Import Data

In [1]:

```
from collections import Counter

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

from sklearn import preprocessing
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
```

In [2]:

```
# Download data
df_house_price_sample=pd.read_csv('house_price_sample_submission.csv')
df_house_price_test=pd.read_csv('house_price_test.csv')
df_house_price_train=pd.read_csv('house_price_train.csv')
df_house_price_train.drop(columns='Id',inplace=True)
df_house_price_test.drop(columns='Id',inplace=True)
df_house_price_sample.drop(columns='Id',inplace=True)
df_house_price_train.head() # X_train, Y_train - Split for training - by 'SalePrice'
```

Out[2]:

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

5 rows × 80 columns

In [3]:

```
df_house_price_test.head() # X_test
```

Out[3]:

	MSSubClass	MSZoning	LotFrontage	LotArea	Street	Alley	LotShape	LandContour	Utilities	LotConfig	...	ScreenPorch	PoolAre
0	20	RH	80.0	11622	Pave	NaN	Reg	Lvl	AllPub	Inside	...	120	
1	20	RL	81.0	14267	Pave	NaN	IR1	Lvl	AllPub	Corner	...	0	
2	60	RL	74.0	13830	Pave	NaN	IR1	Lvl	AllPub	Inside	...	0	
3	60	RL	78.0	9978	Pave	NaN	IR1	Lvl	AllPub	Inside	...	0	
4	120	RL	43.0	5005	Pave	NaN	IR1	HLS	AllPub	Inside	...	144	

5 rows × 79 columns

In [4]:

```
df_house_price_sample.head() # Y_test
```

Out[4]:

	SalePrice
0	169277.052498
1	187758.393989
2	183583.683570
3	179317.477511
4	150730.079977

In [5]:

```
X=df_house_price_train.drop(columns='SalePrice')
Y=df_house_price_train['SalePrice']
```

Exploratory Data Analysis

Initial Data Exploration:

Objects

In [6]:

```
# Object Dataframe Preview
df_house_price_train_objects=df_house_price_train.select_dtypes(include=['object'])
df_house_price_train_objects.head()
```

Out[6]:

	MSZoning	Street	Alley	LotShape	LandContour	Utilities	LotConfig	LandSlope	Neighborhood	Condition1	...	GarageType	Garag
0	RL	Pave	NaN	Reg	Lvl	AllPub	Inside	Gtl	CollgCr	Norm	...	Attchd	
1	RL	Pave	NaN	Reg	Lvl	AllPub	FR2	Gtl	Veenker	Feedr	...	Attchd	
2	RL	Pave	NaN	IR1	Lvl	AllPub	Inside	Gtl	CollgCr	Norm	...	Attchd	
3	RL	Pave	NaN	IR1	Lvl	AllPub	Corner	Gtl	Crawfor	Norm	...	Detchd	
4	RL	Pave	NaN	IR1	Lvl	AllPub	FR2	Gtl	NoRidge	Norm	...	Attchd	

5 rows × 43 columns

In [7]:

```
df_house_price_train_objects.describe()
```

Out[7]:

	MSZoning	Street	Alley	LotShape	LandContour	Utilities	LotConfig	LandSlope	Neighborhood	Condition1	...	GarageType
count	1460	1460	91	1460	1460	1460	1460	1460	1460	1460	...	1379
unique	5	2	2	4	4	2	5	3	25	9	...	6
top	RL	Pave	Grvl	Reg	Lvl	AllPub	Inside	Gtl	NAmes	Norm	...	Attchd
freq	1151	1454	50	925	1311	1459	1052	1382	225	1260	...	870

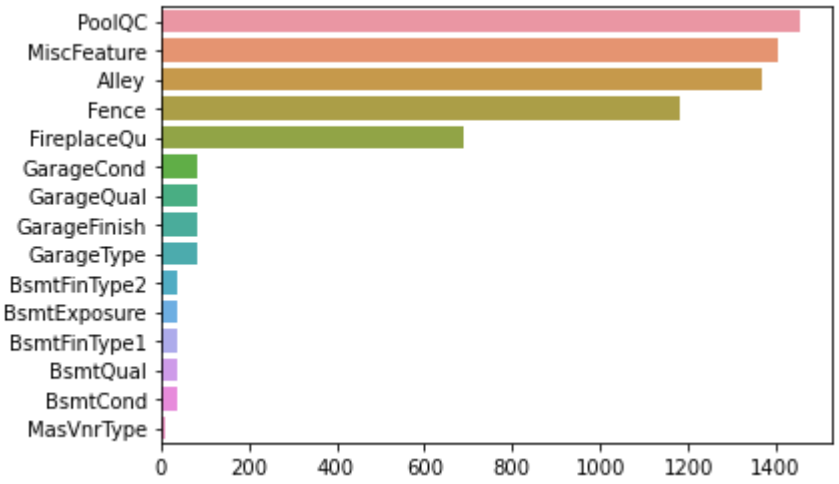
4 rows × 43 columns

In [8]:

```
# Bar chart of missing values for objects dataset
df_house_price_train_objects_missing_data=df_house_price_train_objects.isnull().sum().sort_values(ascending=False).head(15)
sns.barplot(y=df_house_price_train_objects_missing_data.index,x=df_house_price_train_objects_missing_data)
```

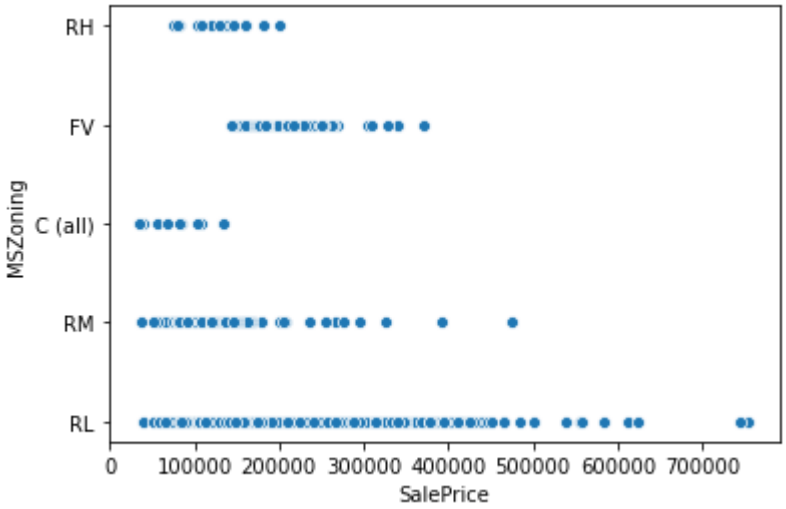
Out[8]:

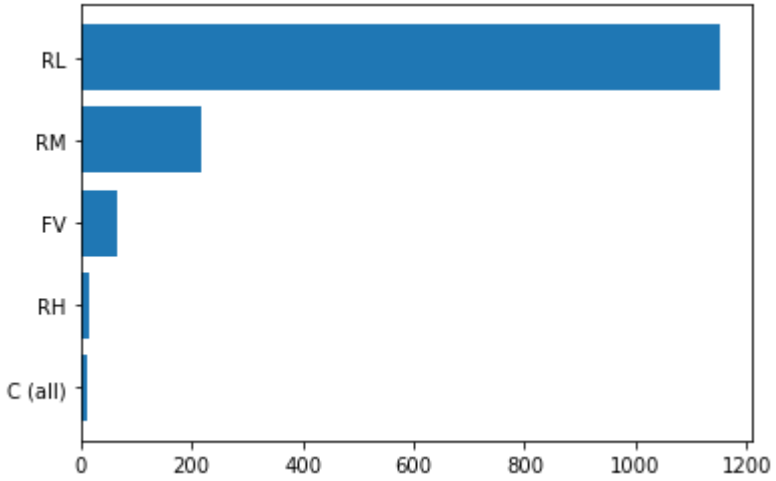
```
<matplotlib.axes._subplots.AxesSubplot at 0x179d32fc4f0>
```

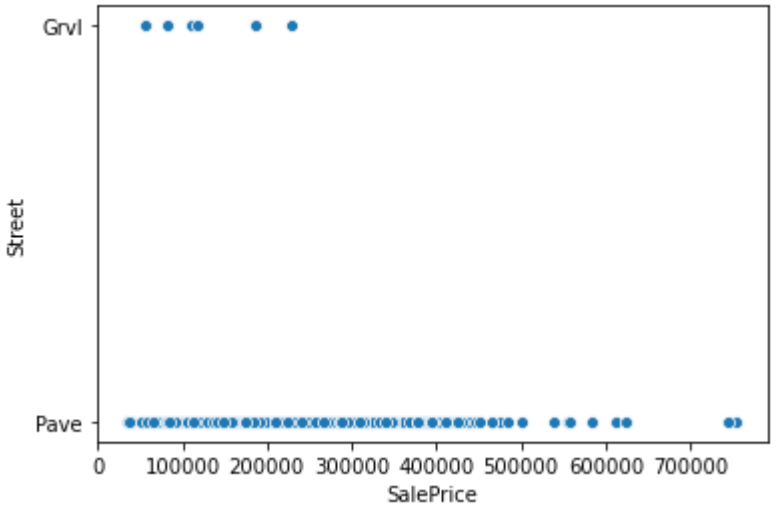


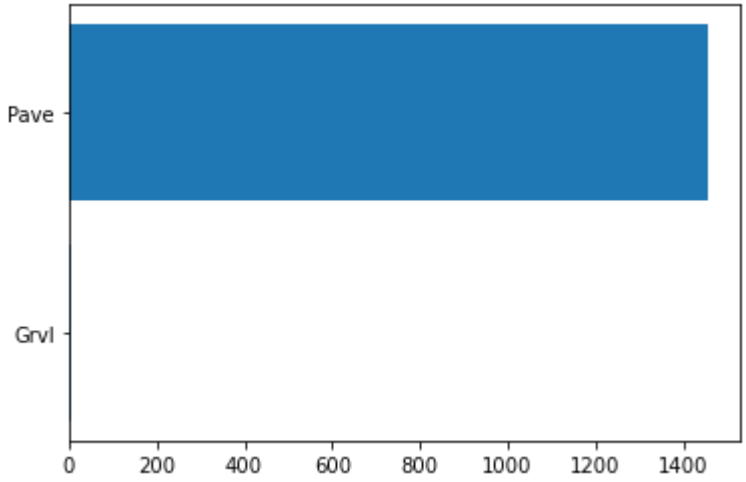
In [9]:

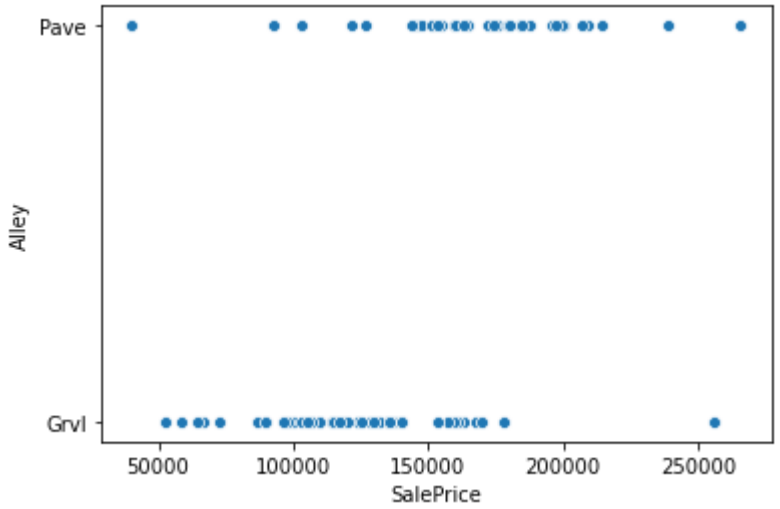
```
# Scatterplot of Object Dataframe for visualization of categorical distribution
for i in range(len(df_house_price_train_objects.columns)):
    sns.scatterplot(y=df_house_price_train_objects.columns[i],x=df_house_price_train['SalePrice'],data=df_house_price_tra
in_objects)
    plt.show()
    df_house_price_train_object_sorted=df_house_price_train_objects[df_house_price_train_objects.columns[i]].value_counts
(ascending=True)
    y_pos = np.arange(len(df_house_price_train_object_sorted))
    plt.barh(y_pos,df_house_price_train_object_sorted,tick_label=df_house_price_train_object_sorted.index)
    plt.show()
```

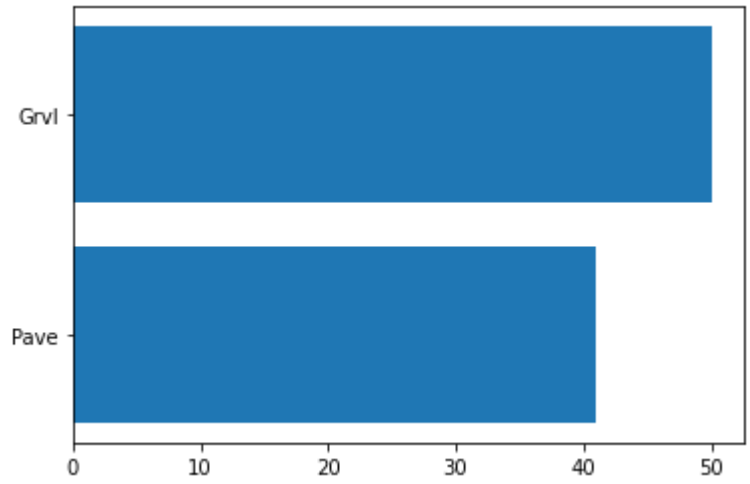

RH
FV
C (all)
RM
RL
MSZoning
0
100000
200000
300000
400000
500000
600000
700000
SalePrice

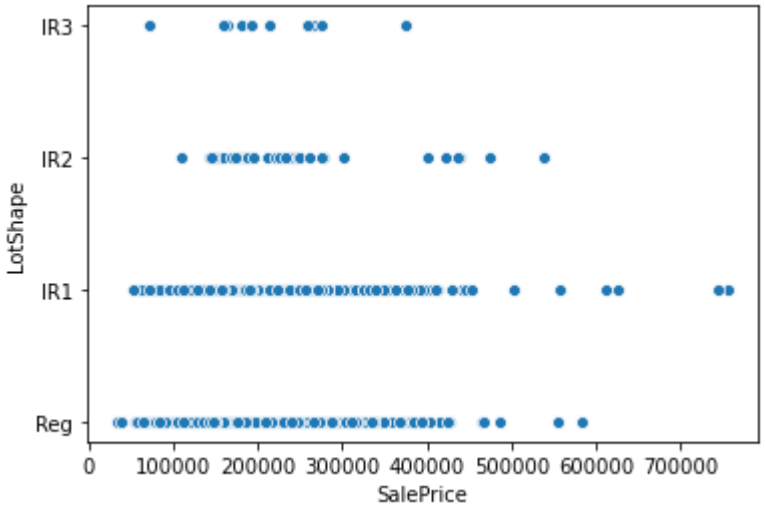

RL
RM
FV
RH
C (all)
0
200
400
600
800
1000
1200

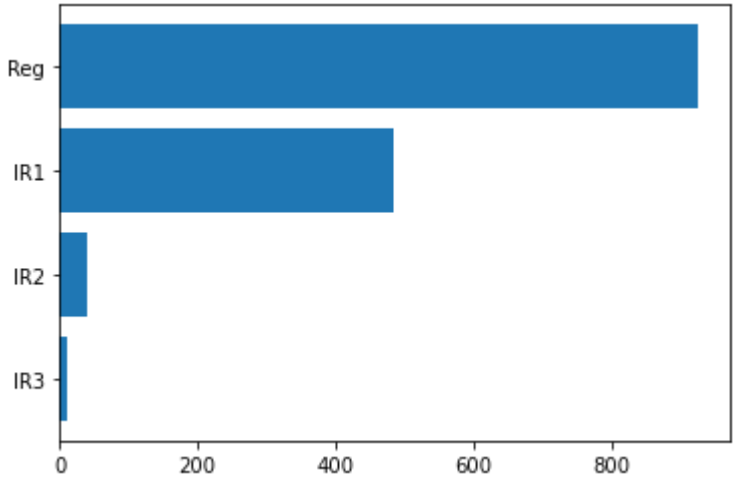

Grvl
Pave
Street
0
100000
200000
300000
400000
500000
600000
700000
SalePrice

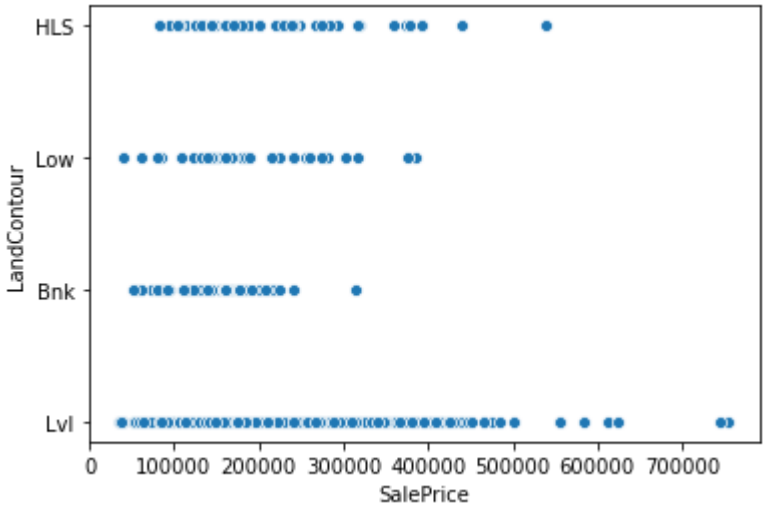

Pave
Grvl
0
200
400
600
800
1000
1200
1400

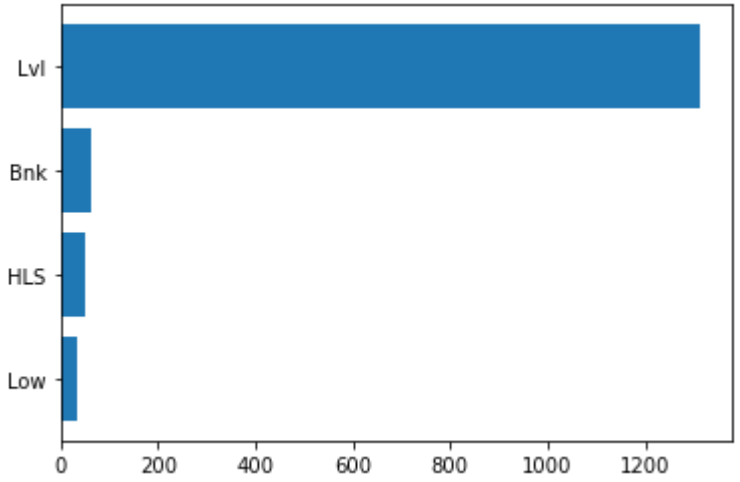

Pave
Grvl
Alley
50000
100000
150000
200000
250000
SalePrice

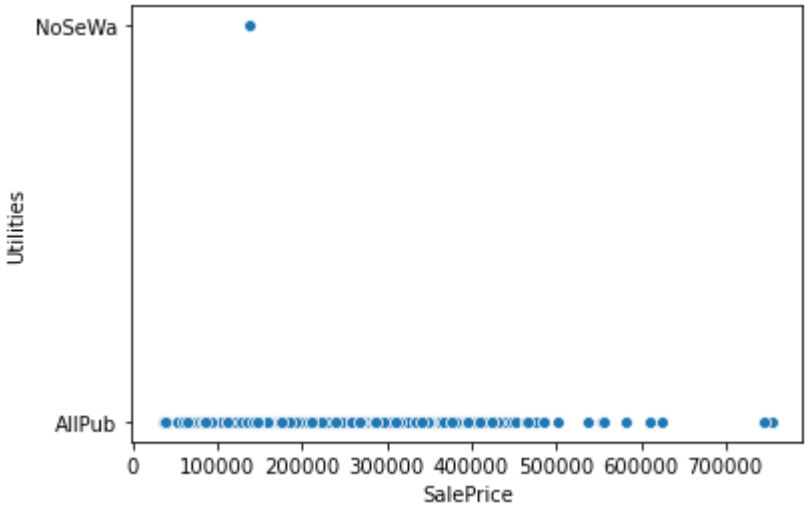

Grvl
Pave
0
10
20
30
40
50

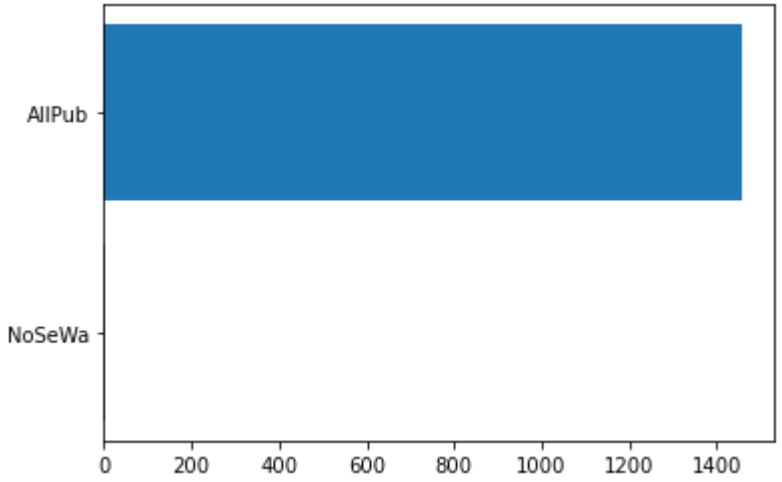

IR3
IR2
IR1
Reg
LotShape
0
100000
200000
300000
400000
500000
600000
700000
SalePrice

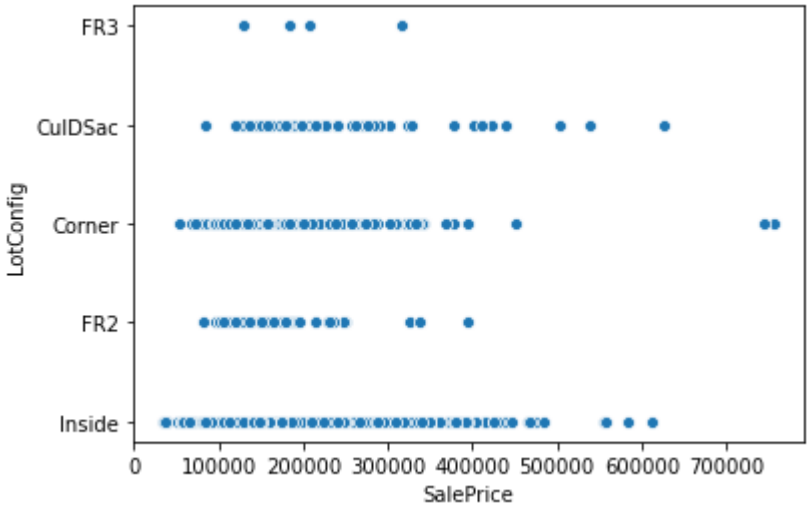

Reg
IR1
IR2
IR3
0
200
400
600
800

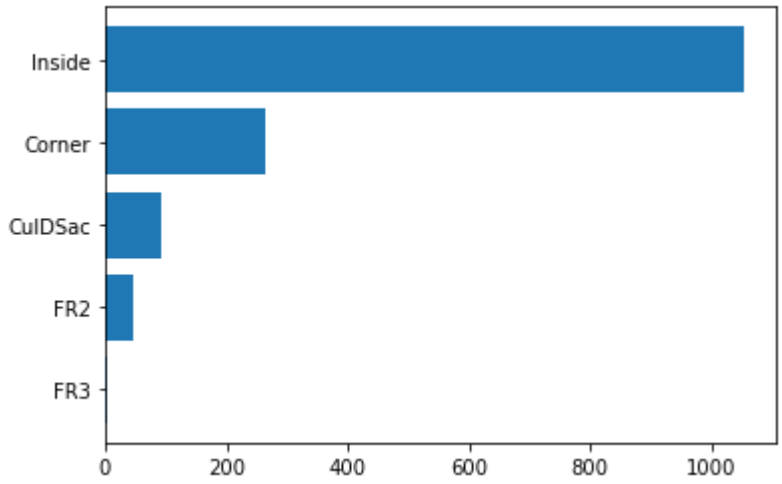

HLS
Low
Bnk
Lvl
LandContour
0
100000
200000
300000
400000
500000
600000
700000
SalePrice

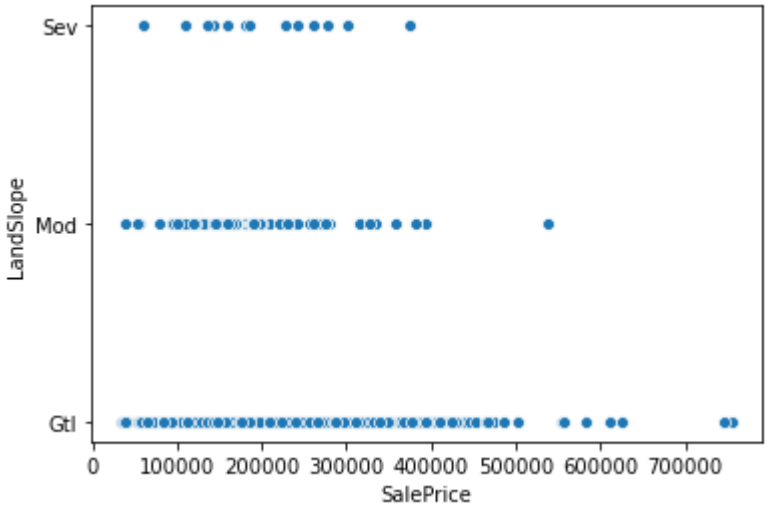

Lvl
Bnk
HLS
Low
0
200
400
600
800
1000
1200

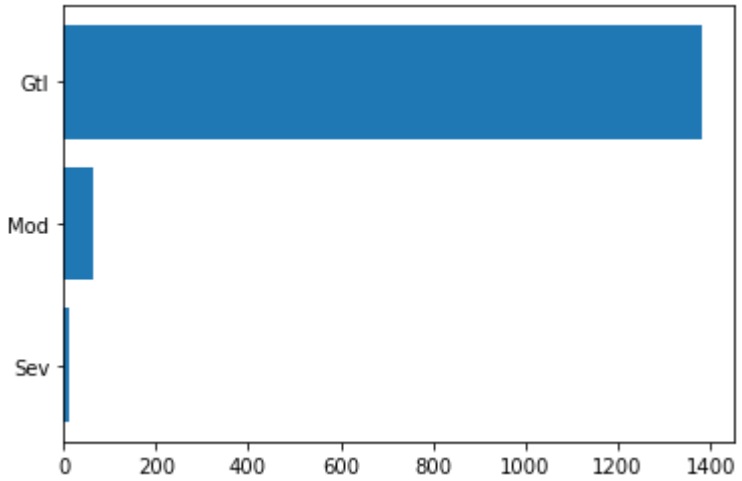

NoSeWa
AllPub
Utilities
0
100000
200000
300000
400000
500000
600000
700000
SalePrice

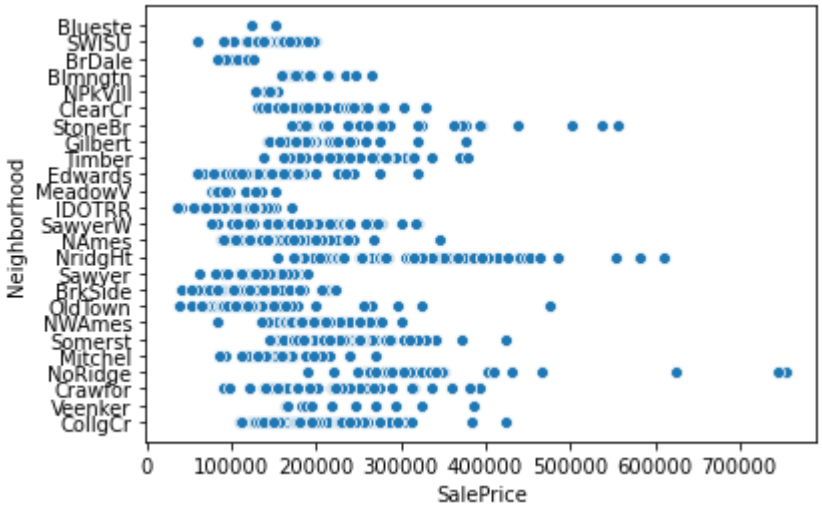

AllPub
NoSeWa
0
200
400
600
800
1000
1200
1400

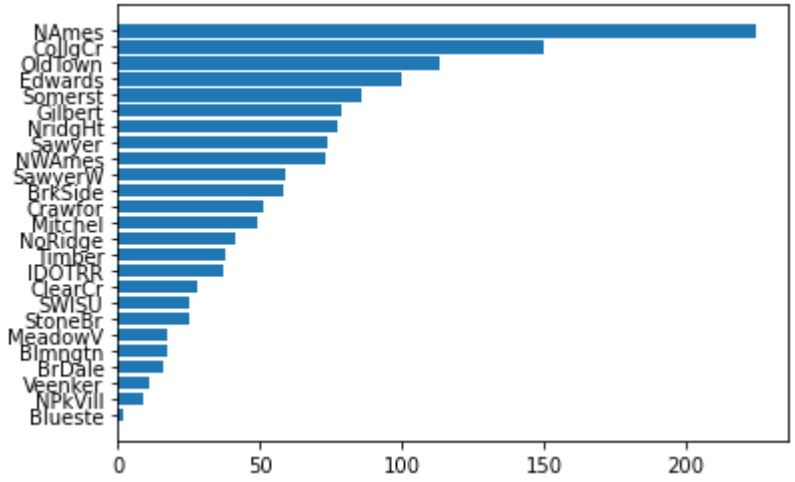

FR3
CulDSac
Corner
FR2
Inside
LotConfig
0
100000
200000
300000
400000
500000
600000
700000
SalePrice

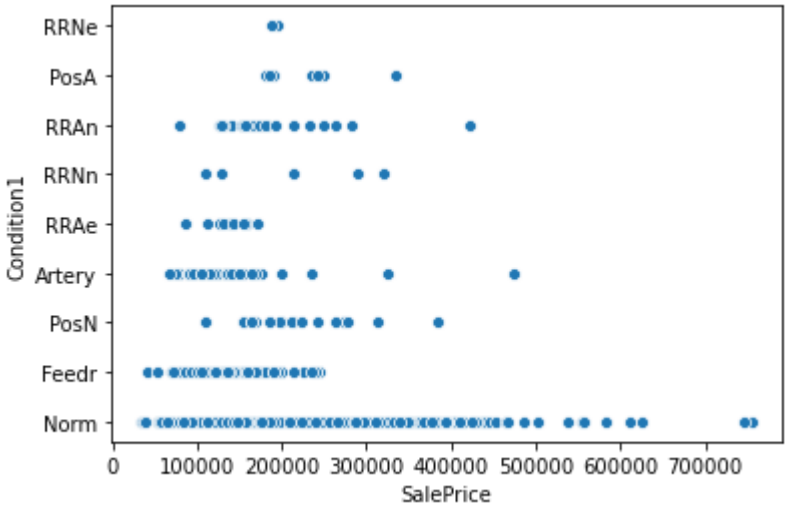

Inside
Corner
CulDSac
FR2
FR3
0
200
400
600
800
1000

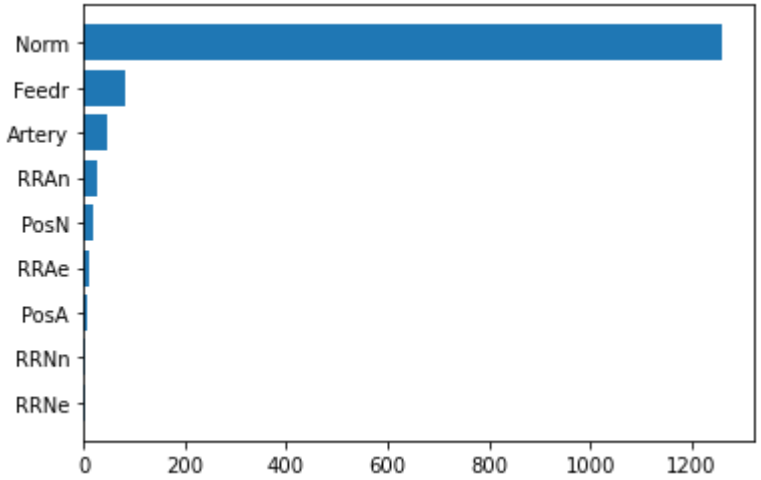

Sev
Mod
Gtl
LandSlope
0
100000
200000
300000
400000
500000
600000
700000
SalePrice

