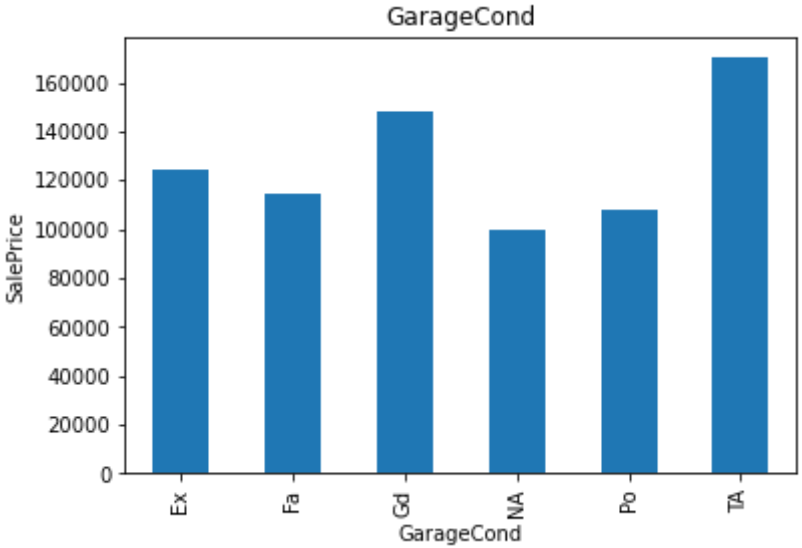

Functional
SalePrice
160000
140000
120000
100000
80000
60000
40000
20000
0
Maj1
Maj2
Min1
Min2
Mod
Sev
Typ
Functional

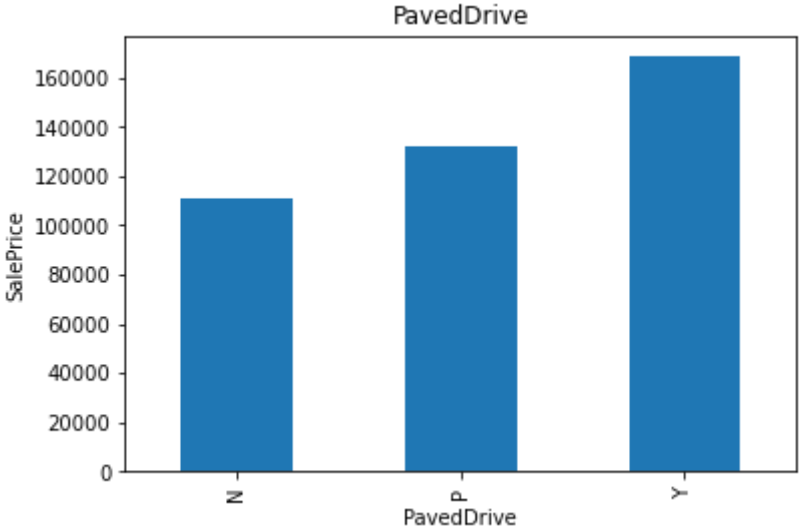

FireplaceQu
SalePrice
300000
250000
200000
150000
100000
50000
0
Ex
Fa
Gd
NA
Po
TA
FireplaceQu

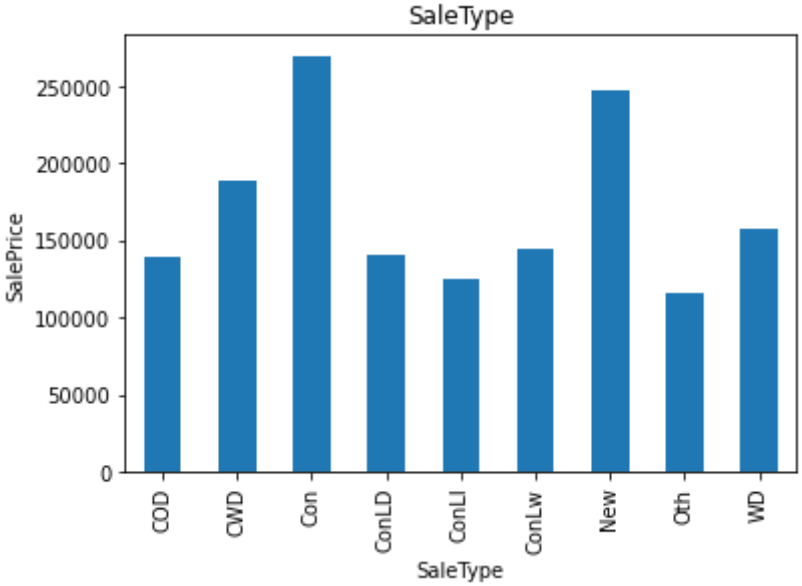

GarageType
SalePrice
200000
150000
100000
50000
0
2Types
Attchd
Basment
BuiltIn
CarPort
Detchd
NA
GarageType

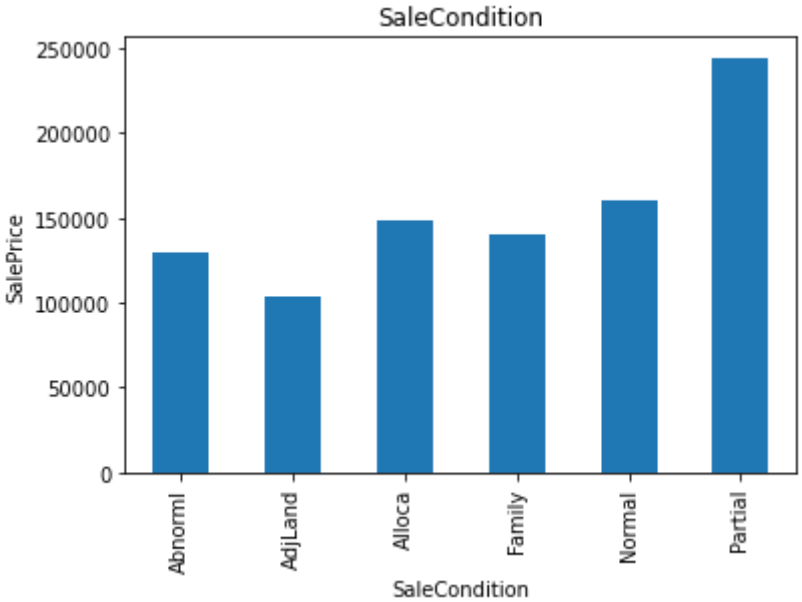

GarageFinish
SalePrice
200000
150000
100000
50000
0
Fin
NA
RFn
Unf
GarageFinish


GarageQual
SalePrice
200000
175000
150000
125000
100000
75000
50000
25000
0
Ex
Fa
Gd
NA
Po
TA
GarageQual