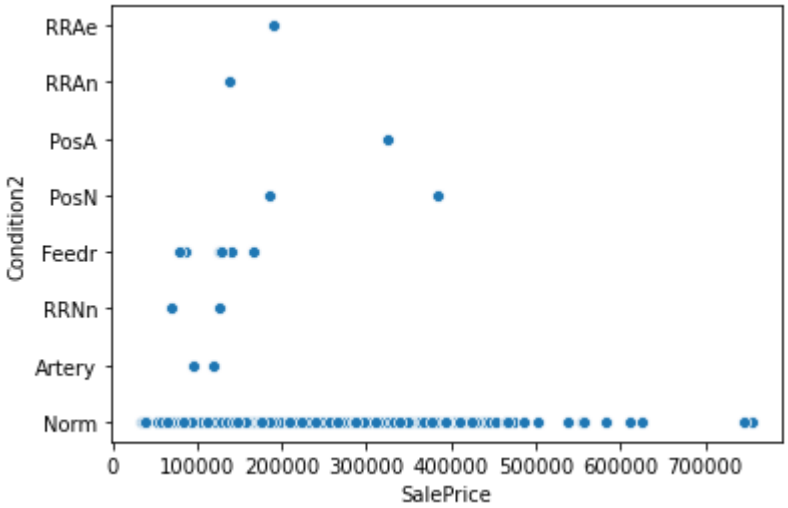

Gtl
Mod
Sev
0
200
400
600
800
1000
1200
1400

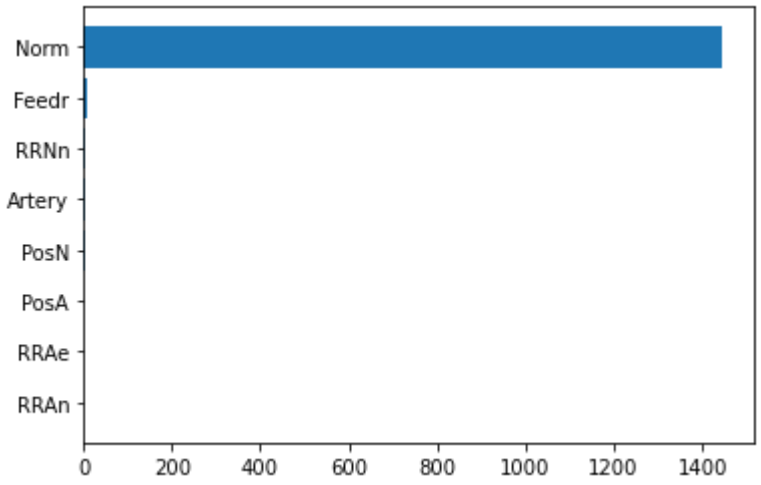

Blueste
SWISU
BrDale
Blmngtn
NPkVill
ClearCr
StoneBr
Gilbert
Timber
Edwards
MeadowV
IDOTRR
SawyerW
NAmes
NridgHt
Sawyer
BrkSide
OldTown
NWAmes
Somerst
Mitchel
NoRidge
Crawfor
Veenker
CollgCr
Neighborhood
0
100000
200000
300000
400000
500000
600000
700000
SalePrice

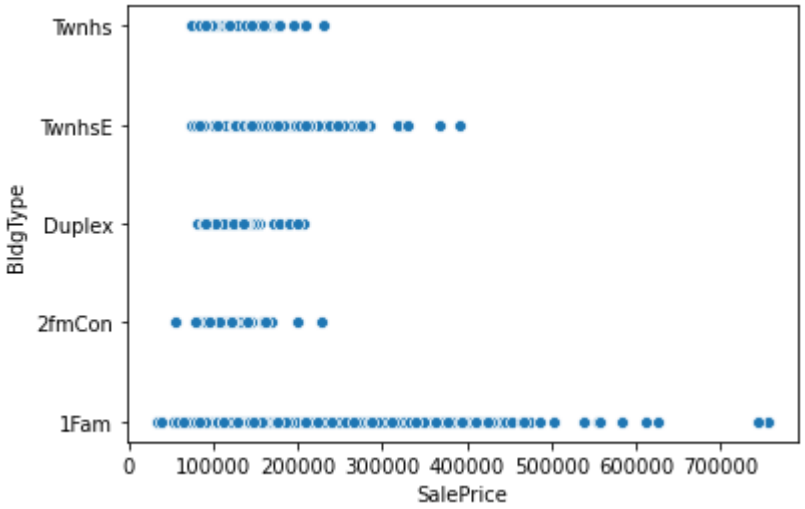

NAmes
CollgCr
OldTown
Edwards
Somerst
Gilbert
NridgHt
Sawyer
NWAmes
SawyerW
BrkSide
Crawfor
Mitchel
NoRidge
Timber
IDOTRR
ClearCr
SWISU
StoneBr
MeadowV
Blmngtn
BrDale
Veenker
NPkVill
Blueste
0
50
100
150
200

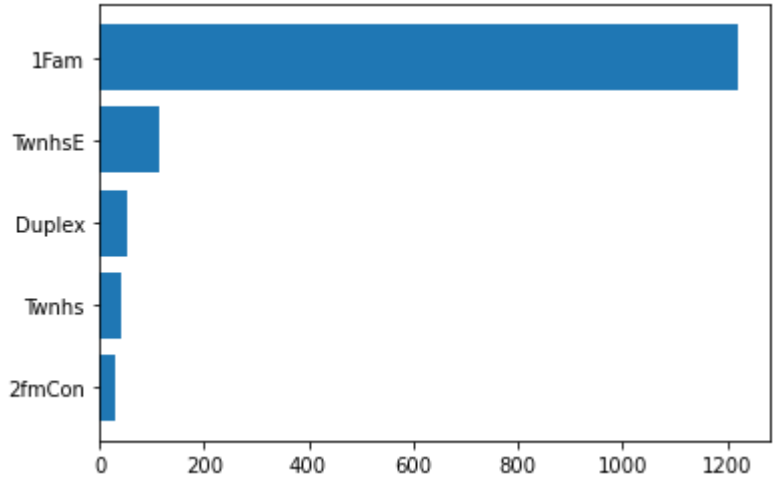

RRNe
PosA
RRAn
RRNn
RRAe
Artery
PosN
Feedr
Norm
Condition1
0
100000
200000
300000
400000
500000
600000
700000
SalePrice

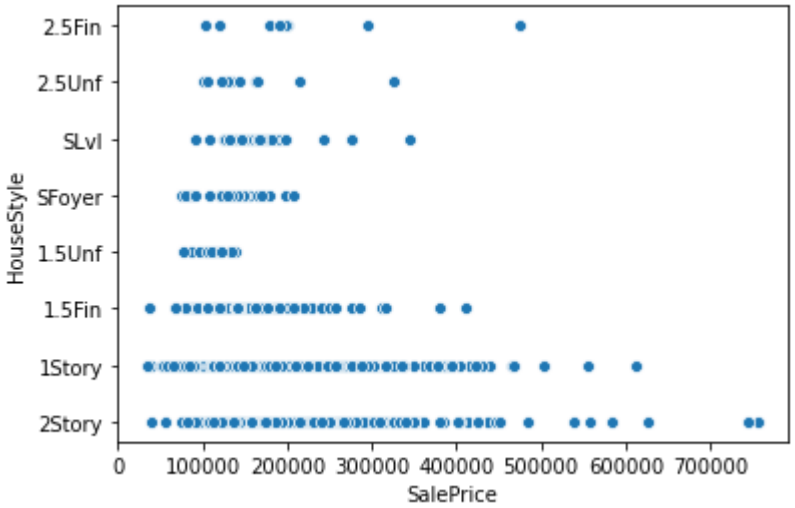

Norm
Feedr
Artery
RRAn
PosN
RRAe
PosA
RRNn
RRNe
0
200
400
600
800
1000
1200

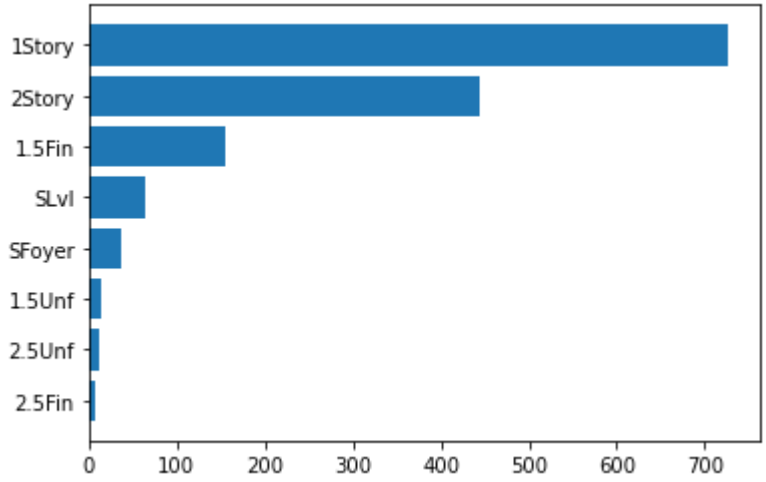

RRAe
RRAn
PosA
PosN
Feedr
RRNn
Artery
Norm
Condition2
0
100000
200000
300000
400000
500000
600000
700000
SalePrice

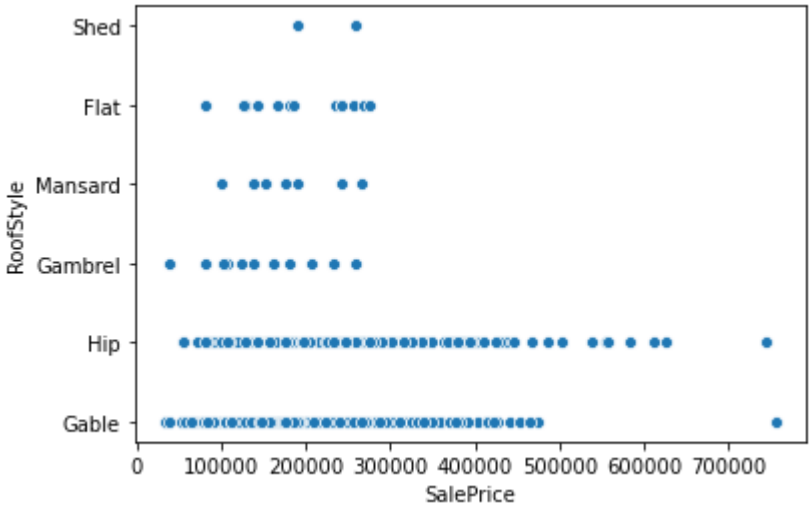

Norm
Feedr
RRNn
Artery
PosN
PosA
RRAe
RRAn
0
200
400
600
800
1000
1200
1400

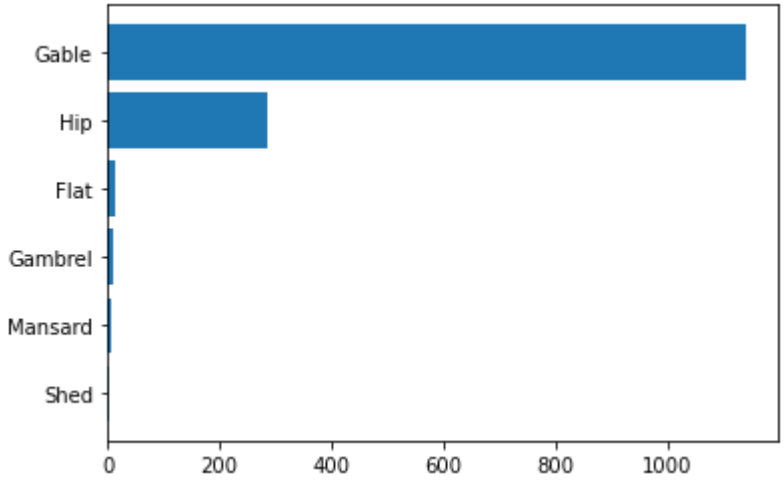

Twnhs
TwnhsE
Duplex
2fmCon
1Fam
BldgType
0
100000
200000
300000
400000
500000
600000
700000
SalePrice

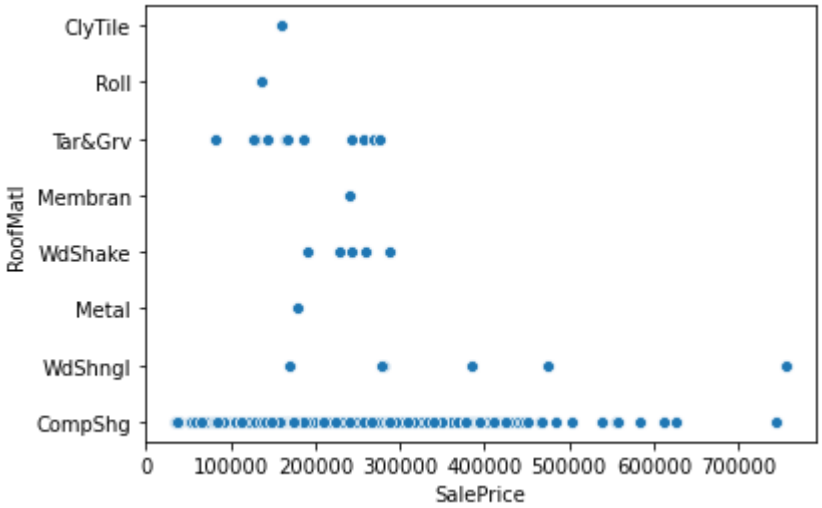

1Fam
TwnhsE
Duplex
Twnhs
2fmCon
0
200
400
600
800
1000
1200

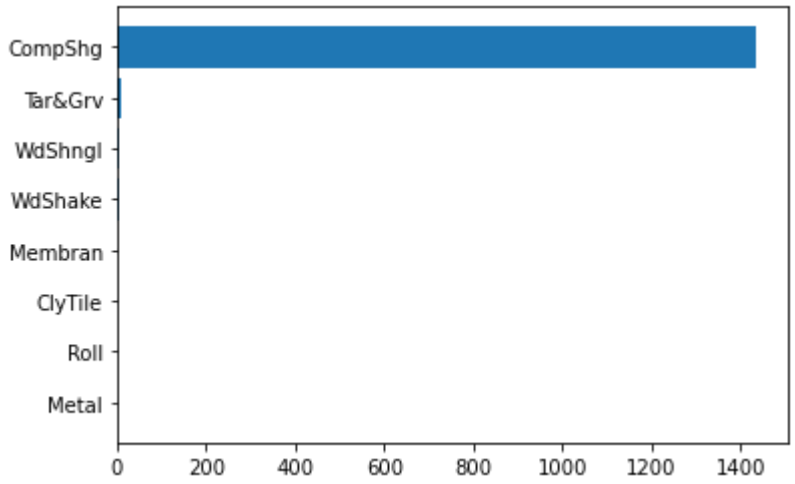

2.5Fin
2.5Unf
SLvl
SFoyer
1.5Unf
1.5Fin
1Story
2Story
HouseStyle
0
100000
200000
300000
400000
500000
600000
700000
SalePrice

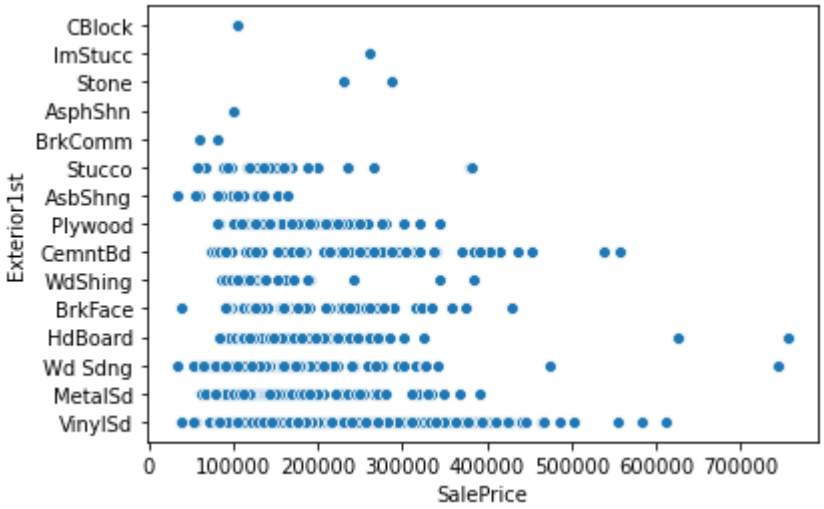

1Story
2Story
1.5Fin
SLvl
SFoyer
1.5Unf
2.5Unf
2.5Fin
0
100
200
300
400
500
600
700

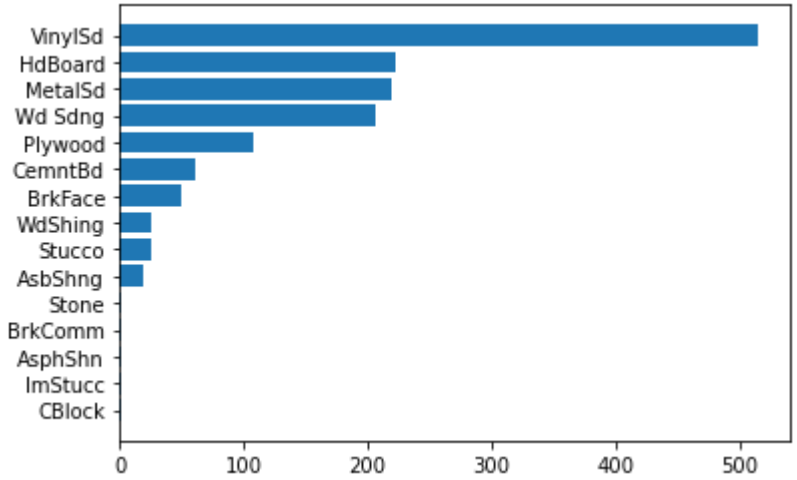

Shed
Flat
Mansard
Gambrel
Hip
Gable
RoofStyle
0
100000
200000
300000
400000
500000
600000
700000
SalePrice

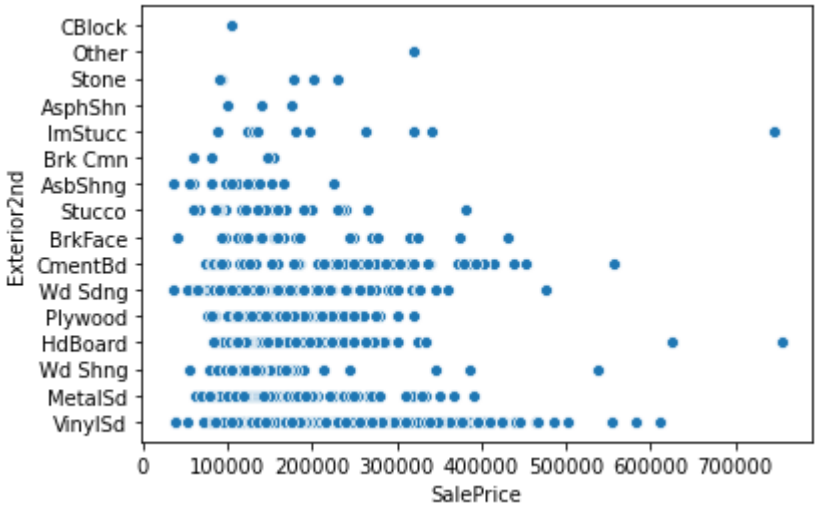

Gable
Hip
Flat
Gambrel
Mansard
Shed
0
200
400
600
800
1000

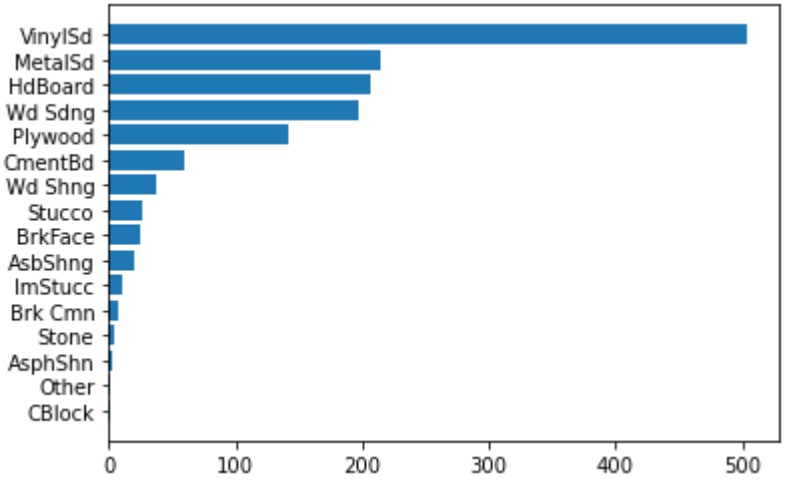

ClyTile
Roll
Tar&Grv
Membran
WdShake
Metal
WdShngl
CompShg
RoofMatl
0
100000
200000
300000
400000
500000
600000
700000
SalePrice

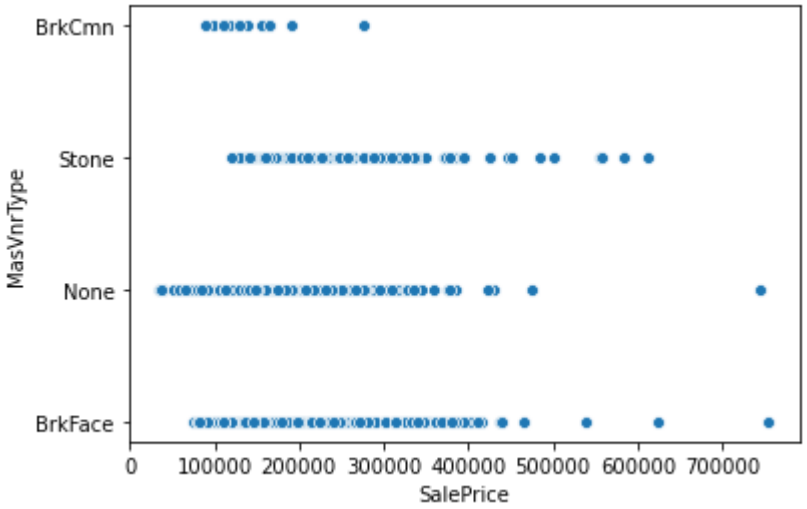

CompShg
Tar&Grv
WdShngl
WdShake
Membran
ClyTile
Roll
Metal
0
200
400
600
800
1000
1200
1400

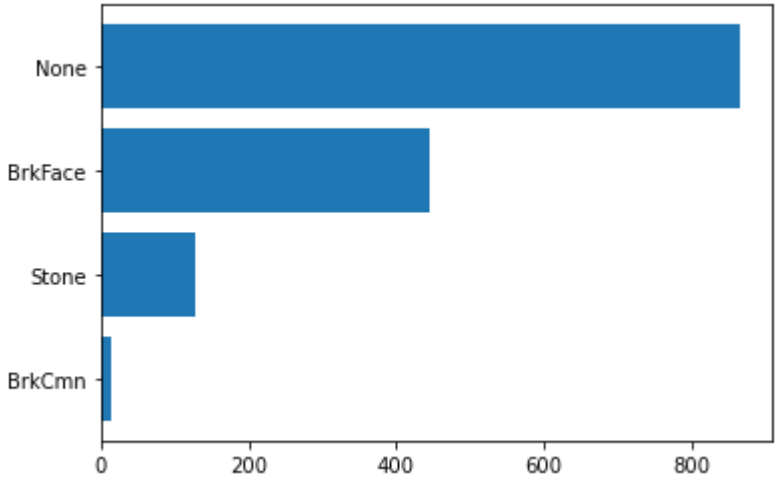

CBlock
ImStucc
Stone
AsphShn
BrkComm
Stucco
AsbShng
Plywood
CemntBd
WdShing
BrkFace
HdBoard
Wd Sdng
MetalSd
VinylSd
Exterior1st
0
100000
200000
300000
400000
500000
600000
700000
SalePrice

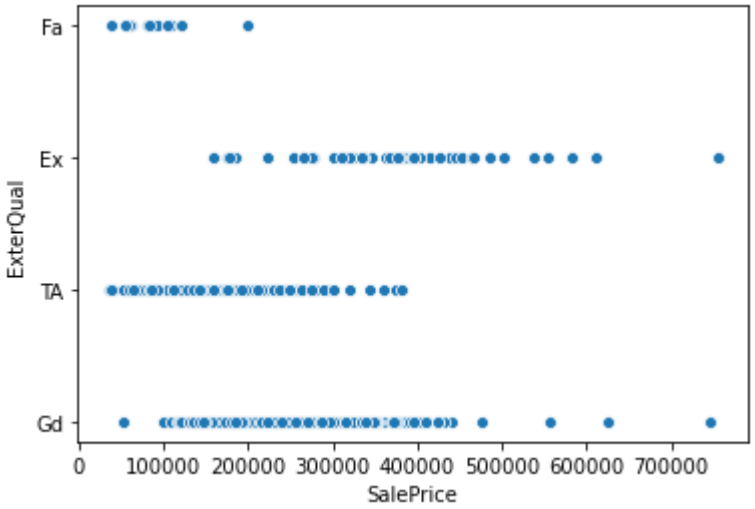

VinylSd
HdBoard
MetalSd
Wd Sdng
Plywood
CemntBd
BrkFace
WdShing
Stucco
AsbShng
Stone
BrkComm
AsphShn
ImStucc
CBlock
0
100
200
300
400
500

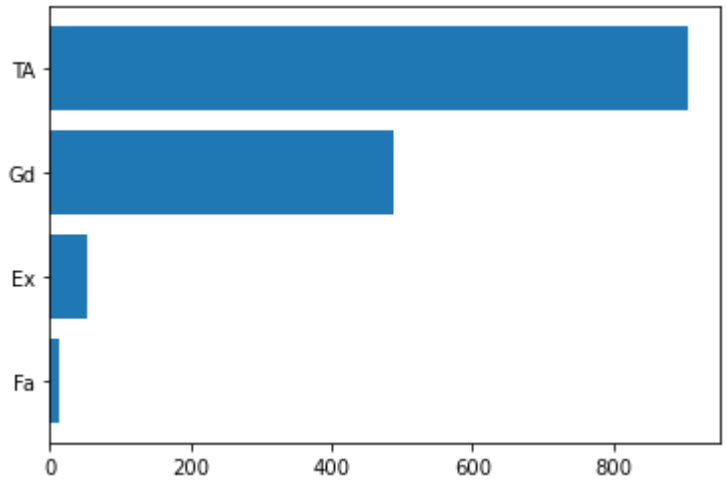

CBlock
Other
Stone
AsphShn
ImStucc
Brk Cmn
AsbShng
Stucco
BrkFace
CmentBd
Wd Sdng
Plywood
HdBoard
Wd Shng
MetalSd
VinylSd
Exterior2nd
0
100000
200000
300000
400000
500000
600000
700000
SalePrice

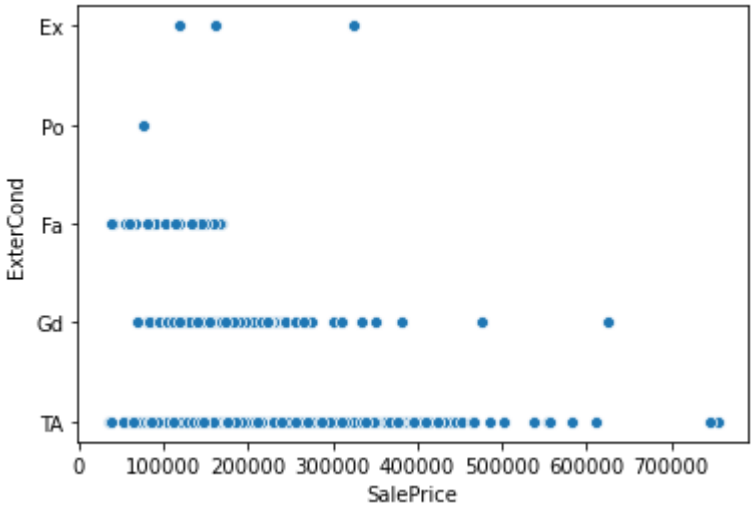

VinylSd
MetalSd
HdBoard
Wd Sdng
Plywood
CmentBd
Wd Shng
Stucco
BrkFace
AsbShng
ImStucc
Brk Cmn
Stone
AsphShn
Other
CBlock
0
100
200
300
400
500

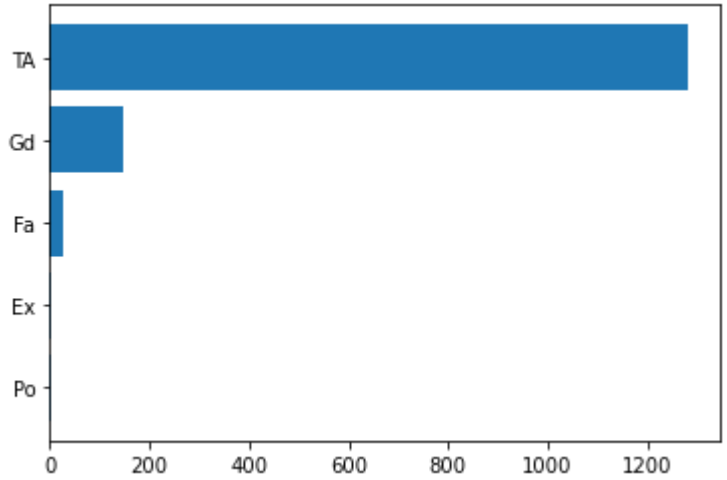

BrkCmn
Stone
None
BrkFace
MasVnrType
0
100000
200000
300000
400000
500000
600000
700000
SalePrice

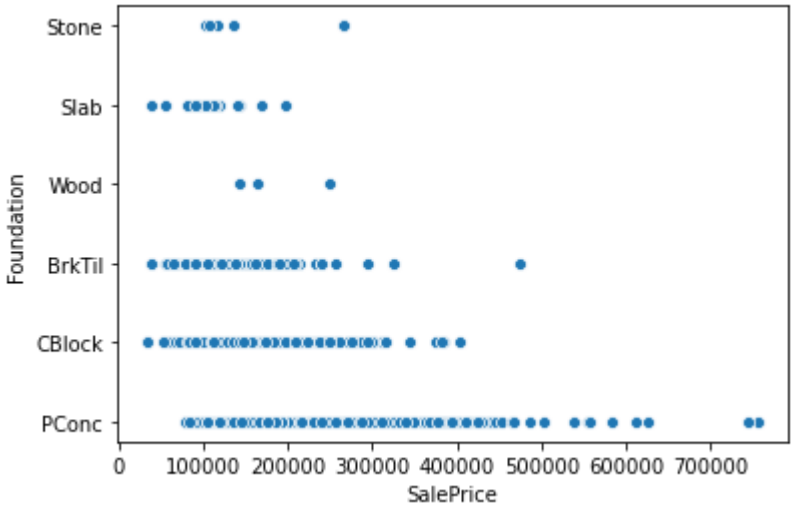

None
BrkFace
Stone
BrkCmn
0
200
400
600
800

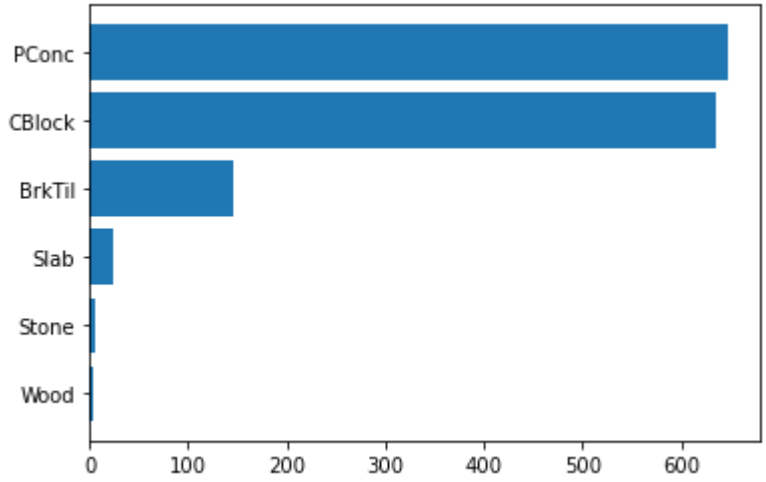

Fa
Ex
TA
Gd
ExterQual
0
100000
200000
300000
400000
500000
600000
700000
SalePrice

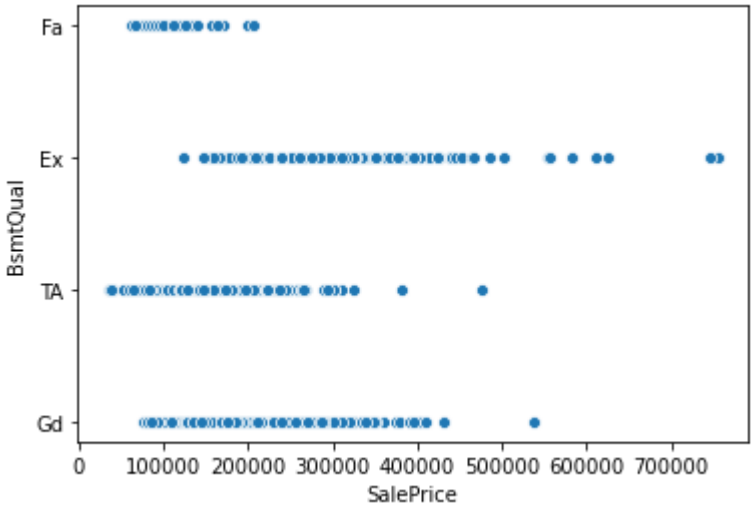

TA
Gd
Ex
Fa
0
200
400
600
800

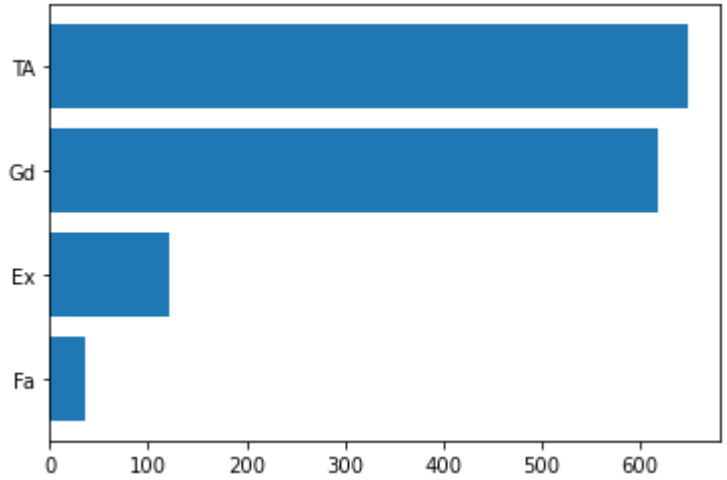

Ex
Po
Fa
Gd
TA
ExterCond
0
100000
200000
300000
400000
500000
600000
700000
SalePrice

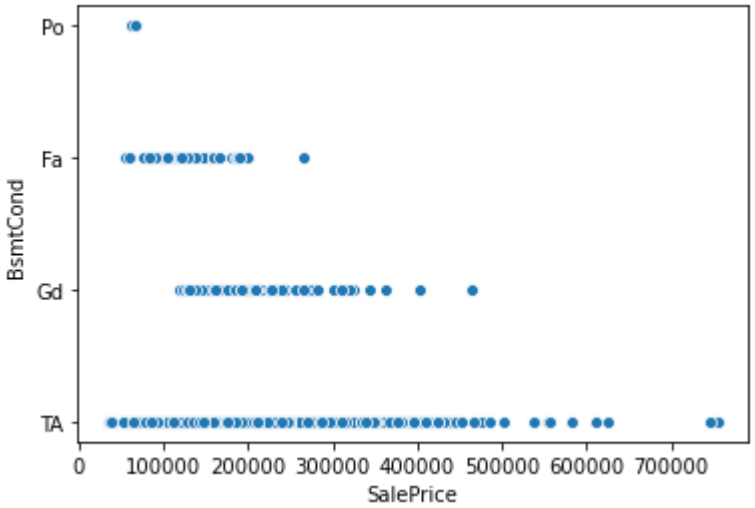

TA
Gd
Fa
Ex
Po
0
200
400
600
800
1000
1200

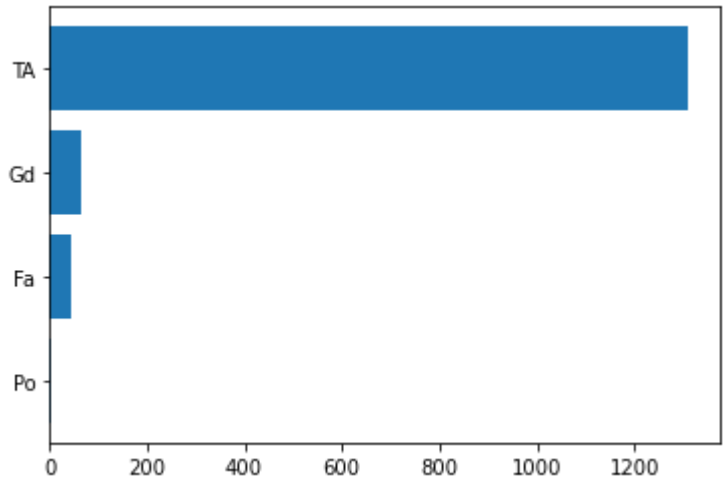

Stone
Slab
Wood
BrkTil
CBlock
PConc
Foundation
0
100000
200000
300000
400000
500000
600000
700000
SalePrice

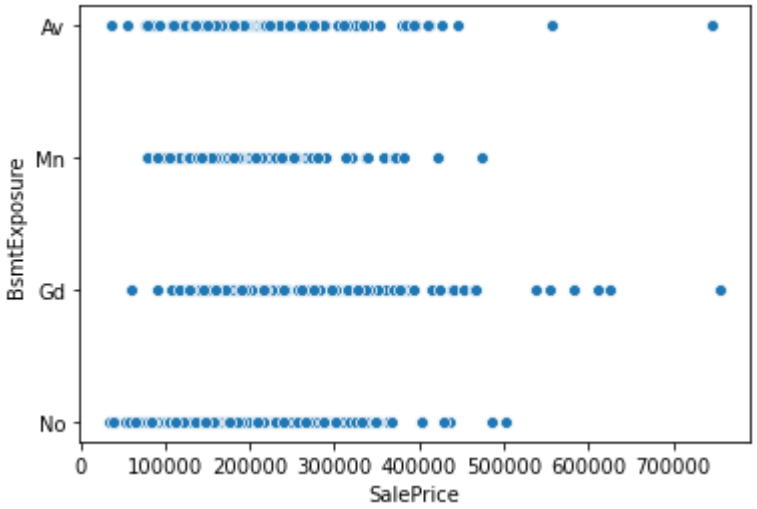

PConc
CBlock
BrkTil
Slab
Stone
Wood
0
100
200
300
400
500
600

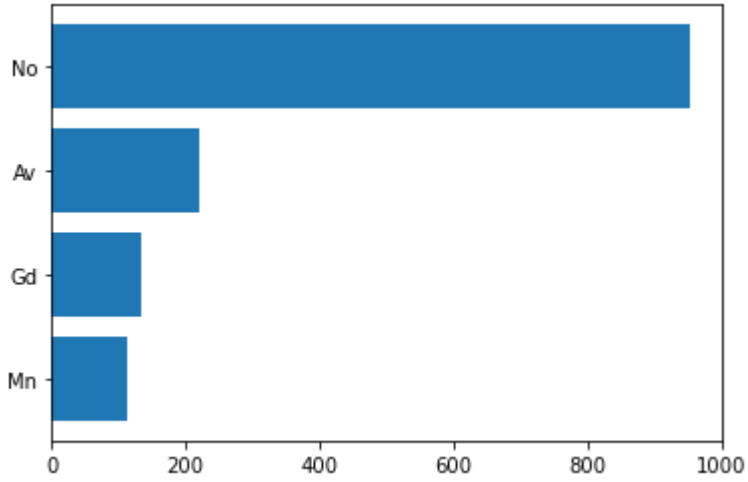

Fa
Ex
TA
Gd
BsmtQual
0
100000
200000
300000
400000
500000
600000
700000
SalePrice

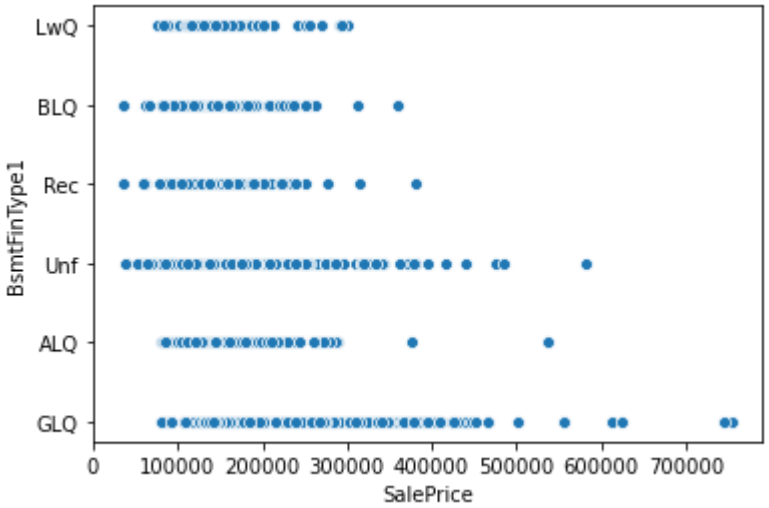

TA
Gd
Ex
Fa
0
100
200
300
400
500
600

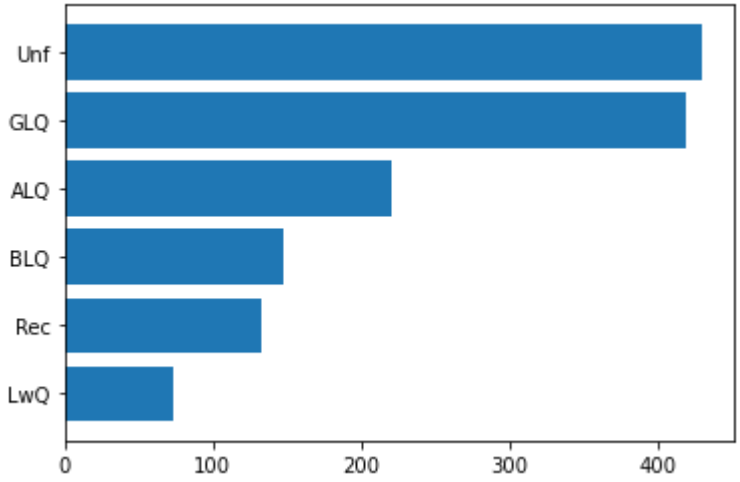

Po
Fa
Gd
TA
BsmtCond
0
100000
200000
300000
400000
500000
600000
700000
SalePrice

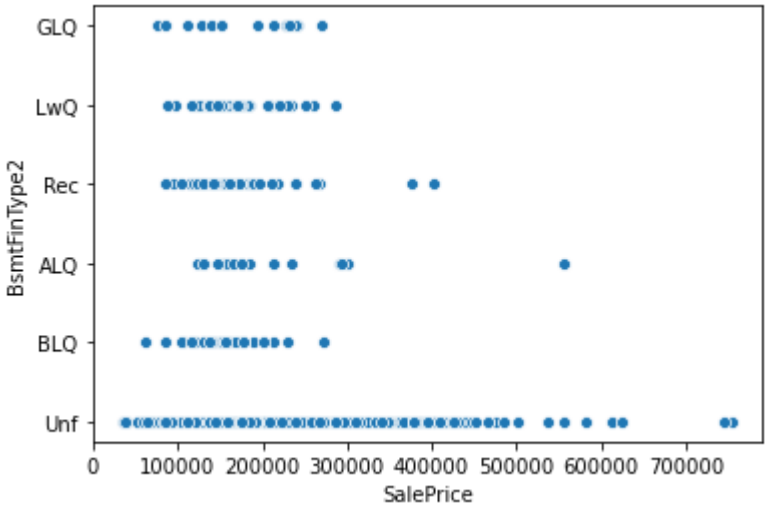

TA
Gd
Fa
Po
0
200
400
600
800
1000
1200

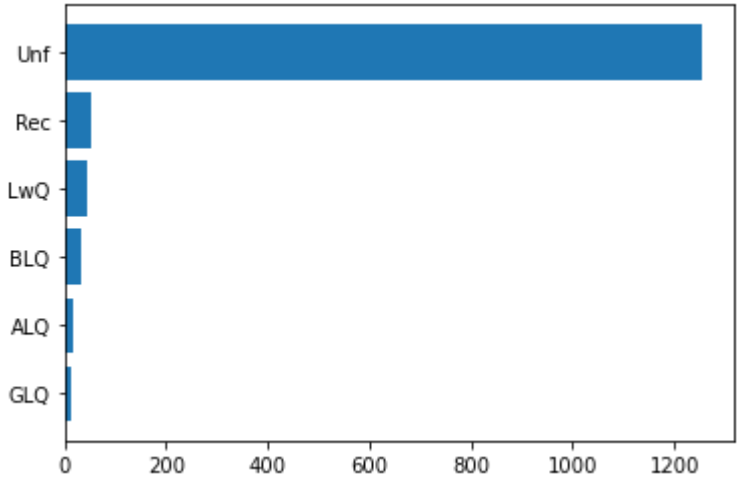

Av
Mn
Gd
No
BsmtExposure
0
100000
200000
300000
400000
500000
600000
700000
SalePrice

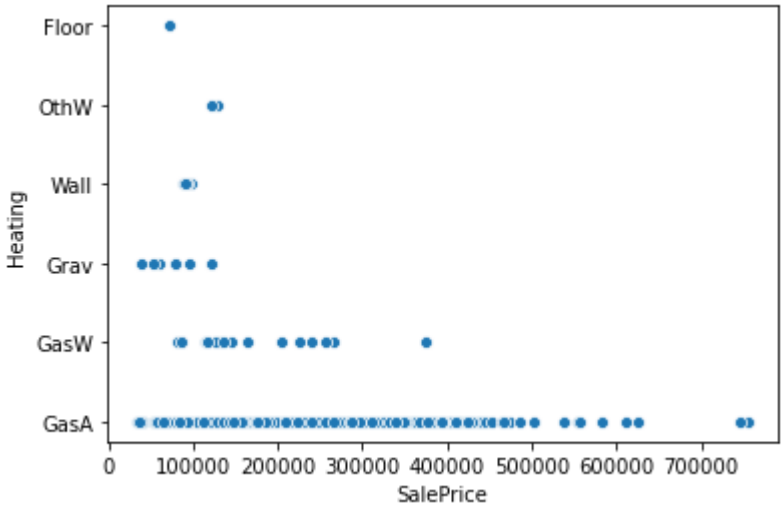

No
Av
Gd
Mn
0
200
400
600
800
1000

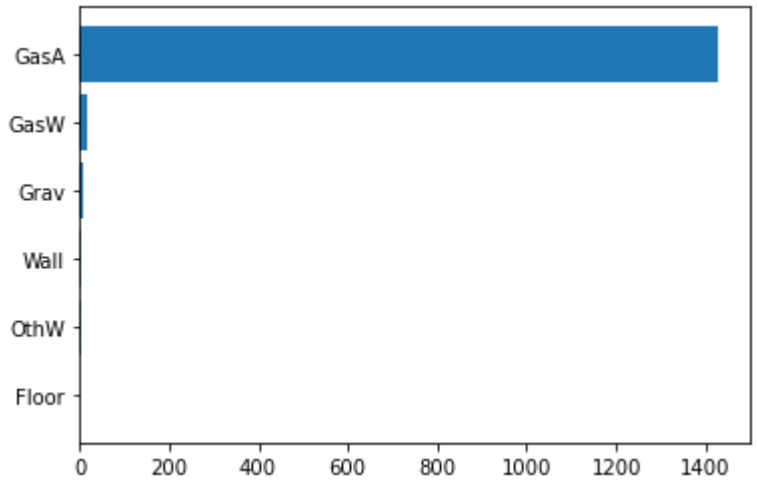

LwQ
BLQ
Rec
Unf
ALQ
GLQ
BsmtFinType1
0
100000
200000
300000
400000
500000
600000
700000
SalePrice

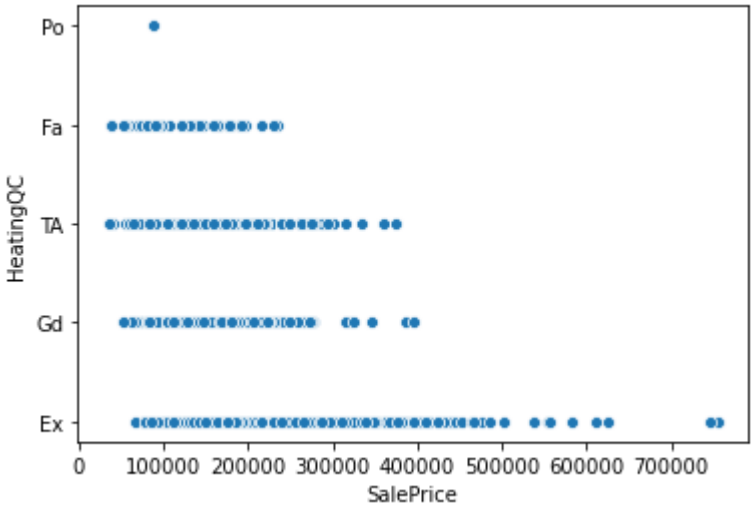

Unf
GLQ
ALQ
BLQ
Rec
LwQ
0
100
200
300
400

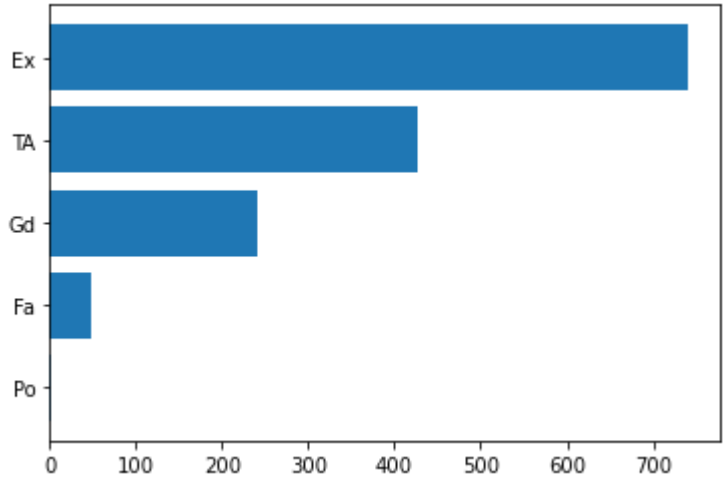

GLQ
LwQ
Rec
ALQ
BLQ
Unf
BsmtFinType2
0
100000
200000
300000
400000
500000
600000
700000
SalePrice

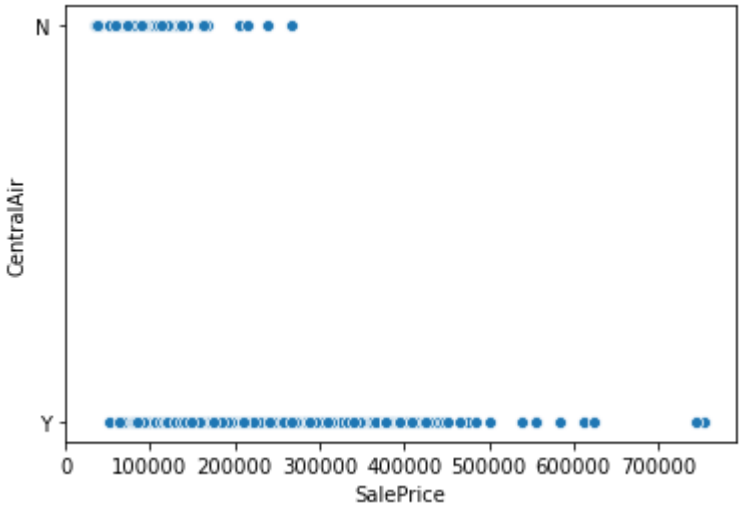

Unf
Rec
LwQ
BLQ
ALQ
GLQ
0
200
400
600
800
1000
1200

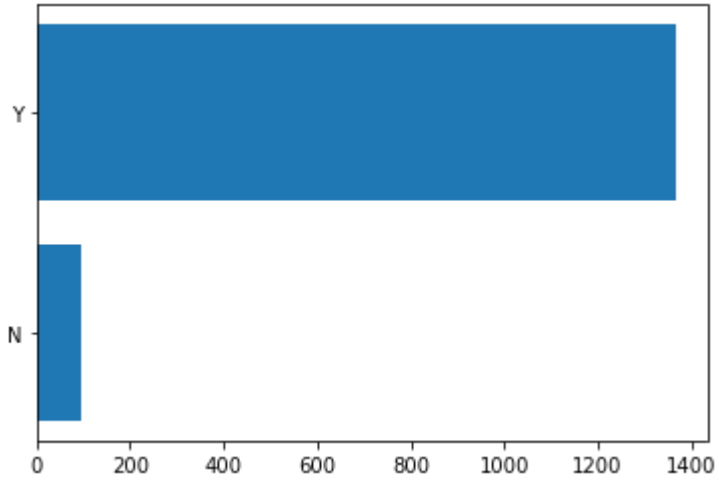

Floor
OthW
Wall
Grav
GasW
GasA
Heating
0
100000
200000
300000
400000
500000
600000
700000
SalePrice

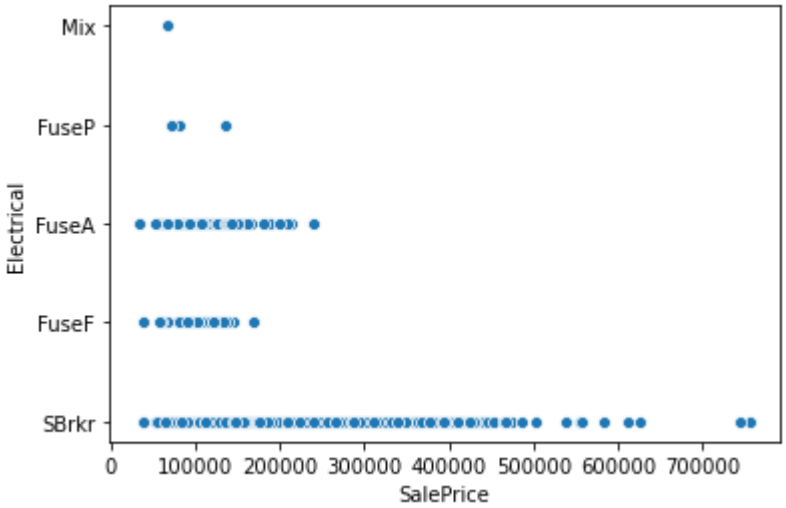

GasA
GasW
Grav
Wall
OthW
Floor
0
200
400
600
800
1000
1200
1400

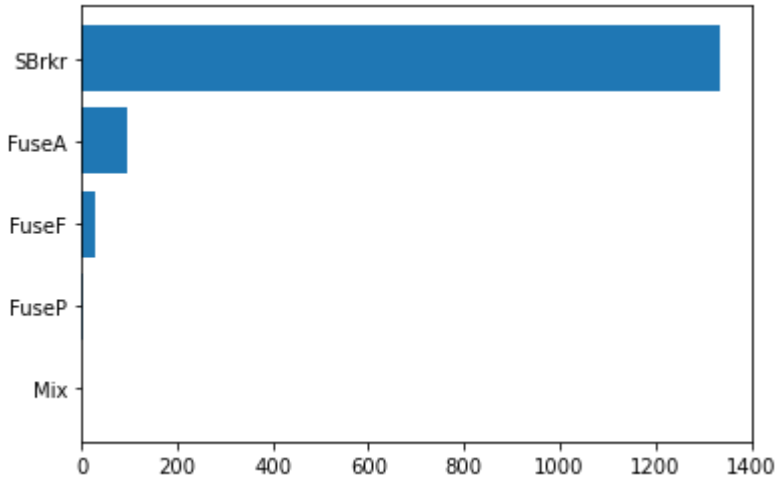

Po
Fa
TA
Gd
Ex
HeatingQC
0
100000
200000
300000
400000
500000
600000
700000
SalePrice

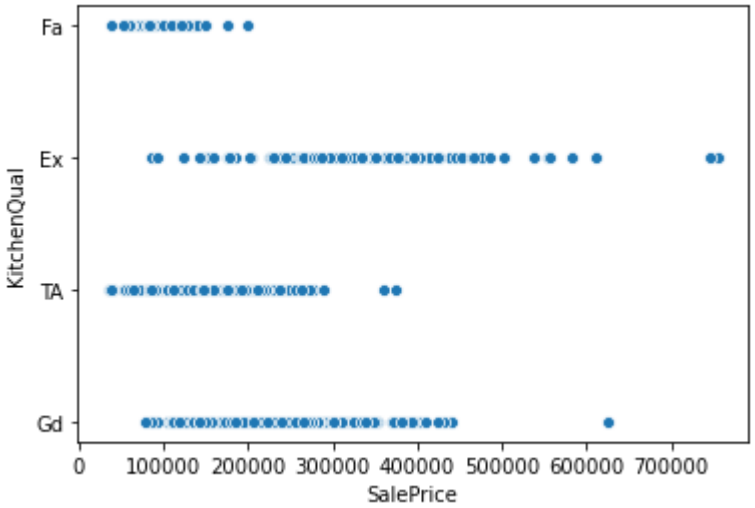

Ex
TA
Gd
Fa
Po
0
100
200
300
400
500
600
700

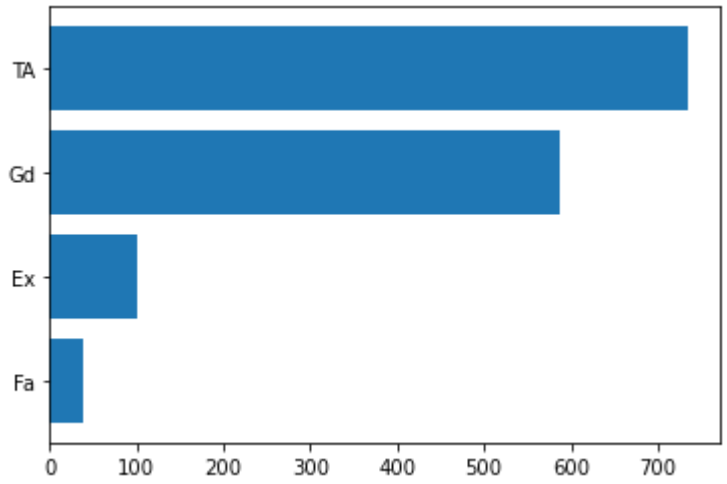

N
Y
CentralAir
0
100000
200000
300000
400000
500000
600000
700000
SalePrice

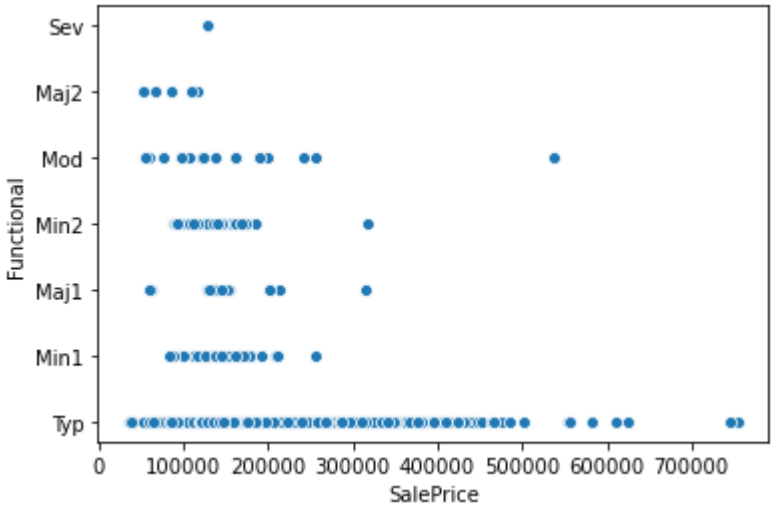

Y
N
0
200
400
600
800
1000
1200
1400

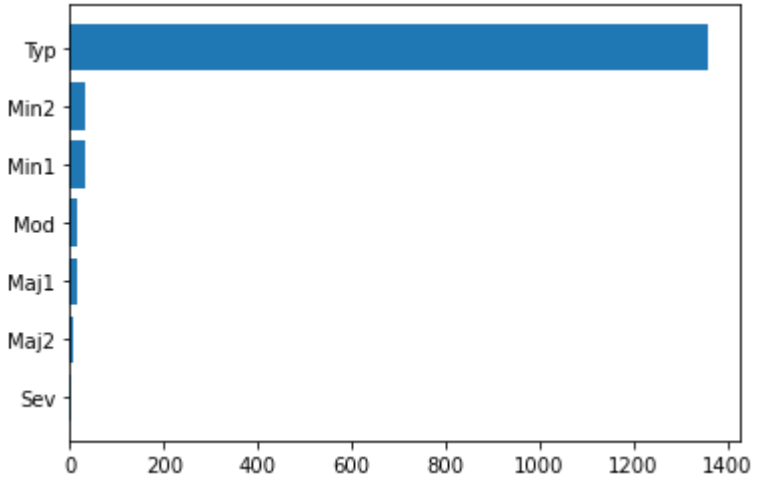

Mix
FuseP
FuseA
FuseF
SBrkr
Electrical
0
100000
200000
300000
400000
500000
600000
700000
SalePrice

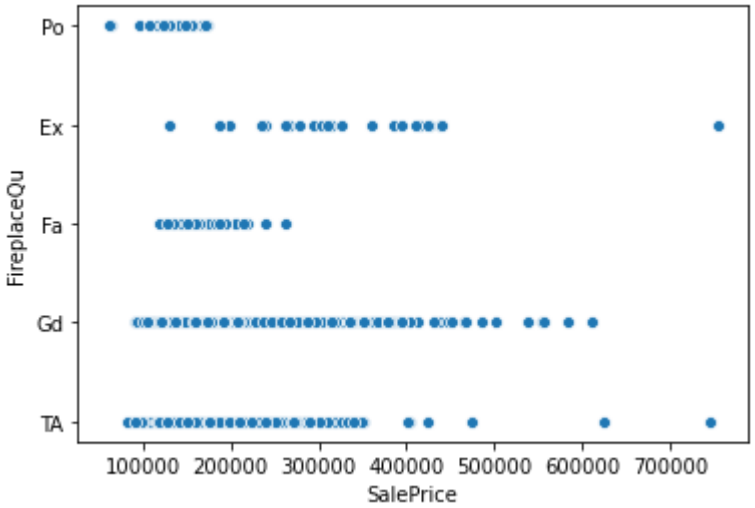

SBrkr
FuseA
FuseF
FuseP
Mix
0
200
400
600
800
1000
1200
1400

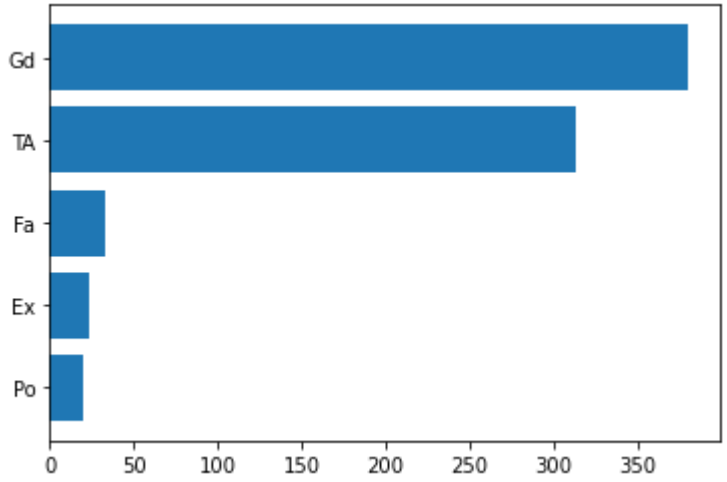

Fa
Ex
TA
Gd
KitchenQual
0
100000
200000
300000
400000
500000
600000
700000
SalePrice

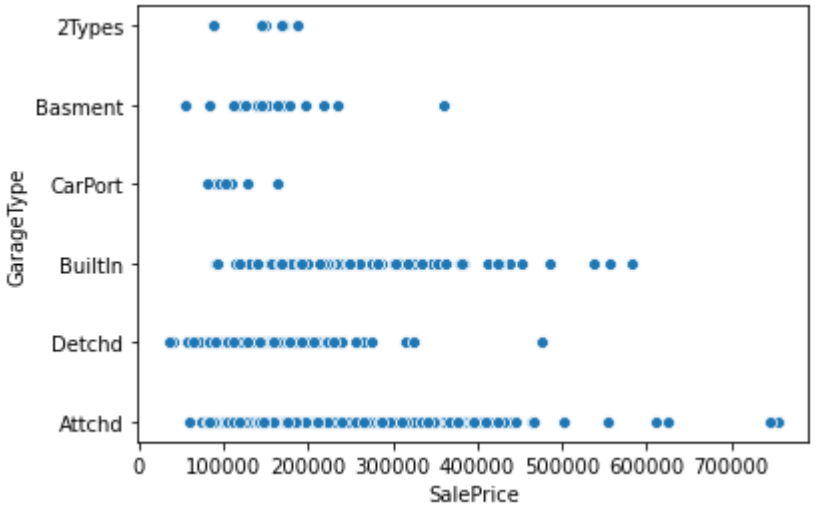

TA
Gd
Ex
Fa
0
100
200
300
400
500
600
700

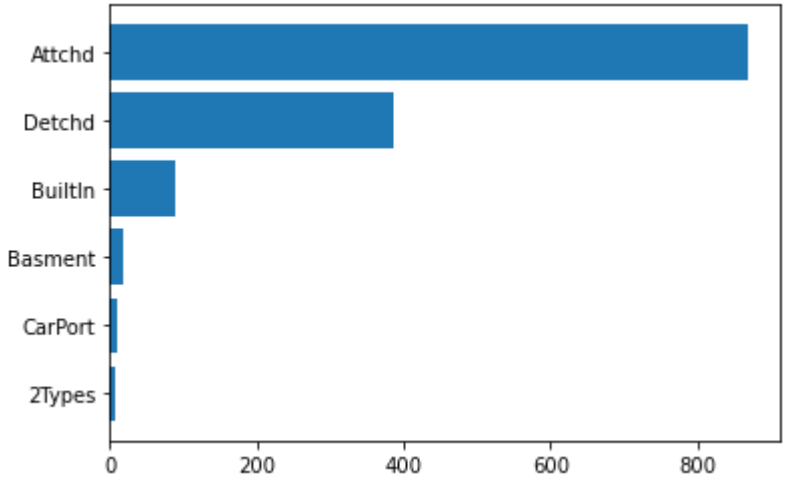

Sev
Maj2
Mod
Min2
Maj1
Min1
Typ
Functional
0
100000
200000
300000
400000
500000
600000
700000
SalePrice