GarageCond
SalePrice
160000
140000
120000
100000
80000
60000
40000
20000
0
Ex
Fa
Gd
NA
Po
TA
GarageCond

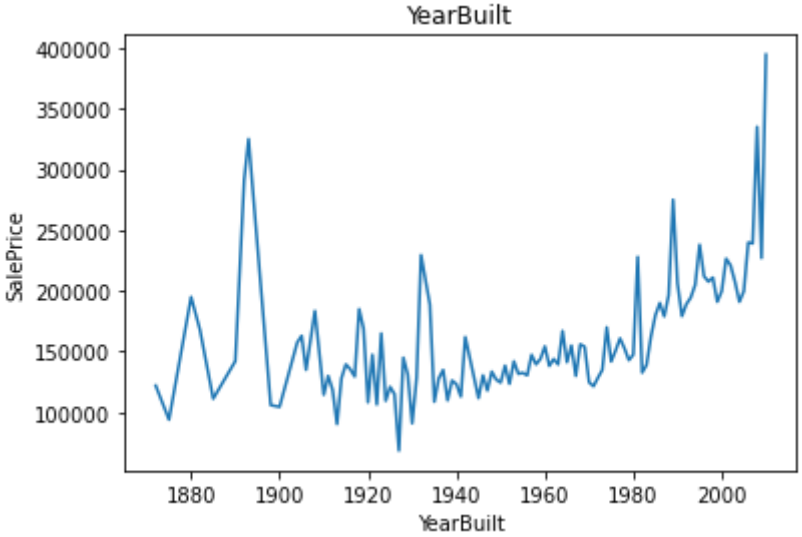

PavedDrive
SalePrice
160000
140000
120000
100000
80000
60000
40000
20000
0
N
P
Y
PavedDrive

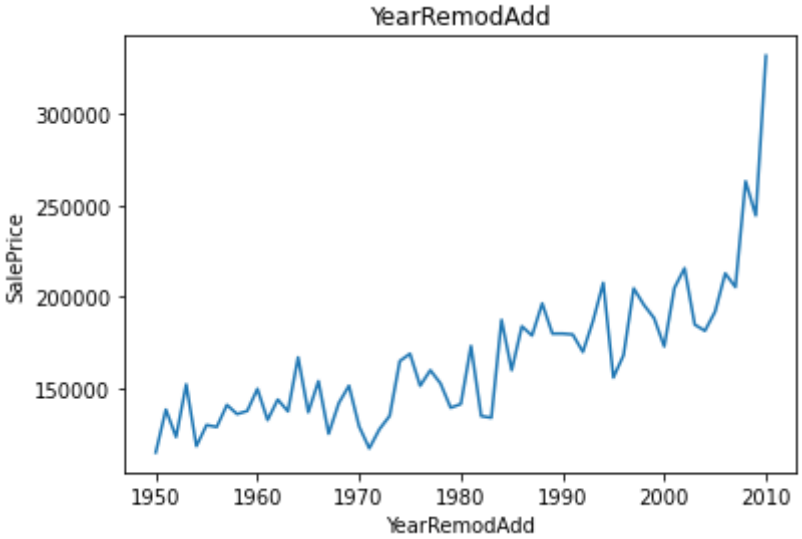

SaleType
SalePrice
250000
200000
150000
100000
50000
0
COD
CWD
Con
ConLD
ConLI
ConLw
New
Oth
WD
SaleType

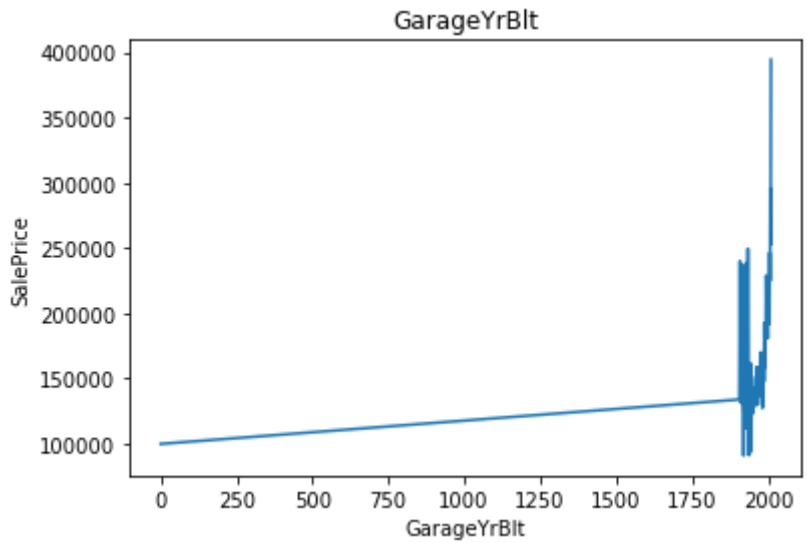


In [25]:

```
for feature in year_feature:
    data=df1.copy()
    data.groupby(feature)['SalePrice'].median().plot()
    plt.xlabel(feature)
    plt.ylabel('SalePrice')
    plt.title(feature)
    plt.show()
```


YearBuilt
400000
350000
300000
250000
200000
150000
100000
SalePrice
1880
1900
1920
1940
1960
1980
2000
YearBuilt


YearRemodAdd
300000
250000
200000
150000
SalePrice
1950
1960
1970
1980
1990
2000
2010
YearRemodAdd