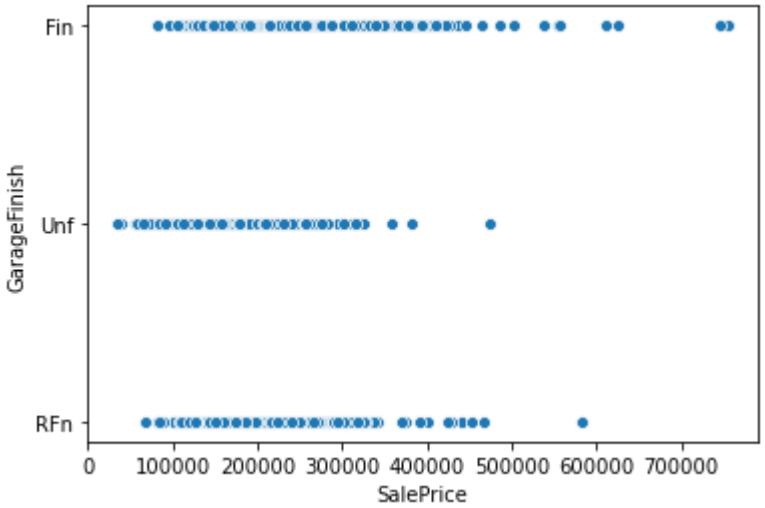

Typ
Min2
Min1
Mod
Maj1
Maj2
Sev
0
200
400
600
800
1000
1200
1400

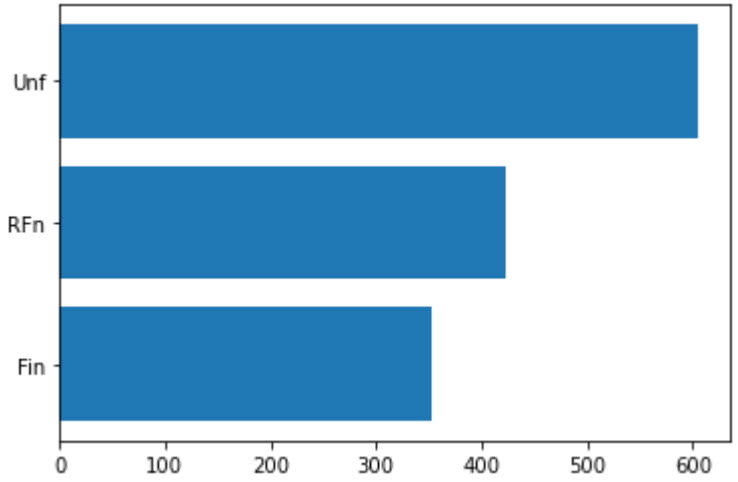

Po
Ex
Fa
Gd
TA
FireplaceQu
100000
200000
300000
400000
500000
600000
700000
SalePrice

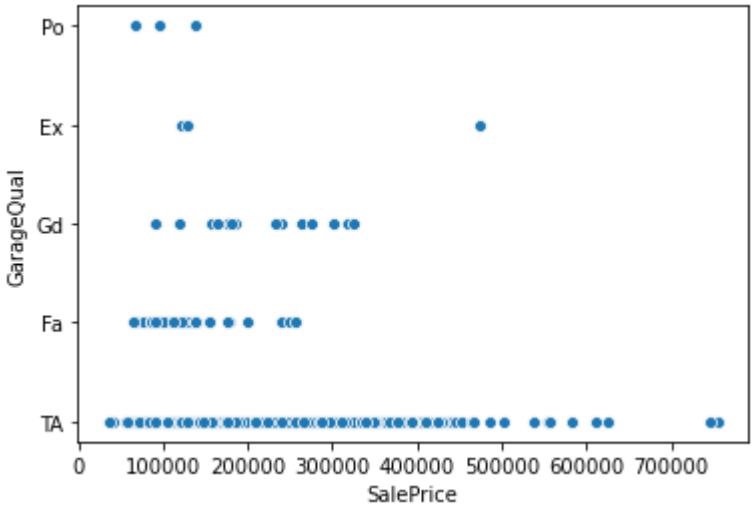

Gd
TA
Fa
Ex
Po
0
50
100
150
200
250
300
350

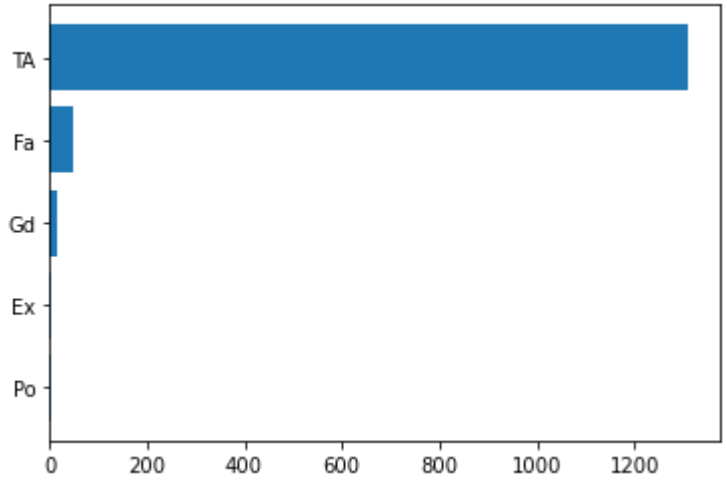

2Types
Basment
CarPort
BuiltIn
Detchd
Attchd
GarageType
0
100000
200000
300000
400000
500000
600000
700000
SalePrice

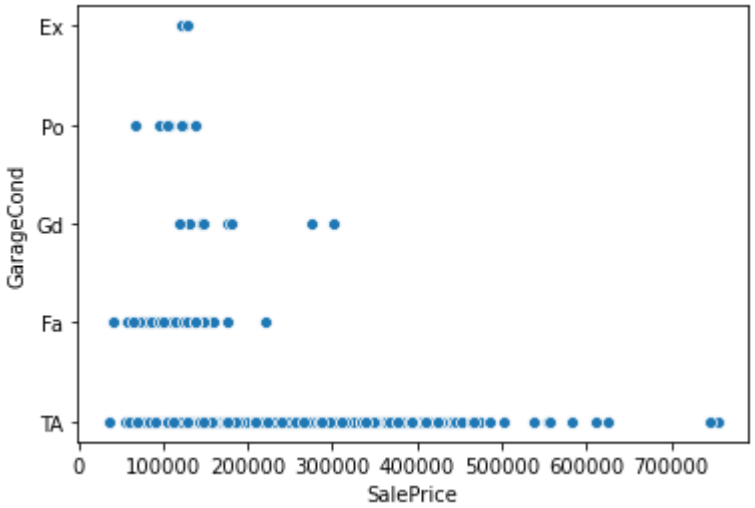

Attchd
Detchd
BuiltIn
Basment
CarPort
2Types
0
200
400
600
800

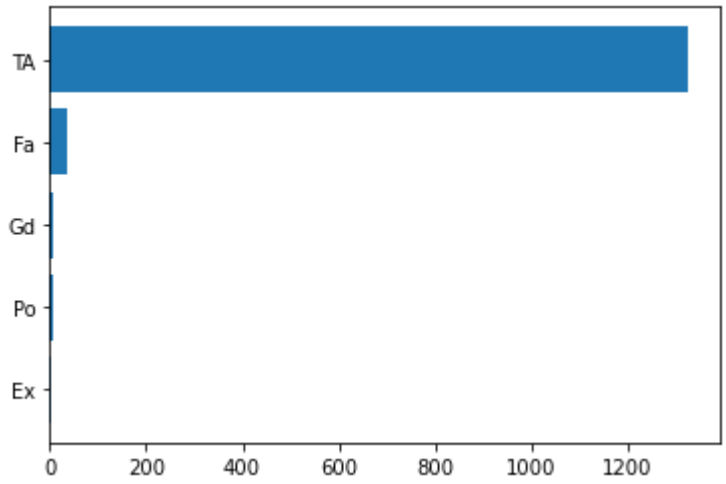

Fin
Unf
RFn
GarageFinish
0
100000
200000
300000
400000
500000
600000
700000
SalePrice

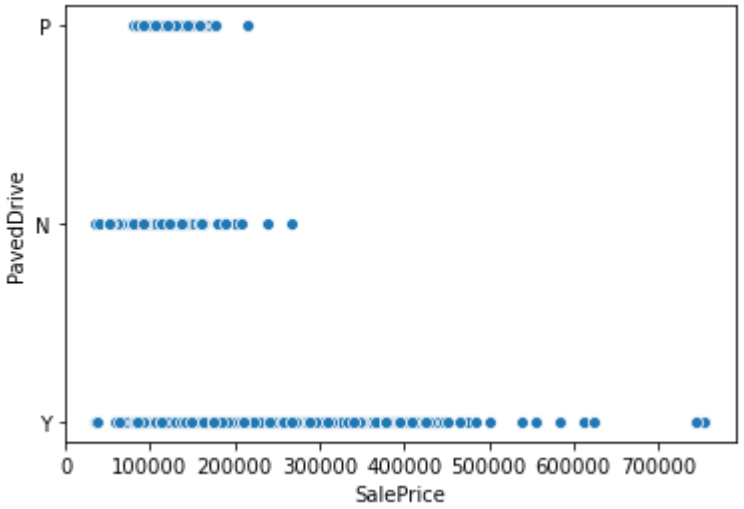

Unf
RFn
Fin
0
100
200
300
400
500
600

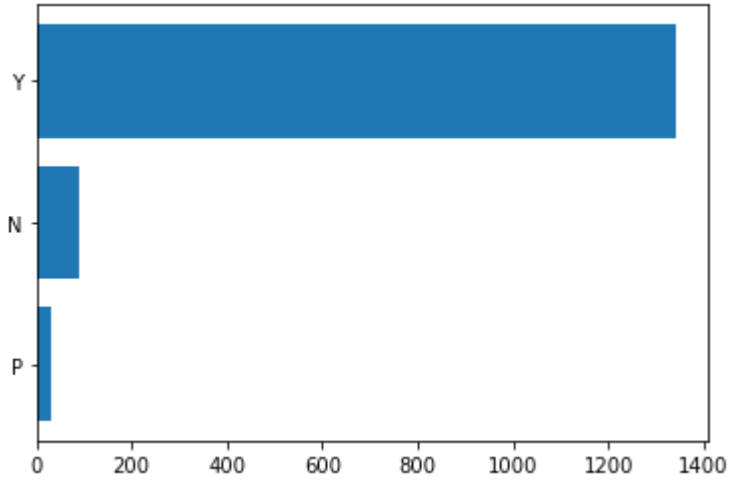

Po
Ex
Gd
Fa
TA
GarageQual
0
100000
200000
300000
400000
500000
600000
700000
SalePrice

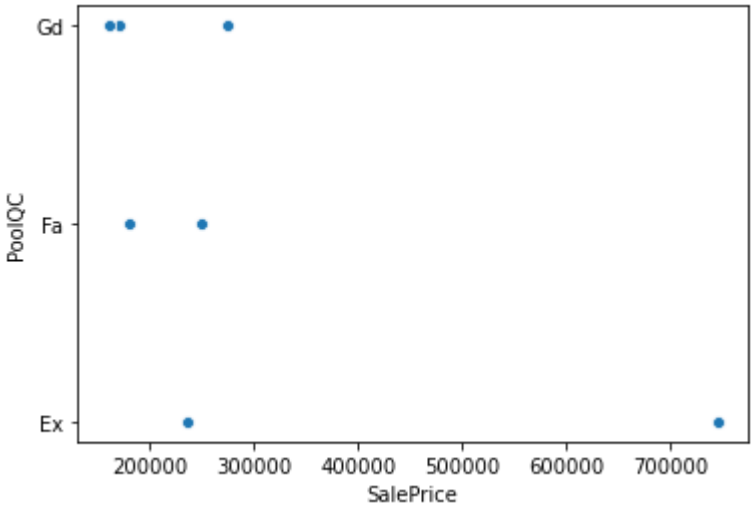

TA
Fa
Gd
Ex
Po
0
200
400
600
800
1000
1200

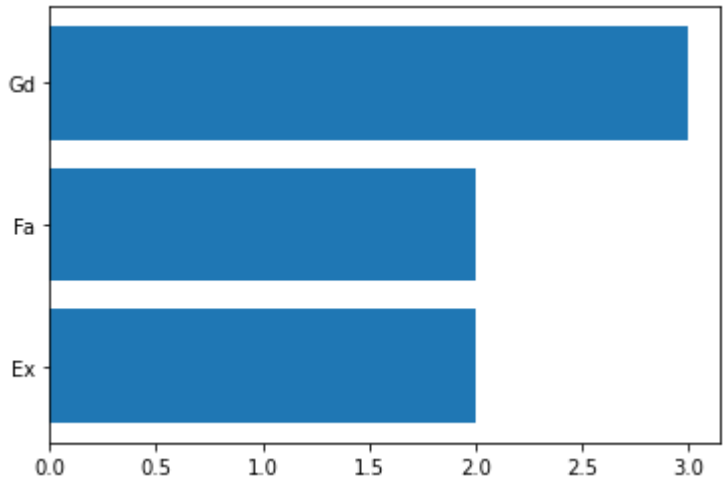

Ex
Po
Gd
Fa
TA
GarageCond
0
100000
200000
300000
400000
500000
600000
700000
SalePrice

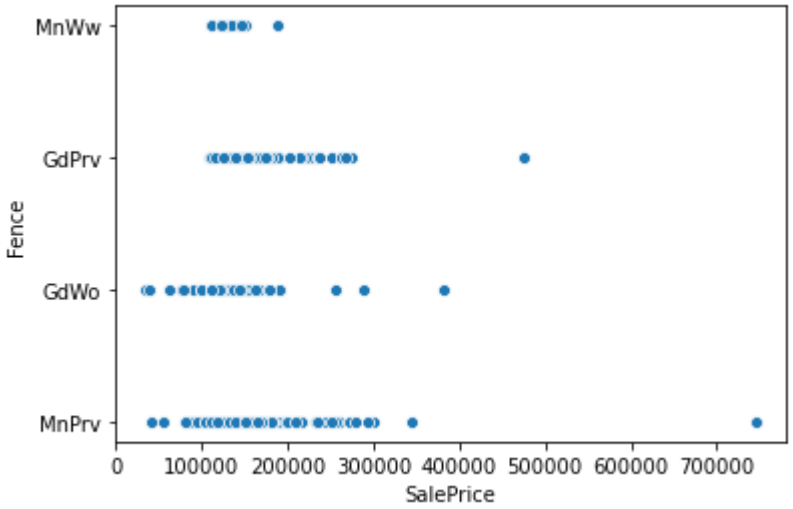

TA
Fa
Gd
Po
Ex
0
200
400
600
800
1000
1200

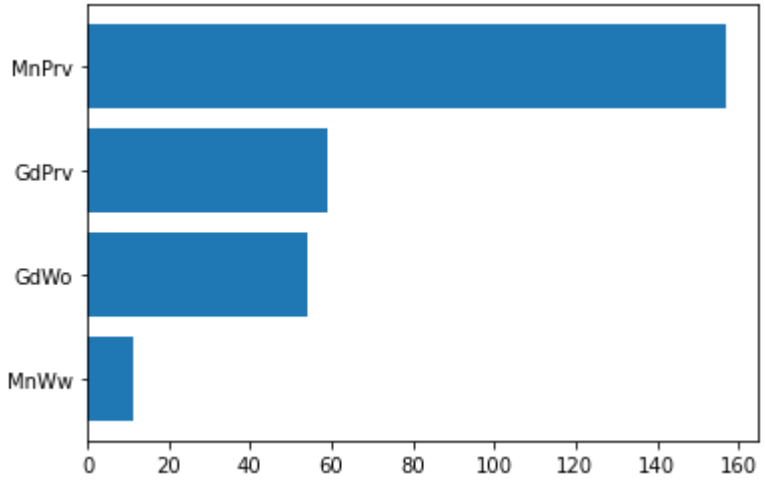

P
N
Y
PavedDrive
0
100000
200000
300000
400000
500000
600000
700000
SalePrice

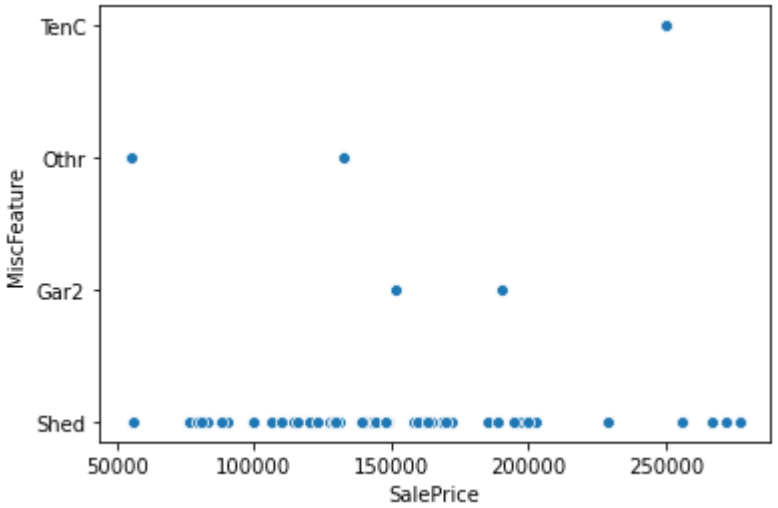

Y
N
P
0
200
400
600
800
1000
1200
1400

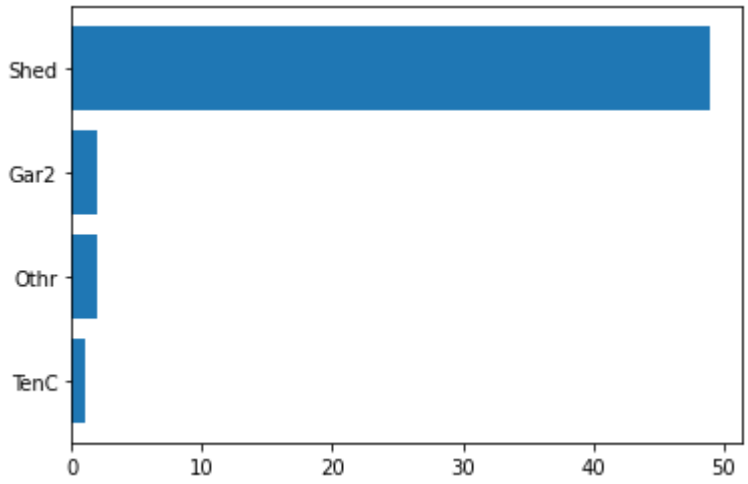

Gd
Fa
Ex
PoolQC
200000
300000
400000
500000
600000
700000
SalePrice

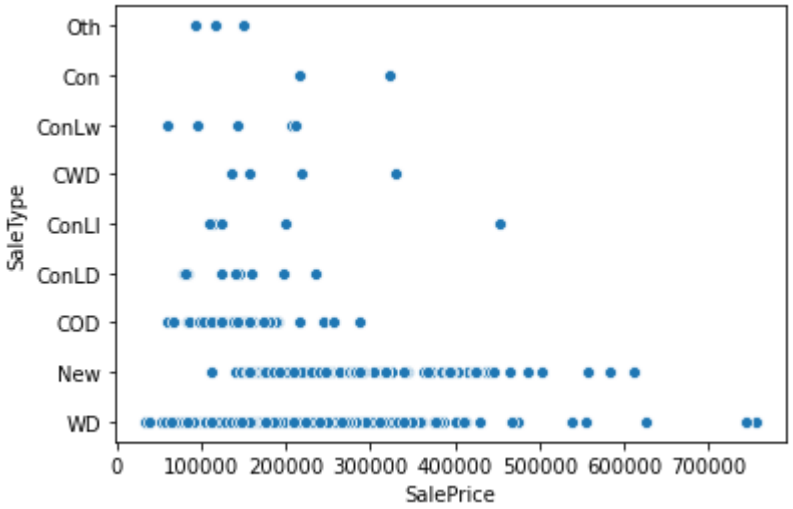

Gd
Fa
Ex
0.0
0.5
1.0
1.5
2.0
2.5
3.0

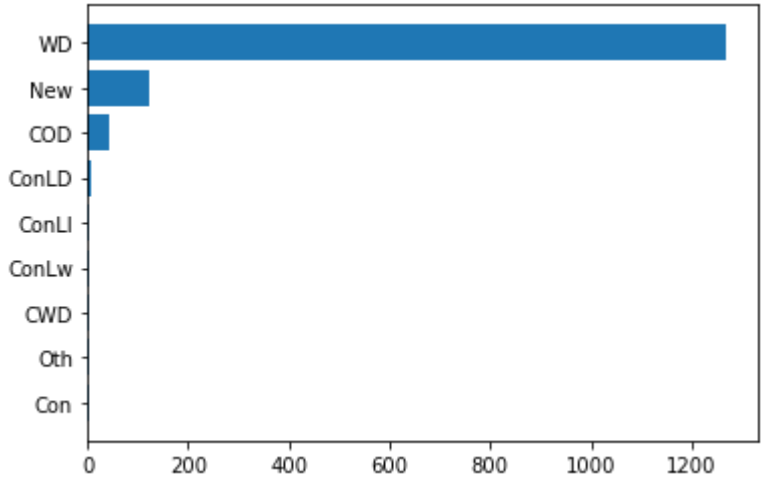

MnWw
GdPrv
GdWo
MnPrv
Fence
0
100000
200000
300000
400000
500000
600000
700000
SalePrice

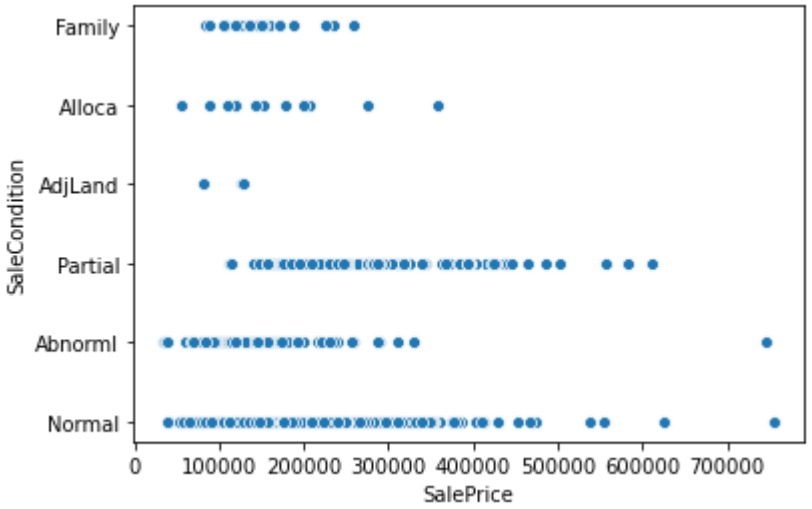

MnPrv
GdPrv
GdWo
MnWw
0
20
40
60
80
100
120
140
160

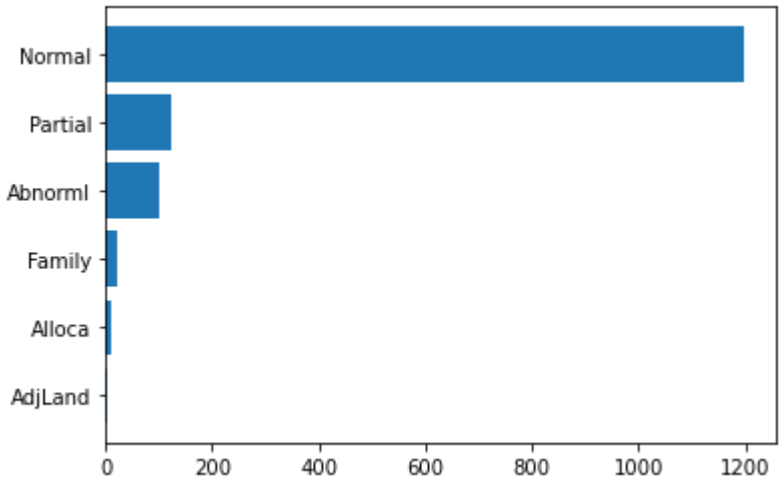

TenC
Othr
Gar2
Shed
MiscFeature
50000
100000
150000
200000
250000
SalePrice


Shed
Gar2
Othr
TenC
0
10
20
30
40
50


Oth
Con
ConLw
CWD
ConLI
ConLD
COD
New
WD
SaleType
0
100000
200000
300000
400000
500000
600000
700000
SalePrice


WD
New
COD
ConLD
ConLI
ConLw
CWD
Oth
Con
0
200
400
600
800
1000
1200


Family
Alloca
AdjLand
Partial
Abnorml
Normal
SaleCondition
0
100000
200000
300000
400000
500000
600000
700000
SalePrice


Normal
Partial
Abnorml
Family
Alloca
AdjLand
0
200
400
600
800
1000
1200

In [10]:

```
## Fill NANs with '0's
df_house_price_train_objects_filled=df_house_price_train_objects.fillna('0')
```

In [11]:

```
# Label encode 'df_house_price_train_objects_filled' dataframe
le = preprocessing.LabelEncoder()
```

In [12]:

```
df_house_price_train_objects_filled.columns
# 43 # range(len(df_house_price_train_objects_filled.columns))
```

Out[12]:

```
Index(['MSZoning', 'Street', 'Alley', 'LotShape', 'LandContour', 'Utilities',
       'LotConfig', 'LandSlope', 'Neighborhood', 'Condition1', 'Condition2',
       'BldgType', 'HouseStyle', 'RoofStyle', 'RoofMatl', 'Exterior1st',
       'Exterior2nd', 'MasVnrType', 'ExterQual', 'ExterCond', 'Foundation',
       'BsmtQual', 'BsmtCond', 'BsmtExposure', 'BsmtFinType1', 'BsmtFinType2',
       'Heating', 'HeatingQC', 'CentralAir', 'Electrical', 'KitchenQual',
       'Functional', 'FireplaceQu', 'GarageType', 'GarageFinish', 'GarageQual',
       'GarageCond', 'PavedDrive', 'PoolQC', 'Fence', 'MiscFeature',
       'SaleType', 'SaleCondition'],
      dtype='object')
```

In [13]:

```
house_price_train_objects_table_encoded=[le.fit_transform(df_house_price_train_objects_filled[df_house_price_train_object
s_filled.columns[i]]) for i in range(len(df_house_price_train_objects_filled.columns))]
```

In [14]:

```
df_house_price_train_objects_table_encoded=pd.DataFrame(house_price_train_objects_table_encoded,index=df_house_price_train
n_objects_filled.columns).transpose()
df_house_price_train_objects_table_encoded['SalePrice']=df_house_price_train['SalePrice']
df_house_price_train_objects_table_encoded.head()
```

Out[14]:

	MSZoning	Street	Alley	LotShape	LandContour	Utilities	LotConfig	LandSlope	Neighborhood	Condition1	...	GarageFinish	Gara
0	3	1	0	3	3	0	4	0	5	2	...	2	
1	3	1	0	3	3	0	2	0	24	1	...	2	
2	3	1	0	0	3	0	4	0	5	2	...	2	
3	3	1	0	0	3	0	0	0	6	2	...	3	
4	3	1	0	0	3	0	2	0	15	2	...	2	

5 rows × 44 columns

In [15]:

```
df_house_price_train_objects_table_encoded.describe()
```

Out[15]:

	MSZoning	Street	Alley	LotShape	LandContour	Utilities	LotConfig	LandSlope	Neighborhood	Con
count	1460.000000	1460.000000	1460.000000	1460.000000	1460.000000	1460.000000	1460.000000	1460.000000	1460.000000	1460.
mean	3.028767	0.995890	0.090411	1.942466	2.777397	0.000685	3.019178	0.062329	12.251370	2.
std	0.632017	0.063996	0.372151	1.409156	0.707666	0.026171	1.622634	0.276232	6.013735	0.
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.
25%	3.000000	1.000000	0.000000	0.000000	3.000000	0.000000	2.000000	0.000000	7.000000	2.
50%	3.000000	1.000000	0.000000	3.000000	3.000000	0.000000	4.000000	0.000000	12.000000	2.
75%	3.000000	1.000000	0.000000	3.000000	3.000000	0.000000	4.000000	0.000000	17.000000	2.
max	4.000000	1.000000	2.000000	3.000000	3.000000	1.000000	4.000000	2.000000	24.000000	8.