In [26]:

```
df1.drop(["Id","YrSold","YearRemodAdd","GarageYrBlt","MasVnrArea","LotFrontage","BsmtFinSF2","BsmtUnfSF","FireplaceQu"],inplace=Tr
```

In [27]:

```
df1.head()
```

Out[27]:

MSSubClass	MSZoning	LotArea	Street	LotShape	LandContour	Utilities	LotConfig	LandSlope	Neighborhood	Condition1	Condition2	BldgType	H

	MSSubClass	MSZoning	LotArea	Street	LotShape	LandContour	Utilities	LotConfig	LandSlope	Neighborhood	Condition1	Condition2	BldgType	H
0	60	RL	8450	Pave	Reg	Lvl	AllPub	Inside	Gtl	CollgCr	Norm	Norm	1Fam	
1	20	RL	9600	Pave	Reg	Lvl	AllPub	FR2	Gtl	Veenker	Feedr	Norm	1Fam	
2	60	RL	11250	Pave	IR1	Lvl	AllPub	Inside	Gtl	CollgCr	Norm	Norm	1Fam	
3	70	RL	9550	Pave	IR1	Lvl	AllPub	Corner	Gtl	Crawfor	Norm	Norm	1Fam	
4	60	RL	14260	Pave	IR1	Lvl	AllPub	FR2	Gtl	NoRidge	Norm	Norm	1Fam	

In [28]:

```
from sklearn import preprocessing
encoder = preprocessing.LabelEncoder()
for i in df1.columns:
    if isinstance(df1[i][0], str):
            df1[i] = encoder.fit_transform(df1[i])
```

In [29]:

```
X=df1.drop("SalePrice",axis=1)
y=df1["SalePrice"]
```

In [30]:

```
X
```

Out[30]:

	MSSubClass	MSZoning	LotArea	Street	LotShape	LandContour	Utilities	LotConfig	LandSlope	Neighborhood	Condition1	Condition2	BldgType
0	60	3	8450	1	3	3	0	4	0	5	2	2	(
1	20	3	9600	1	3	3	0	2	0	24	1	2	(
2	60	3	11250	1	0	3	0	4	0	5	2	2	(
3	70	3	9550	1	0	3	0	0	0	6	2	2	(
4	60	3	14260	1	0	3	0	2	0	15	2	2	(
...	...	...	...	...	...	...	...	...	...	...	...	...	..
1455	60	3	7917	1	3	3	0	4	0	8	2	2	(
1456	20	3	13175	1	3	3	0	4	0	14	2	2	(

	MSSubClass	MSZoning	LotArea	Street	LotShape	LandContour	Utilities	LotConfig	LandSlope	Neighborhood	Condition1	Condition2	BldgType
1457	70	3	9042	1	3	3	0	4	0	6	2	2	(
1458	20	3	9717	1	3	3	0	4	0	12	2	2	(
1459	20	3	9937	1	3	3	0	4	0	7	2	2	(

1460 rows × 67 columns

In [31]:

```
y
```

Out[31]:

```
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 [32]:

```
X.shape
```

Out[32]:

```
(1460, 67)
```

In [33]:

```
y.shape
```

Out[33]:

```
(1460,)
```

Training and model

In [34]:

```
from sklearn.model_selection import train_test_split
```

In [35]:
```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.33, random_state=31)
```

In [36]:
```
from sklearn.linear_model import LinearRegression
```

In [37]:
```
lin_model=LinearRegression()
```

In [38]:
```
lin_model.fit(X_train,y_train)
```

Out[38]:
```
LinearRegression()
```

In [39]:
```
lin_model.score(X_test,y_test)
```

Out[39]:
```
0.826869032380648
```

In [40]:
```
from sklearn import ensemble
reg = ensemble.GradientBoostingRegressor(learning_rate = 0.1, n_estimators=1000)
reg.fit(X_train, y_train)
reg.score(X_test, y_test)
```

Out[40]:
```
0.8394266335892154
```

In [41]:
```
pred =reg.predict(X_test)
pred[:5]
```

Out[41]:
```
array([214468.52641887, 130638.79080997, 123790.31028825, 154227.60774091,
       117955.81232661])
```

Working with test dataset

In [42]:
```
df2=pd.read_csv("test1.csv")
```

In [43]:
```
df2.head()
```

Out[43]:

	Id	MSSubClass	MSZoning	LotFrontage	LotArea	Street	Alley	LotShape	LandContour	Utilities	LotConfig	LandSlope	Neighborhood	Condition1
0	1461	20	RH	80.0	11622	Pave	NaN	Reg	Lvl	AllPub	Inside	Gtl	NAmes	Feedı
1	1462	20	RL	81.0	14267	Pave	NaN	IR1	Lvl	AllPub	Corner	Gtl	NAmes	Norm
2	1463	60	RL	74.0	13830	Pave	NaN	IR1	Lvl	AllPub	Inside	Gtl	Gilbert	Norm
3	1464	60	RL	78.0	9978	Pave	NaN	IR1	Lvl	AllPub	Inside	Gtl	Gilbert	Norm
4	1465	120	RL	43.0	5005	Pave	NaN	IR1	HLS	AllPub	Inside	Gtl	StoneBr	Norm

Handling Missing Values in Data

In [44]:
```
df2.isnull().sum().sort_values(ascending=False).head(35)
```

Out[44]:
```
PoolQC          1456
MiscFeature     1408
Alley           1352
Fence           1169
FireplaceQu      730
LotFrontage      227
GarageYrBlt       78
GarageQual        78
GarageFinish      78
GarageCond        78
GarageType        76
BsmtCond          45
BsmtQual          44
BsmtExposure      44
BsmtFinType1      42
BsmtFinType2      42
MasVnrType        16
MasVnrArea        15
MSZoning           4
BsmtHalfBath       2
Utilities          2
Functional         2
BsmtFullBath       2
BsmtFinSF1         1
```

```
BsmtFinSF2         1
BsmtUnfSF          1
KitchenQual        1
TotalBsmtSF        1
Exterior2nd        1
GarageCars         1
Exterior1st        1
GarageArea         1
SaleType           1
MiscVal            0
BedroomAbvGr       0
dtype: int64
```

In [45]:

```
df2 = df2.drop(['PoolQC','MiscFeature','Alley','Fence','LotFrontage'],axis=1)
```

In [46]:

```
#separating the data acc to dtypes
numeric_data = df2.select_dtypes(include=[np.number])
categorical_data = df2.select_dtypes(exclude=[np.number])
```

In [47]:

```
#filling null values
for c in categorical_data.columns:
    df2[c].fillna('NA', inplace=True)

df2['MasVnrType'].fillna('None', inplace=True)
```

In [48]:

```
for c in numeric_data.columns:
    df2[c].fillna(0, inplace=True)
```

In [49]:

```
df2.isnull().sum().sort_values(ascending=False).head(35)
```

Out[49]:

```
Id               0
GarageType       0
Fireplaces       0
Functional       0
TotRmsAbvGrd     0
KitchenQual      0
KitchenAbvGr     0
BedroomAbvGr     0
HalfBath         0
```

```
FullBath         0
BsmtHalfBath     0
BsmtFullBath     0
GrLivArea        0
LowQualFinSF     0
2ndFlrSF         0
1stFlrSF         0
Electrical       0
FireplaceQu      0
GarageYrBlt      0
HeatingQC        0
GarageFinish     0
SaleType         0
YrSold           0
MoSold           0
MiscVal          0
PoolArea         0
ScreenPorch      0
3SsnPorch        0
EnclosedPorch    0
OpenPorchSF      0
WoodDeckSF       0
PavedDrive       0
GarageCond       0
GarageQual       0
GarageArea       0
dtype: int64
```

Visualizing the test dataset

In [50]:

```
klib.corr_mat(df2)
```

Out[50]:

	Id	MSSubClass	LotArea	OverallQual	OverallCond	YearBuilt	YearRemodAdd	MasVnrArea	BsmtFinSF1	BsmtFinSF2	BsmtUnfSF	Tota
Id	1.00	0.00	0.05	-0.06	0.01	-0.06	-0.08	-0.02	-0.05	0.02	0.00	
MSSubClass	0.00	1.00	-0.36	0.03	-0.07	0.04	0.05	-0.01	-0.06	-0.08	-0.11	
LotArea	0.05	-0.36	1.00	0.11	-0.10	0.05	0.04	0.19	0.19	0.05	0.07	
OverallQual	-0.06	0.03	0.11	1.00	-0.10	0.62	0.59	0.45	0.32	-0.03	0.24	
OverallCond	0.01	-0.07	-0.10	-0.10	1.00	-0.36	0.02	-0.14	-0.06	0.04	-0.14	

	Id	MSSubClass	LotArea	OverallQual	OverallCond	YearBuilt	YearRemodAdd	MasVnrArea	BsmtFinSF1	BsmtFinSF2	BsmtUnfSF	Tota
YearBuilt	-0.06	0.04	0.05	0.62	-0.36	1.00	0.63	0.30	0.31	-0.01	0.11	
YearRemodAdd	-0.08	0.05	0.04	0.59	0.02	0.63	1.00	0.21	0.18	-0.06	0.15	
MasVnrArea	-0.02	-0.01	0.19	0.45	-0.14	0.30	0.21	1.00	0.34	0.04	0.06	
BsmtFinSF1	-0.05	-0.06	0.19	0.32	-0.06	0.31	0.18	0.34	1.00	-0.06	-0.46	
BsmtFinSF2	0.02	-0.08	0.05	-0.03	0.04	-0.01	-0.06	0.04	-0.06	1.00	-0.26	
BsmtUnfSF	0.00	-0.11	0.07	0.24	-0.14	0.11	0.15	0.06	-0.46	-0.26	1.00	
TotalBsmtSF	-0.04	-0.20	0.28	0.56	-0.18	0.43	0.31	0.43	0.55	0.08	0.41	
1stFlrSF	-0.02	-0.25	0.46	0.48	-0.17	0.34	0.24	0.44	0.47	0.07	0.28	
2ndFlrSF	-0.01	0.31	-0.01	0.20	-0.02	0.03	0.18	0.06	-0.19	-0.10	-0.01	
LowQualFinSF	-0.02	0.01	-0.01	-0.07	-0.01	-0.10	-0.06	-0.05	-0.07	-0.02	0.07	
GrLivArea	-0.03	0.07	0.37	0.56	-0.16	0.29	0.35	0.41	0.22	-0.02	0.23	
BsmtFullBath	-0.03	0.02	0.09	0.22	-0.03	0.24	0.15	0.20	0.63	0.17	-0.37	
BsmtHalfBath	0.01	-0.00	-0.01	-0.04	0.05	-0.02	-0.08	0.01	0.09	0.12	-0.12	
FullBath	-0.06	0.15	0.15	0.51	-0.24	0.47	0.48	0.24	0.11	-0.07	0.26	
HalfBath	-0.05	0.18	0.08	0.27	-0.12	0.30	0.24	0.18	-0.02	-0.03	-0.03	
BedroomAbvGr	0.00	0.01	0.18	0.05	-0.03	-0.04	-0.00	0.05	-0.12	-0.04	0.20	
KitchenAbvGr	-0.02	0.24	-0.03	-0.13	-0.09	-0.10	-0.14	-0.07	-0.09	-0.03	0.10	
TotRmsAbvGrd	0.00	0.04	0.29	0.35	-0.13	0.13	0.20	0.27	0.06	-0.06	0.24	
Fireplaces	-0.04	-0.06	0.28	0.38	-0.04	0.19	0.15	0.30	0.33	0.08	-0.04	
GarageYrBlt	-0.02	-0.13	0.09	0.27	0.04	0.25	0.15	0.12	0.14	0.06	-0.01	
GarageCars	-0.06	-0.05	0.26	0.60	-0.18	0.54	0.43	0.35	0.29	0.01	0.15	
GarageArea	-0.05	-0.11	0.32	0.57	-0.16	0.48	0.38	0.37	0.32	0.02	0.15	
WoodDeckSF	0.02	-0.02	0.16	0.27	0.04	0.23	0.23	0.17	0.24	0.13	-0.07	
OpenPorchSF	-0.00	-0.03	0.16	0.29	-0.10	0.21	0.26	0.16	0.14	-0.01	0.11	

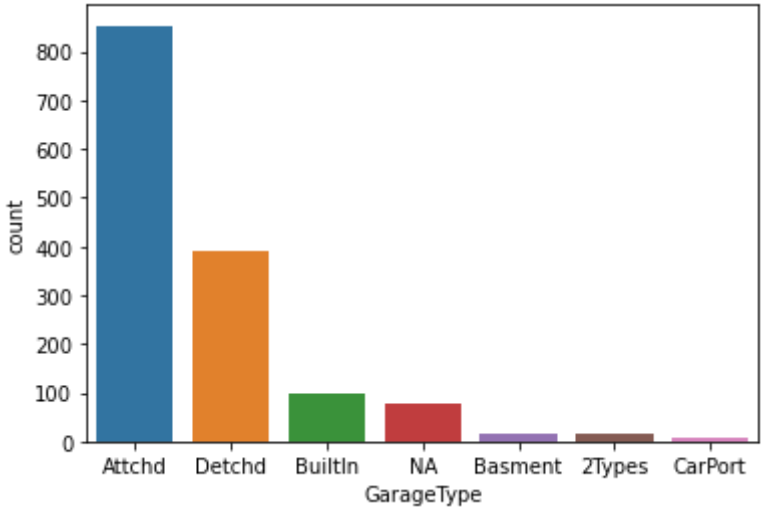
	Id	MSSubClass	LotArea	OverallQual	OverallCond	YearBuilt	YearRemodAdd	MasVnrArea	BsmtFinSF1	BsmtFinSF2	BsmtUnfSF	Tota
EnclosedPorch	0.02	-0.03	0.10	-0.16	0.07	-0.36	-0.24	-0.11	-0.10	0.03	0.01	
3SsnPorch	-0.03	-0.03	-0.00	0.00	0.07	-0.01	0.03	0.01	0.09	-0.01	-0.05	
ScreenPorch	0.03	-0.07	0.09	0.02	0.03	-0.03	-0.05	0.07	0.13	0.04	-0.08	
PoolArea	0.05	-0.02	0.14	-0.01	-0.04	-0.00	-0.03	-0.01	0.01	0.05	-0.03	
MiscVal	-0.01	-0.05	0.14	0.03	0.01	0.01	0.00	0.11	0.17	-0.01	0.00	
MoSold	0.14	0.01	0.01	-0.01	-0.01	0.02	0.01	0.00	0.01	-0.00	0.01	
YrSold	-0.97	-0.01	-0.05	-0.01	0.02	-0.01	0.03	-0.03	0.03	-0.01	-0.04	

In [51]:

```
sns.countplot(df2.GarageType)
```

Out[51]:

```
<AxesSubplot:xlabel='GarageType', ylabel='count'>
```

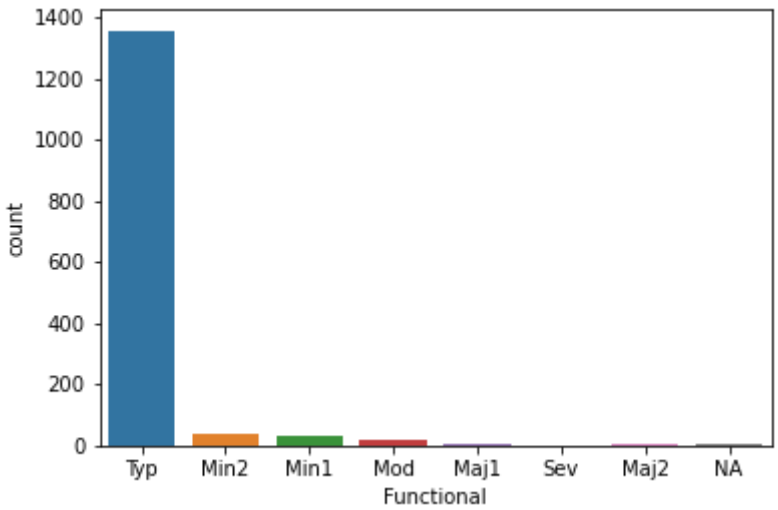


In [52]:

```
sns.countplot(df2.Functional)
```

Out[52]:

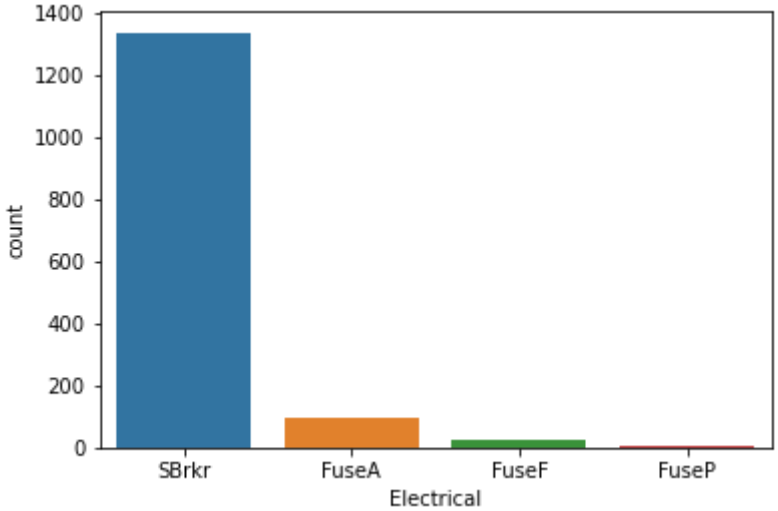
```
<AxesSubplot:xlabel='Functional', ylabel='count'>
```



for i in df2.columns: print('Attribute name:',i) print('----------') print(df2[i].value_counts()) print('----------')

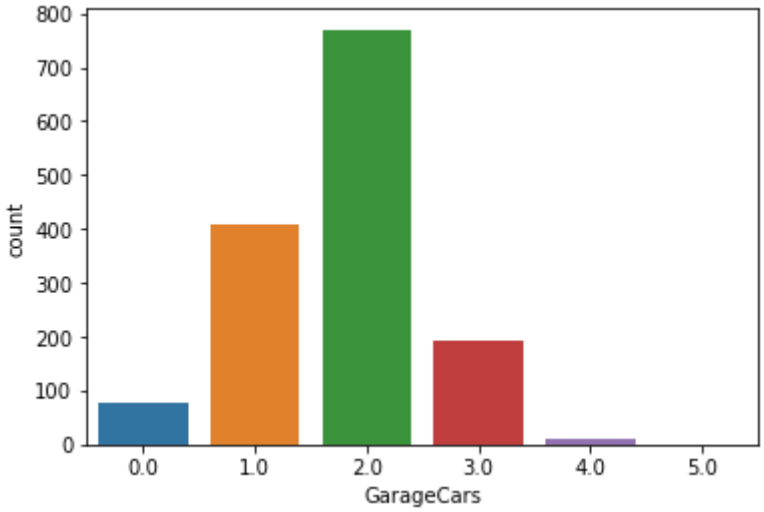
In [54]:
```
sns.countplot(x="Electrical", data=df2)
```

Out[54]: <AxesSubplot:xlabel='Electrical', ylabel='count'>



In [55]:
```
sns.countplot(x="GarageCars", data=df2)
```

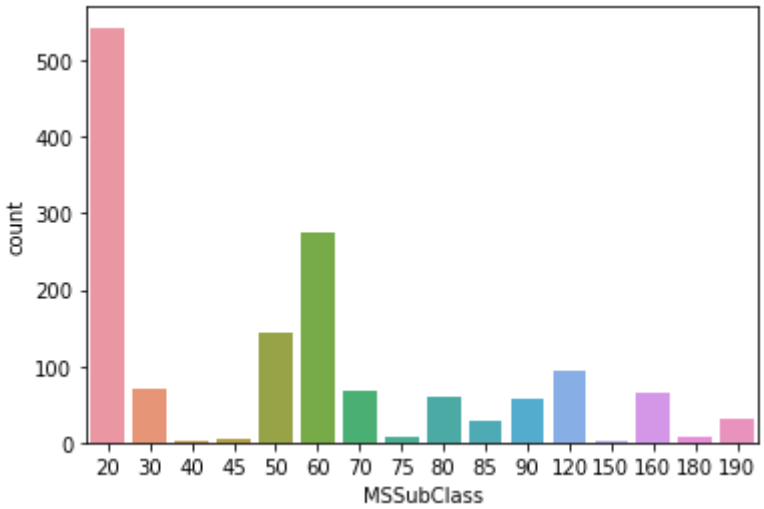
Out[55]: <AxesSubplot:xlabel='GarageCars', ylabel='count'>



In [56]:

```
sns.countplot(df2.MSSubClass)
```

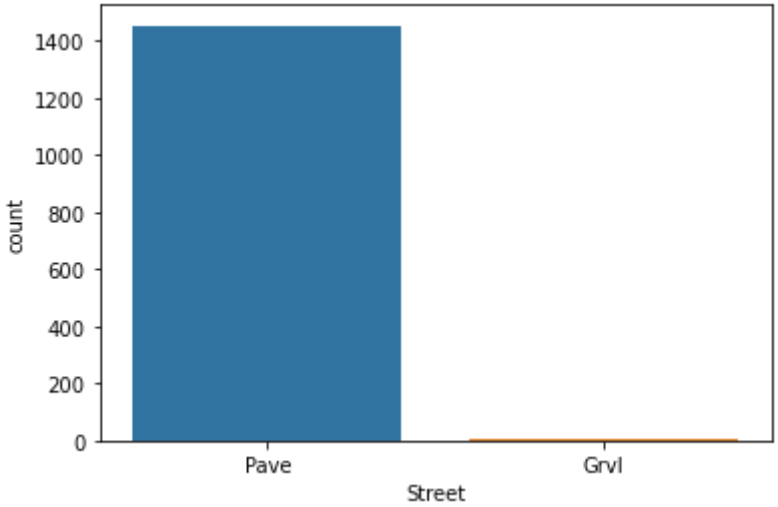
Out[56]: <AxesSubplot:xlabel='MSSubClass', ylabel='count'>



In [57]:

```
sns.countplot(df2.Street)
```

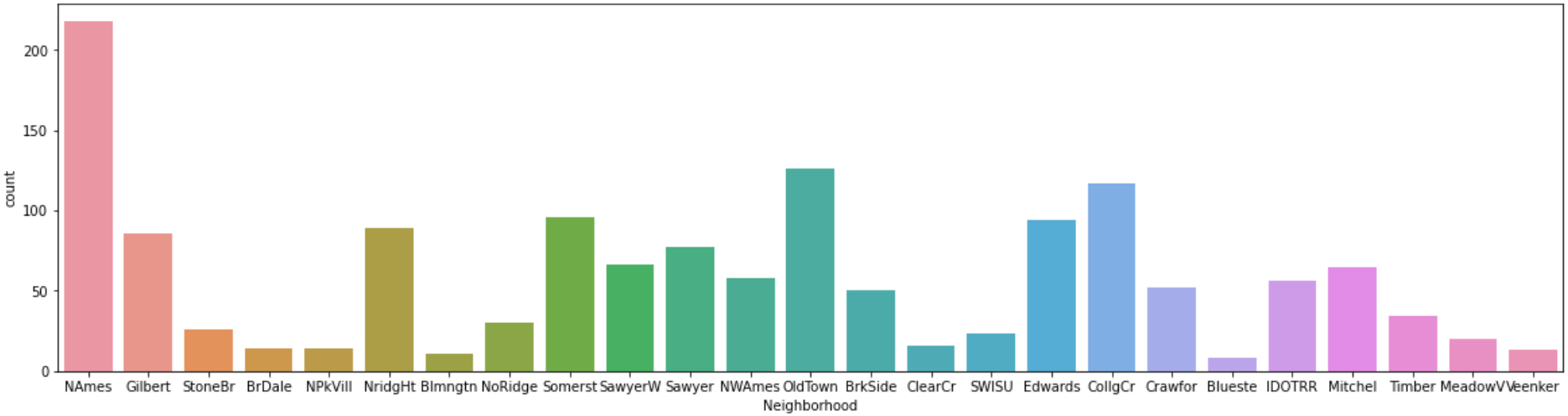
Out[57]: <AxesSubplot:xlabel='Street', ylabel='count'>



In [58]:

```
plt.figure(figsize=(20,5))
sns.countplot(df2.Neighborhood)
```

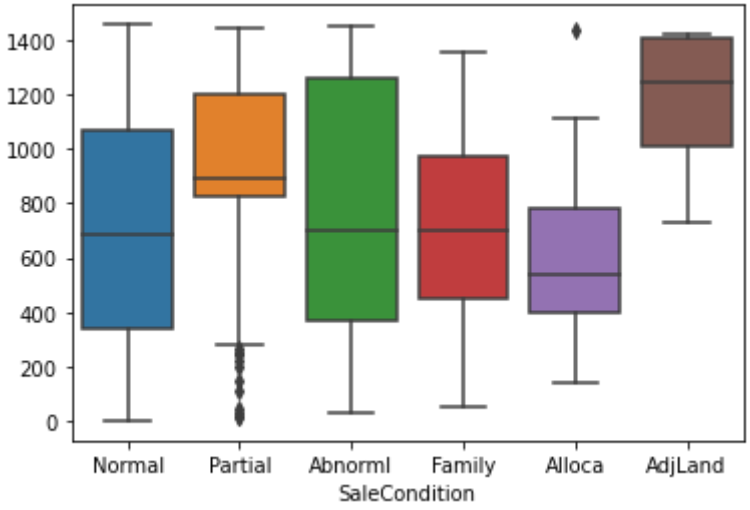
Out[58]: <AxesSubplot:xlabel='Neighborhood', ylabel='count'>



In [59]:

```
sns.boxplot(x=df2.SaleCondition,y=df2.index)
```

Out[59]: <AxesSubplot:xlabel='SaleCondition'>



sns.pairplot(df2,size=10)

In [60]:
```
df2.drop("Id",axis=1,inplace=True)
```

In [61]:
```
df2.drop(["YearRemodAdd","MasVnrArea","YrSold","GarageYrBlt","BsmtFinSF2","BsmtUnfSF","FireplaceQu"],axis=1,inplace=True)
```

In [62]:
```
df2.head()
```

Out[62]:

	MSSubClass	MSZoning	LotArea	Street	LotShape	LandContour	Utilities	LotConfig	LandSlope	Neighborhood	Condition1	Condition2	BldgType	H
0	20	RH	11622	Pave	Reg	Lvl	AllPub	Inside	Gtl	NAmes	Feedr	Norm	1Fam	
1	20	RL	14267	Pave	IR1	Lvl	AllPub	Corner	Gtl	NAmes	Norm	Norm	1Fam	
2	60	RL	13830	Pave	IR1	Lvl	AllPub	Inside	Gtl	Gilbert	Norm	Norm	1Fam	
3	60	RL	9978	Pave	IR1	Lvl	AllPub	Inside	Gtl	Gilbert	Norm	Norm	1Fam	
4	120	RL	5005	Pave	IR1	HLS	AllPub	Inside	Gtl	StoneBr	Norm	Norm	TwnhsE	

In [63]:
```
from sklearn import preprocessing
```

```
encoder = preprocessing.LabelEncoder()
for i in df2.columns:
    if isinstance(df2[i][0], str):
            df2[i] = encoder.fit_transform(df2[i])
```

In [64]:

```
df2
```

Out[64]:

	MSSubClass	MSZoning	LotArea	Street	LotShape	LandContour	Utilities	LotConfig	LandSlope	Neighborhood	Condition1	Condition2	BldgType
0	20	3	11622	1	3	3	0	4	0	12	1	2	(
1	20	4	14267	1	0	3	0	0	0	12	2	2	(
2	60	4	13830	1	0	3	0	4	0	8	2	2	(
3	60	4	9978	1	0	3	0	4	0	8	2	2	(
4	120	4	5005	1	0	1	0	4	0	22	2	2	4
...	...	...	...	...	...	...	...	...	...	...	...	...	..
1454	160	5	1936	1	3	3	0	4	0	10	2	2	3
1455	160	5	1894	1	3	3	0	4	0	10	2	2	4
1456	20	4	20000	1	3	3	0	4	0	11	2	2	(
1457	85	4	10441	1	3	3	0	4	0	11	2	2	(
1458	60	4	9627	1	3	3	0	4	1	11	2	2	(

1459 rows × 67 columns

In [65]:

```
x1=df2
```

In [66]:

```
#assing the value of predicted output from train dataset
new_y=reg.predict(x1)
```

In [67]:

```
#adding dependent variable in test dataset
```

```
df2["SalePrice"]=new_y
```

In [68]:

```
df2
```

Out[68]:

	MSSubClass	MSZoning	LotArea	Street	LotShape	LandContour	Utilities	LotConfig	LandSlope	Neighborhood	Condition1	Condition2	BldgType
0	20	3	11622	1	3	3	0	4	0	12	1	2	(
1	20	4	14267	1	0	3	0	0	0	12	2	2	(
2	60	4	13830	1	0	3	0	4	0	8	2	2	(
3	60	4	9978	1	0	3	0	4	0	8	2	2	(
4	120	4	5005	1	0	1	0	4	0	22	2	2	4
...	...	...	...	...	...	...	...	...	...	...	...	...	..
1454	160	5	1936	1	3	3	0	4	0	10	2	2	3
1455	160	5	1894	1	3	3	0	4	0	10	2	2	4
1456	20	4	20000	1	3	3	0	4	0	11	2	2	(
1457	85	4	10441	1	3	3	0	4	0	11	2	2	(
1458	60	4	9627	1	3	3	0	4	1	11	2	2	(

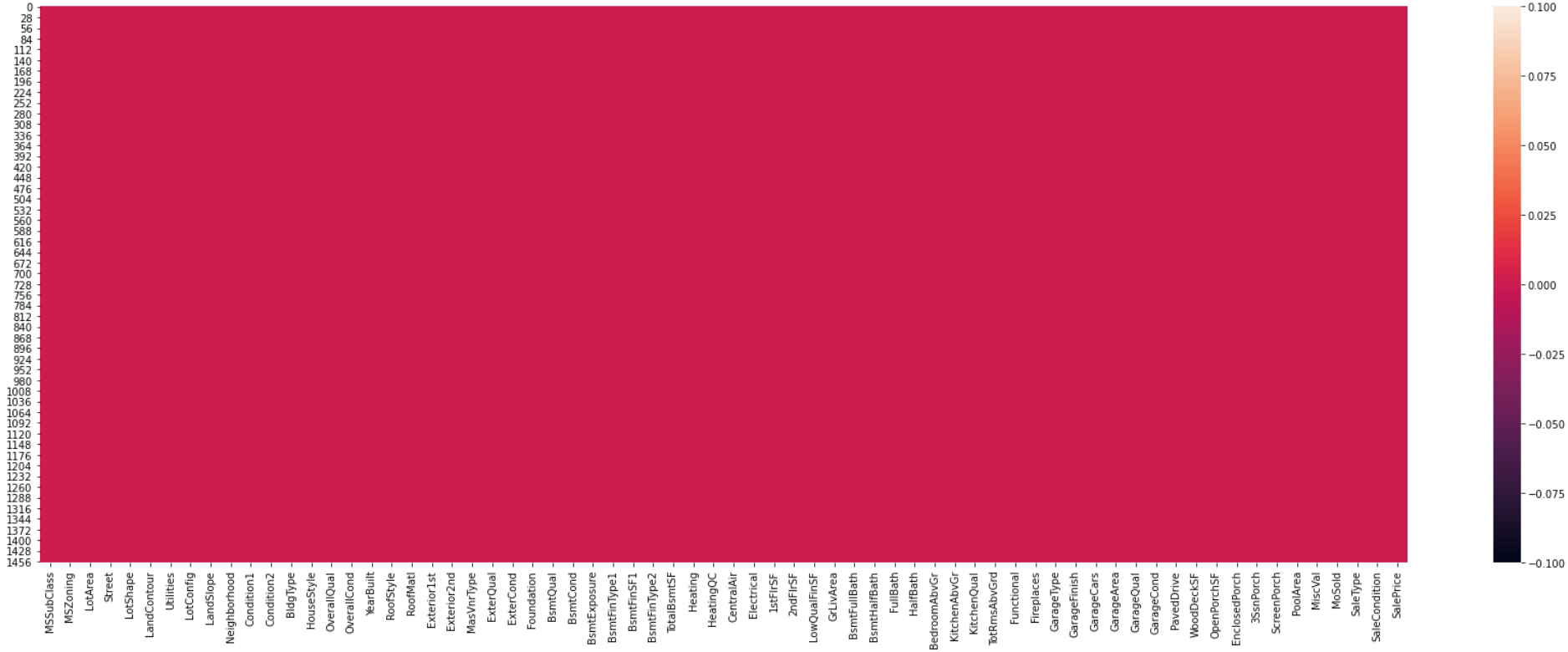
1459 rows × 68 columns

In [69]:

```
#checking null values
plt.figure(figsize=(30,10))
sns.heatmap(df2.isnull())
```

Out[69]:

```
<AxesSubplot:>
```



In [70]:

```
new_data=[df1,df2]
```

In [71]:

```
#merge both train and test dataset
new_data=pd.concat(new_data)
```

In [72]:

```
new_data
```

Out[72]:

	MSSubClass	MSZoning	LotArea	Street	LotShape	LandContour	Utilities	LotConfig	LandSlope	Neighborhood	Condition1	Condition2	BldgType
0	60	3	8450	1	3	3	0	4	0	5	2	2	(
1	20	3	9600	1	3	3	0	2	0	24	1	2	(
2	60	3	11250	1	0	3	0	4	0	5	2	2	(

	MSSubClass	MSZoning	LotArea	Street	LotShape	LandContour	Utilities	LotConfig	LandSlope	Neighborhood	Condition1	Condition2	BldgType
3	70	3	9550	1	0	3	0	0	0	6	2	2	(
4	60	3	14260	1	0	3	0	2	0	15	2	2	(
...	...	...	...	...	...	...	...	...	...	...	...	...	..
1454	160	5	1936	1	3	3	0	4	0	10	2	2	3
1455	160	5	1894	1	3	3	0	4	0	10	2	2	4
1456	20	4	20000	1	3	3	0	4	0	11	2	2	(
1457	85	4	10441	1	3	3	0	4	0	11	2	2	(
1458	60	4	9627	1	3	3	0	4	1	11	2	2	(

2919 rows × 68 columns

In [73]:

```
new_data.isnull().sum()
```

Out[73]:

```
MSSubClass       0
MSZoning         0
LotArea          0
Street           0
LotShape         0
                ..
MiscVal          0
MoSold           0
SaleType         0
SaleCondition    0
SalePrice        0
Length: 68, dtype: int64
```

In [74]:

```
X=new_data.drop("SalePrice",axis=1)
y=new_data["SalePrice"]
```

Training and Modeling

In [75]:
```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=20)
```

In [76]:
```
from sklearn.linear_model import LinearRegression
lin_model=LinearRegression()
lin_model.fit(X_train,y_train)
lin_model.score(X_test,y_test)
```

Out[76]:
```
0.8804259070082157
```

In [77]:
```
from sklearn import ensemble
reg = ensemble.GradientBoostingRegressor(learning_rate = 0.1, n_estimators=1000)
reg.fit(X_train, y_train)
reg.score(X_test, y_test)
```

Out[77]:
```
0.9442042559426979
```

In [78]:
```
pred=lin_model.predict(X_test)
```

In [79]:
```
pred[:5]
```

Out[79]:
```
array([125255.76980313, 191803.55414176, 115132.88487725, 182525.0665402 ,
       107398.84632386])
```

In [80]:
```
pred_reg=reg.predict(X_test)
```

In [81]:
```
pred_reg[:5]
```

Out[81]:
```
array([126023.94492582, 187048.14678974, 125491.0126407 , 192018.10466226,
       113948.5209683 ])
```