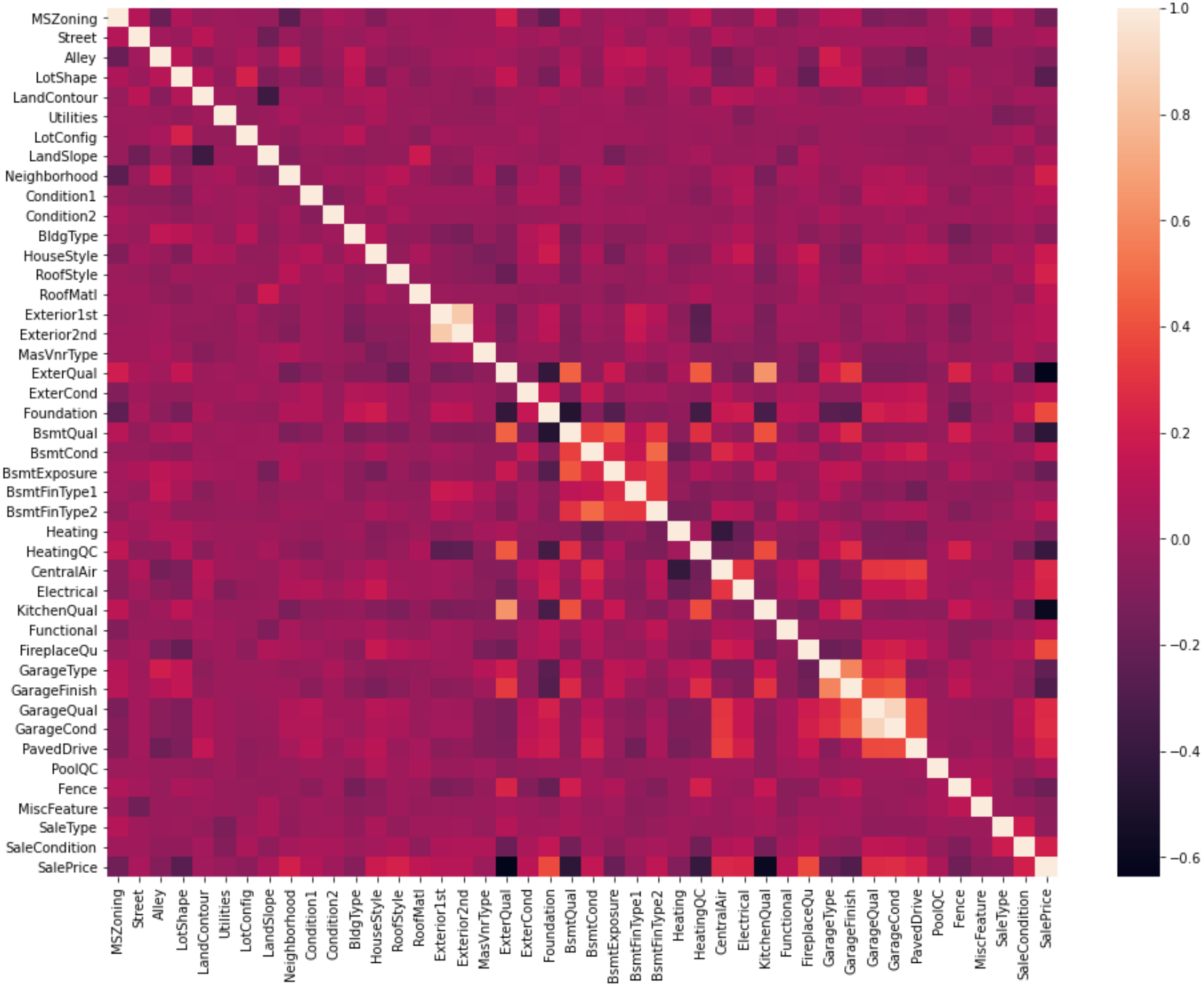
8 rows × 44 columns

In [16]:

```
## Correlation
df_house_price_train_objects_table_encoded_fit=preprocessing.StandardScaler().fit(df_house_price_train_objects_table_enco
ded).transform(df_house_price_train_objects_table_encoded)
df_house_price_train_objects_table_encoded_corr=pd.DataFrame(data=df_house_price_train_objects_table_encoded_fit,columns=
df_house_price_train_objects_table_encoded.columns).corr()
```

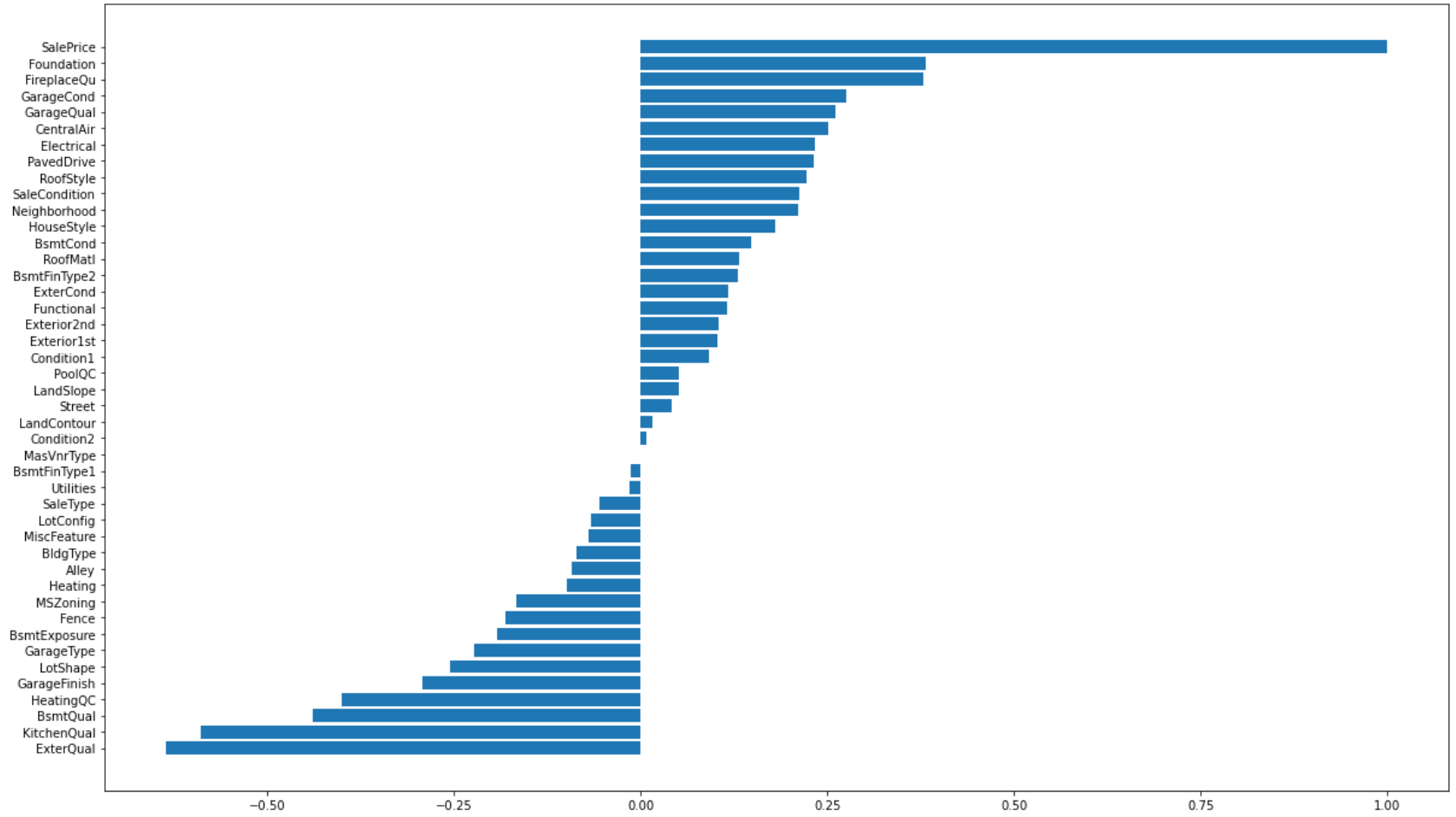
In [17]:

```
fig,ax=plt.subplots(figsize=(16,12))
sns.heatmap(df_house_price_train_objects_table_encoded_corr)
plt.show()
```


MSZoning
Street
Alley
LotShape
LandContour
Utilities
LotConfig
LandSlope
Neighborhood
Condition1
Condition2
BldgType
HouseStyle
RoofStyle
RoofMatl
Exterior1st
Exterior2nd
MasVnrType
ExterQual
ExterCond
Foundation
BsmtQual
BsmtCond
BsmtExposure
BsmtFinType1
BsmtFinType2
Heating
HeatingQC
CentralAir
Electrical
KitchenQual
Functional
FireplaceQu
GarageType
GarageFinish
GarageQual
GarageCond
PavedDrive
PoolQC
Fence
MiscFeature
SaleType
SaleCondition
SalePrice
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6

In [18]:

```
df_house_price_train_objects_table_encoded_corr_saleprice=df_house_price_train_objects_table_encoded_corr['SalePrice'].sort_values()
y_pos = np.arange(len(df_house_price_train_objects_table_encoded_corr_saleprice))
fig,ax=plt.subplots(figsize=(20,12))
plt.barh(y_pos, df_house_price_train_objects_table_encoded_corr_saleprice,tick_label=df_house_price_train_objects_table_encoded_corr_saleprice.index)
plt.show()
```



Numbers

In [19]:

```
# Numbers Dataset Preview
df_house_price_train_numbers=df_house_price_train.select_dtypes(include=['number'])
df_house_price_train_numbers.head()
```

Out[19]:

	MSSubClass	LotFrontage	LotArea	OverallQual	OverallCond	YearBuilt	YearRemodAdd	MasVnrArea	BsmtFinSF1	BsmtFinSF2	...
0	60	65.0	8450	7	5	2003	2003	196.0	706	0	...
1	20	80.0	9600	6	8	1976	1976	0.0	978	0	...
2	60	68.0	11250	7	5	2001	2002	162.0	486	0	...
3	70	60.0	9550	7	5	1915	1970	0.0	216	0	...
4	60	84.0	14260	8	5	2000	2000	350.0	655	0	...

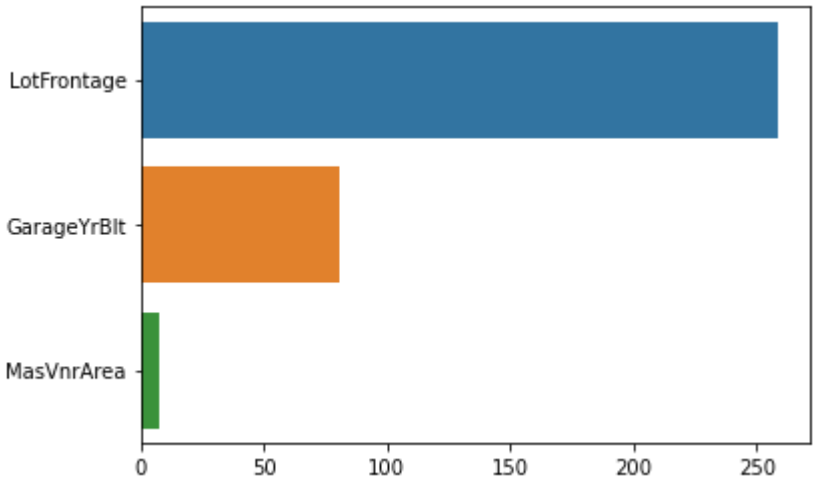
5 rows × 37 columns

In [20]:

```
# Bar plot of missing values in the number dataset
df_house_price_train_numbers_missing_data=df_house_price_train_numbers.isnull().sum().sort_values(ascending=False).head(3
)
sns.barplot(y=df_house_price_train_numbers_missing_data.index,x=df_house_price_train_numbers_missing_data)
```

Out[20]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x179d37c2c70>
```



In [21]:

```
# Create dataframe of continuous values
df_house_price_train_numbers_continuous = df_house_price_train_numbers[['LotFrontage', 'LotArea',
       'YearBuilt', 'YearRemodAdd', 'MasVnrArea', 'BsmtFinSF1', 'BsmtFinSF2',
       'BsmtUnfSF', 'TotalBsmtSF', '1stFlrSF', '2ndFlrSF', 'LowQualFinSF',
       'GrLivArea' ,
       'GarageYrBlt', 'GarageArea', 'WoodDeckSF', 'OpenPorchSF',
       'EnclosedPorch', '3SsnPorch', 'ScreenPorch', 'PoolArea', 'MiscVal',
       'SalePrice']]
df_house_price_train_numbers_continuous.head()
```

Out[21]:

	LotFrontage	LotArea	YearBuilt	YearRemodAdd	MasVnrArea	BsmtFinSF1	BsmtFinSF2	BsmtUnfSF	TotalBsmtSF	1stFlrSF	...	Ga
0	65.0	8450	2003	2003	196.0	706	0	150	856	856	...	
1	80.0	9600	1976	1976	0.0	978	0	284	1262	1262	...	
2	68.0	11250	2001	2002	162.0	486	0	434	920	920	...	
3	60.0	9550	1915	1970	0.0	216	0	540	756	961	...	
4	84.0	14260	2000	2000	350.0	655	0	490	1145	1145	...	

5 rows × 23 columns

In [22]:

```
df_house_price_train_numbers_continuous.describe()
```

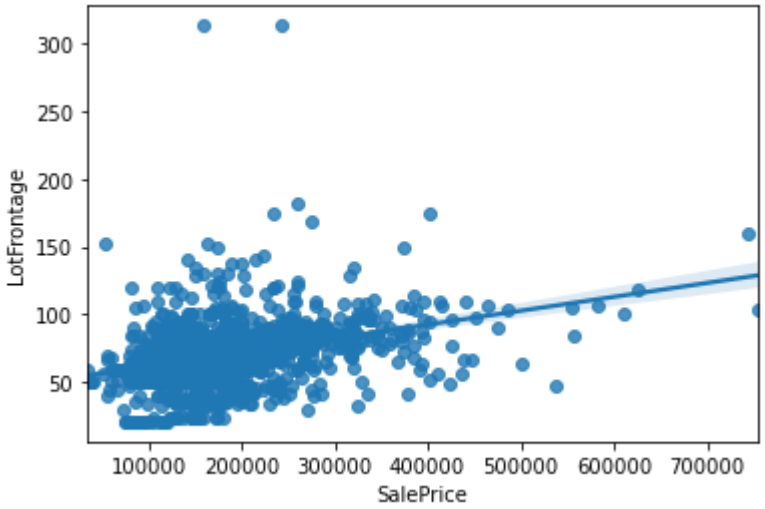
Out[22]:

	LotFrontage	LotArea	YearBuilt	YearRemodAdd	MasVnrArea	BsmtFinSF1	BsmtFinSF2	BsmtUnfSF	TotalBsmtSF	
count	1201.000000	1460.000000	1460.000000	1460.000000	1452.000000	1460.000000	1460.000000	1460.000000	1460.000000	14
mean	70.049958	10516.828082	1971.267808	1984.865753	103.685262	443.639726	46.549315	567.240411	1057.429452	11
std	24.284752	9981.264932	30.202904	20.645407	181.066207	456.098091	161.319273	441.866955	438.705324	3
min	21.000000	1300.000000	1872.000000	1950.000000	0.000000	0.000000	0.000000	0.000000	0.000000	3
25%	59.000000	7553.500000	1954.000000	1967.000000	0.000000	0.000000	0.000000	223.000000	795.750000	8
50%	69.000000	9478.500000	1973.000000	1994.000000	0.000000	383.500000	0.000000	477.500000	991.500000	10
75%	80.000000	11601.500000	2000.000000	2004.000000	166.000000	712.250000	0.000000	808.000000	1298.250000	13
max	313.000000	215245.000000	2010.000000	2010.000000	1600.000000	5644.000000	1474.000000	2336.000000	6110.000000	46

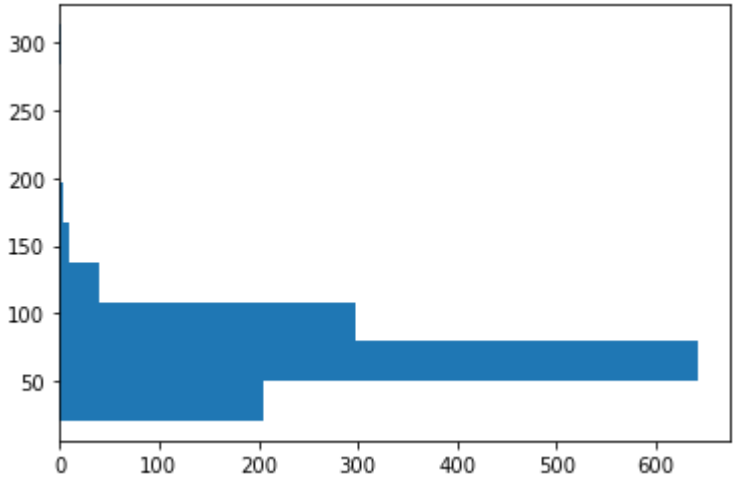
8 rows × 23 columns

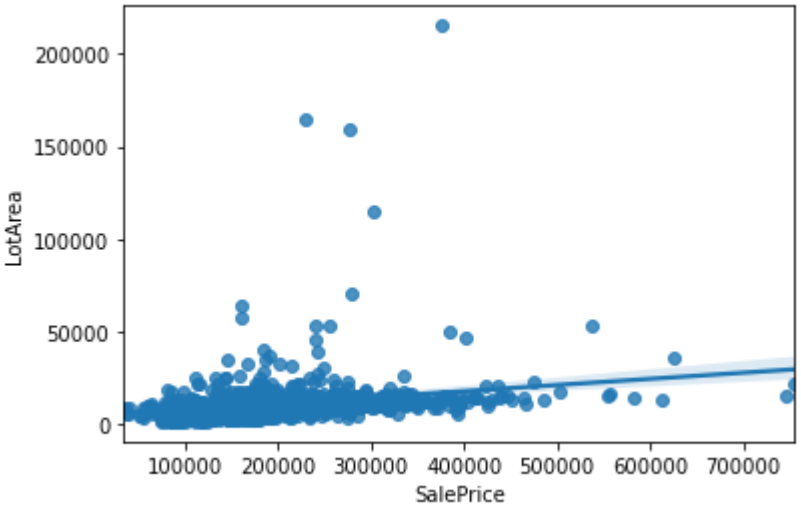
In [23]:

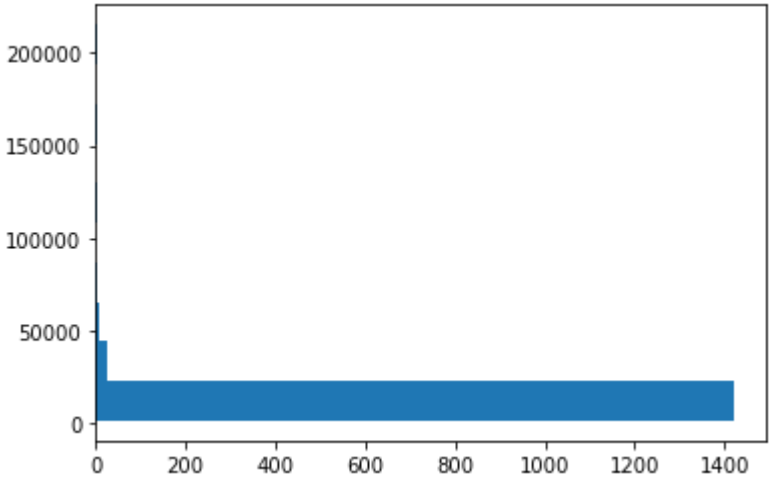
```
for i in range(len(df_house_price_train_numbers_continuous.columns)):
#     fig,ax=plt.subplots(4,1)
#     sns.despine(left=True)
    sns.regplot(y=df_house_price_train_numbers_continuous.columns[i],x='SalePrice',data=df_house_price_train_numbers_cont
inuous)
    plt.show()
    plt.hist(df_house_price_train_numbers_continuous[df_house_price_train_numbers_continuous.columns[i]],orientation='hor
izontal')
    plt.show()
#     plt.setp(ax)
#     plt.tight_layout()
```

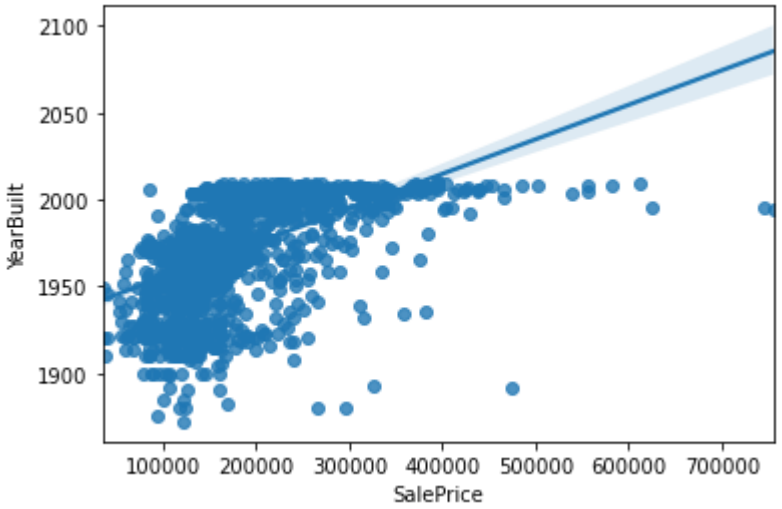


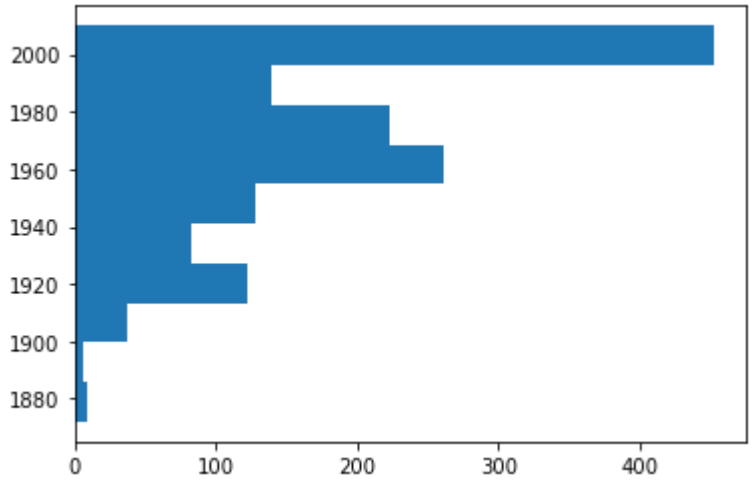

```
C:\Users\marky\anaconda3\envs\geo_env\lib\site-packages\numpy\lib\histograms.py:839: RuntimeWarning: invalid
value encountered in greater_equal
  keep = (tmp_a >= first_edge)
C:\Users\marky\anaconda3\envs\geo_env\lib\site-packages\numpy\lib\histograms.py:840: RuntimeWarning: invalid
value encountered in less_equal
  keep &= (tmp_a <= last_edge)
```

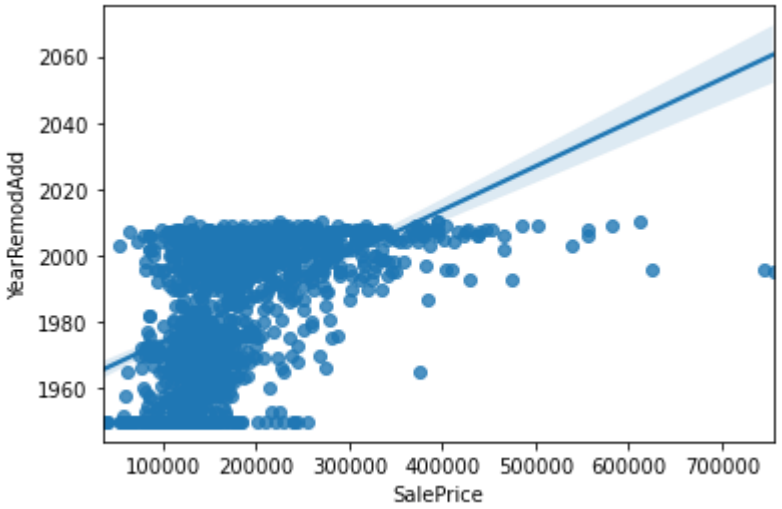


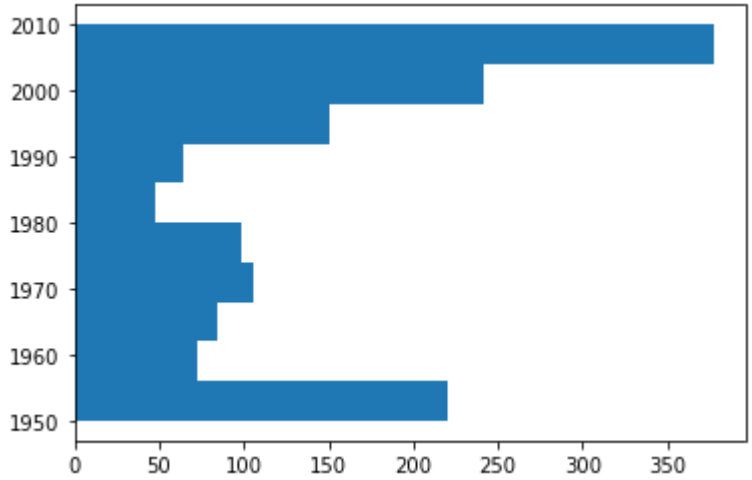

200000
150000
100000
50000
0
LotArea
100000
200000
300000
400000
500000
600000
700000
SalePrice

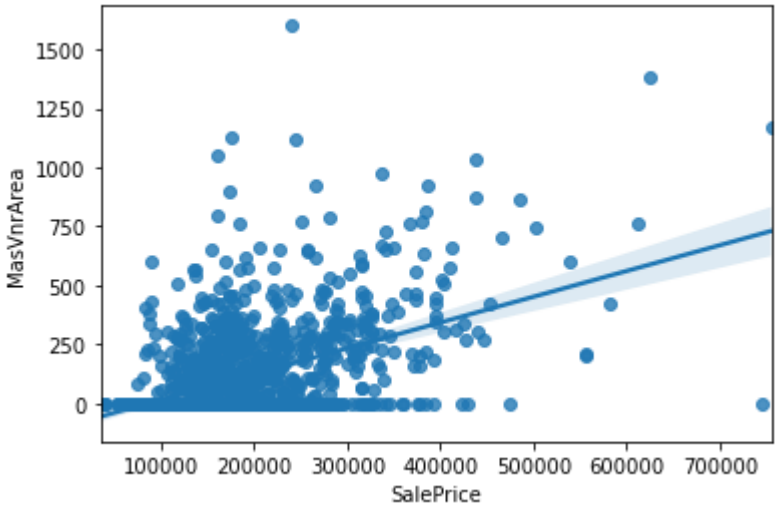

200000
150000
100000
50000
0
0
200
400
600
800
1000
1200
1400


2100
2050
2000
1950
1900
YearBuilt
100000
200000
300000
400000
500000
600000
700000
SalePrice

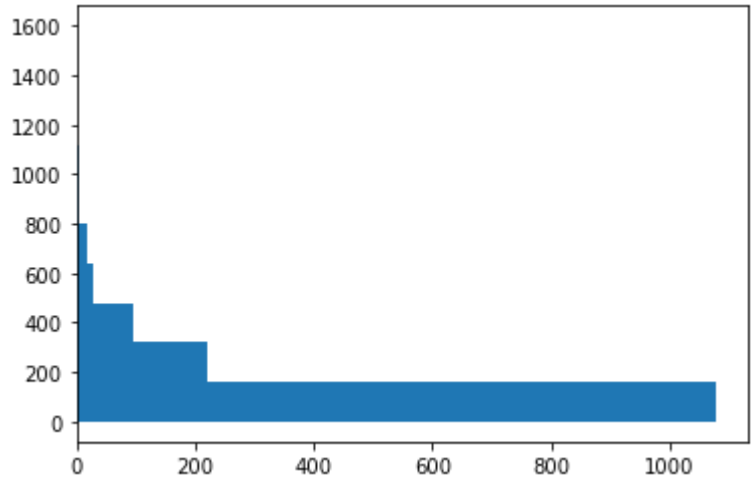

2000
1980
1960
1940
1920
1900
1880
0
100
200
300
400

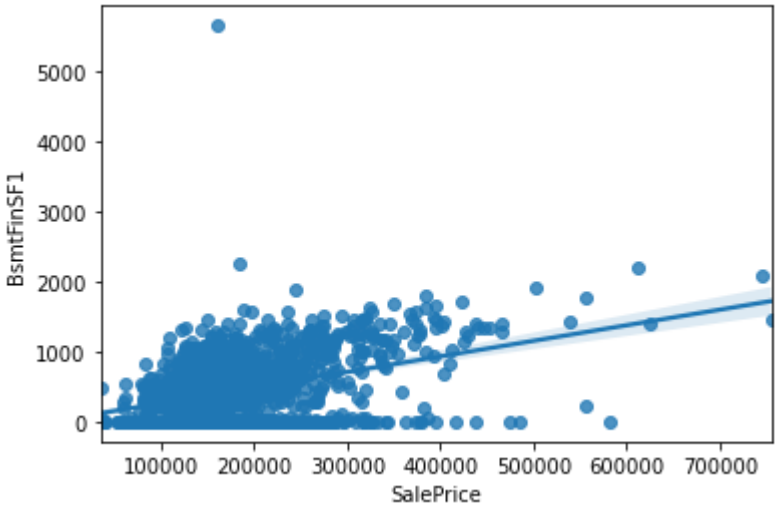

2060
2040
2020
2000
1980
1960
YearRemodAdd
100000
200000
300000
400000
500000
600000
700000
SalePrice

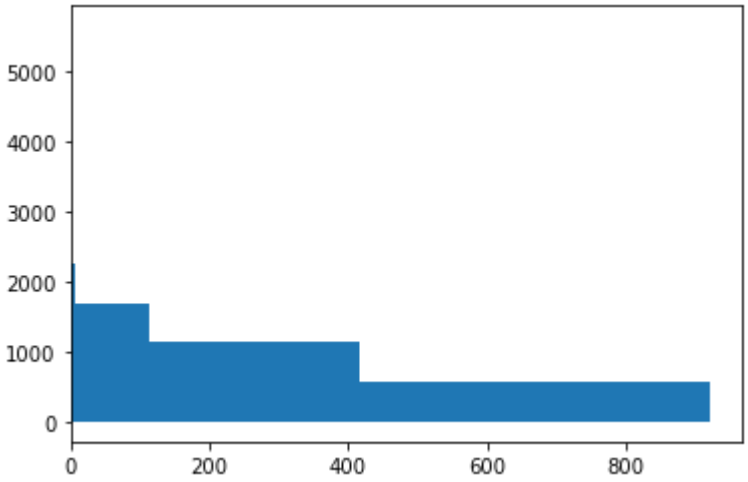

2010
2000
1990
1980
1970
1960
1950
0
50
100
150
200
250
300
350

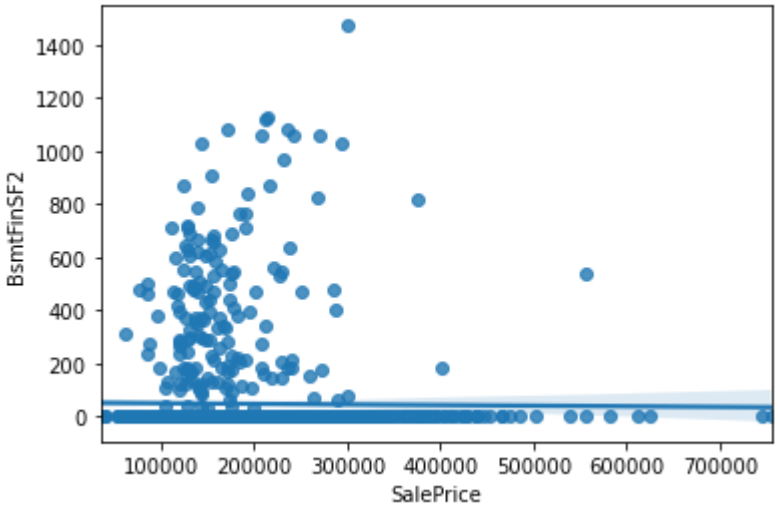


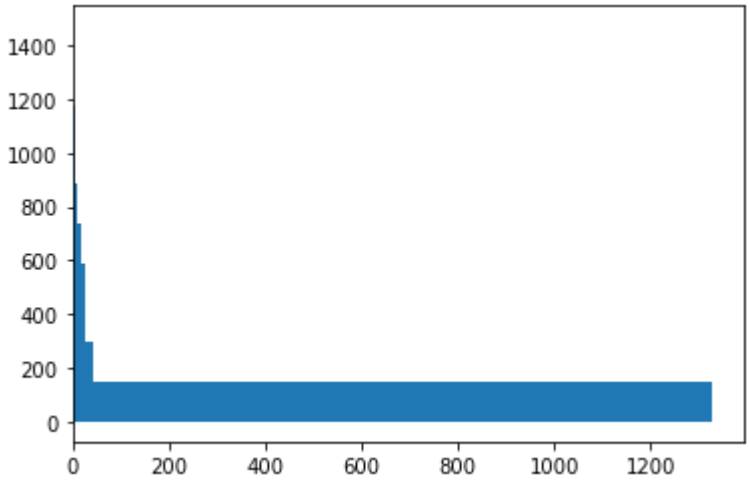

```
C:\Users\marky\anaconda3\envs\geo_env\lib\site-packages\numpy\lib\histograms.py:839: RuntimeWarning: invalid
value encountered in greater_equal
  keep = (tmp_a >= first_edge)
C:\Users\marky\anaconda3\envs\geo_env\lib\site-packages\numpy\lib\histograms.py:840: RuntimeWarning: invalid
value encountered in less_equal
  keep &= (tmp_a <= last_edge)
```

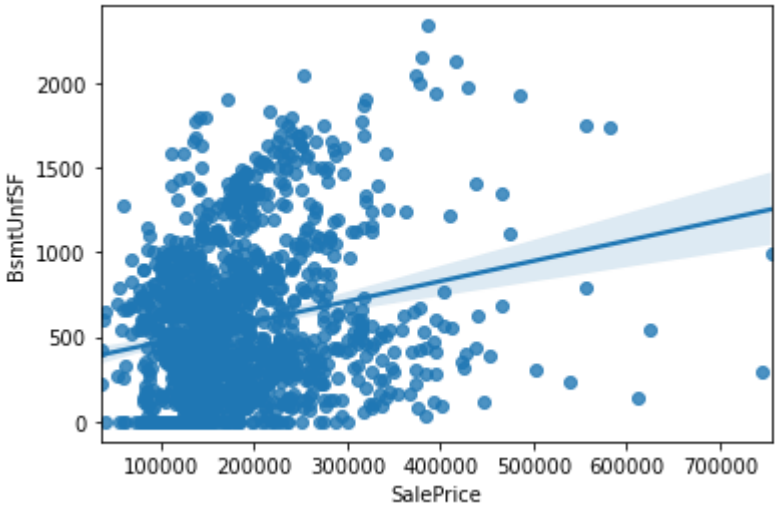

1600
1400
1200
1000
800
600
400
200
0
0
200
400
600
800
1000

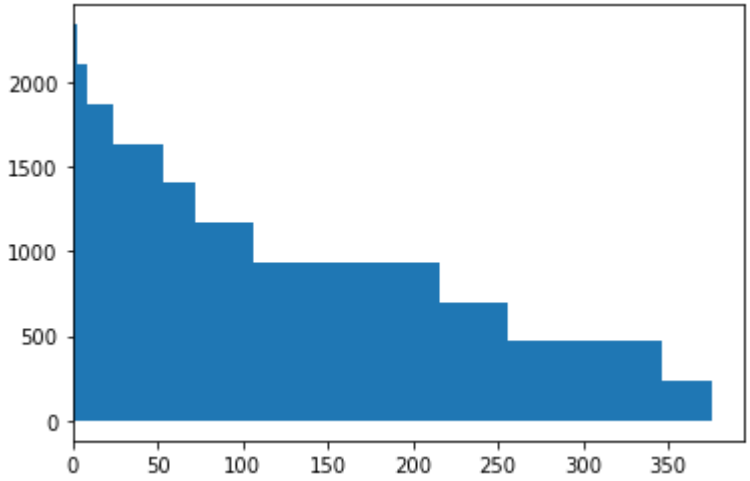

5000
4000
3000
2000
1000
0
BsmtFinSF1
100000
200000
300000
400000
500000
600000
700000
SalePrice

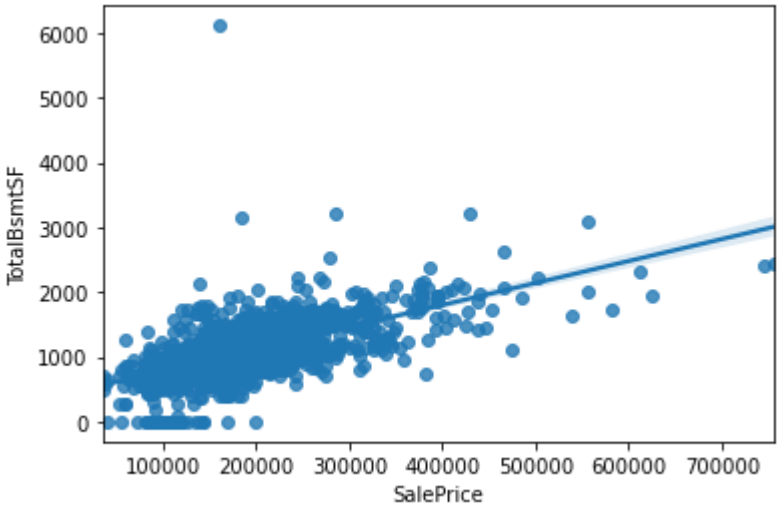

5000
4000
3000
2000
1000
0
0
200
400
600
800

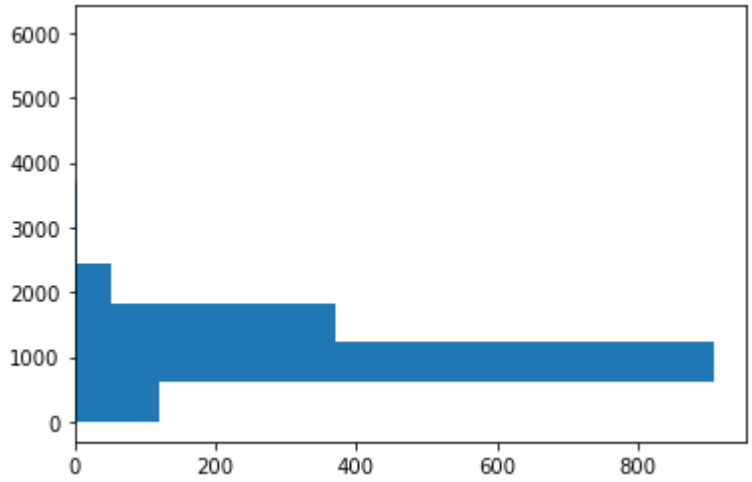

1400
1200
1000
800
600
400
200
0
BsmtFinSF2
100000
200000
300000
400000
500000
600000
700000
SalePrice

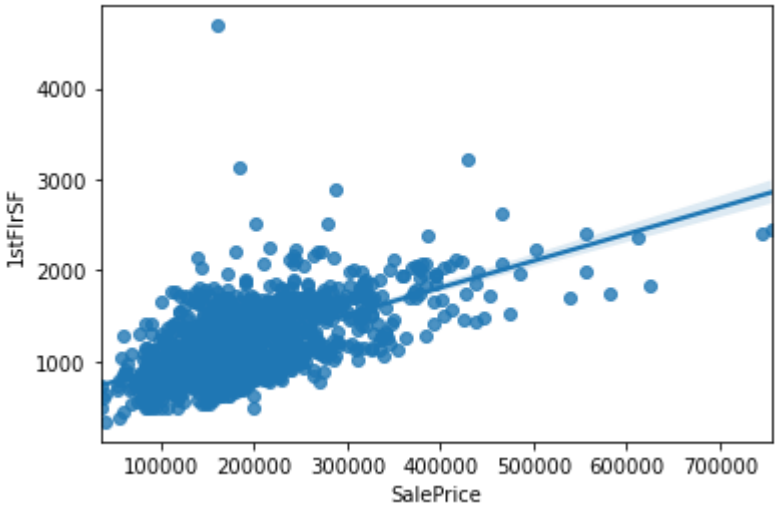

1400
1200
1000
800
600
400
200
0
0
200
400
600
800
1000
1200

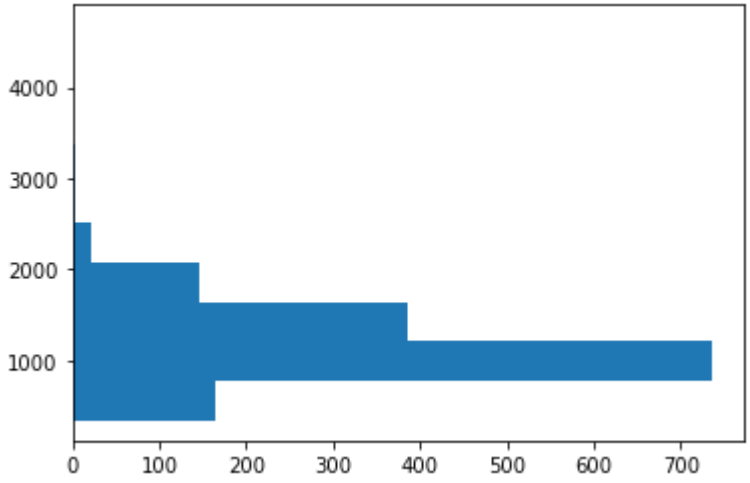

BsmtUnfSF
2000
1500
1000
500
0
100000
200000
300000
400000
500000
600000
700000
SalePrice

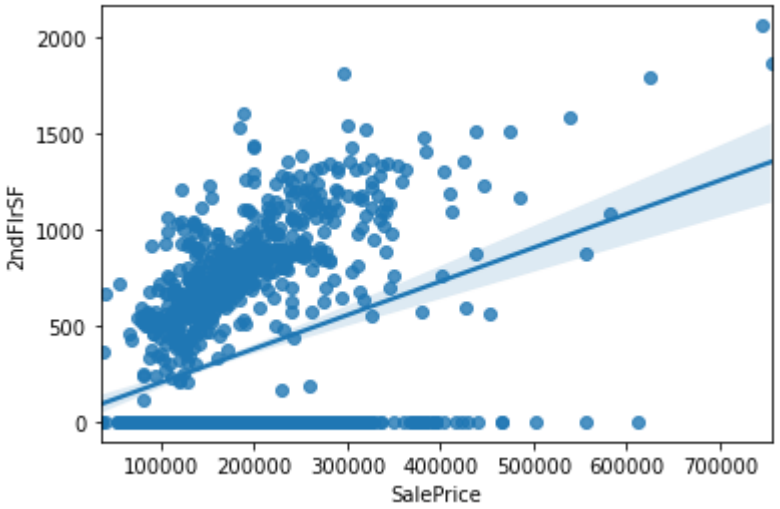

2000
1500
1000
500
0
0
50
100
150
200
250
300
350

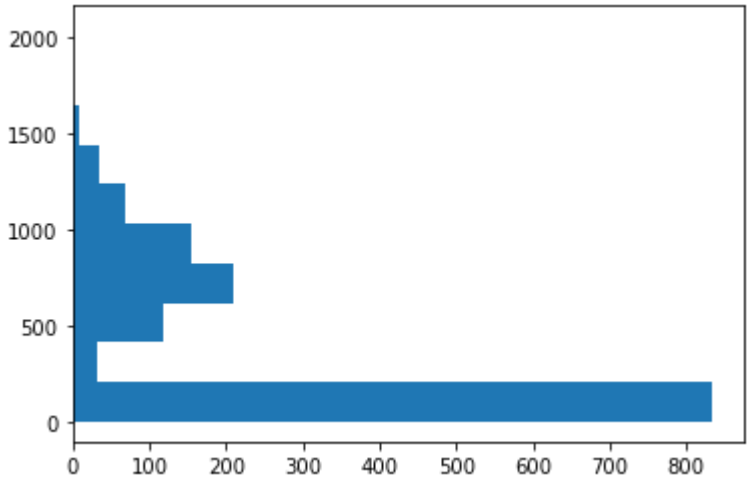

TotalBsmtSF
SalePrice
0
1000
2000
3000
4000
5000
6000
100000
200000
300000
400000
500000
600000
700000

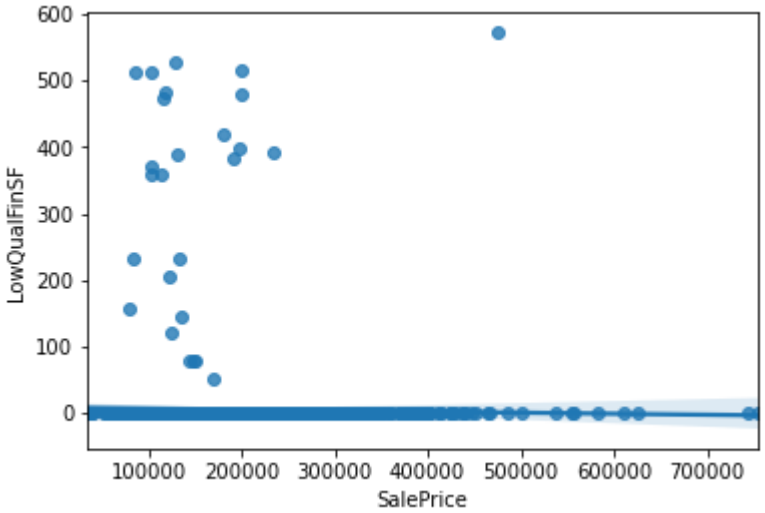

0
1000
2000
3000
4000
5000
6000
0
200
400
600
800

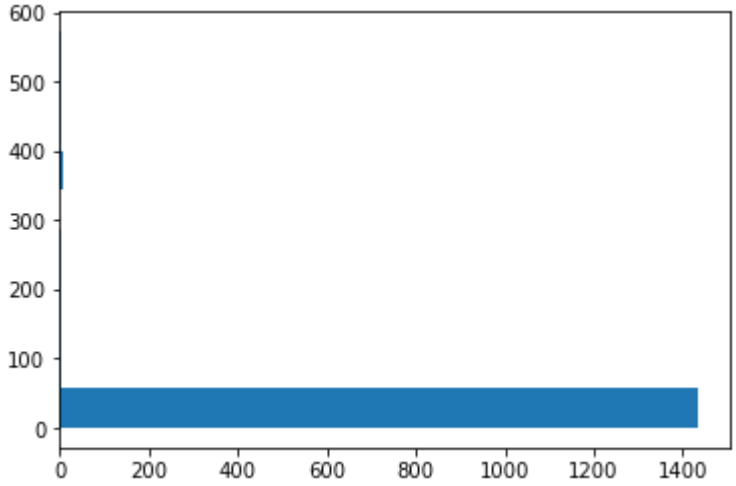

4000
3000
2000
1000
1stFlrSF
100000
200000
300000
400000
500000
600000
700000
SalePrice

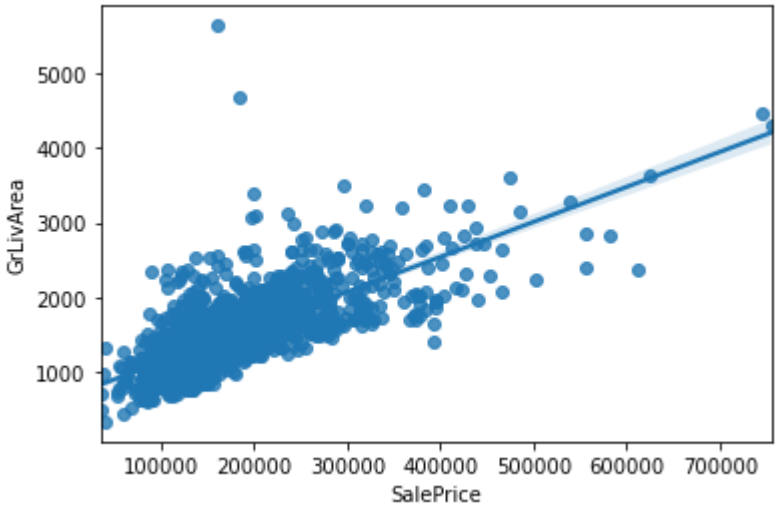

4000
3000
2000
1000
0
100
200
300
400
500
600
700

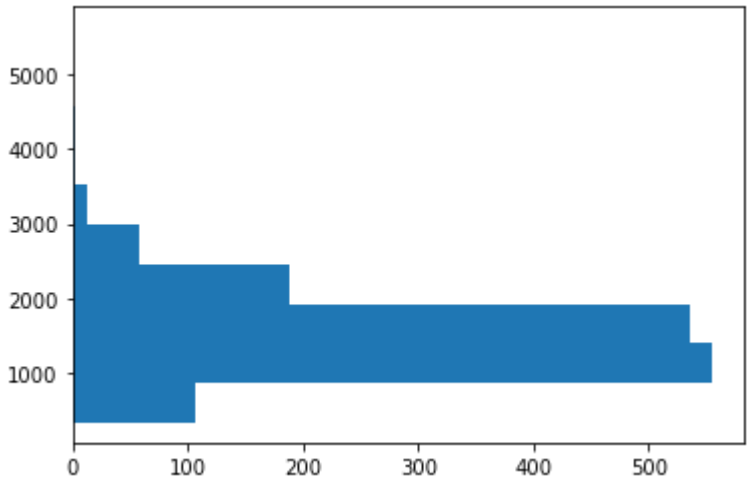

2ndFlrSF
SalePrice
0
500
1000
1500
2000
100000
200000
300000
400000
500000
600000
700000

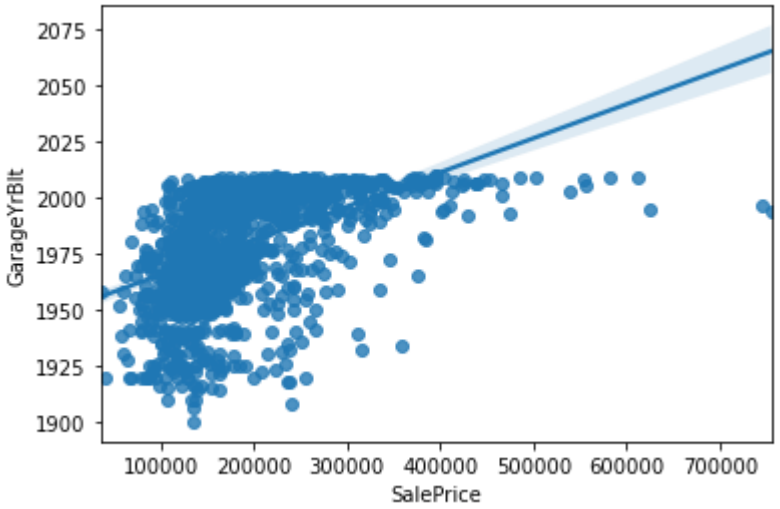

0
500
1000
1500
2000
0
100
200
300
400
500
600
700
800


LowQualFinSF
SalePrice
600
500
400
300
200
100
0
100000
200000
300000
400000
500000
600000
700000

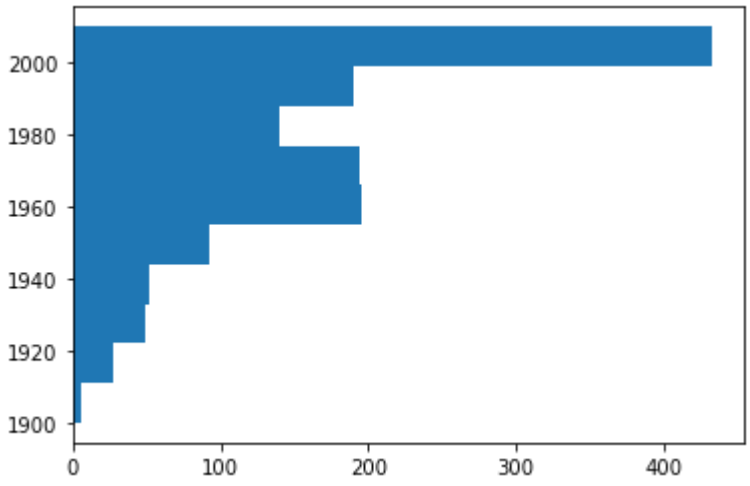

600
500
400
300
200
100
0
0
200
400
600
800
1000
1200
1400

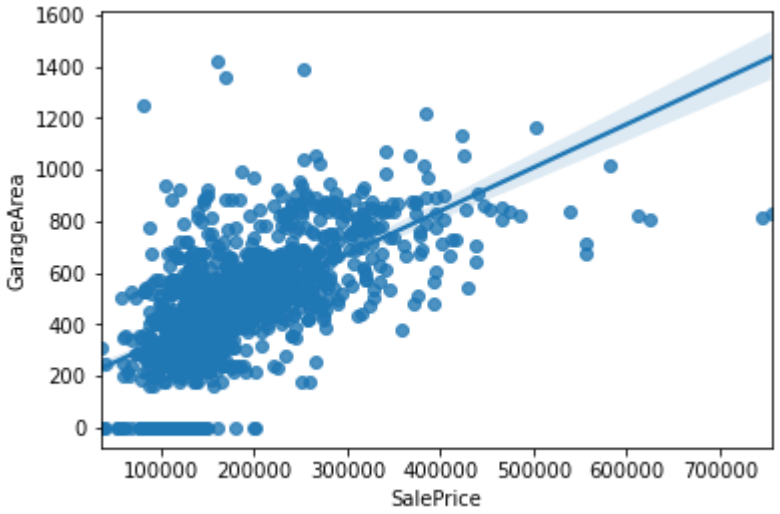

5000
4000
3000
2000
1000
GrLivArea
100000
200000
300000
400000
500000
600000
700000
SalePrice

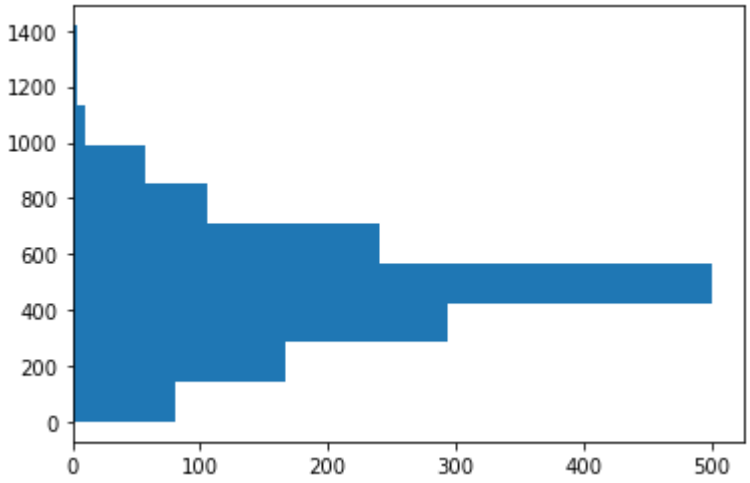

5000
4000
3000
2000
1000
0
100
200
300
400
500

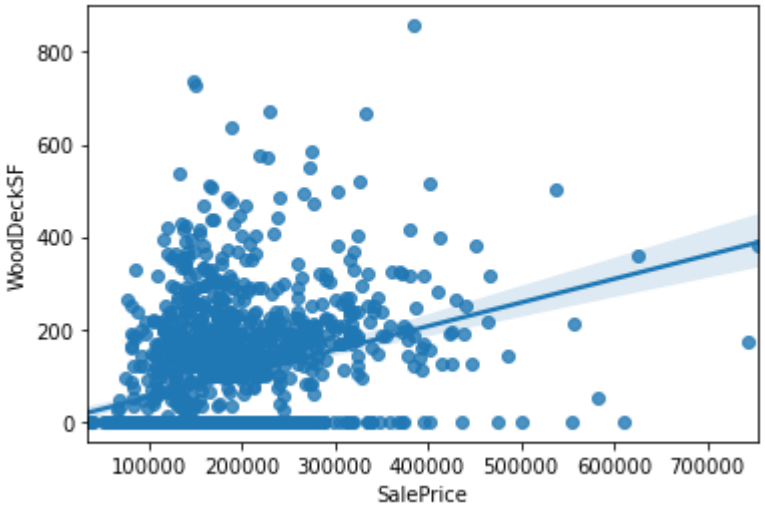


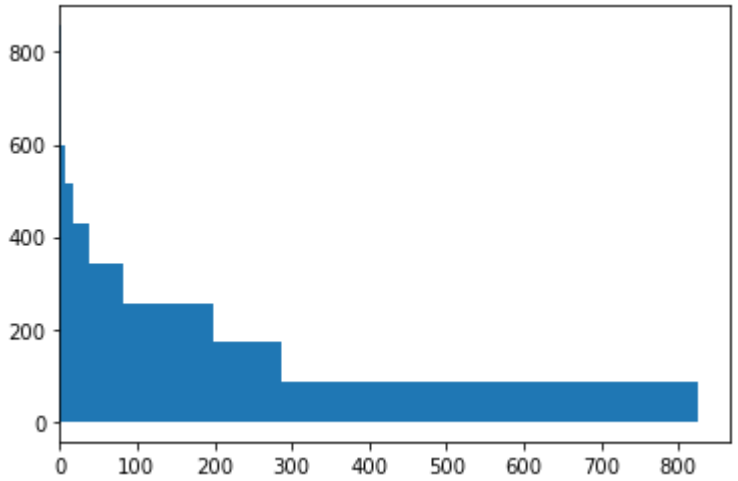

```
C:\Users\marky\anaconda3\envs\geo_env\lib\site-packages\numpy\lib\histograms.py:839: RuntimeWarning: invalid
value encountered in greater_equal
  keep = (tmp_a >= first_edge)
C:\Users\marky\anaconda3\envs\geo_env\lib\site-packages\numpy\lib\histograms.py:840: RuntimeWarning: invalid
value encountered in less_equal
  keep &= (tmp_a <= last_edge)
```

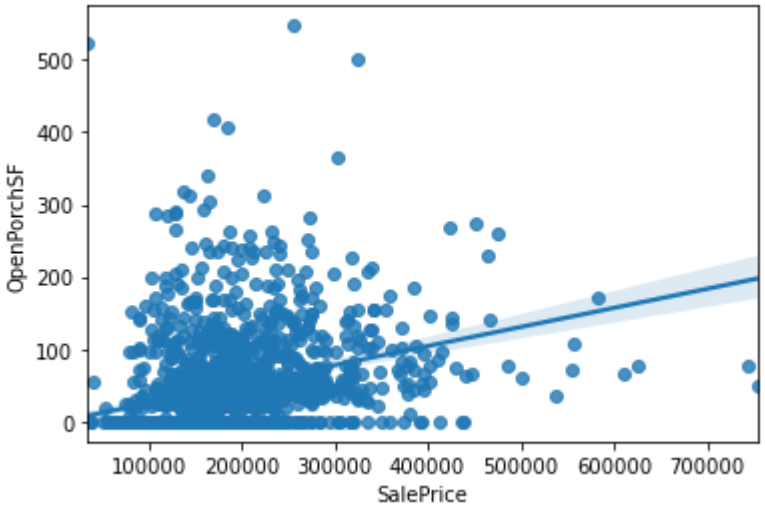


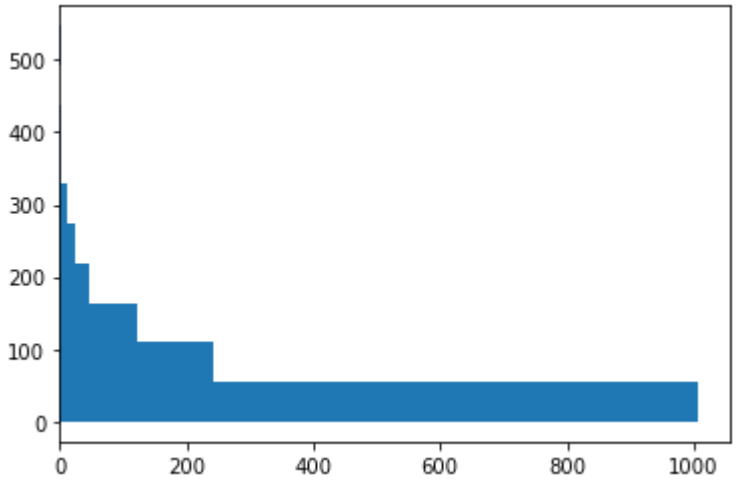

GarageArea
SalePrice
0
200
400
600
800
1000
1200
1400
1600
100000
200000
300000
400000
500000
600000
700000

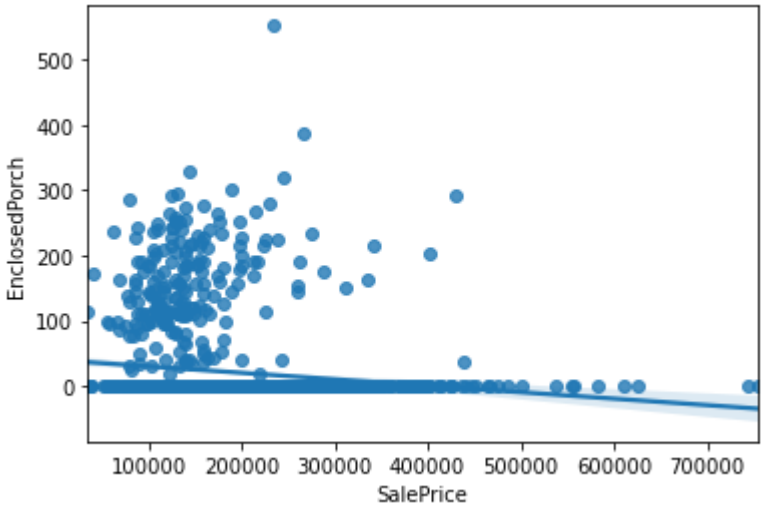

0
200
400
600
800
1000
1200
1400
0
100
200
300
400
500

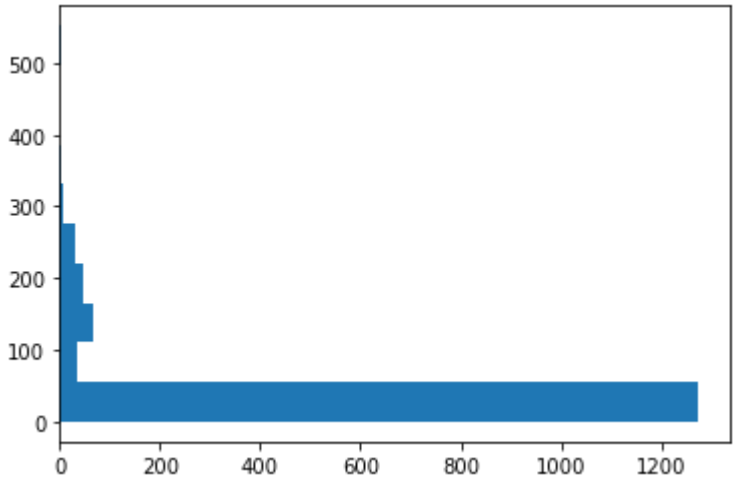

WoodDeckSF
SalePrice
0
200
400
600
800
100000
200000
300000
400000
500000
600000
700000

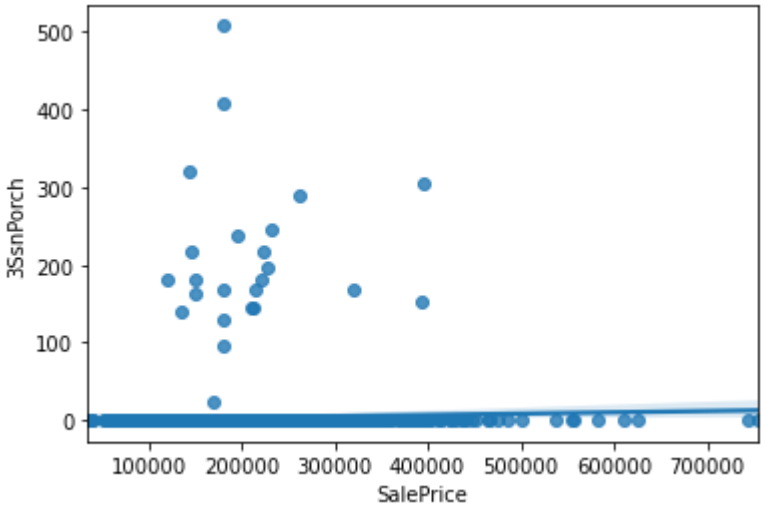

0
200
400
600
800
0
100
200
300
400
500
600
700
800

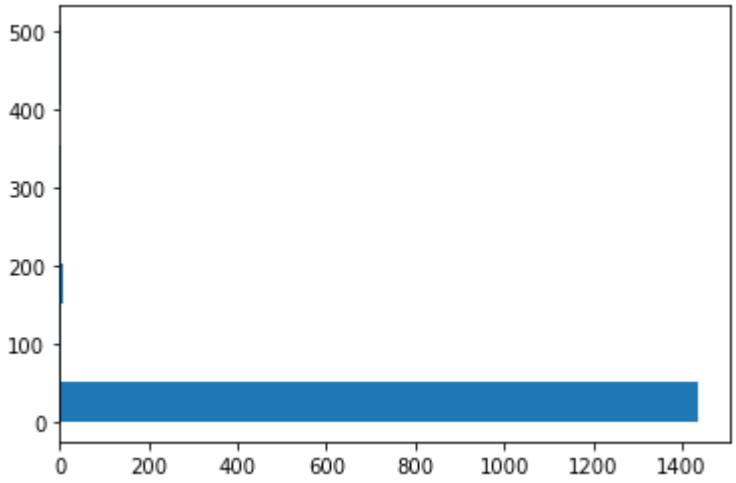

OpenPorchSF
SalePrice
0
100
200
300
400
500
100000
200000
300000
400000
500000
600000
700000

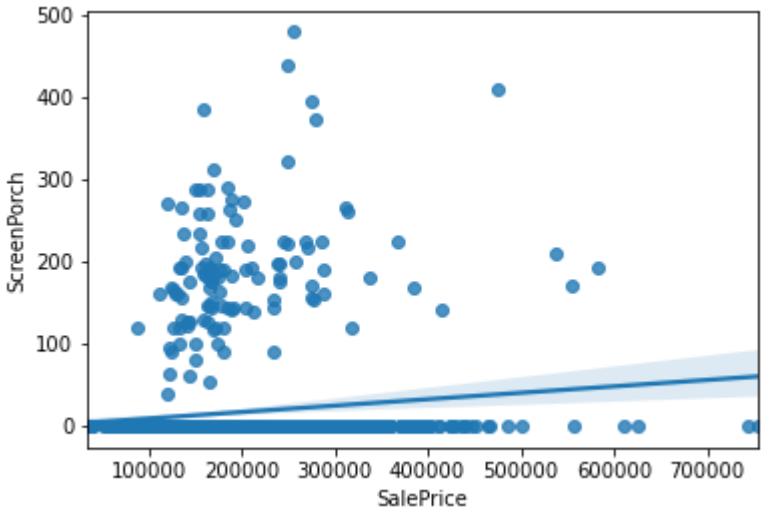

0
100
200
300
400
500
0
200
400
600
800
1000

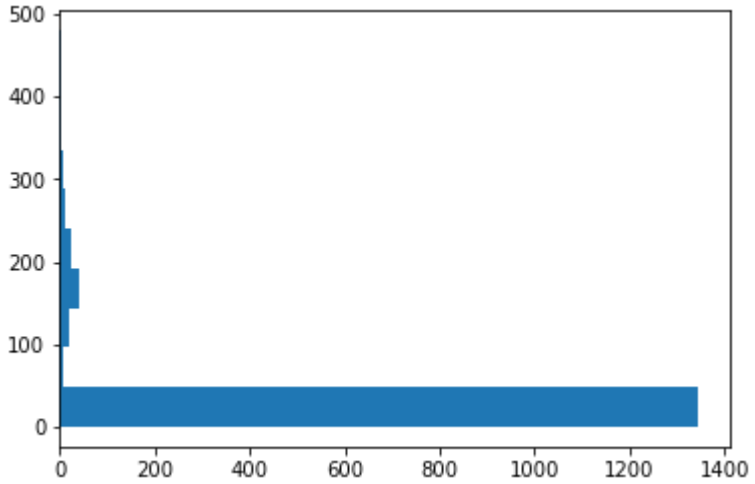

EnclosedPorch
SalePrice
500
400
300
200
100
0
100000
200000
300000
400000
500000
600000
700000

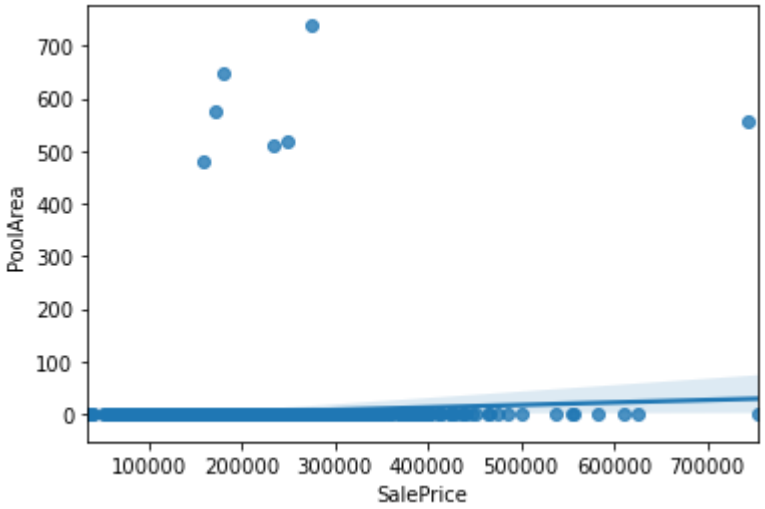

500
400
300
200
100
0
0
200
400
600
800
1000
1200

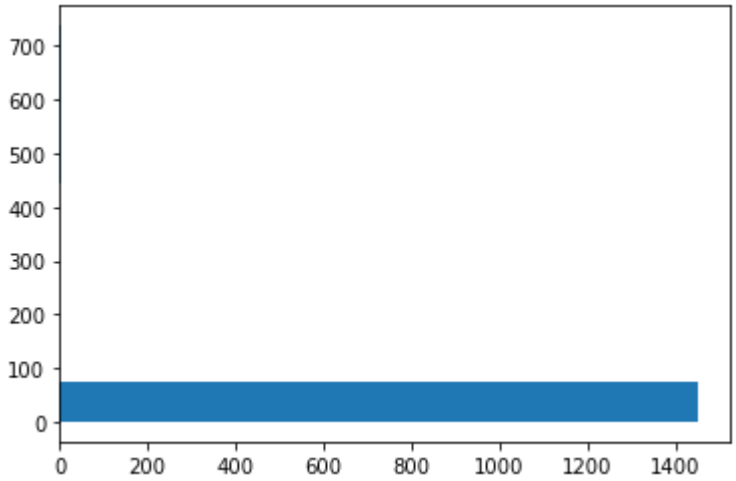

3SsnPorch
SalePrice
0
100
200
300
400
500
100000
200000
300000
400000
500000
600000
700000

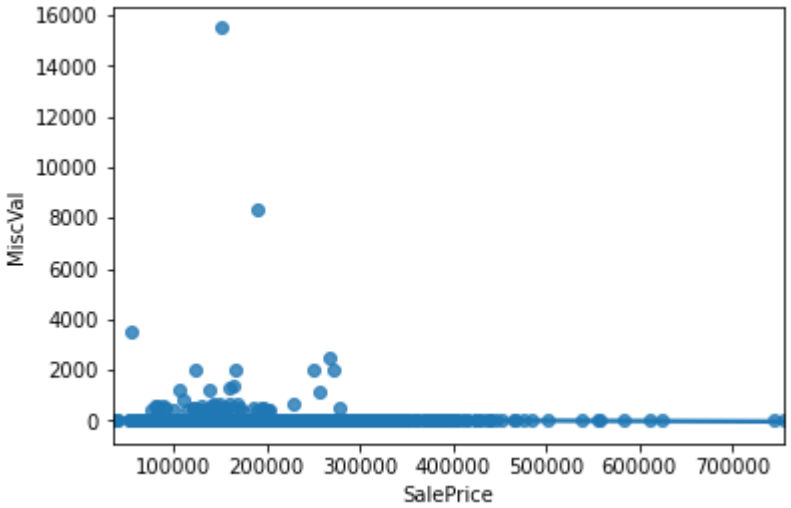

0
100
200
300
400
500
0
200
400
600
800
1000
1200
1400

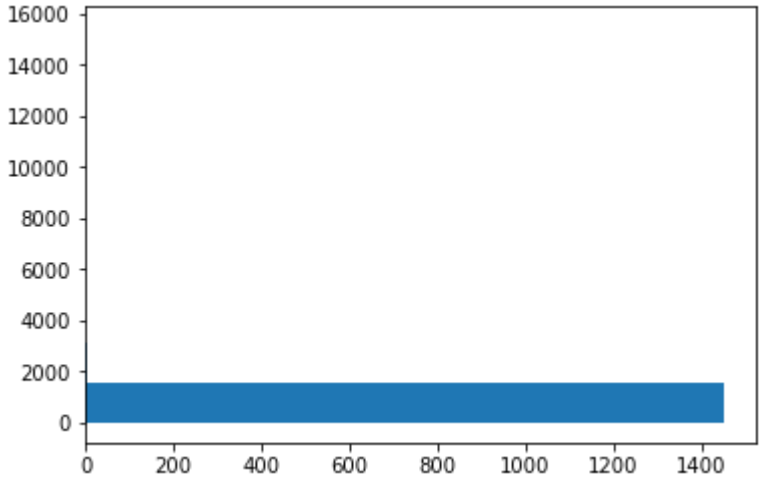

500
400
300
200
100
0
ScreenPorch
100000
200000
300000
400000
500000
600000
700000
SalePrice

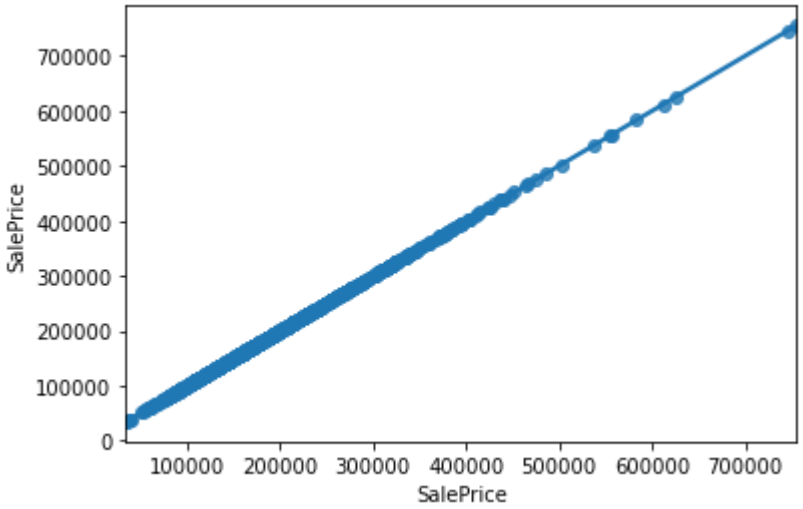

500
400
300
200
100
0
0
200
400
600
800
1000
1200
1400

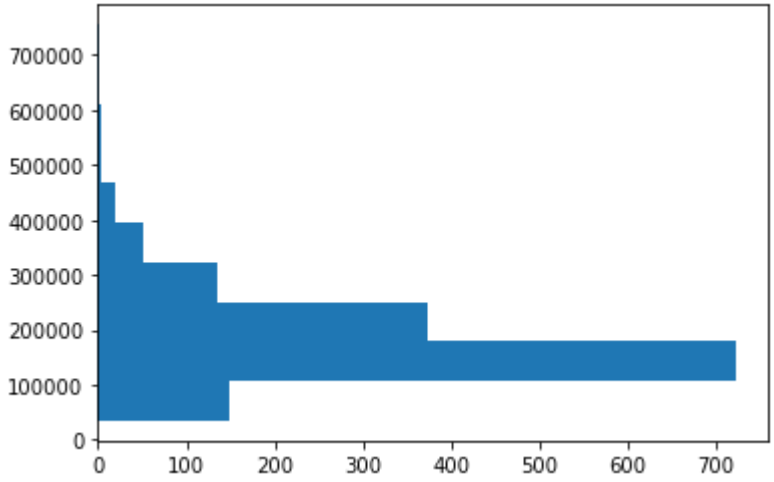

PoolArea
SalePrice
0
100
200
300
400
500
600
700
100000
200000
300000
400000
500000
600000
700000


0
100
200
300
400
500
600
700
0
200
400
600
800
1000
1200
1400


16000
14000
12000
10000
8000
6000
4000
2000
0
MiscVal
100000
200000
300000
400000
500000
600000
700000
SalePrice

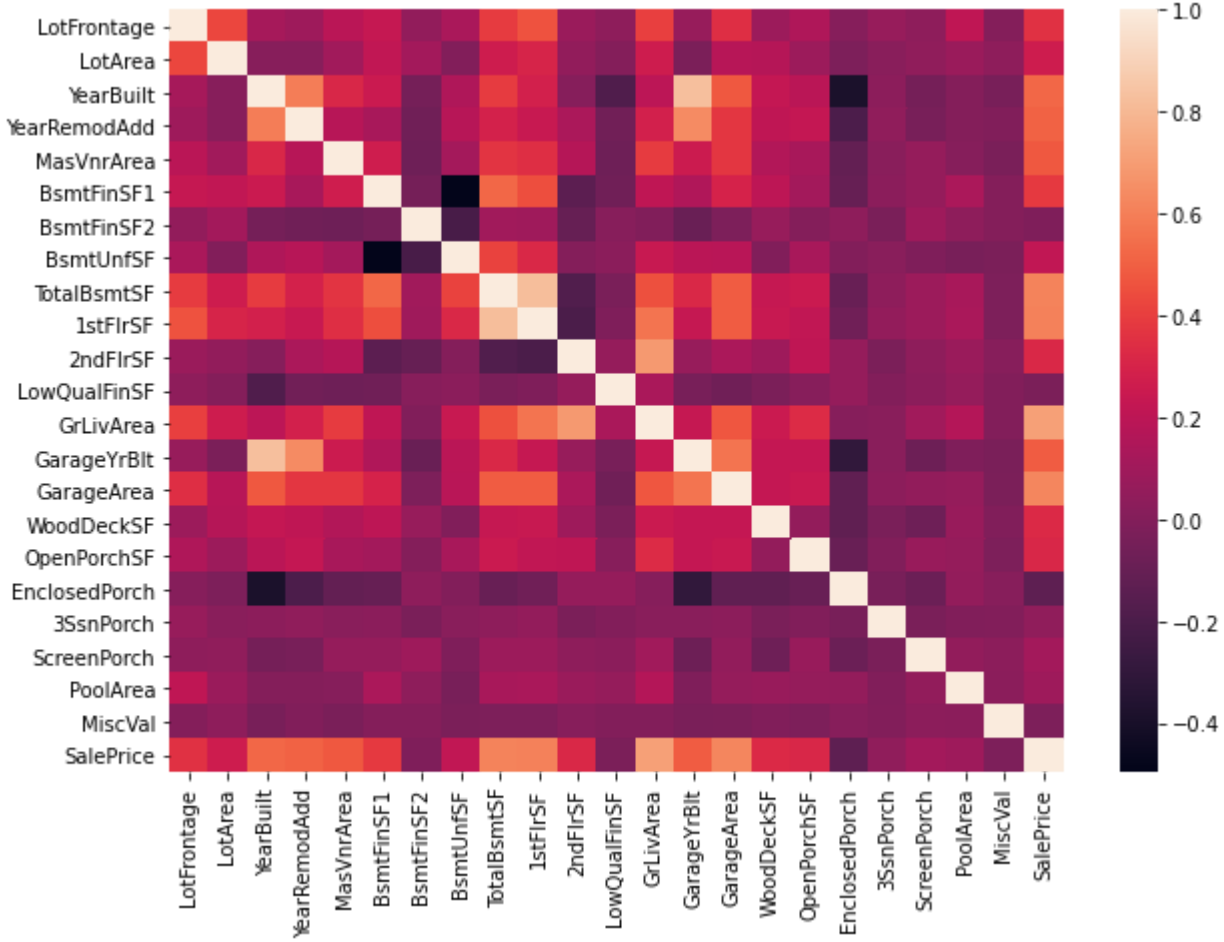

16000
14000
12000
10000
8000
6000
4000
2000
0
0
200
400
600
800
1000
1200
1400


700000
600000
500000
400000
300000
200000
100000
0
SalePrice
100000 200000 300000 400000 500000 600000 700000
SalePrice


700000
600000
500000
400000
300000
200000
100000
0
0 100 200 300 400 500 600 700

In [24]:

```
plt.subplots(figsize=(10,7))
df_house_price_train_numbers_continuous_fit=preprocessing.StandardScaler().fit(df_house_price_train_numbers_continuous).transform(df_house_price_train_numbers_continuous)
df_house_price_train_numbers_continuous_corr=pd.DataFrame(data=df_house_price_train_numbers_continuous,columns=df_house_price_train_numbers_continuous.columns).corr()
sns.heatmap(df_house_price_train_numbers_continuous_corr)
plt.show()
```

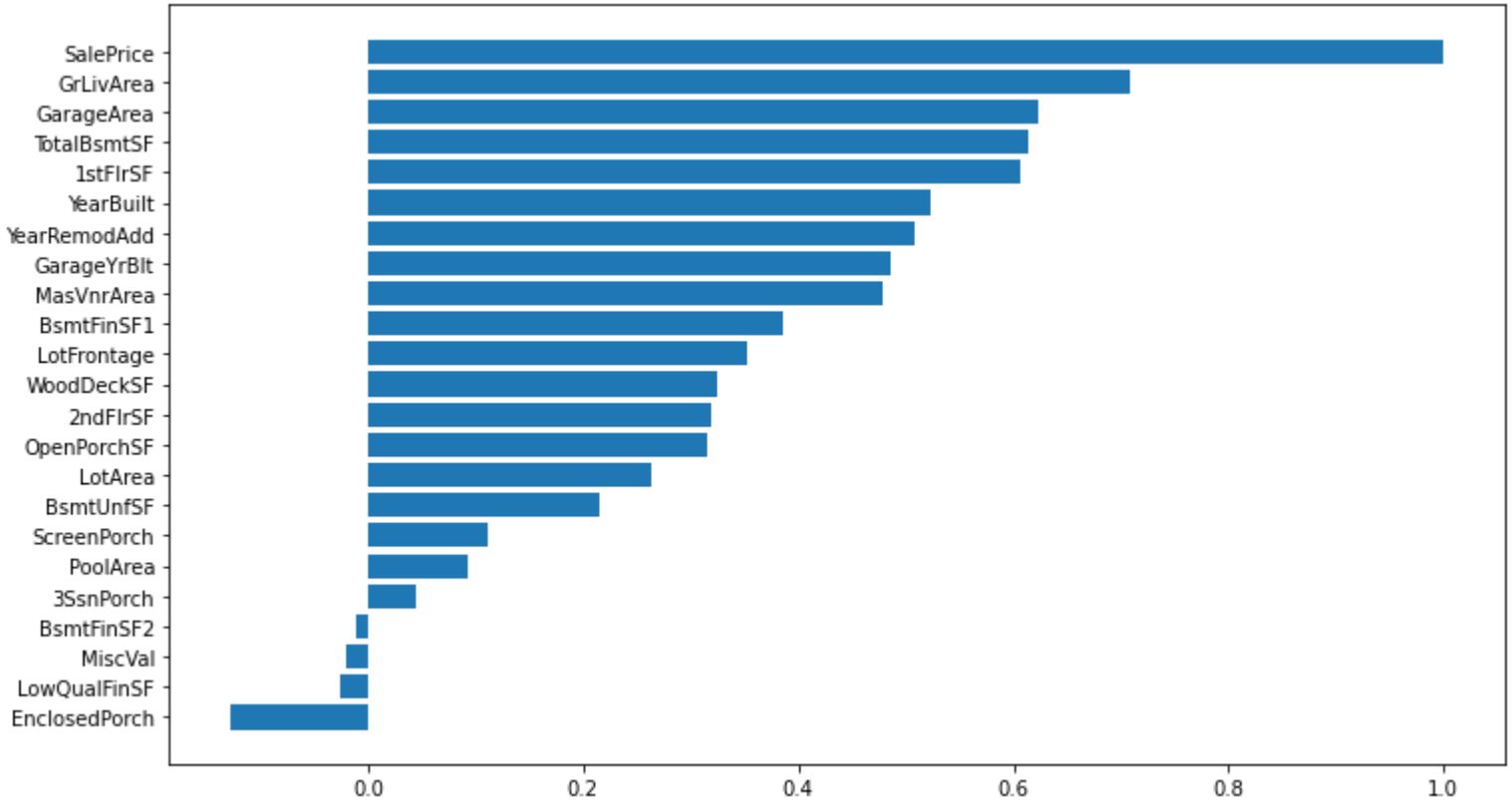


In [25]:

```
df_house_price_train_numbers_continuous_corr_saleprice=df_house_price_train_numbers_continuous_corr['SalePrice'].sort_val
ues()
```

In [26]:

```
plt.subplots(figsize=(12,7))
y_pos = np.arange(len(df_house_price_train_numbers_continuous_corr_saleprice))
plt.barh(y_pos, df_house_price_train_numbers_continuous_corr_saleprice,tick_label=df_house_price_train_numbers_continuous
_corr_saleprice.index)
plt.show()
```



In [27]:

```
df_house_price_train_numbers_ordinal_columns = list((Counter(df_house_price_train_numbers) - Counter(df_house_price_train
_numbers_continuous)).elements())
```

In [28]:

```
df_house_price_train_numbers_ordinal=df_house_price_train_numbers[df_house_price_train_numbers_ordinal_columns]
```

In [29]:

```
df_house_price_train_numbers_ordinal['SalePrice']=df_house_price_train_numbers_continuous['SalePrice']
```

```
<ipython-input-29-049508acd956>:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#r
eturning-a-view-versus-a-copy
  df_house_price_train_numbers_ordinal['SalePrice']=df_house_price_train_numbers_continuous['SalePrice']
```

In [30]:

```
df_house_price_train_numbers_ordinal
```

Out[30]:

	MSSubClass	OverallQual	OverallCond	BsmtFullBath	BsmtHalfBath	FullBath	HalfBath	BedroomAbvGr	KitchenAbvGr	TotRmsA
0	60	7	5	1	0	2	1	3	1	
1	20	6	8	0	1	2	0	3	1	
2	60	7	5	1	0	2	1	3	1	
3	70	7	5	1	0	1	0	3	1	
4	60	8	5	1	0	2	1	4	1	
...	...	...	...	...	...	...	...	...	...	
1455	60	6	5	0	0	2	1	3	1	
1456	20	6	6	1	0	2	0	3	1	
1457	70	7	9	0	0	2	0	4	1	
1458	20	5	6	1	0	1	0	2	1	
1459	20	5	6	1	0	1	1	3	1	

1460 rows × 15 columns

In [31]:

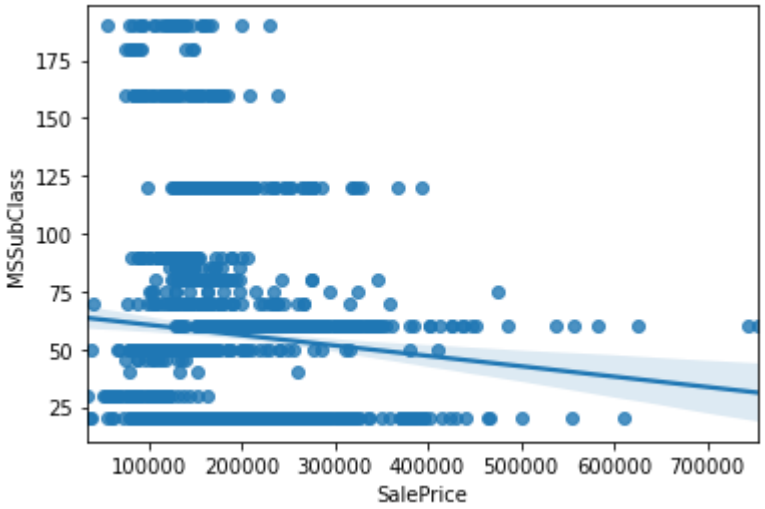
```
df_house_price_train_numbers_ordinal.describe()
```

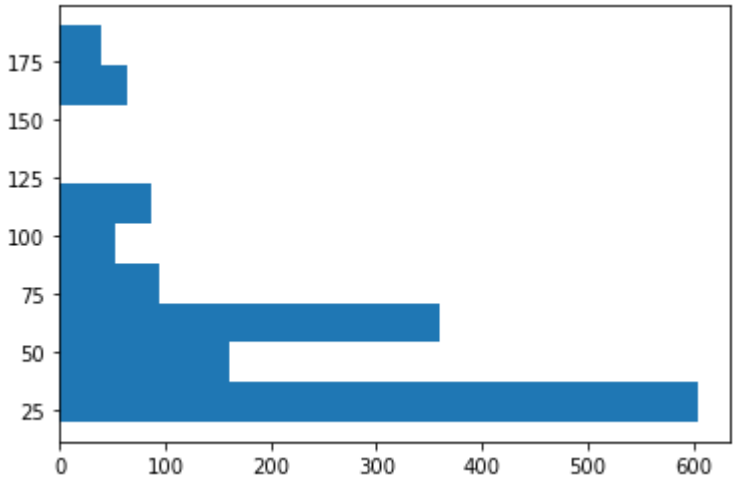
Out[31]:

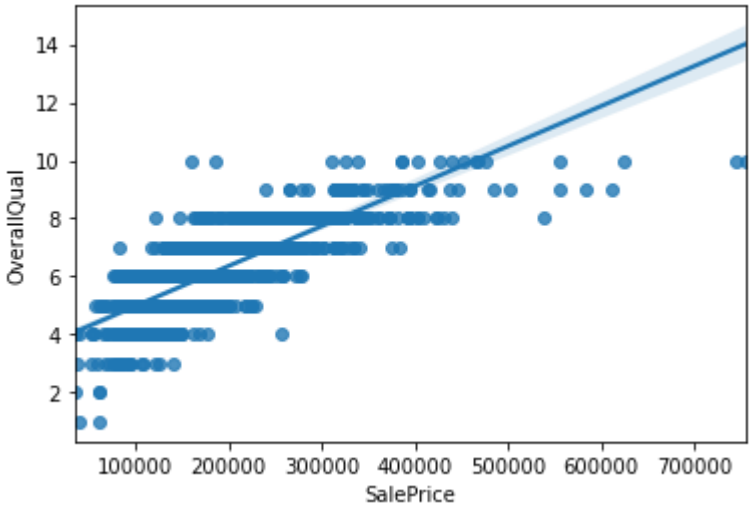
	MSSubClass	OverallQual	OverallCond	BsmtFullBath	BsmtHalfBath	FullBath	HalfBath	BedroomAbvGr	KitchenAbvGr
count	1460.000000	1460.000000	1460.000000	1460.000000	1460.000000	1460.000000	1460.000000	1460.000000	1460.000000
mean	56.897260	6.099315	5.575342	0.425342	0.057534	1.565068	0.382877	2.866438	1.046575
std	42.300571	1.382997	1.112799	0.518911	0.238753	0.550916	0.502885	0.815778	0.220338
min	20.000000	1.000000	1.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	20.000000	5.000000	5.000000	0.000000	0.000000	1.000000	0.000000	2.000000	1.000000
50%	50.000000	6.000000	5.000000	0.000000	0.000000	2.000000	0.000000	3.000000	1.000000
75%	70.000000	7.000000	6.000000	1.000000	0.000000	2.000000	1.000000	3.000000	1.000000
max	190.000000	10.000000	9.000000	3.000000	2.000000	3.000000	2.000000	8.000000	3.000000

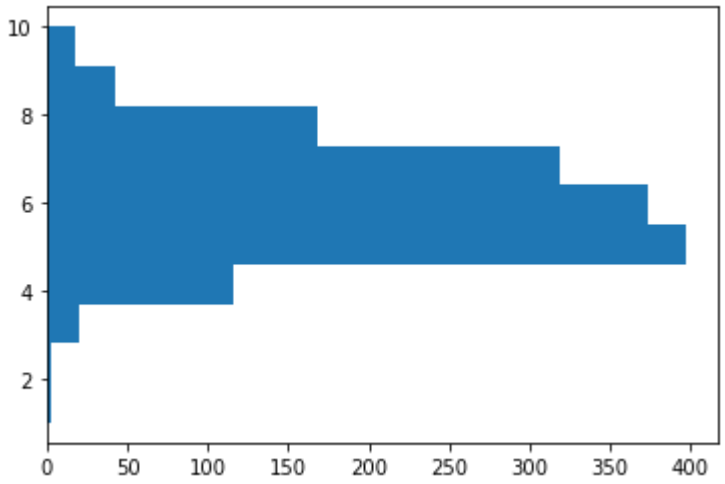
In [32]:

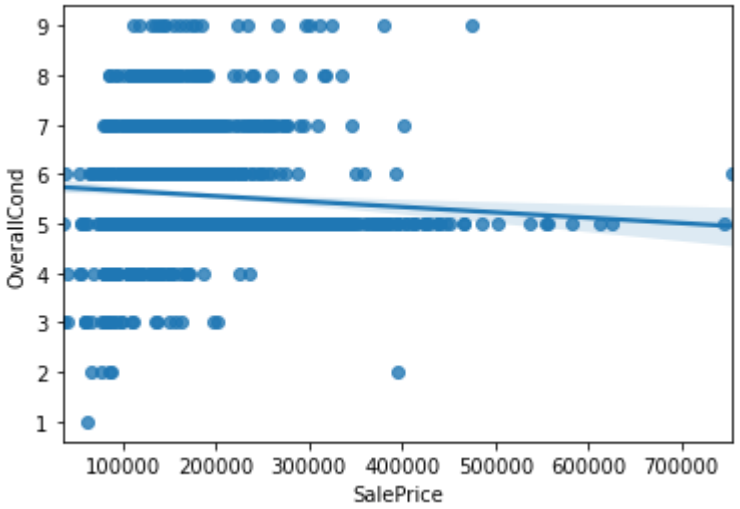
```
for i in range(len(df_house_price_train_numbers_ordinal.columns)):
    sns.regplot(y=df_house_price_train_numbers_ordinal.columns[i],x='SalePrice',data=df_house_price_train_numbers_ordinal
)
    plt.show()
    df_house_price_train_numbers_ordinal_sorted=df_house_price_train_numbers_ordinal[df_house_price_train_numbers_ordinal
.columns[i]].value_counts(ascending=True)
    y_pos = np.arange(len(df_house_price_train_numbers_ordinal_sorted))
    plt.hist(df_house_price_train_numbers_ordinal[df_house_price_train_numbers_ordinal.columns[i]],orientation='horizonta
l')
    plt.show()
```

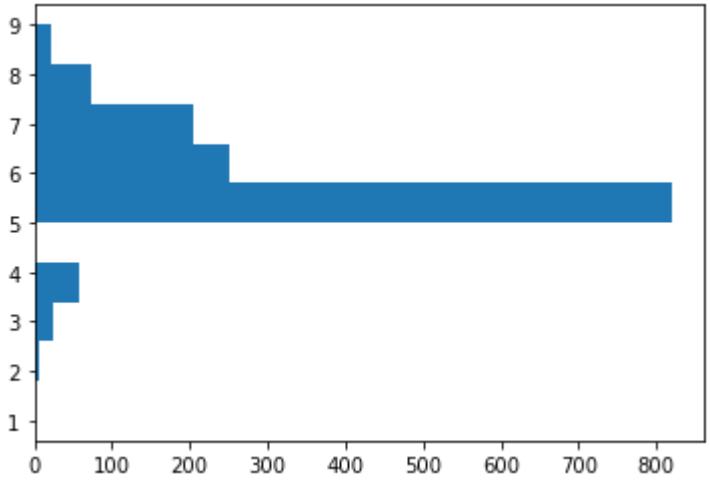

MSSubClass
175
150
125
100
75
50
25
100000
200000
300000
400000
500000
600000
700000
SalePrice

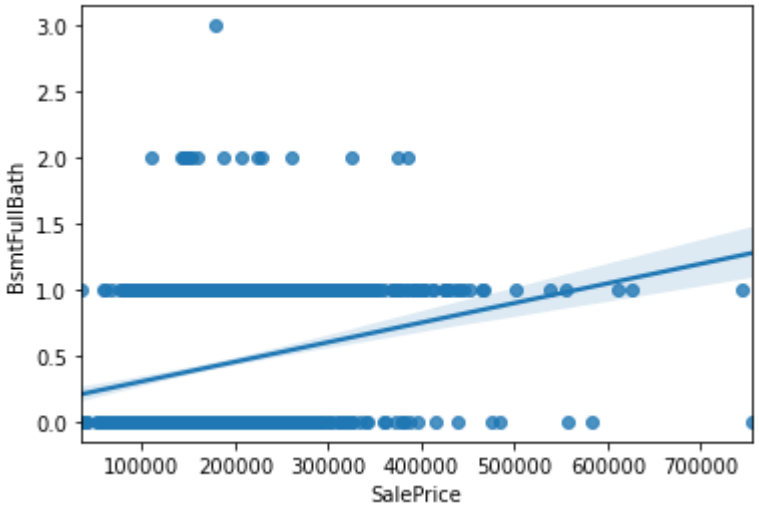

175
150
125
100
75
50
25
0
100
200
300
400
500
600

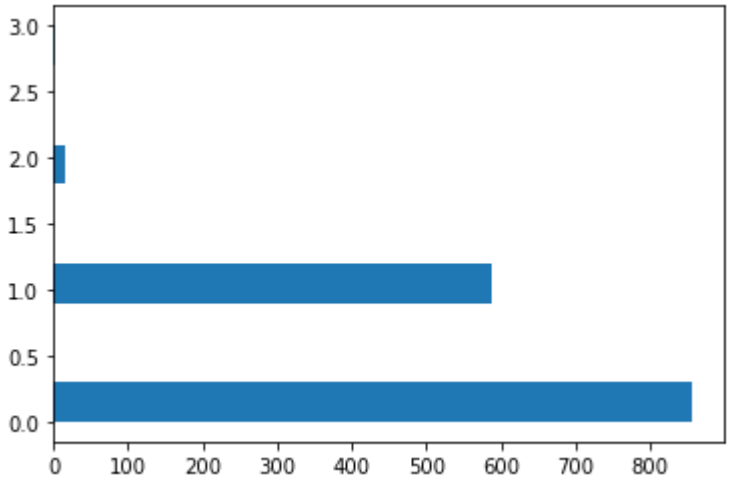

OverallQual
SalePrice
14
12
10
8
6
4
2
100000
200000
300000
400000
500000
600000
700000

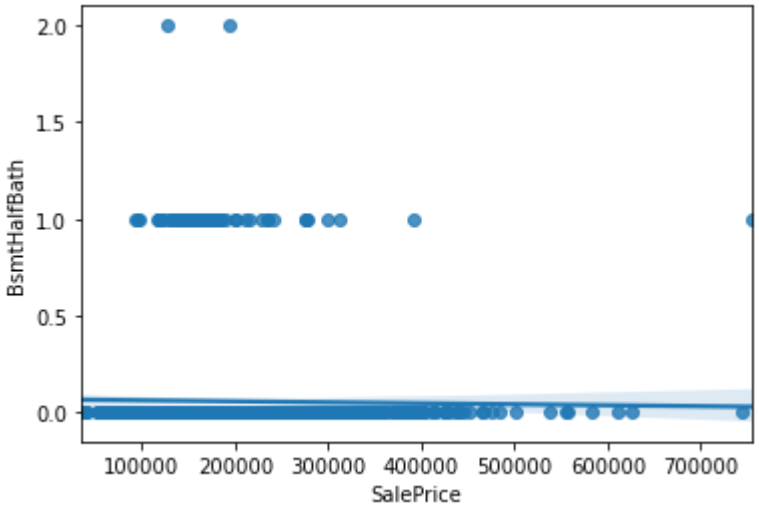

10
8
6
4
2
0
50
100
150
200
250
300
350
400

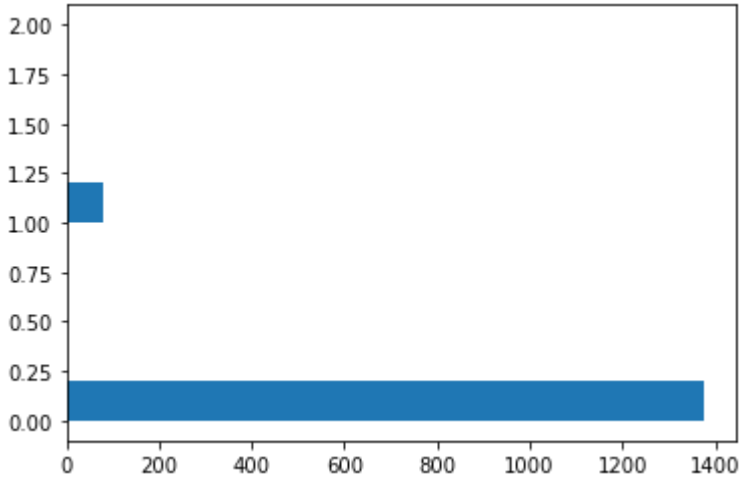

OverallCond
SalePrice
9
8
7
6
5
4
3
2
1
100000
200000
300000
400000
500000
600000
700000

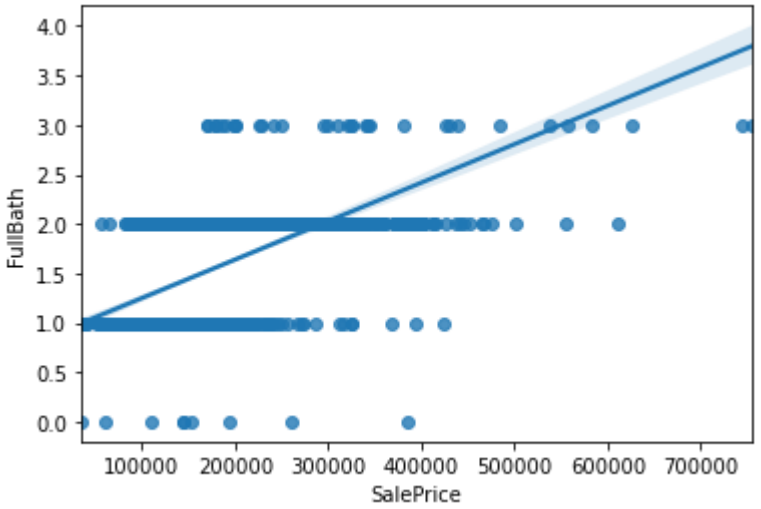

9
8
7
6
5
4
3
2
1
0
100
200
300
400
500
600
700
800

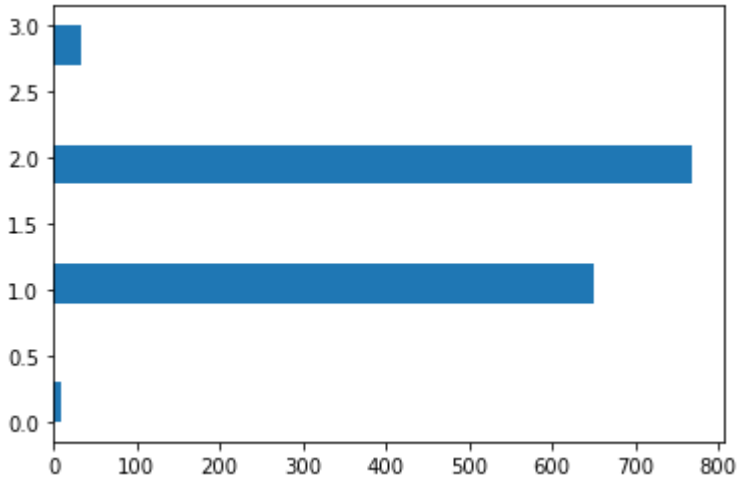

BsmtFullBath
SalePrice
3.0
2.5
2.0
1.5
1.0
0.5
0.0
100000
200000
300000
400000
500000
600000
700000

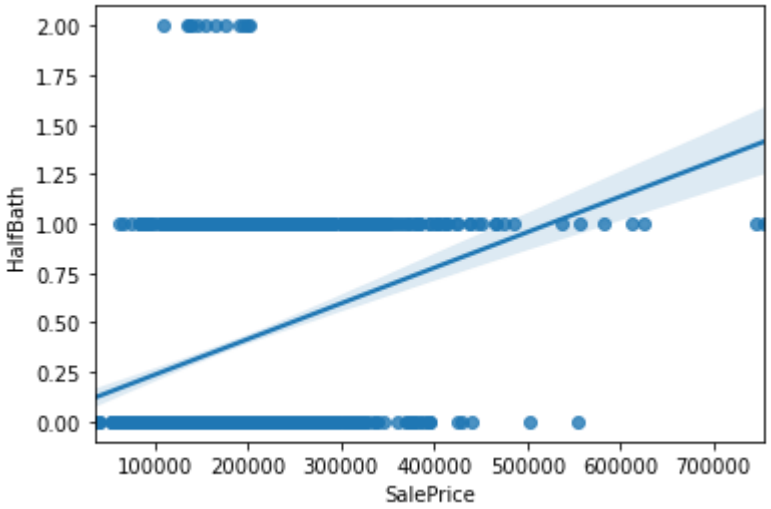

3.0
2.5
2.0
1.5
1.0
0.5
0.0
0
100
200
300
400
500
600
700
800

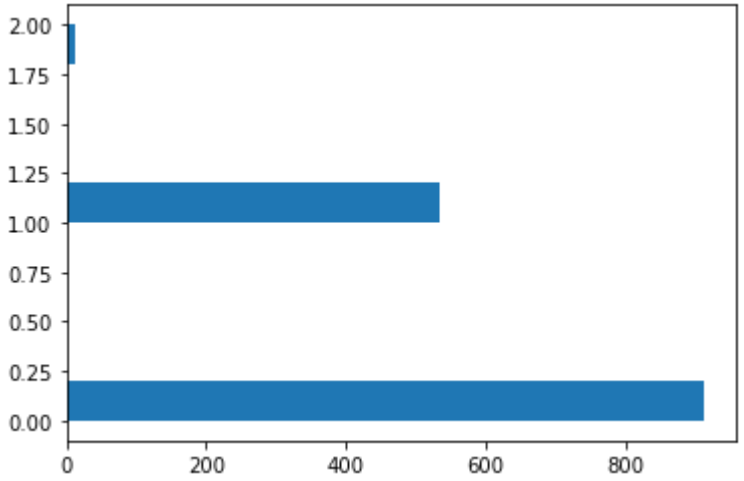

2.0
1.5
1.0
0.5
0.0
BsmtHalfBath
100000
200000
300000
400000
500000
600000
700000
SalePrice

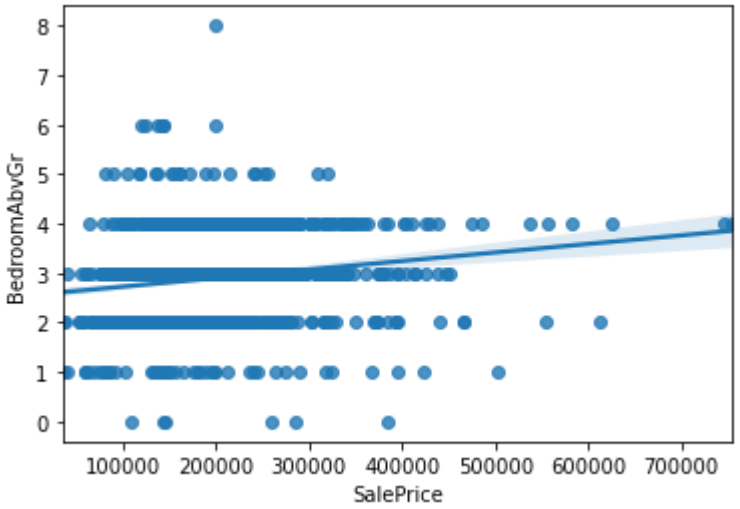

2.00
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00
0
200
400
600
800
1000
1200
1400

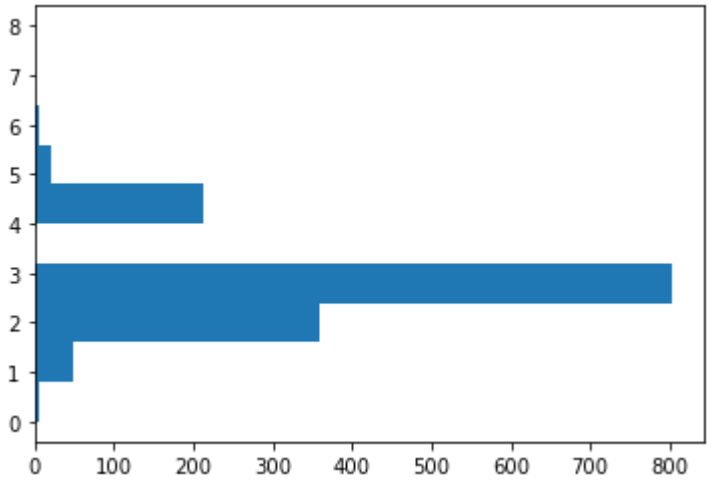

4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
FullBath
100000
200000
300000
400000
500000
600000
700000
SalePrice

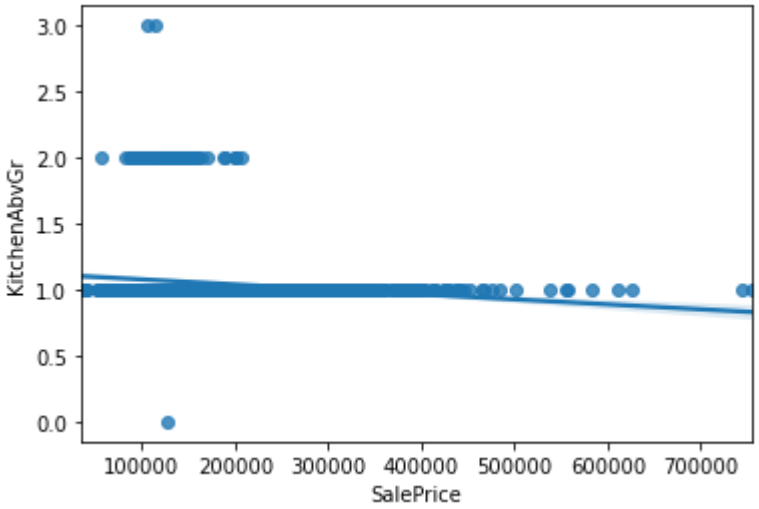

3.0
2.5
2.0
1.5
1.0
0.5
0.0
0
100
200
300
400
500
600
700
800

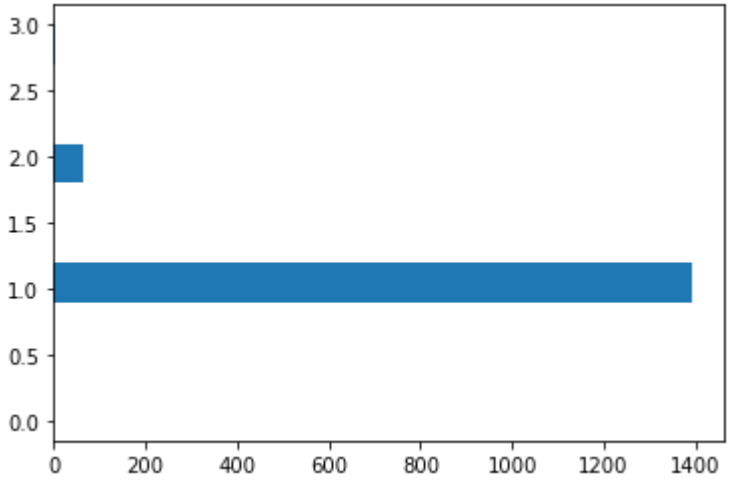

2.00
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00
HalfBath
100000
200000
300000
400000
500000
600000
700000
SalePrice

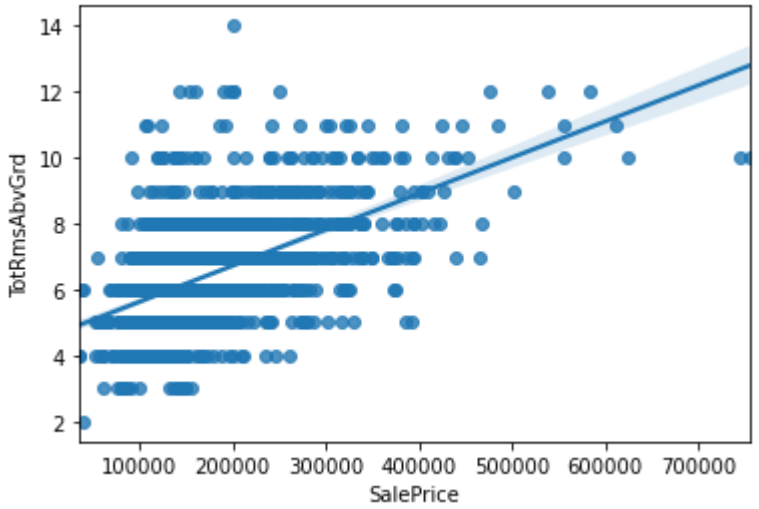

2.00
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00
0
200
400
600
800

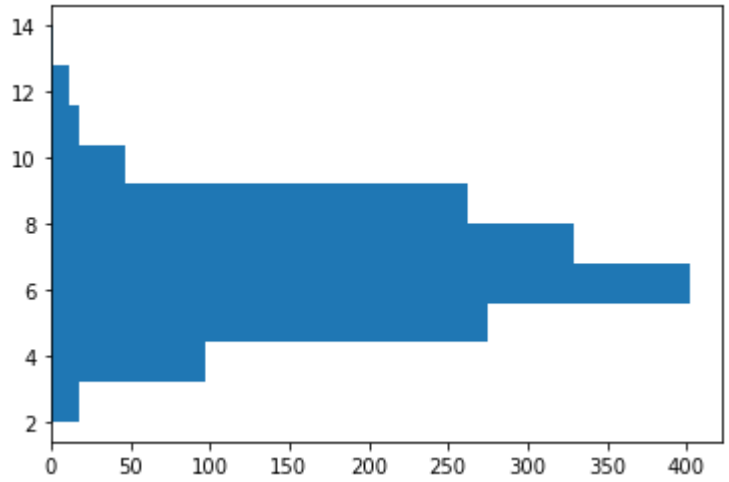

BedroomAbvGr
SalePrice
0
1
2
3
4
5
6
7
8
100000
200000
300000
400000
500000
600000
700000

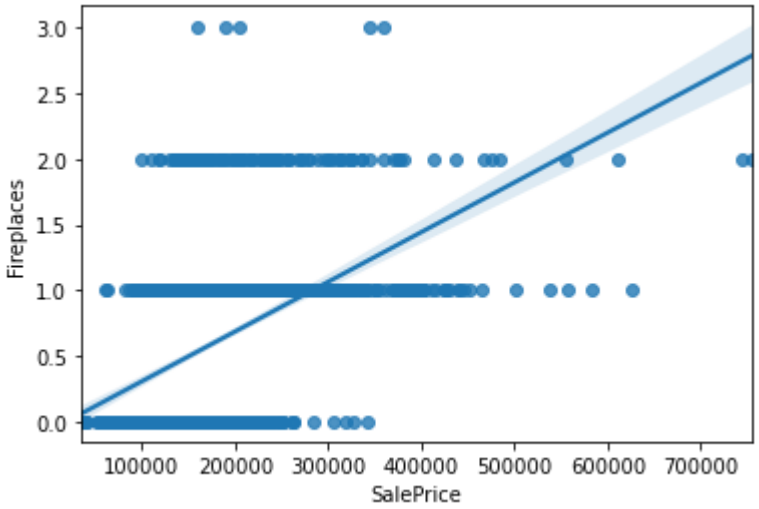

0
1
2
3
4
5
6
7
8
0
100
200
300
400
500
600
700
800

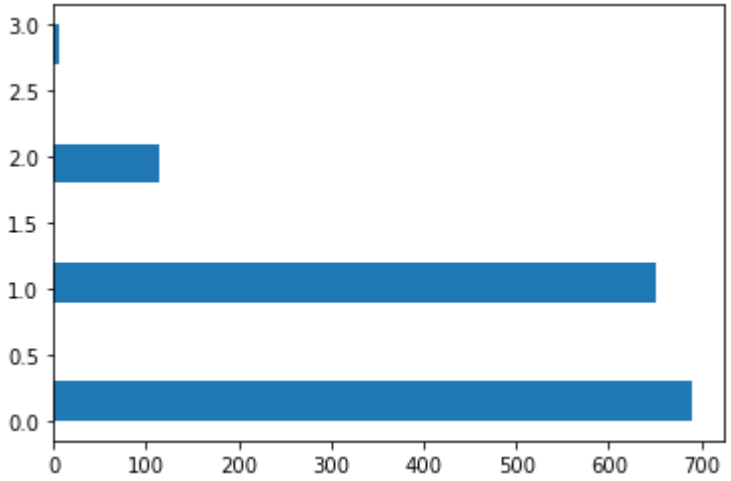

KitchenAbvGr
SalePrice
3.0
2.5
2.0
1.5
1.0
0.5
0.0
100000
200000
300000
400000
500000
600000
700000

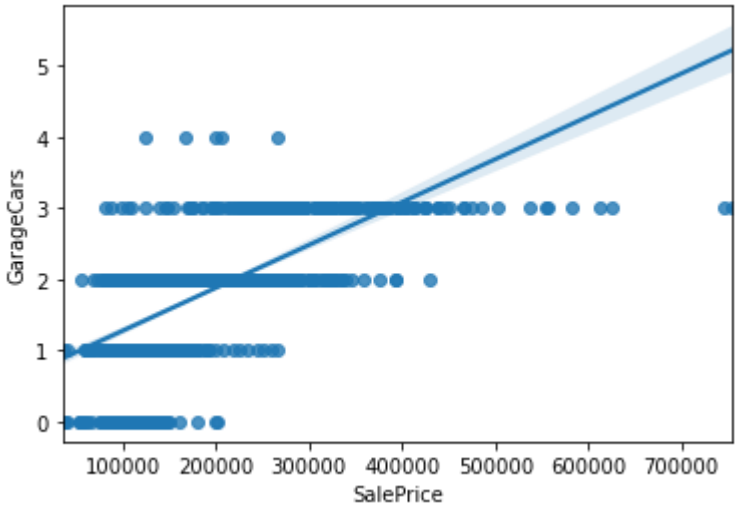

3.0
2.5
2.0
1.5
1.0
0.5
0.0
0
200
400
600
800
1000
1200
1400

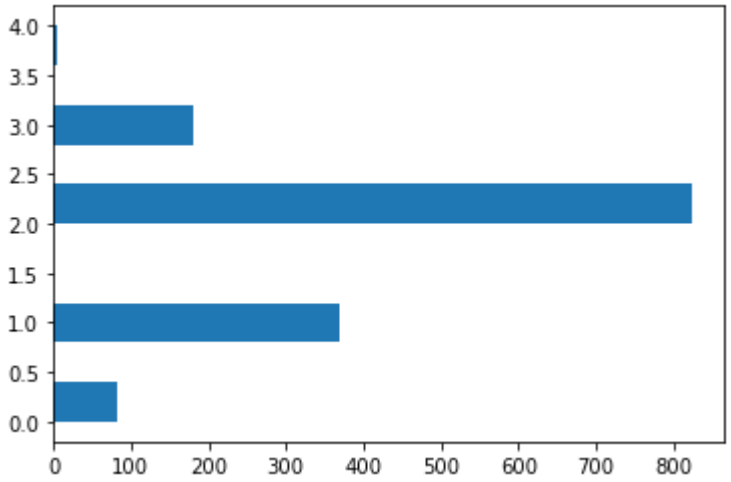

TotRmsAbvGrd
SalePrice
14
12
10
8
6
4
2
100000
200000
300000
400000
500000
600000
700000

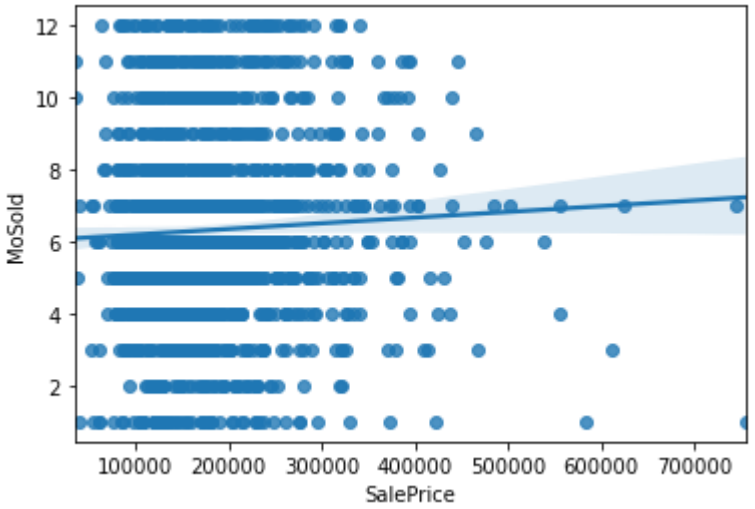

14
12
10
8
6
4
2
0
50
100
150
200
250
300
350
400

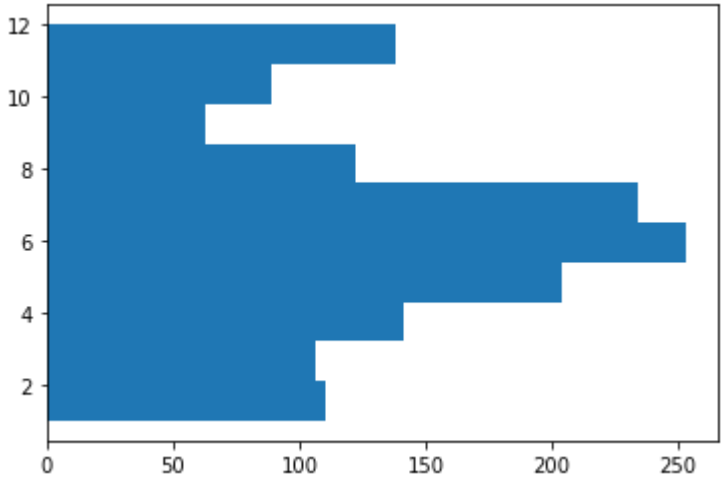

3.0
2.5
2.0
1.5
1.0
0.5
0.0
Fireplaces
100000
200000
300000
400000
500000
600000
700000
SalePrice

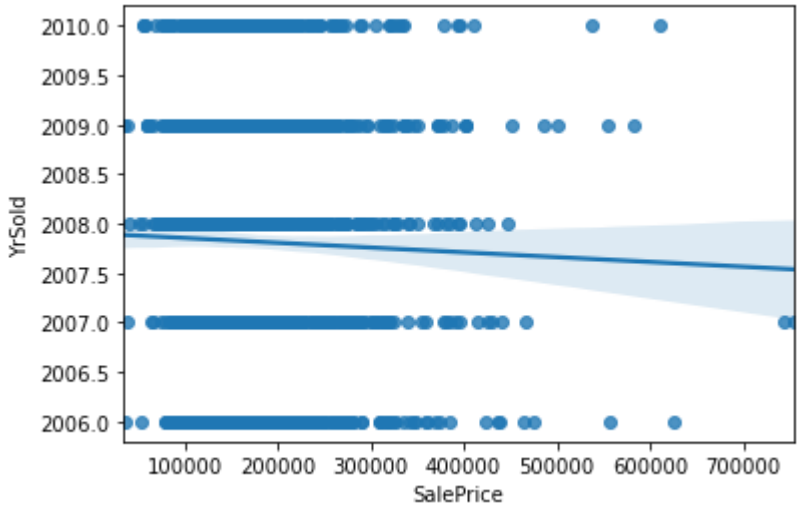

3.0
2.5
2.0
1.5
1.0
0.5
0.0
0
100
200
300
400
500
600
700

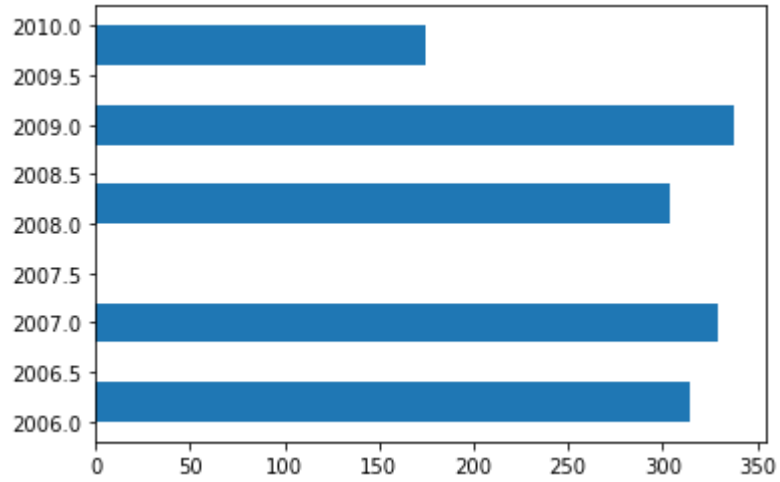

GarageCars
SalePrice
0
1
2
3
4
5
100000
200000
300000
400000
500000
600000
700000

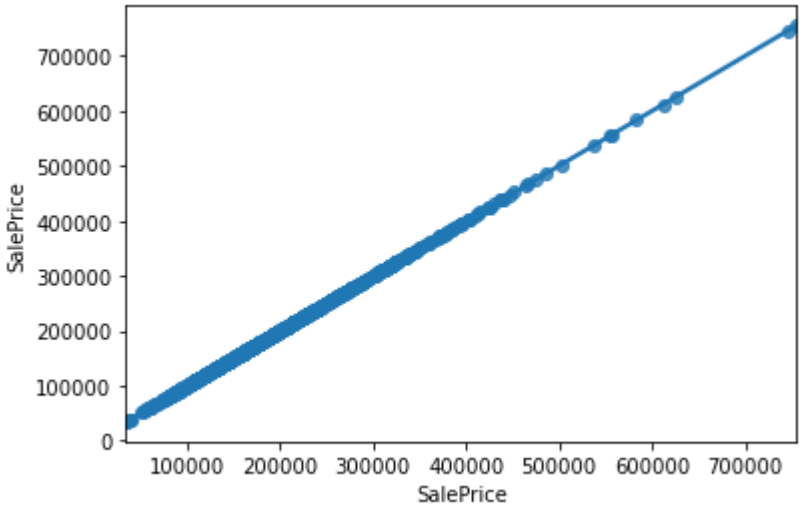

0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
0
100
200
300
400
500
600
700
800

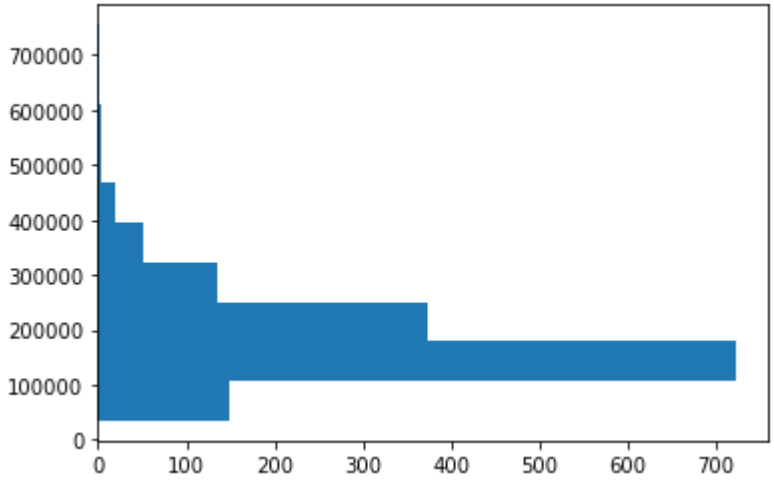

MoSold
SalePrice
12
10
8
6
4
2
100000
200000
300000
400000
500000
600000
700000


12
10
8
6
4
2
0
50
100
150
200
250


2010.0
2009.5
2009.0
2008.5
2008.0
2007.5
2007.0
2006.5
2006.0
YrSold
100000
200000
300000
400000
500000
600000
700000
SalePrice

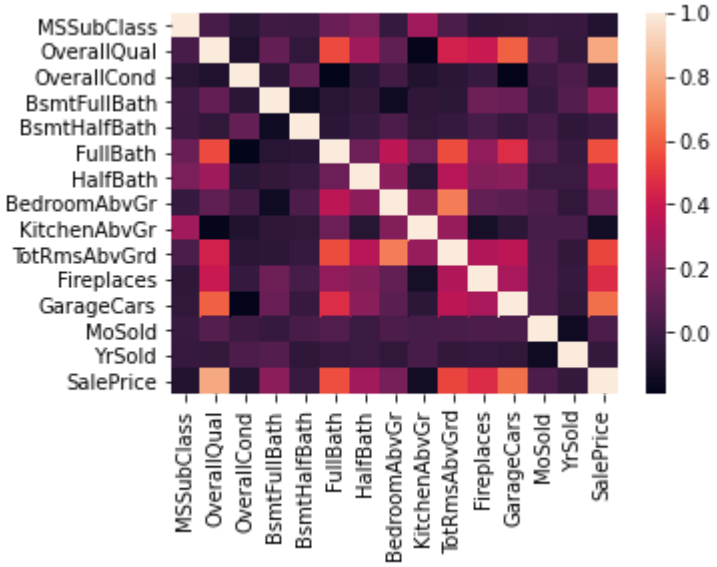

2010.0
2009.5
2009.0
2008.5
2008.0
2007.5
2007.0
2006.5
2006.0
0
50
100
150
200
250
300
350


700000
600000
500000
400000
300000
200000
100000
0
SalePrice
100000 200000 300000 400000 500000 600000 700000
SalePrice


700000
600000
500000
400000
300000
200000
100000
0
0 100 200 300 400 500 600 700

In [33]:

```
plt.subplots(figsize=(5,3.5))
df_house_price_train_numbers_ordinal_fit=preprocessing.StandardScaler().fit(df_house_price_train_numbers_ordinal).transfo
rm(df_house_price_train_numbers_ordinal)
df_house_price_train_numbers_ordinal_corr=pd.DataFrame(data=df_house_price_train_numbers_ordinal_fit,columns=df_house_pri
ce_train_numbers_ordinal.columns).corr()
sns.heatmap(df_house_price_train_numbers_ordinal_corr)
plt.show()
```

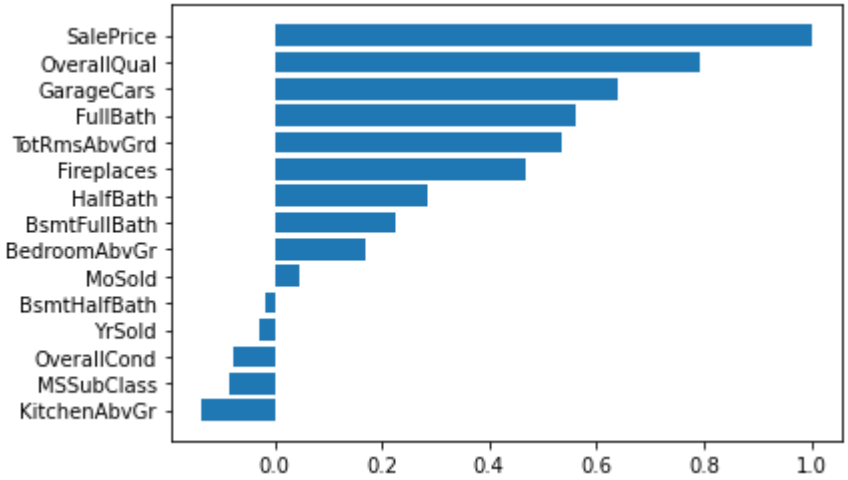


In [34]:

```
df_house_price_train_numbers_ordinal_corr_saleprice=df_house_price_train_numbers_ordinal_corr['SalePrice'].sort_values(as
cending=True)
```

In [35]:

```
y_pos = np.arange(len(df_house_price_train_numbers_ordinal_corr_saleprice))
plt.barh(y_pos, df_house_price_train_numbers_ordinal_corr_saleprice,tick_label=df_house_price_train_numbers_ordinal_corr_
saleprice.index)
plt.show()
```



Data Cleaning

Find and replace 'NAN's to '0's

In [36]:

```
df_house_price_train_objects_table_encoded.isnull().sum().sort_values(ascending=False)
```

Out[36]:

```
SalePrice        0
SaleCondition    0
ExterCond        0
ExterQual        0
MasVnrType       0
Exterior2nd      0
Exterior1st      0
RoofMatl         0
RoofStyle        0
HouseStyle       0
BldgType         0
Condition2       0
Condition1       0
Neighborhood     0
LandSlope        0
LotConfig        0
Utilities        0
LandContour      0
LotShape         0
Alley            0
Street           0
Foundation       0
BsmtQual         0
BsmtCond         0
GarageType       0
SaleType         0
MiscFeature      0
Fence            0
PoolQC           0
PavedDrive       0
GarageCond       0
GarageQual       0
GarageFinish     0
FireplaceQu      0
BsmtExposure     0
Functional       0
KitchenQual      0
Electrical       0
CentralAir       0
HeatingQC        0
Heating          0
```

```
BsmtFinType2     0
BsmtFinType1     0
MSZoning         0
dtype: int64
```

In [37]:

```
df_house_price_train_numbers=df_house_price_train_numbers.fillna('0')
df_house_price_train_numbers.isnull().sum().sort_values(ascending=False)
```

Out[37]:

```
SalePrice        0
BsmtHalfBath     0
GrLivArea        0
LowQualFinSF     0
2ndFlrSF         0
1stFlrSF         0
TotalBsmtSF      0
BsmtUnfSF        0
BsmtFinSF2       0
BsmtFinSF1       0
MasVnrArea       0
YearRemodAdd     0
YearBuilt        0
OverallCond      0
OverallQual      0
LotArea          0
LotFrontage      0
BsmtFullBath     0
FullBath         0
YrSold           0
HalfBath         0
MoSold           0
MiscVal          0
PoolArea         0
ScreenPorch      0
3SsnPorch        0
EnclosedPorch    0
OpenPorchSF      0
WoodDeckSF       0
GarageArea       0
GarageCars       0
GarageYrBlt      0
Fireplaces       0
TotRmsAbvGrd     0
KitchenAbvGr     0
BedroomAbvGr     0
MSSubClass       0
dtype: int64
```

Combine Dataframes

In [38]:

```
df_house_price_train_objects_table_encoded.drop(columns='SalePrice',inplace=True)
#df_house_price_train_numbers.drop(columns='SalePrice',inplace=True)
```

Compare Dataframes

In [39]:

```
# Cleaned Dataframe
df_house_prices=pd.concat([df_house_price_train_objects_table_encoded,df_house_price_train_numbers], axis=1)
df_house_prices.head()
```

Out[39]:

	MSZoning	Street	Alley	LotShape	LandContour	Utilities	LotConfig	LandSlope	Neighborhood	Condition1	...	WoodDeckSF	Ope
0	3	1	0	3	3	0	4	0	5	2	...	0	
1	3	1	0	3	3	0	2	0	24	1	...	298	
2	3	1	0	0	3	0	4	0	5	2	...	0	
3	3	1	0	0	3	0	0	0	6	2	...	0	
4	3	1	0	0	3	0	2	0	15	2	...	192	

5 rows × 80 columns

In [40]:

```
# Vs Original Dataframe
df_house_price_train.head()
```

Out[40]:

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

5 rows × 80 columns

In [54]:

```
# Re-order cleaned dataframe to match original dataframe
df_house_prices=df_house_prices[df_house_price_train.columns]
df_house_prices.head()
```

Out[54]:

	MSSubClass	MSZoning	LotFrontage	LotArea	Street	Alley	LotShape	LandContour	Utilities	LotConfig	...	PoolArea	PoolQC	Fe
0	60	3	65	8450	1	0	3	3	0	4	...	0	0	
1	20	3	80	9600	1	0	3	3	0	2	...	0	0	
2	60	3	68	11250	1	0	0	3	0	4	...	0	0	
3	70	3	60	9550	1	0	0	3	0	0	...	0	0	
4	60	3	84	14260	1	0	0	3	0	2	...	0	0	

5 rows × 80 columns

In [ ]:

```
# Remove outliers
```

In [42]:

```
# Determine PCA
```

In [43]:

```
# Potential Dimension Reduction
```

In [44]:

```
# Repeat for non-linear set
```

In [45]:

```
#
#sns.pairplot(df_house_price_train_numbers_continous)
```

In [ ]:

```

```

In [ ]:

```

```

In [ ]:

```

```

In [46]:

```
# Clean Data
```

In [47]:

```
# Visualize Data
```

In [48]:

```
# Clean Data
```

In [49]:

```
# Regression Model
```

In [50]:

```
# Regression Performance
```

In [51]:

```
# Create Clusters (Optional)
```

In [52]:

```
# Map Clusters (Optional)
```

In [53]:

```
# Data Stats (Optional)
```